Avoka Transact

Transact Manager (Manager v5.0)

Search this documentation

Contents

Recently Updated Pages

Transact Manager (Manager v5.1)
May 31, 2019 • updated by Anonymous • view change

Transact Manager (Manager v17.10)
May 31, 2019 • updated by Anonymous • view change

Managing Form Attachments (Manager v5.0)
May 30, 2019 • updated by Anonymous • view change

Delivery Configuration (Manager v5.0)
May 30, 2019 • updated by Anonymous • view change

Importing Forms from Transact Composer (Manager v5.0)
May 30, 2019 • updated by Anonymous • view change

Transact Manager Administration Guide (Manager v5.0)

This document provides a guide for system administrators managing a Transact Manager system. Before following any part of the document, ensure you have correctly installed the Transact Manager (TM) Instance. For installation instructions, please see the Transaction Manager Installation Guide.

Search this Document

Overview

Transaction Manager is the sophisticated Enterprise Forms Management System that acts as the cornerstone of Avoka Transact.

Over the years, many organizations have discovered the benefits of transforming their traditional paper-based forms into streamlined, web-based electronic forms. Though transforming a paper-based form into an electronic form is a great first step, to truly realize the benefits of modern web-based forms, organizations must provide several other features to their form.

The following list documents several common features that an organization would need to implement to realize the true benefits of web-based electronic forms.

- Provide user manageable profiles that are able to pre-populate forms
- Create and build an automated receipt generation system
- Create a submission tracking system
- Produce a submission management system

Transact Manager combined with Adobe LiveCycle ES, provides organizations with all these features and so much more. As a configurable web-based solution, Transact Manager manages the complete life cycle of forms created in Transact Composer and Transact Maestro.

The Form Life Cycle

There are three phases that make up the life cycle of a form in the Transact Manager solution. These three phases are identified and documented below.

Phase 1 - Form Definition & Publishing

The first phase of the form's life cycle is all about defining the form and publishing it to the organization's website. Links to Transaction Manager forms are typically embedded within an organization's website. When the organization's users click these links, they are redirected to the Transaction Manager server. the server retrieves the appropriate form definition, pre-populates it with fields and data, and then renders/presents the form to the user.

Phase 2 - Form Submission

The second phase of the life cycle concerns the submission process of the form. Once a user has completed the form (ie entered all mandatory fields and clicked the submit button), it is submitted (sent) back to the Transact Manager server. It is at this stage that the completed form (submission) is processed.

Transaction Manager supports several post-submission services that can be applied to be activated at this stage of the form's life cycle. Listed below are several of the post-submission services that can be applied at this stage.

- Attachments
- Payments
- Signature services

Once a submission has been processed, TM generates a receipt that can be emailed or downloaded by the user.

Registered users of the Transaction Manager Example Self Service Space may view any previous submissions and download receipts at their convenience. The Transact Manager Example Self Service Space is where registered TM users can manage their submissions.

Phase 3 - Submission Delivery

The third and final phase of the form life cycle is the delivery of the completed form, the delivery of the XML Data associated with the completed form, and the delivery of any relevant attachments that the form user may have attached to their submission to the intended recipient of the form.

Transact Manager provides several delivery options, including email, web services delivery mechanisms and custom LiveCycle ES processes. Along with these options, Transact Manager also provides an option for organizations to implement their own stand-alone organization-based delivery manager to handle the delivery of forms created for their organization.

Core Capabilities

Transact Manager has several core capabilities. These core capabilities are identified and documented below.

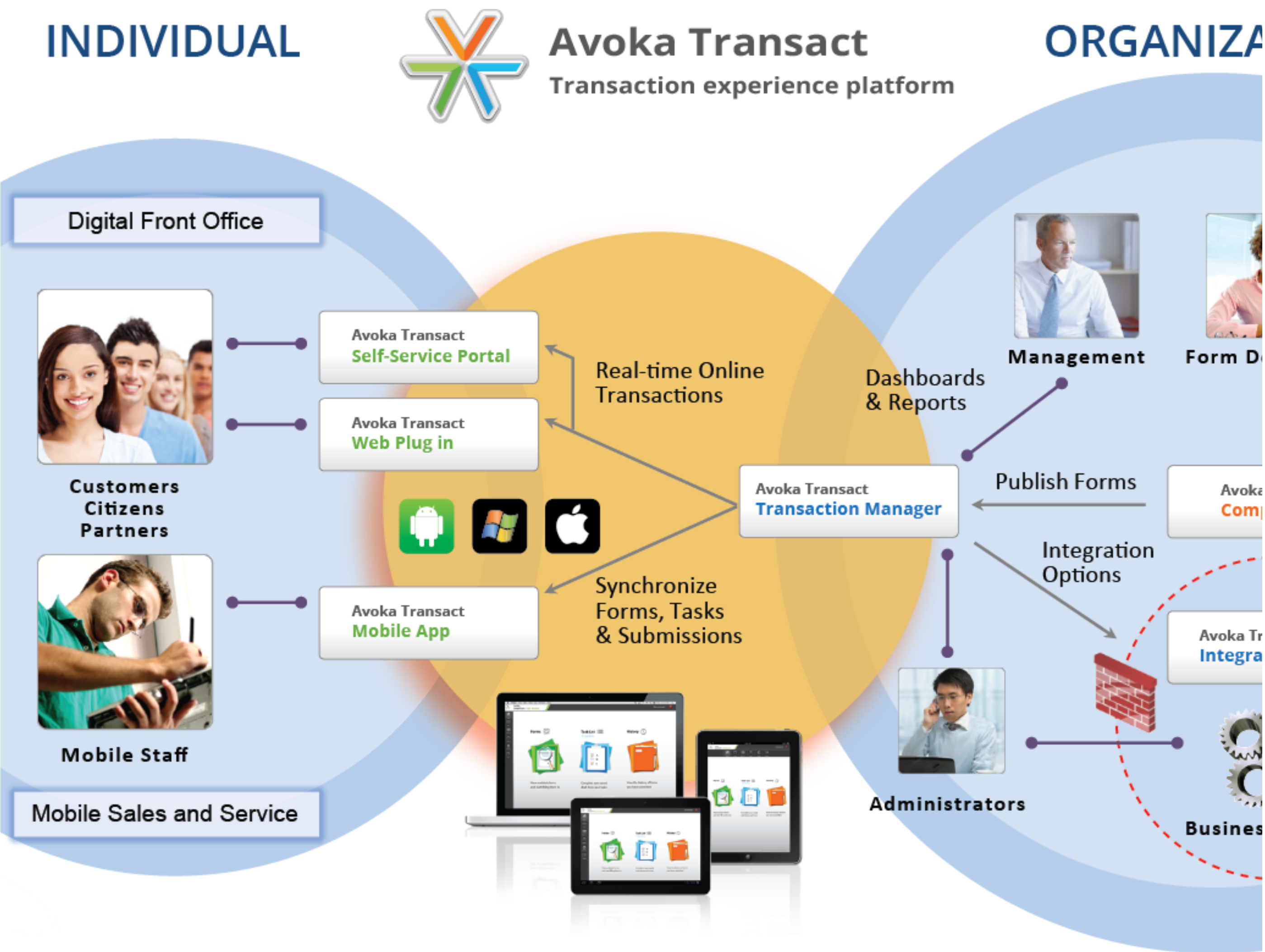
- **Form Hosting -** Multiple forms can be hosted centrally and be made accessible to your users. You can do this by embedding links to the forms into your web pages or by using form search engines. Transact Manager provides the tools required to manage forms and process submissions made by users.

- **Data Pre-Population -** Users can rapidly complete forms that are pre-populated with information. This information can come from one or more Transact Manager profiles or organization provided data retrieved via a web service.

- **Receipting** - Users are provided with an electronic receipt that includes the submitted form, payment and submission information. Receipts can be viewed online, emailed, or downloaded and printed.
- **Payment** - If a submission needs to be accompanied by a payment, Transact Manager can be configured to read the correct payment amount from the form and process the payment using a secure credit card payment process.
- **Supporting Documentation -** Submissions can be supported by additional documents that may be uploaded electronically or submitted in paper form. Business logic in the forms can be applied to supporting documents (ie plans must be submitted for building work over $10000).
- **Submission Delivery -** Completed submissions can be delivered by email to one or more addresses. These addresses can be assigned by web services or by using a custom LiveCycle process. The delivery process can be configured separately for each form.
- **Form Configuration and Version Control -** Forms provide fine-grained control over behavior, including form properties, form metadata, and custom LiveCycle rendering orchestrations. Forms also support version control enabling parallel development and multiple versions of the same form. Transact Manager also provides a sophisticated facility for promoting forms from development, to test, and finally production on separate servers.
- **Task Assignment -** To facilitate more flexible workflows, forms along with their prefilled data can be assigned to a user or a group of users. This task is then presented to the user on their to-do list.
- **Customized User Experience -** Transact Manager provides an Example Self Service Space where users can manage their submissions. Custom-branded user spaces based on the Example Spaces are easy to create and deploy.
- **Submission Tracking -** All submissions can be tracked from form download to submission delivery. Customer support can view forms and attachments to provide user assistance during the submission process.
- **Reporting -** Transaction Manager provides multiple charts and standard reports for reporting on submissions. Additional reports can be configured using advanced reporting modules. Reports can be displayed on a screen or can be automatically sent out via email.

Transact Modules

The image below illustrates the major modules of the Transact solution.



Transact System Modules

The major Transact system components are identified and documented below.

- **Web Plugin** – provides a form space for customer facing sales and service transactions. This form space is linked to from your (organization) website, but can also be embedded on your website if the form is fairly small. Web Plugin supports unauthenticated and authenticated users.
- **Work Space** – provides a work space for business staff working on tasks, jobs, completing ad-hoc forms, and providing help desk support. The work space is generally used by authenticated users but also supports unauthenticated transactions as well.
- **TransactField App** – provides a mobile app and PC application which enables business field staff to complete forms offline. Once online, the application allows them to sync their work with the Transact Manager server, and receive new work tasks to complete. The TransactField App supports authenticated users only.
- **Transact Manager**– provides an Enterprise Forms Management System enabling a sophisticated customer interaction and system integration hub between form transactions and back-end business processes.
- **Business Reports** – provides a Business Intelligence Reporting Tool (BIRT) for scheduled and ad-hoc reporting.

Transaction Manager supports a flexible deployment of its components to multiple application server instances to support scalability. In addition, one or more LiveCycle instances may be used for rendering, with real-time health monitoring and automated load balancing.

Security (Manager v5.0)

This chapter deals with the security model implemented in the Transact Manager platform.

Search

Concepts

A *user account* in Transact Manager (TM) is an account of a user who is able to successfully log on to TM. A user account may be authorized to log on to the management console and/or one or more spaces in Transact Manager. This authorization is controlled by assigning the user account to one or more spaces (one of these spaces may be the management console).

In addition, each user account can belong to one or more Transact Manager roles. Roles are collections of user accounts with similar access rights. Similarly, user accounts can belong to one or more groups in Transact Manager. A group is a collection of users who are gathered based on something other than permissions (eg a common goal).

An administrator in Transact Manager is a user account that is associated with/has authorization to access the Transact Manager management console space.

A space user is a user account that is associated with at least one user space. Note that these categories are not mutually exclusive: An administrator can also be associated with many user spaces, and can also have global access or be restricted to data associated with one or more specific organizations.

Transact Manager controls access to certain content or actions using *permissions*. When requesting restricted content, Transact Manager will only allow this this content to be accessed if the required permissions are present. Permissions can be granted to one or more roles, and a role is characterized by the permissions it contains. If a user account belongs to several roles, they will have all permissions that are set for at least one of their roles.

As users make changes to Transact Manager, these changes are tracked and written to the *audit log*. This makes it possible to view an object's history and review a specific user's changes.

To provide flexibility around the security configuration, Transact Manager supports one or multiple *security managers*. A security manager contains such settings as how users are authenticated (against TM-managed users and/or Lightweight Directory access Protocol (LDAP)), lockout and password settings, session timeout, SSO configuration and user self-registration settings.Each security manager contains one or more *authentication managers* responsible for authenticating a user during login (by default, there is an authentication manager for authentication against an LDAP directory and another authentication manager for authentication against the TM user database).

Each space in TM can be associated with its own security manager or use the default security manager defined in the system.

Permissions

Each permission in Transact Manager represents the right to access restricted content, for example editing a form or viewing an error log.

Permissions are specific to a space; the Transact Manager management console defines a fixed set of permissions, and each user space can define its own permissions and implement security rules.

A full list of the available permissions in Transact Manager can be found in the Permissions (Manager v5.0)page.

Roles

Transaction Manager comes with a super administrator role with all permissions related to the Transaction Manager management console. As a Transact Manager Administrator, you are able to create new roles and modify existing roles to suit your needs.

Standard Roles

When Transaction Manager starts up for the first time (or whenever there are no roles defined in the Transaction Manager database), it will automatically create the following set of default roles.

- A root administrator role with all permissions
- A form developer role
- An operations monitoring role,
- An organization user manager role
- A system support role

As an administrator, you have permissions to customize these roles to suit your specific needs, or set up new roles.

Root Administrator

This is the most powerful role in Transact Manager. Root administrators have all permissions related to the Transact Manager management console assigned to them and can access all restricted content. Critical operations such as deleting organizations and editing deployment properties should be accessible to these users only.

Developer / System Manager

These two roles have identical permissions. Developer/System Manager users have a wide range of permissions available to them to manage TM configurations. Users in these roles are able to edit organizations, forms and all related entities. Developers and System Managers are also able to view requests, view submissions as well as view a wide range of system configuration settings. These roles are very powerful but cannot access submission data, user accounts and are not able to modify roles.

Operations

Organization operation administrators can view submissions (included submitted XML data) and reports as well as various logs. Organization operation administrators are given these permission to allow them to handle support requests submitted by space users.
However, they cannot access forms and associated configurations, nor can they access the details for user accounts stored in Transact Manager.

Organization User Manager

Organization user administrators can create and configure users for their organization. This is a role that might be used in addition to other roles to give existing administrators access to user accounts, though it can also be used as a stand-alone role.

System Support

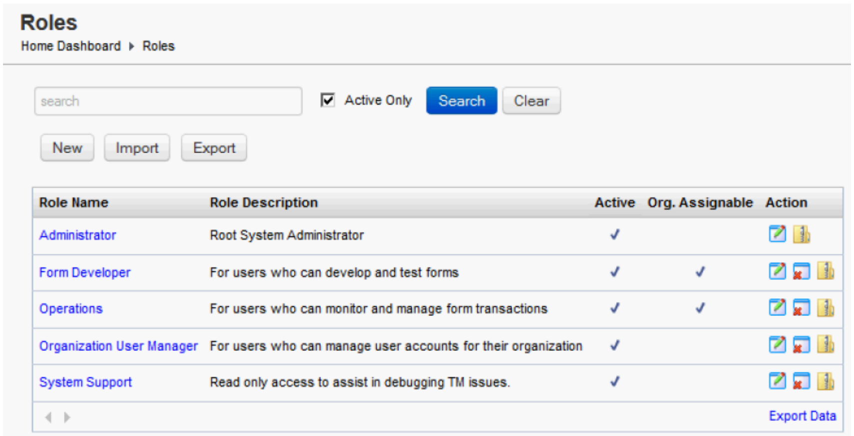
The system support role can be used by Avoka staff for debugging purposes. By default, this role has extensive read-only permissions to aid in debugging. However, XML data submitted by users cannot be viewed.

Transaction Data Access

This role provides access to transaction data entered by users and allows administrators to edit this data and make submission status changes. As this allows access to very sensitive information, this role should be assigned very carefully.

Browsing the List of Roles

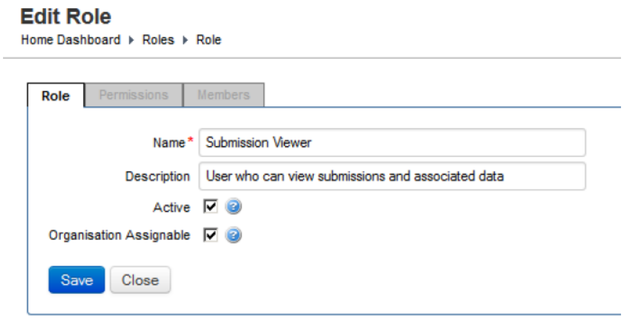
Transact Manager allows administrators to view the list of available TM roles. To view this list of roles, go to "Security > Roles". You can filter this list by using the search box and entering the name of the role and a description of the role, and exporting the search results to Excel by clicking the "Export Data" link below the search text box. The screenshot below illustrates the search roles functionality.



Using the role page (documented in the above screenshot), you can add, modify, assign permissions to and manage the members of a role by using the edit icon ( ). To delete a role, use the delete icon ( ). It is also at this page that as an administrator, you are able to import and export roles.

Adding a Role

The screenshot below documents the process of a new role being created (see below for an example with the fields filled in).



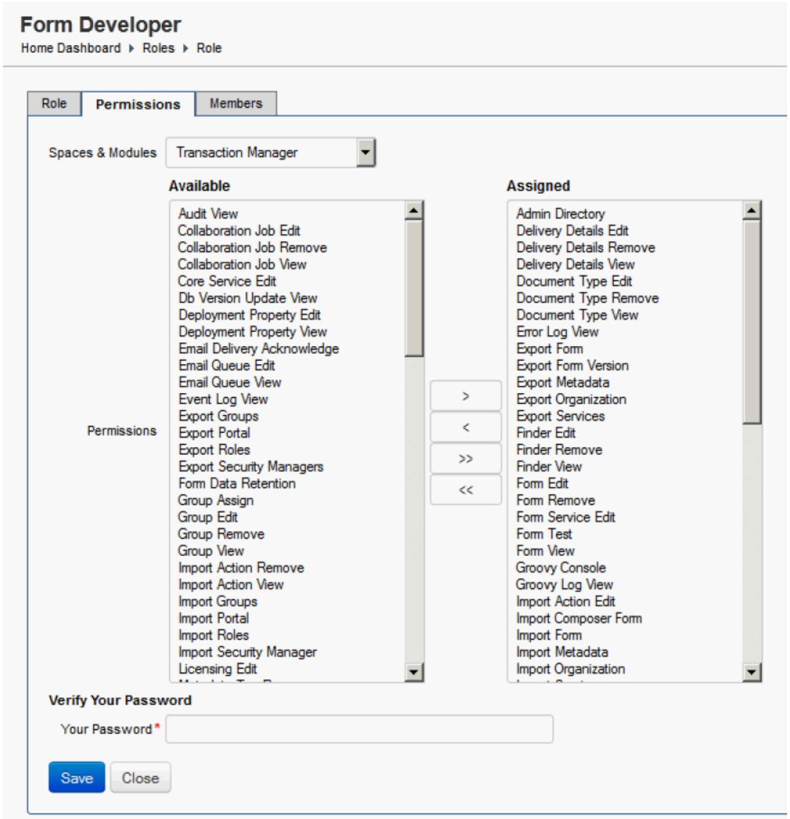
Creating a role

Follow the steps below to create a new role.

1. Navigate to the Roles page (Security > Roles)
2. Click the "New"button. Clicking this button will redirect you to the "Edit Role"page.
3. Provide a name for your new role.
4. Provide a description of your role (This step is optional).
5. Select the "Active"checkbox. If you fail to select this checkbox your new role will be considered "Inactive". Inactive roles cannot be assigned to users.
6. Select the "Organization Assignable" checkbox if you want to allow organization administrators the permission to assign this new role to other administrators within the same organization. If you do not select this checkbox, even organization administrators with permissions to assign roles will be unable to see or assign this new role.
7. Click save

Once you have saved the role, it is time to assign permissions to the role.

Setting Up Permissions for a Role



Assigning permissions to a role

Follow the steps below to assign permissions to a role.

1. Navigate to the "Roles" page (Security > Roles)
2. Select the role you want to assign permissions to and click Edit
3. Select a space from the "Spaces and Modules" drop down. Selecting a space will update the "Permissions" list to include only those permissions belonging to the selected space
4. Select the "Permissions"tab. Selecting this tab will display all of the available permissions and the permissions that are currently assigned to that space (the above screenshot documents the "Permissions" list page for the 'Transaction Manager' space).
5. Select a permission from the 'available permissions' list that you want to assign to the selected space and use the ">" button to assign the selected permission to the space (to select multiple permissions, use the Shift and Ctrl keys, on Windows).
6. Enter your administration password
7. Click save

Note: Using the ">>" button will assign all available permissions to the selected role.

All permissions assigned to a role are loaded when the user logs in. In other words, any changes will be visible to users as soon as they log out and then log back in to the space.

Removing permissions from a role

Follow the steps below to remove permissions from a role.

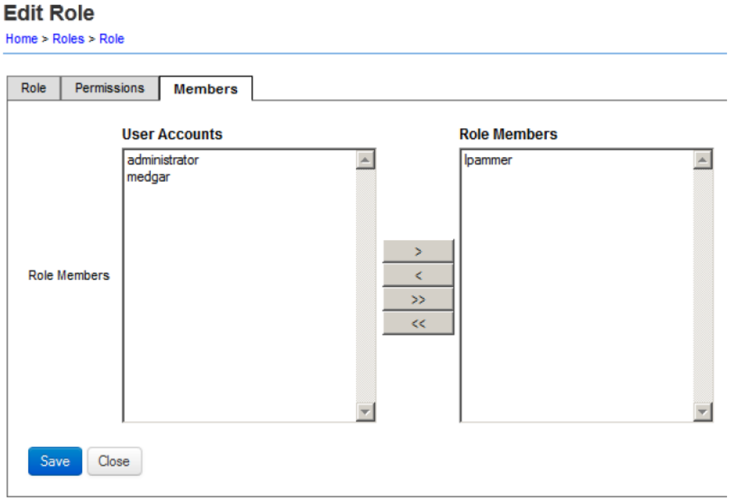
1. Navigate to the "Roles" page (Security > Roles)
2. Select the role you want to remove permissions from and click Edit
3. Select a space from the "Spaces and Modules" drop down. Selecting a space will update the "Permissions" list to include only those permissions belonging to the selected space
4. Select the "Permissions"tab. Selecting this tab will display all of the available permissions and the permissions that are currently assigned to that space (the above screenshot documents the "Permissions" list page for the 'Transaction Manager' space).
5. Select a permission from the 'assigned permissions' list that you want to remove from the selected space and use the "<" button to remove the selected permission from the space (to select multiple permissions, use the Shift and Ctrl keys, on Windows).
6. Enter your administration password
7. Click save

Note: Using the "<<" button will remove all assigned permissions from the selected role.

All permissions assigned to a role are loaded when the user logs in. In other words, any changes will be visible to users as soon as they log out and then log back in to the space.

Managing Role Members

As an administrator, you are able to view and modify the list of users who are currently assigned particular roles (role members).



Managing role membership

Follow the steps below to enter the managing role membership page.

1. Navigate to the Roles page (Security > Roles)
2. Select a role
3. Click the 'Edit Role' icon ( )
4. Switch to the 'Members' tab

Select one or more user accounts (on Windows, use the Shift and Ctrl keys to select multiple user accounts) and use the ">" and "<" buttons to assign and revoke the role from all selected users.

You can also assign the role to all users by using the ">>" button; likewise, the "<<" button will remove the role from all users.
Once you have finished modifying the list of members, enter your administrator password (if required) and save the role.

Deactivating a Role

You can deactivate a role to ensure its permissions are no longer applied without deleting the role (e.g. so it can later be reactivated if needed). To do so, edit the role, untick the "Active" checkbox and save. Inactive roles are not available for assignment to users and do not confer any permissions until they are reactivated.
The "Security > Roles" page also hides inactive roles by default. You can include them in searches by unticking the "Active Only" checkbox.

Deleting a Role

You can delete a role by going to "Security > Roles" and clicking the "Delete" icon ( ) next to the role. You will be asked to confirm this action before the role is deleted. This action cannot be undone.

Importing / Exporting Roles

You can export roles and permissions assigned to them, to be imported to a different Transaction Manager instance or for backup purposes.
To learn about how to export and import roles, please refer to Multi-Environment Support (Manager v5.0)#_sec_MigratingRoles.

Groups

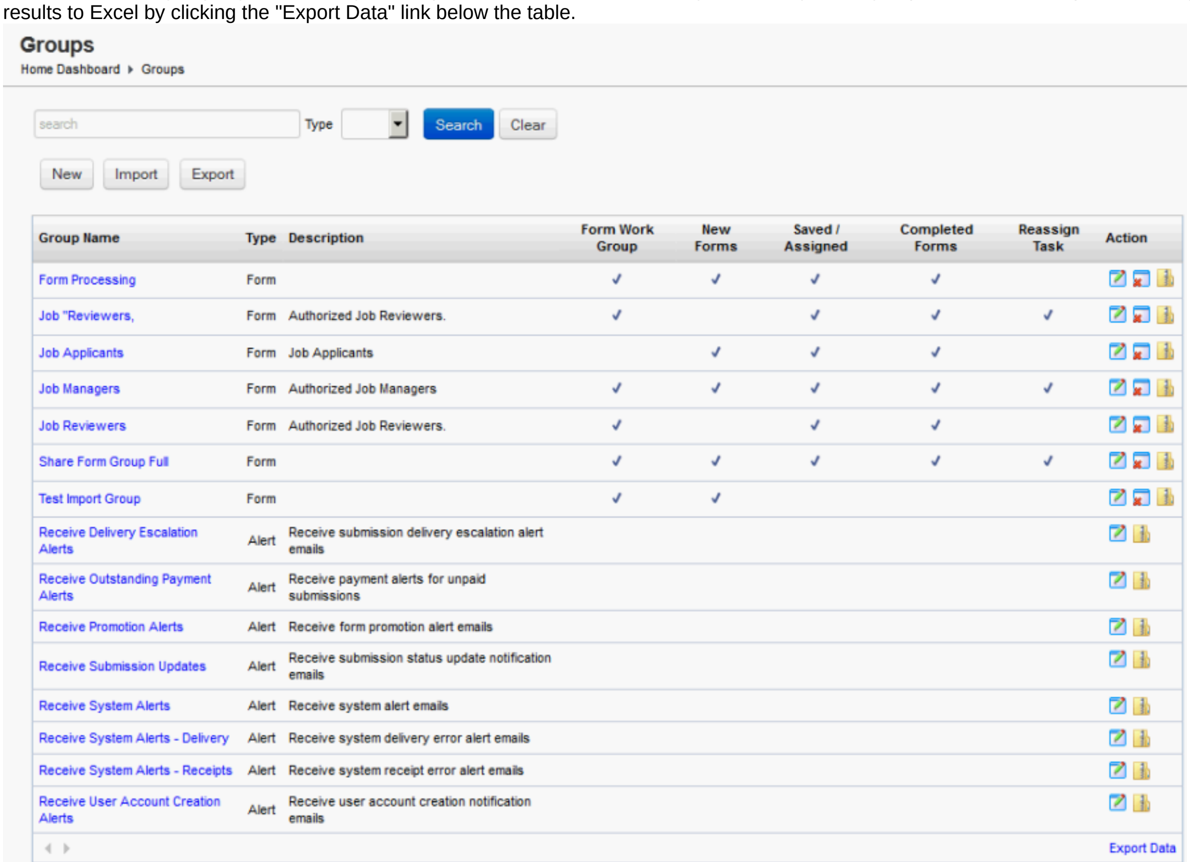
Groups can be used to define sets of users who have something in common other than a shared permission set as provided by roles. Transaction Manager provides a set of pre-defined groups which users can be assigned to in order to receive email notifications about certain events.
You can also create form work groups to control access to a set of forms.

Form Work Groups

Groups that control access to forms are called form work groups. All users within a form work group share the forms assigned to the group to a certain extent; the exact nature of what is being shared is controlled on the group level.
There are three aspects of form sharing that can be configured for a form work group. Users can simply share access to a set of forms that all of them can open and submit, but that are not accessible to non-members. Secondly, users can share tasks and incomplete (e.g. saved) submissions for the forms in the group, meaning that each group member can see drafts submitted by other members and complete these draft forms even if they were not the initial submitters. Finally, the form work group can be configured to allow users to see all completed submissions of group forms by other group members and, if configured, access the receipt for these submissions.
Each of these three settings can be turned on independently, allowing for flexible form work group configuration. Each group applies only to forms assigned to the group, and confers its privileges only to user accounts who are members of the group.

Browsing the List of Groups

To view the list of groups, go to "Security > Groups". You can filter the list using a substring of the group name and description, and export the search results to Excel by clicking the "Export Data" link below the table.



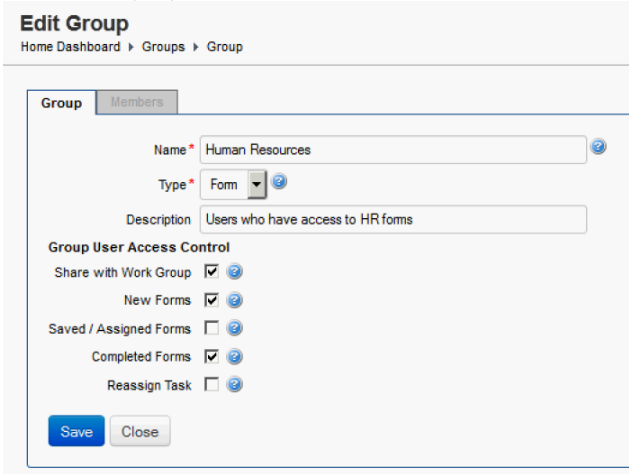
Group Name	Type	Description	Form Work Group	New Forms	Saved / Assigned	Completed Forms	Reassign Task	Action
Form Processing	Form		✓	✓	✓	✓		
Job "Reviewers,	Form	Authorized Job Reviewers.	✓		✓	✓	✓	
Job Applicants	Form	Job Applicants		✓	✓	✓		
Job Managers	Form	Authorized Job Managers	✓	✓	✓	✓	✓	
Job Reviewers	Form	Authorized Job Reviewers.	✓		✓	✓		
Share Form Group Full	Form		✓	✓	✓	✓	✓	
Test Import Group	Form		✓	✓				
Receive Delivery Escalation Alerts	Alert	Receive submission delivery escalation alert emails						
Receive Outstanding Payment Alerts	Alert	Receive payment alerts for unpaid submissions						
Receive Promotion Alerts	Alert	Receive form promotion alert emails						
Receive Submission Updates	Alert	Receive submission status update notification emails						
Receive System Alerts	Alert	Receive system alert emails						
Receive System Alerts - Delivery	Alert	Receive system delivery error alert emails						
Receive System Alerts - Receipts	Alert	Receive system receipt error alert emails						
Receive User Account Creation Alerts	Alert	Receive user account creation notification emails						

Browsing the list of groups

As you see, there is already a set of alert groups defined. These groups have the read-only flag set and cannot be renamed or deleted.
Using this page, you can add, modify ( ) or delete ( ) groups.

Adding a Group

To add a new group, click the "New" button. You will be directed to the "Edit Group" page (see below for an example of a form work group).

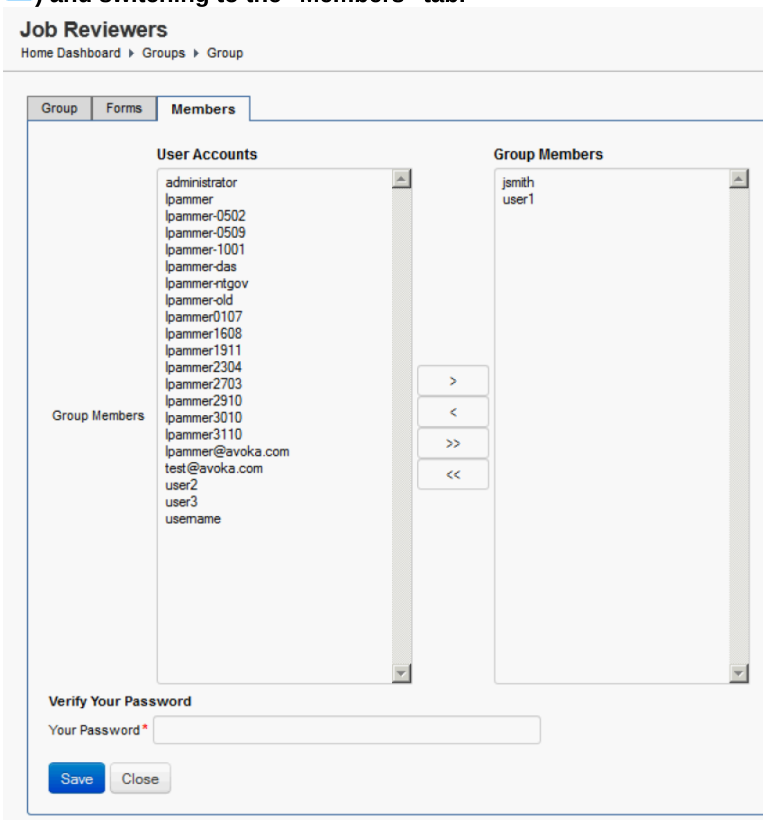


Creating a custom group

Fill in a unique group name and an optional description.
Choose the group type (you cannot currently define custom alerts, so you should select "Form" to create a form work group).
When creating form work groups, you can define the access rules described in _sec_formworkgroups. Tick "New Forms" to allow users to actively start filling in forms belonging to the group (otherwise they cannot access the forms at all, except when tasks are assigned to the group and the group is configured to share tasks and incomplete submissions). Tick "Saved / Assigned Forms" to allow sharing of tasks assigned to the group as well as incomplete submissions for these forms. Tick "Completed Forms" to allow sharing of the list of completed submissions and access to receipts.Tick "Reassign Task" to allow members to reassign task submissions.
Save the group. Now you can start assigning forms and users to the form work group.

Managing the List of Group Members

You can view and modify the list of group members by going to "Security > Groups", clicking the "Edit Group" icon ( ) and switching to the "Members" tab.



Managing group membership

The list of all current group members is displayed in "Group Members".

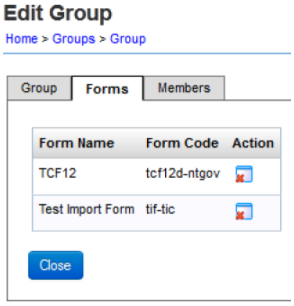
Select one or more user accounts (on Windows, use the Shift and Ctrl keys to select multiple user accounts) and use the ">" and "<" buttons to add and remove the selected members.

You can also add all users to the group by using the ">>" button; likewise, the "<<" button will remove all users from the group.

When you have finished modifying the list of members, enter your administrator password (if required) and save the group.

Viewing Forms Assigned to a Form Group

Form groups control access to one or more forms. To see which forms are currently associated with a form group, edit the group and switch to the "Forms" tab.



Viewing the list of forms associated with a form group

You can remove forms from the group by clicking the "Remove Form From Group" icon ( ) next to the form. To add a form to the group, please follow the instructions in chapter "Form Configuration", section "Form Group Access" .

Deleting a Group

You can delete a group by going to "Security > Groups" and clicking the "Delete" icon ( ) next to the group. You will be asked to confirm this action before the group is deleted. This action cannot be undone.

Pre-Defined Groups

Transaction Manager comes with several pre-defined alert groups:

1. Receive Delivery Escalation Alerts
   Members of this group will be notified when submissions with secure email delivery have not been acknowledged after a certain period (for the exact rules, see chapter "Delivery Configuration", section "Secure Email Delivery").
2. Receive Outstanding Payment Alerts
   Members of this group will be notified when there are submissions with outstanding payments for their organization. This group is relevant only if administrators wish to monitor submissions for those that are abandoned at the payment stage (e.g. to contact the applicant and offer assistance).
3. Receive Promotion Alerts
   Members of this group will be notified when a form belonging to their organization is promoted (for the exact rules, see Form Configuration (Manager v5.0)#_sec_promotingaformversion).
4. Receive Submission Updates
   Members of this group will be notified when the processing status for a submission they have made has been updated. This group is relevant only to space users, not administrators.
5. Receive System Alerts
   Members of this group will be notified when the System Monitoring job (see Scheduled Jobs (Manager v5.0)#_sec_systemmonitoringjob) finds that important system jobs are impacted.
6. Receive System Alerts - Delivery
   Members of this group will be notified when submission delivery fails.
7. Receive System Alerts - Receipts
   Members of this group will be notified when submission receipt generation fails.
8. Receive User Account Creation Alerts
   Members of this group will be notified when a user has applied for an account on a user space (this will be discussed in _sec_customizinguserenrolment).

Dynamically Loaded Groups

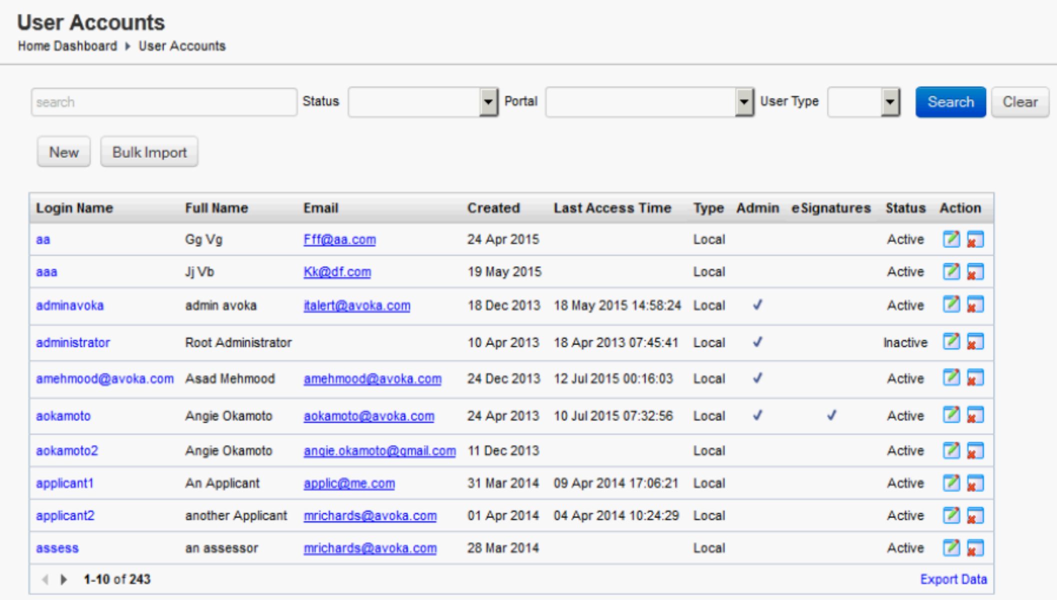
When TM users are managed externally (LDAP or SSO authentication), TM can load the user's group associations at login time. If groups with identical names are set up in TM, these dynamically loaded group associations can be used to control form work group access.
For LDAP users, groups are usually defined as a distinguished name (e.g. "CN=mygroup,OU=Distribution Groups,OU=MyBusiness,DC=mycompany"). Since these names are somewhat unwieldy, TM parses out the group name (to be exact, it takes the string before the first comma ("CN=mygroup") and parses out the string following the first equals sign, resulting in a group name of "mygroup".
For SSO users, the group names can be passed in when creating the SSO token. As Groovy scripts are used to customize SSO behavior, you can tailor the scripts to parse out the group names in any way you need. Please refer to the Groovy services guide ("System > Groovy Services Guide") for more information on the Groovy scripts involved in SSO authentication.

User Accounts

A user account is a login to a Transaction Manager space. A user account is characterized by some personal data, their credentials, user type ("Local", "LDAP" or "SSO"), one or more user profiles, the spaces they can log on to, the organizations they can access and the roles and groups they belong to.

Browsing the List of User Accounts

To view the list of user accounts set up in Transaction Manager, go to "Security > User Accounts". You can filter the list by organization, a substring of the user's login name and full name, user type and account status ("Active", "Inactive", "Locked", "Pending", "Rejected"). In addition, you can choose to show only user accounts associated with a specific space.
You can export the search results to Excel by clicking the "Export Data" link below the table.

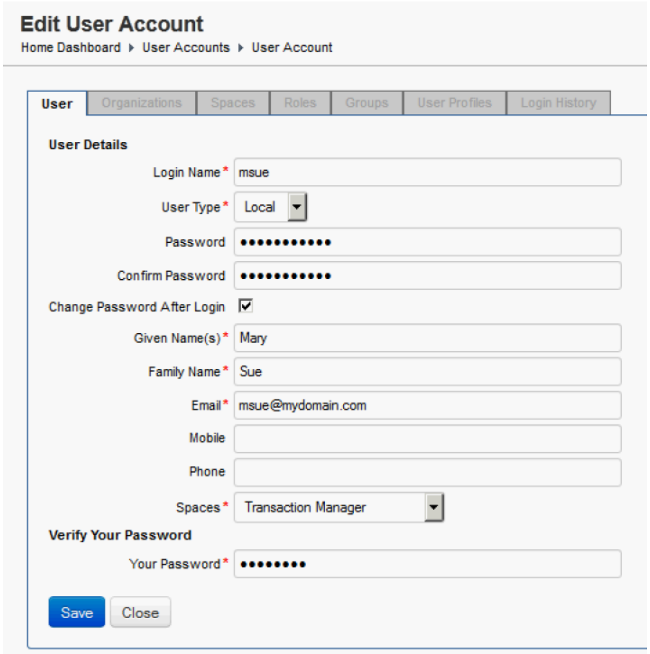


Browsing the list of user accounts

From this page, you can add, edit (, also allows you to assign spaces, roles and groups to the user) and delete ( ) user accounts. You can also import a list of user accounts.

Adding a User Account

To add a user account, click the "New" button. You will be taken to the "Edit User Account" page (an example with all fields filled in is shown below).



Creating a user account

Choose the appropriate user type from the "User Type" dropdown. For users managed by Transaction Manager, use "Local" (selected by default). If you would like to manually set up an LDAP user, choose "LDAP" (note that you will get a restricted field set as the user is managed in the LDAP directory in this scenario - see _sec_authenticationproviders for more on user authentication); similarly, choose "SSO" for an SSO user account (only applicable if you are using an SSO security manager).

Fill in the personal details (given and family names, email address and phone numbers).

Enter a unique login name and the initial password. By default, the password must be at least twelve, but at most 20 characters long and contain at least one special character and mixed case letters (the exact requirements depend on the security manager configuration for the chosen space).

If "Change Password After Login" is ticked, the new user will be forced to change his initial password when logging on, and will be unable to access any restricted content until this has been done.

Choose a space from the "Spaces" dropdown. This will be the space (either the management console or one of the user spaces) that the user will be associated with initially. More spaces can be added later as described in _sec_changespacesforuseraccount".

For security reasons, you must enter your own password to create the user.

Once you have filled in all the data, save the new user account.

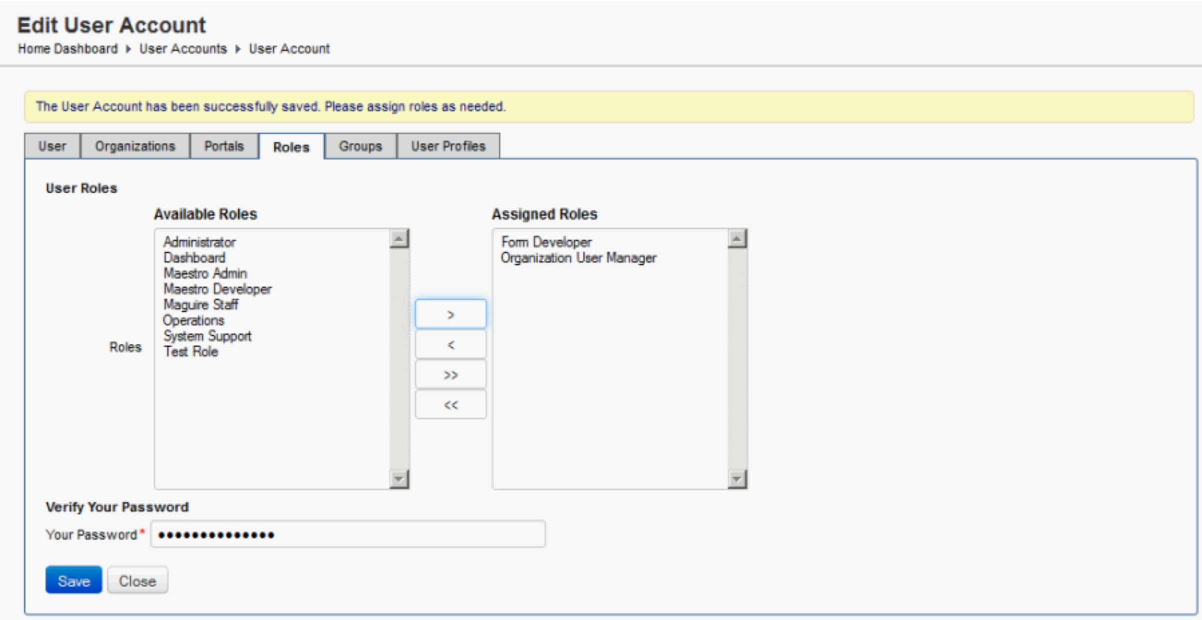
Note that the user will not have access permissions until they have been assigned a role. When creating a new user, Transaction Manager will automatically open the "Roles" tab for you so you can assign roles as needed.

The following sections describe how to assign one or more roles and spaces to a user account and set up organization access. These steps are essential for setting up functional administrator users.

Changing the Roles Assigned to a User Account

There are several scenarios where you might want to change the roles assigned to a user account: A new user account needs to be set up with its initial set of roles, and existing user accounts may need further access rights. If you suspect that a user account may have been compromised, you can remove all their roles and spaces and make the user account inactive as well, making it unable to log on or perform any restricted actions.

To change the roles assigned to a user account, go to "Security > User Accounts", click the "Edit" icon ( ) next to the user account and switch to the "Roles" tab.



Changing the roles of assigned to a user account

Please note that there are precautions built into the Transaction Manager management console that prevent organization administrators from assigning themselves inappropriate roles. If an organization administrator edits the role assignments for a user account, they will see only roles for which "Organization Assignable" is ticked. All other roles will be hidden. Global administrators, on the other hand, can see and assign all roles to other user accounts (if their own roles contain the permissions around role assignment).

To change the roles assigned to a user account, select roles (on Windows, use the Shift and Ctrl keys to select multiple roles) and use the ">" and "<" button to assign and remove them.

You can also add all available roles to a user account by using the ">>" button; likewise, the "<<" button will remove all roles, leaving the user account without any privileges.

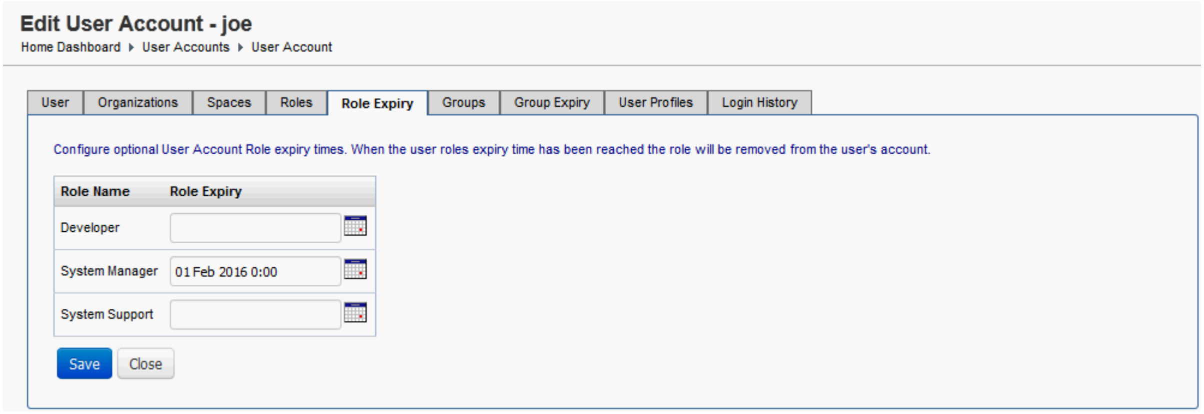
For security reasons, you may have to enter your own password to save changes to the roles.

Once all roles have been assigned appropriately, save the user account.

The roles assigned to a user account are loaded when the user logs in, so changes will be visible to the user after logging out of the space and logging in again.

Managing Role Expiry

Sometimes it makes sense to assign a role to a user for a limited amount of time (e.g. to allow Avoka support to help debug a problem). TM allows you to configure an expiry time for user roles so you don't have to remember to remove the roles yourself. To manage role expiry times for a user, edit the user and switch to the "Role Expiry" tab.



Configuring role expiry for a user account

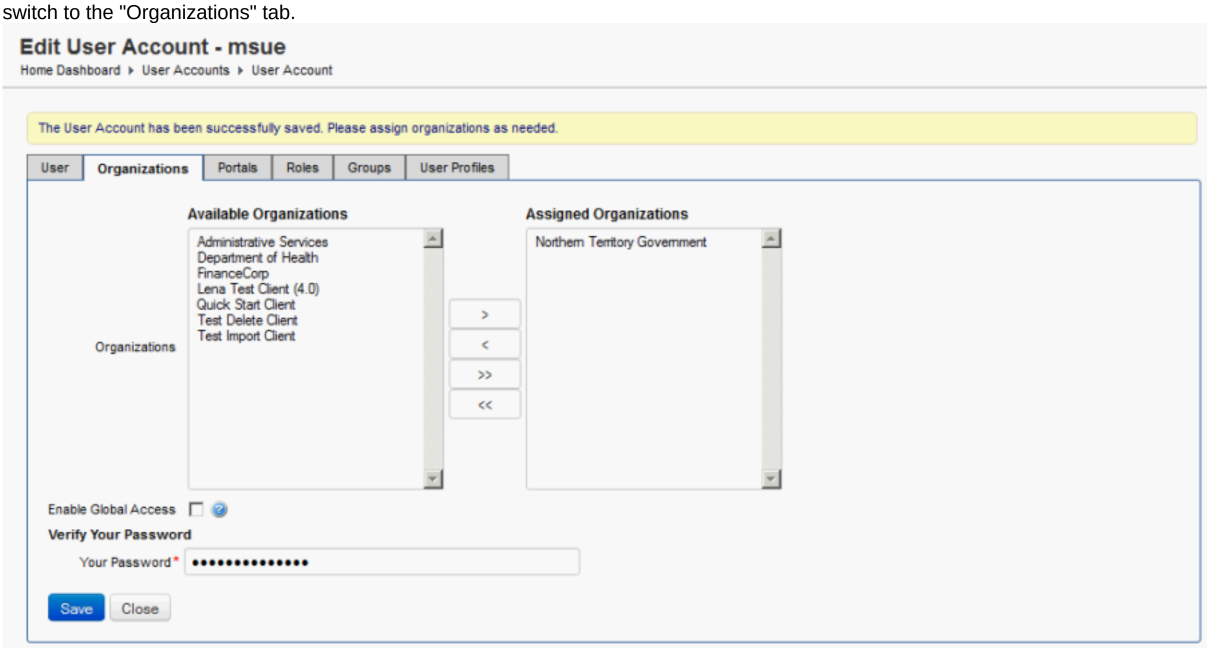
You can enter an expiry time for each role or remove expiry times if needed.

Save your changes. When the expiry time arrives, the background job "Security Policy Manager" (see Scheduled Jobs (Manager v5.0)#_sec_scheduledjob_securitypolicymanager) will remove the role from the user account.

Changing the Organizations Accessible to a User Account

Users associated with the TM management console can be given global access or restricted to one or more organizations. Organization administrators have restricted access privileges; in particular, whenever there is data associated with organizations, they will see only the subset relevant to their own organization(s) (e.g. on the "Forms" page, they will see only their own forms). Organization administrators can also assign only organization assignable roles to other administrators (effectively preventing them from promoting themselves or others to global administrators).

To change the organizations a user account can access, go to "Security > User Accounts", click the "Edit" icon ( ) next to the user account and switch to the "Organizations" tab.



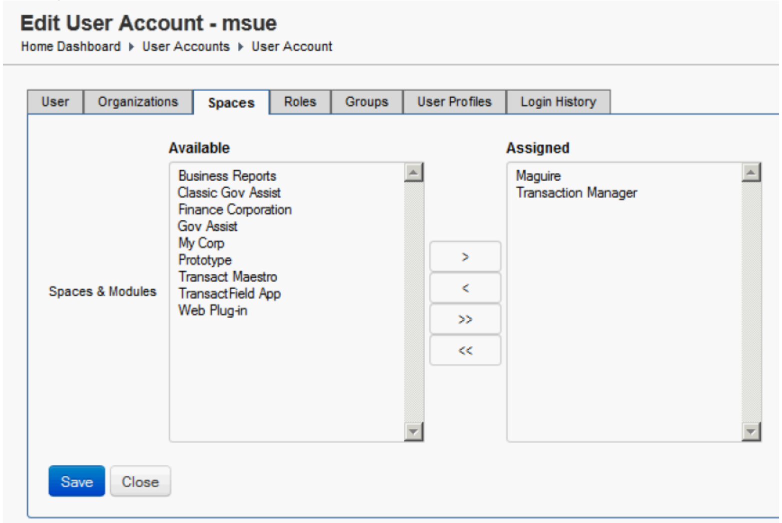
Configuring organization access for a user account

The "Enable Global Access" checkbox controls whether a user is a global administrator or an organization administrator. If ticked, the user automatically has access to all organizations. Otherwise individual organizations must be assigned to the user.
To change the organizations a user account can access, select organizations (on Windows, use the Shift and Ctrl keys to select multiple organizations) and use the ">" and "<" button to assign and remove them.
For security reasons, you have to enter your own password to save changes to the organizations.
Once all organizations have been assigned appropriately, save the user account.

Changing the Spaces Assigned to a User Account

Before a user can log in to the management console or a user space, they need to be associated with the corresponding space record. An existing user can register for a space account if the user space supports this; the management console does not support self-registration.

To change the spaces assigned to a user account, go to "Security > User Accounts", click the "Edit" icon ( ) next to the user account and switch to the "Spaces" tab.

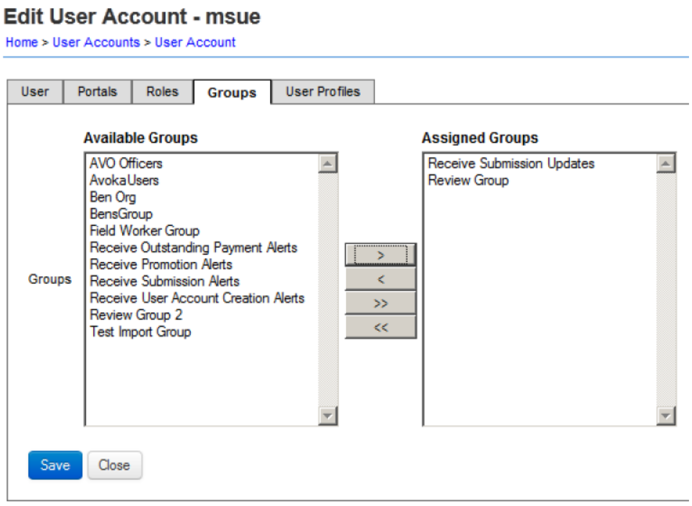


Changing the spaces assigned to a user account

To change the spaces assigned to a user account, select spaces (on Windows, use the Shift and Ctrl keys to select multiple spaces) and use the ">" and "<" button to assign and remove them.
You can also add all available spaces to a user account by using the ">>" button; likewise, the "<<" button will remove all spaces.
Once all spaces have been assigned appropriately, save the user account.
Note: Only global administrators can always assign all spaces to user accounts. Administrators restricted to one or several organizations can assign only the spaces they themselves have been assigned.

Changing a User's Group Associations

To change the groups a user belongs to go to "Security > User Accounts", click the "Edit" icon ( ) next to the user account and switch to the "Groups" tab.

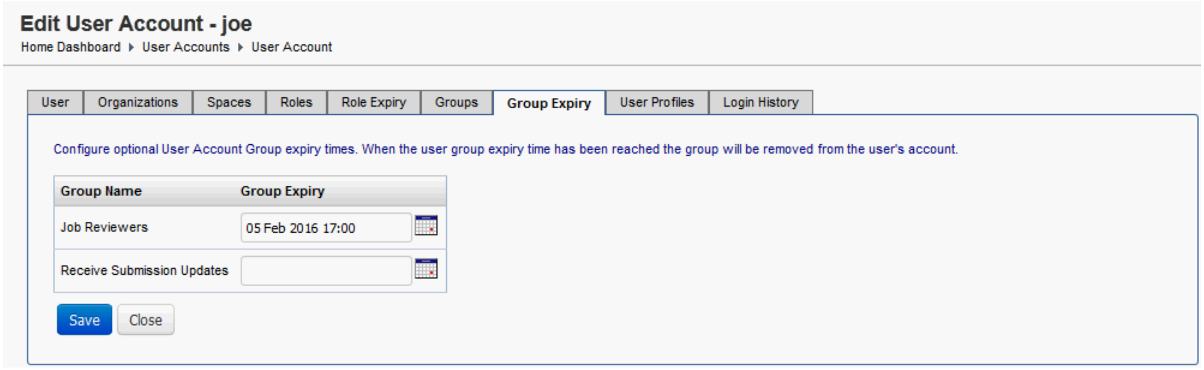


Assigning a user to groups

To change the groups assigned to a user account, select groups (on Windows, use the Shift and Ctrl keys to select multiple groups) and use the ">" and "<" buttons to assign and remove them.
You can also add the user to all available groups by using the ">>" button; likewise, the "<<" button will remove the user from all groups.
Once all groups have been assigned appropriately, save the user account.

Managing Group Expiry

Sometimes it makes sense to assign a user to a group for a limited amount of time. TM allows you to configure an expiry time for user groups so you don't have to remember to remove the user from the groups yourself. To manage group expiry times for a user, edit the user and switch to the "Group Expiry" tab.



Configuring group expiry for a user account

You can enter an expiry time for each group or remove expiry times if needed.

Save your changes. When the expiry time arrives, the background job "Security Policy Manager" (see Scheduled Jobs (Manager v5.0)#_sec_scheduledjob_securitypolicymanager) will remove the user account from the group.

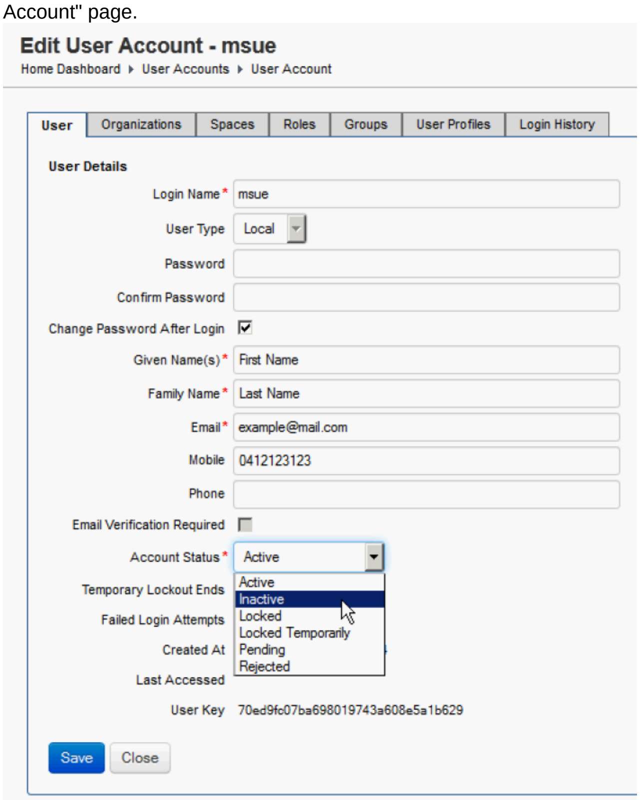
Removing a User Account

If you wish to disable a user account, you have two options: You can deactivate the account, which will make them unable to log on but preserve the user record in Transaction Manager, or you can delete the user account, which irrevocably removes all data associated with the user account.

Note that submissions made by a user will be preserved even after their user account is deleted; however, these submissions will not be associated with a user any more.

Deactivating a User Account

To deactivate a user account, go to "Security > User Account" and click the "Edit" icon ( ) for the user in question. You will be taken to the "Edit User Account" page.



Deactivating a user account

Choose "Inactive" from the "Account Status" dropdown and save. The user account record still exists, but the user can no longer log on or access restricted content.

To reactivate the user, simply edit the user account and choose "Active" from the "Account Status" dropdown, then save.

Deleting a User Account

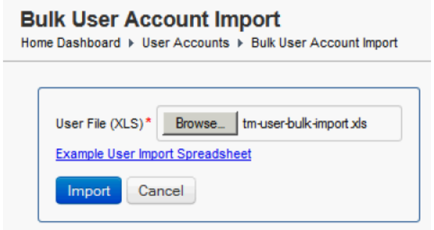
To delete a user account, go to "Security > User Account", locate the user account and click the "Delete" icon ( ). Since this operation will permanently remove the user record, you will be asked to confirm your decision.

Note that you cannot delete the user you are currently logged on as, nor can the system-generated account "Root Administrator" be deleted.

Even after a user account has been deleted, the changes the user made to the system are still visible in the audit log (see _sec_trackchanges below).

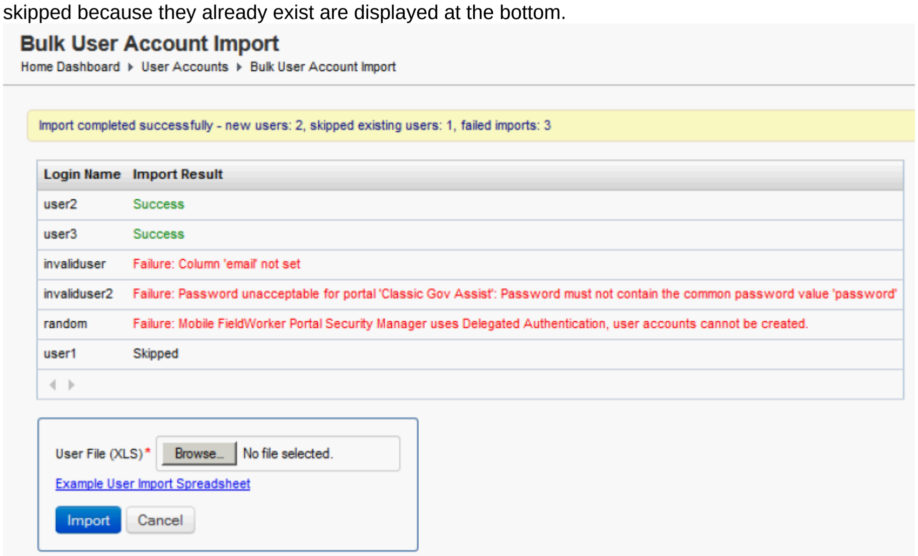
User Account Bulk Import

To make it easier to create a set of users, Transaction Manager allows you to bulk import user accounts defined in an Excel spreadsheet.For security reasons, there are some limitiations: Only local users can be created; users can be associated with spaces used to host forms, but users cannot be associated with other spaces such as the management console. Existing user accounts (with the same login name) are left untouched by the import.
To access user bulk import, go to "Security > User Account" and click "Bulk Import" to access the "Bulk User Account Import" page.



Uploading a set of users defined in a spreadsheet

The XLS file you upload must follow a certain format (specifically the header column names need to be correct and are expected to be in a certain order). You can download an example spreadsheet by clicking on the example file link.
Once you have a suitable XLS file containing the user accounts you want to import, upload the file via the "User File XLS" field and click "Import".
The import file will be processed, and the import results will be displayed (see below for an example). In the detail table, users that were imported successfully are displayed at the top, followed by users for which an error occurred (e.g. mandatory column not set). Finally, user accounts that were skipped because they already exist are displayed at the bottom.



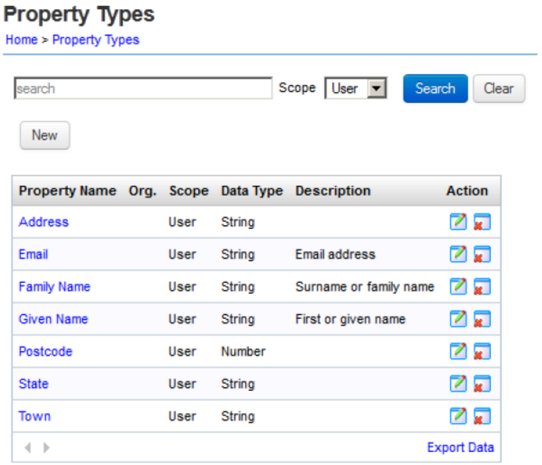
User bulk import results

User Profile Configuration

Users can set up one or more profiles that can be used to pre-populate forms if the user space supports this. Each profile has the same fields, and each field corresponds to a user property (for more on properties and how to set them up, see Organization Configuration (Manager v5.0)#_sec_propertytypes). When you create a user via the management console, a default profile is created automatically.
Examples for user profile fields are first name, last name and phone number. Transaction Manager allows you to define whichever fields you need and map them to locations in a form data configuration so they can be pre-populated into the form.

Viewing the User Profile Properties

To view all the user profile fields, go to "Forms > Property Types" and choose "User" from the "Scope" dropdown (you should see a list similar to the one in the screenshot below).
You can filter this list using a substring of the property name and the data type (see Organization Configuration (Manager v5.0) #_sec_datatypesforaproperty for the list of possible data types), and export the search results to Excel by clicking the "Export Data" link below the table.

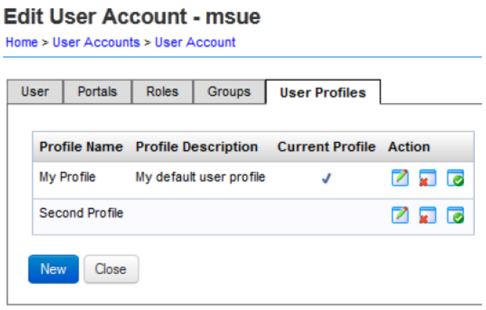


Viewing user property types

This page allows you to add user properties, edit existing user properties ( ) and delete user properties ( ). For more information on how to do this, see Organization Configuration (Manager v5.0)#_sec_propertytypes.

Viewing a User's Profile Data

To view a user's profiles, go to "Security > User Accounts" and click the "Edit" icon ( ) next to the user. You will be directed to the "Edit User Account" page. Switch to the "User Profiles" tab to see all profiles that the user has set up.

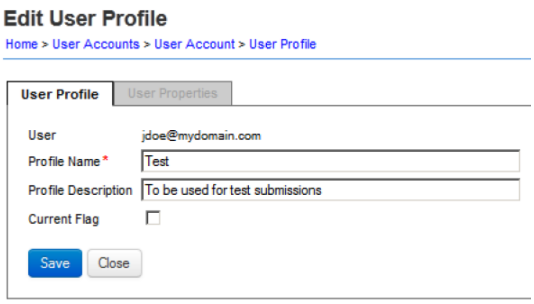


The list of profiles set up for a user

This page allows you to create a new profile and edit ( ) or delete ( ) existing user profiles. You can also promote a profile to the default (current) profile ( ).

Adding a Profile to a User

To add a profile to a user, view the user's profiles as described above and click "New". The screenshot below shows an example where the fields have been filled in.

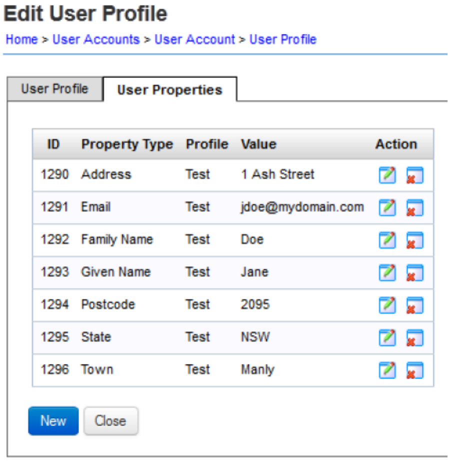


Creating a user profile

Enter a profile name and a description.
The "Current Flag" indicates whether the profile will be used as the default when rendering forms.
Save the profile. The "User Properties" tab should now be enabled allowing you to configure the new user profile.

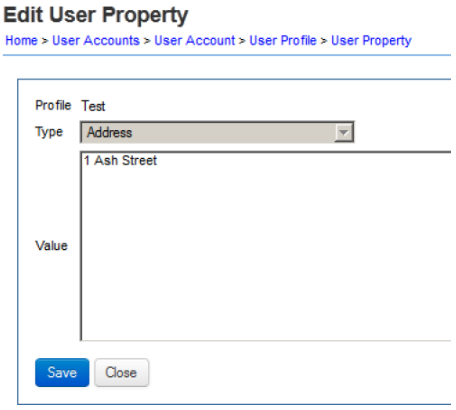
Modifying a User's Profile Data

To modify a user's profile data, view the user's profiles as described in section 2.6.2. Click the "Edit" icon ( ) next to the profile you want to modify. You will be directed to the "Edit User Profile" page. Switch to the "User Properties" tab. You will see all user properties and their current values for this user.



Modifying a user profile

To modify the value of a profile property, click the "Edit" icon ( ) next to it. You will be directed to the "Edit User Property" page, where you can enter the new value and save.



Editing a user property

Generally, you should avoid deleting user properties as this will cause the user profile to be inconsistent with the other profile (since it will be missing the property). If you want to unset a property, simply leave the value field blank.

Deleting a User Profile

Be very careful when deleting user profiles so you do not accidentally remove data from a non-test user.

To delete a user profile, view the list of profiles as described in section 2.6.2. Find the profile you want to delete, and then click the "Delete" icon ( ). You will be asked to confirm this action before the user profile is irrevocably deleted.

Default Role and Administrator

When Transaction Manager starts up, it will check to see if any roles or user accounts exist. If not, it will create an "administrator" role with all available permissions and an administrator that belongs to that role. The credentials for the administrator are "administrator" with an initial password of "password". You will be asked to change the password to a different value when you first log on.
To avoid the possibility of accidentally locking up the system, neither the auto-generated role nor the administrator can be deleted.

Tracking Changes

When a user makes changes to important Transaction Manager entities (e.g. creates a form, changes an organization property, deletes delivery details), these changes are recorded and the logs made available for viewing via the Transaction Manager audit log. This allows you to find out which user was responsible for each change, as well as to view the change history for a particular entity.

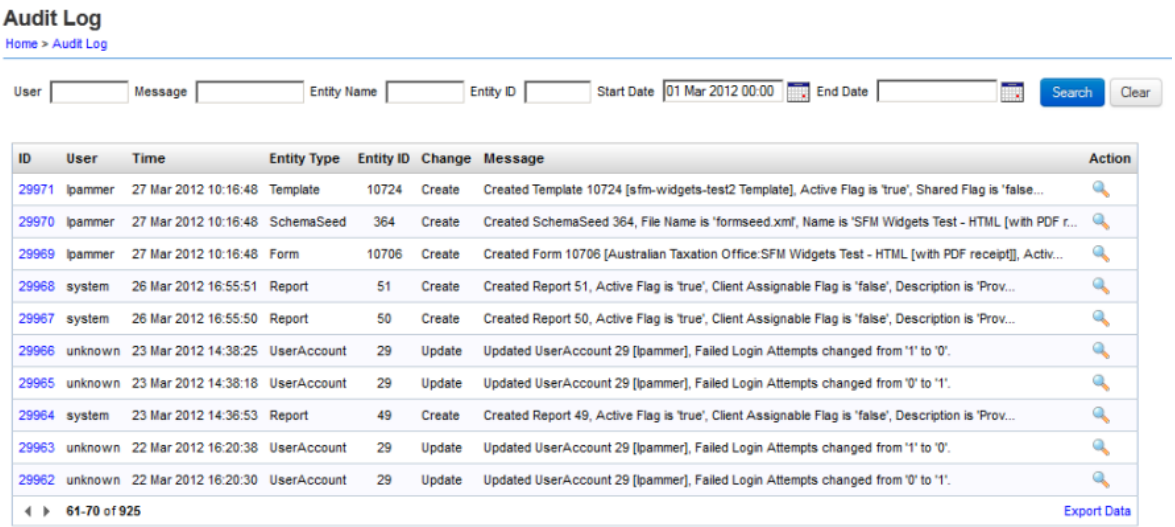
Data Logged for an Audit Log Entry

Each audit log entry contains the same fields; depending on the nature of the change logged, the fields have to be interpreted in different contexts, though.
The fields contained in an audit log entry are as follows:

1. Entity
   The type of object affected by the change, for example "Form" or "UserDetails".
2. Entity ID
   The ID of the entity that was affected by the change. This ID is unique within the entity type (e.g. two different forms always have different IDs, but a form may have the same ID as an organization).
3. User
   This field identifies the actor who made the change. The actor can be an administrator (most common case), but also a space user or even a Transaction Manager job. In any case, the actor's login name (for administrators and space users) or name (for jobs) is logged.
4. Time
   The date and time when the change was made.
5. Message
   A string describing what was changed. For a "create" or "edit" operation, this string will contain all attributes that have changed and their old and new values; for a "delete" operation, the message will only say which object has been deleted.

Browsing the Audit Log

To view the audit log, go to "Security > Audit Log".



ID	User	Time	Entity Type	Entity ID	Change	Message	Action
29971	lpammer	27 Mar 2012 10:16:48	Template	10724	Create	Created Template 10724 [sfm-widgets-test2 Template], Active Flag is 'true', Shared Flag is 'false...	
29970	lpammer	27 Mar 2012 10:16:48	SchemaSeed	364	Create	Created SchemaSeed 364, File Name is 'formseed.xml', Name is 'SFM Widgets Test - HTML [with PDF r...	
29969	lpammer	27 Mar 2012 10:16:48	Form	10706	Create	Created Form 10706 [Australian Taxation Office:SFM Widgets Test - HTML [with PDF receipt]], Activ...	
29968	system	26 Mar 2012 16:55:51	Report	51	Create	Created Report 51, Active Flag is 'true', Client Assignable Flag is 'false', Description is 'Prov...	
29967	system	26 Mar 2012 16:55:50	Report	50	Create	Created Report 50, Active Flag is 'true', Client Assignable Flag is 'false', Description is 'Prov...	
29966	unknown	23 Mar 2012 14:38:25	UserAccount	29	Update	Updated UserAccount 29 [lpammer], Failed Login Attempts changed from '1' to '0'.	
29965	unknown	23 Mar 2012 14:38:18	UserAccount	29	Update	Updated UserAccount 29 [lpammer], Failed Login Attempts changed from '0' to '1'.	
29964	system	23 Mar 2012 14:36:53	Report	49	Create	Created Report 49, Active Flag is 'true', Client Assignable Flag is 'false', Description is 'Prov...	
29963	unknown	22 Mar 2012 16:20:38	UserAccount	29	Update	Updated UserAccount 29 [lpammer], Failed Login Attempts changed from '1' to '0'.	
29962	unknown	22 Mar 2012 16:20:30	UserAccount	29	Update	Updated UserAccount 29 [lpammer], Failed Login Attempts changed from '0' to '1'.	

61-70 of 925 Export Data

The audit log

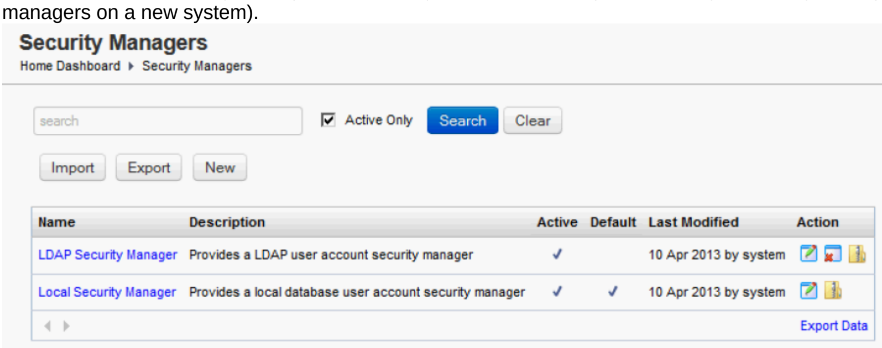
You can filter the list using a substring of the "User" and "Message" attributes, the entity name, the entity ID and the date and time at which the change was made. For an explanation of these fields see the previous section.
Examples:
To view all the changes made to the form property with ID 1000, enter "1000" into the "Entity ID" field and click "Search". This gives you all changes made to objects with an ID of 1000. Should there be several objects, you can narrow the list down by entering "FormProperty" into the "Entity Name" field.
To view all changes made by the administrator with the login name "jsmith", enter "jsmith" into the "User" field and search. You will get up to 1000 changes, with the most recent changes shown first by default. If you want to filter the results, you could enter a start and end date to see only changes made within that date range.
You can also export the search results to Excel by clicking the "Export Data" link below the table.

Security Managers

A security manager contains configuration around user authentication, self-registration and session management. Each space is associated with a security manager (though security managers can be shared between spaces).
On a new installation, the default security manager supports user authentication against the TM user base. In addition, TM provides templates for other security manager for LDAP, ADFS and SSO authentication. You can create new security managers based on a template and configure existing security managers to match your needs. You can also export security managers and import them to other Transaction Manager instances.
Each security manager contains one or more authentication providers, which perform the actual authentication of a user. Multiple authentication providers can be chained (e.g. authentication first against the TM database, then several external LDAP directories). If an authentication provider can authenticate a user successfully, the user is deemed to have logged on successfully, and subsequent authentication providers are not processed.

Viewing the Set of Security Managers

To view the set of security managers defined in your TM instance, go to "Security > Security Managers" (the screenshot below shows the set of security managers on a new system).

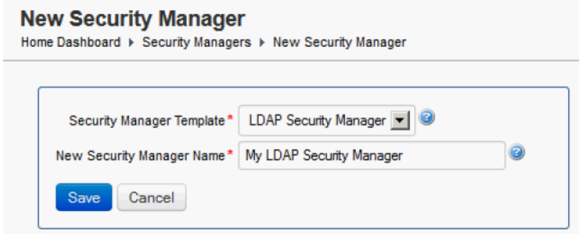


The list of security managers

You can create new security managers, edit and delete existing security managers, export a single or all security managers and import a security manager archive.
Note: There always has to be a default security manager. Therefore, you cannot delete the default security manager.
You can also see when a security manager was last modified and by whom in the "Last Modified" column.

Creating a Security Manager

To create a security manager, go to "Security > Security Managers" and click the "New" button. You will be directed to the "New Security Manager" page.

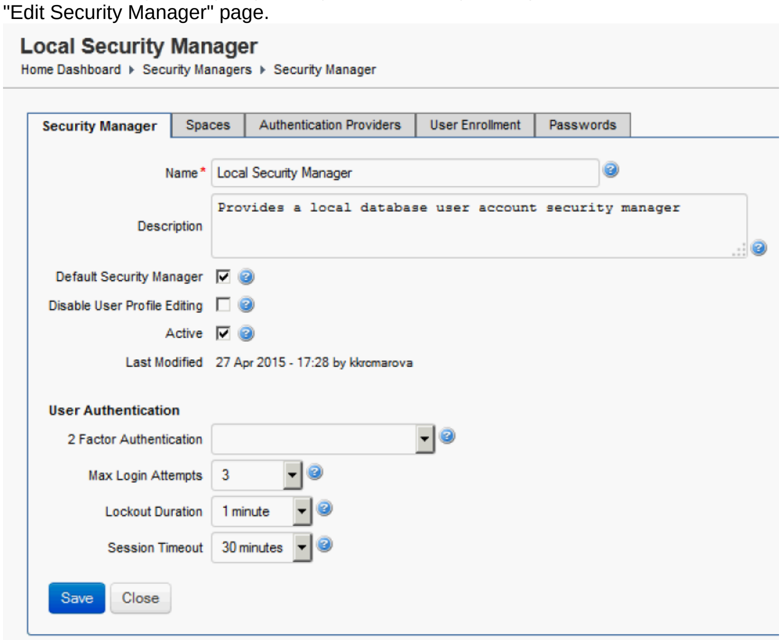


Creating a security manager

Select a template to base your security manager on, and enter a unique name for the security manager.
Save your changes. The security manager will be created with its initial settings based on the selected template. The following section explains the configuration options that are available.

Configuring a Security Manager

To edit a security manager, go to "System > Security Managers" and click the "Edit" icon ( ) next to the security manager. You will be directed to the "Edit Security Manager" page.



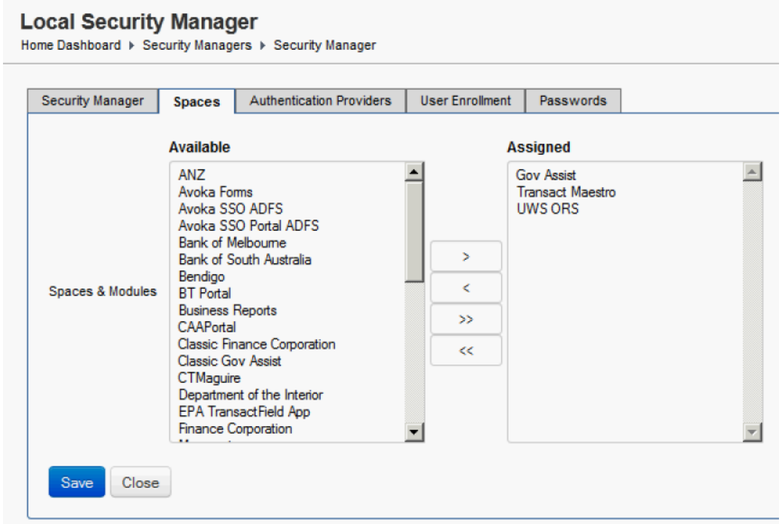
Editing a security manager

Basic Security Manager Settings

The "Security Manager" tab contains some basic settings:
The "Name" field contains the unique name that identifies the security manager.
The "Description" field is optional and is used for display purposes only.
The "Default Security Manager" checkbox should be ticked for the default security manager for your TM system. It is used whenever a space does not explicitly specify a security manager. Make sure that there is always a default security manager.
The "Active" checkbox can be used to deactivate (rather than delete) a security manager. It is then hidden from the default view, but can later be reactivated. Note that deactivating a security manager also reconfigures any spaces that were explicitly set to use this security manager to instead use the system default security manager instead.
The "Disable User Profile Editing" checkbox can be ticked if you would like to prevent users from editing their profile data. This setting applies to all spaces using the security manager.
The "User Authentication" section provides settings related to user authentication and lockout.
The "2 Factor Authentication" dropdown allows you to specify a service that will be used by this security manager for two-factor authentication. See the Transact Integration Guide, chapter "Access Control" for details on two-factor authentication. Do not change this setting unless you have read the section in the Transact Integration Guide as you could lock administrators (including yourself) or users out of the system.
The "Max Login Attempts" field holds the number of successive failed login attempts a user is granted before their user account is locked. Every time a user logs in successfully, the failed login attempts counter is reset to zero.
The "Lockout Duration" field defines what happens when the user exceeds the maximum number of failed login attempts. You can choose to lock the user account for a limited duration (after which the account reactivates automatically and the number of failed login attempts is reset) or forever (account can only be reactivated by an administrator). While an account is locked, login attempts are ignored and do not increment the failed login attempt counter.
The "Session Timeout" field holds the time after which idle sessions shall be discarded and the user will be forced to reauthenticate.

Spaces Associated with the Security Manager

The "Spaces" tab shows the set of spaces that are explicitly configured to use this security manager.

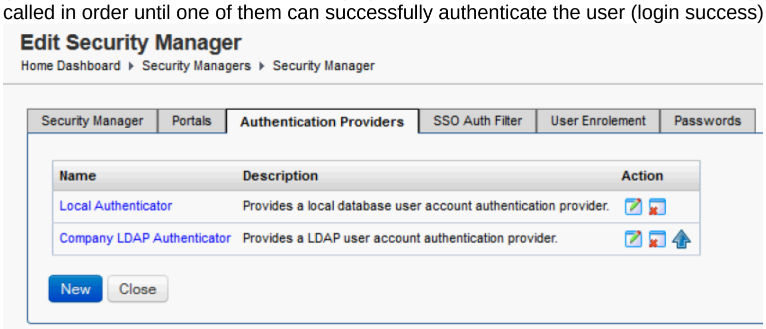


Assigning spaces to a security manager

Note: The default security manager is associated with all spaces that do not explicitly define the security manager. This is not reflected in the "Spaces" tab, which shows only explicit associations between spaces and security managers.

Authentication Providers

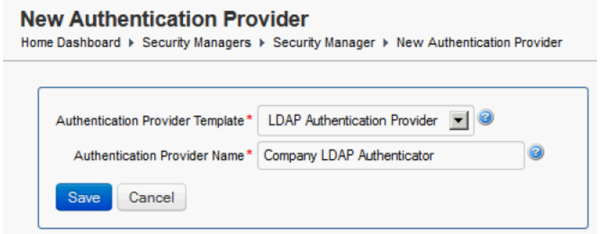
The "Authentication Providers" tab allows you to define the authentication providers that the security manager will use to attempt to authenticate a user (the screenshot below shows an example). You can add one or more authentication providers; if multiple authentication providers are defined, they are called in order until one of them can successfully authenticate the user (login success) or all of them fail to authenticate the user (login failure).



The authentication providers configured for a security manager

In case of login failure, the login failure attempts counter is incremented and user account locked if required.
If login succeeds, the login failure attempts counter is reset to zero and a user session is created with the set of roles associated with the user (if any).
You can add and delete authentication providers. Please ensure there is always at least one authentication provider as the security manager cannot function otherwise.

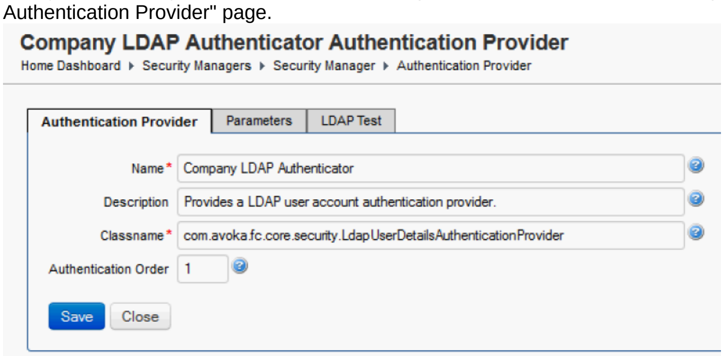
You can reorder the authentication providers by using the "Move Up" icon ( ) next to the authentication provider.
To add an authentication provider, click the "New" button. Fill in the details for the new authentication provider on the "New Authentication Provider" page.



Creating an authentication provider

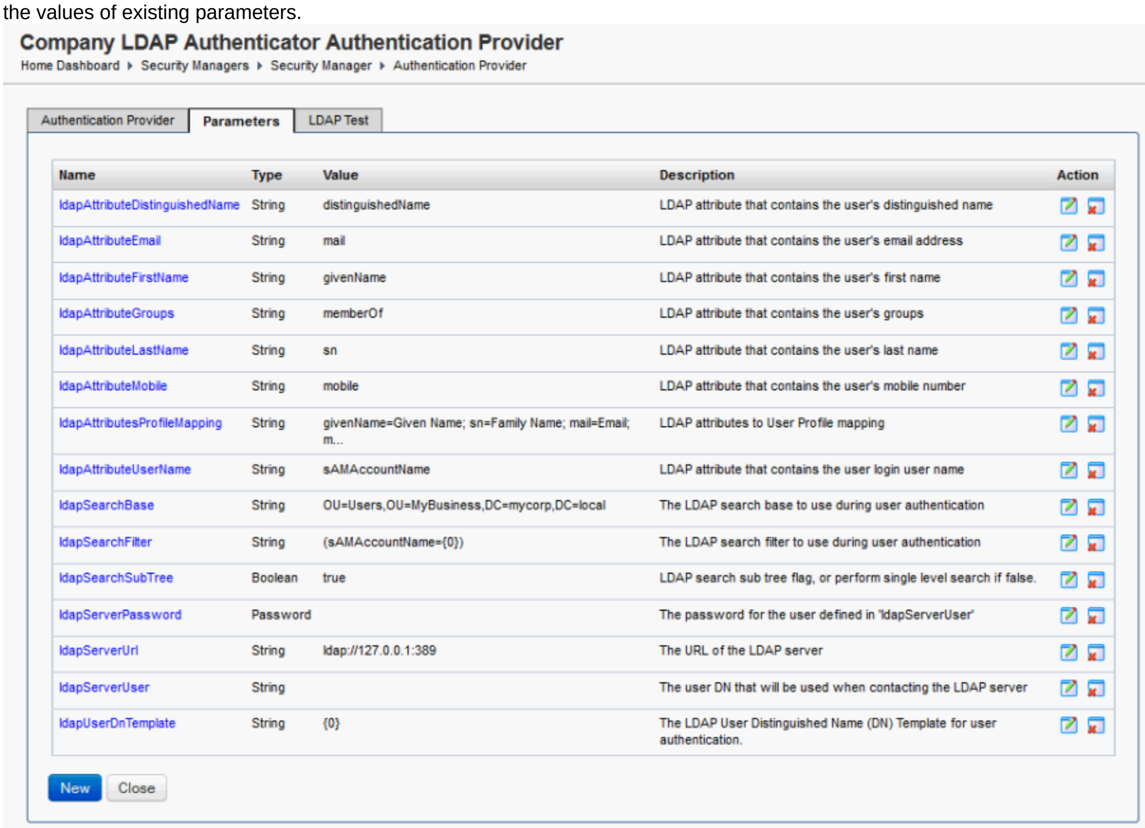
Choose one of the available templates from the "Authentication Provider Template" dropdown. The template defines the initial settings for your authentication provider. Note that the templates are built into TM and do not correspond to the set of authentication providers defined on your system.
Enter the name of the new authentication provider into the "Authentication Provider Name" field and save your changes. A new authentication provider based on the template will be created and added as the last authentication provider for the current security manager.

Now you can configure the new authentication provider. Click on the "Edit" icon ( ) next to it to get to the "<Authentication provider name> Authentication Provider" page.



Editing an authentication provider

You can change the name and description on the "Authentication Provider" tab. Do not change the "Classname" field, and use the arrows on the "Authentication Providers" tab of the parent page to change the ordering instead of changing the "Authentication Order" field.
The "Parameters" tab allows you to configure the settings exposed by the authentication provider. The set of parameters is specific to the authentication provider template. Generally, you should not add or delete parameters (except possibly for Groovy-based authentication parameters) but instead adjust the values of existing parameters.



Name	Type	Value	Description	Action
ldapAttributeDistinguishedName	String	distinguishedName	LDAP attribute that contains the user's distinguished name	
ldapAttributeEmail	String	mail	LDAP attribute that contains the user's email address	
ldapAttributeFirstName	String	givenName	LDAP attribute that contains the user's first name	
ldapAttributeGroups	String	memberOf	LDAP attribute that contains the user's groups	
ldapAttributeLastName	String	sn	LDAP attribute that contains the user's last name	
ldapAttributeMobile	String	mobile	LDAP attribute that contains the user's mobile number	
ldapAttributesProfileMapping	String	givenName=Given Name; sn=Family Name; mail=Email; m...	LDAP attributes to User Profile mapping	
ldapAttributeUserName	String	sAMAccountName	LDAP attribute that contains the user login user name	
ldapSearchBase	String	OU=Users,OU=MyBusiness,DC=mycorp,DC=local	The LDAP search base to use during user authentication	
ldapSearchFilter	String	(sAMAccountName={0})	The LDAP search filter to use during user authentication	
ldapSearchSubTree	Boolean	true	LDAP search sub tree flag, or perform single level search if false.	
ldapServerPassword	Password		The password for the user defined in 'ldapServerUser'	
ldapServerUrl	String	ldap://127.0.0.1:389	The URL of the LDAP server	
ldapServerUser	String		The user DN that will be used when contacting the LDAP server	
ldapUserDnTemplate	String	{0}	The LDAP User Distinguished Name (DN) Template for user authentication.	

The parameters for an LDAP authentication provider

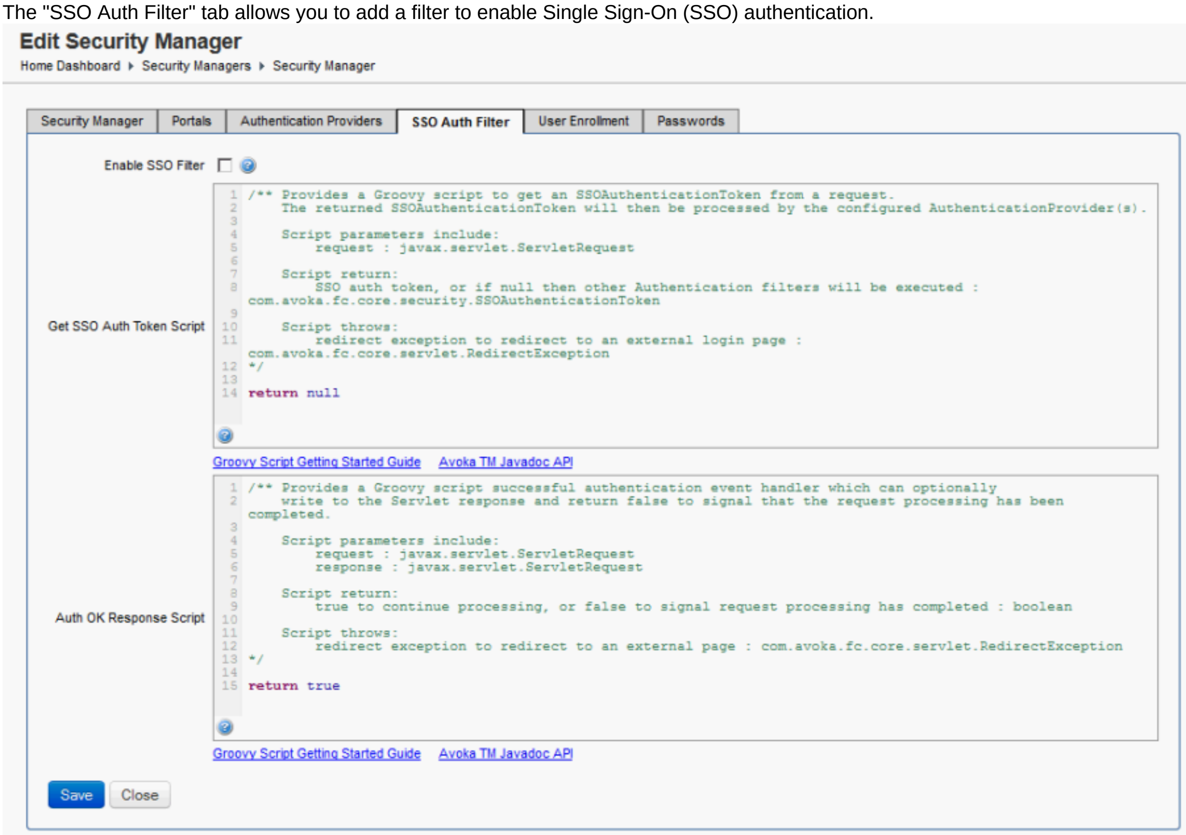
Finally, you can test LDAP based authentication providers using the "LDAP Test" panel.

Testing authentication for an LDAP authentication provider

Choose a space associated with the current security manager.
For testing authentication, enter a user name and password and click "Authenticate User".The authentication provider will attempt to authenticate the user using the current parameter values. The results will be displayed in the "LDAP Results" field.
To do a lookup in the LDAP directory, choose a space, enter a user name and optionally tick "All Attributes" to return the full set of LDAP attributes. Click "Lookup User".The authentication provider will attempt to look up the user in LDAP and retrieve the relevant attributes. The results will be displayed in the "LDAP Results" field.
To add a user to a space, choose a space, enter the user name and click "Add User To Portal". The user must not yet exist in TM.The authentication provider will attempt to load the user details from LDAP and create a new LDAP-based user account for the user, associated with the selected space. The results will be displayed in the "LDAP Results" field.

Adding a Single Sign-On Filter

The "SSO Auth Filter" tab allows you to add a filter to enable Single Sign-On (SSO) authentication.



Configuring SSO for a security manager

To enable the SSO filter, tick the "Enable SSO Filter" checkbox.
Customize the filter behavior by modifying the two Groovy scripts:
The "Get SSO Auth Token Script" field should be modified to retrieve the SSO token from the request. The "Auth OK Response Script" field can be modified to customize the behavior on successful authentication. Please refer to the TM Groovy services guide ("System > Groovy Services Guide" for more information on these scripts.
Save your changes. The effects will be applied immediately – be sure to test any changes thoroughly.

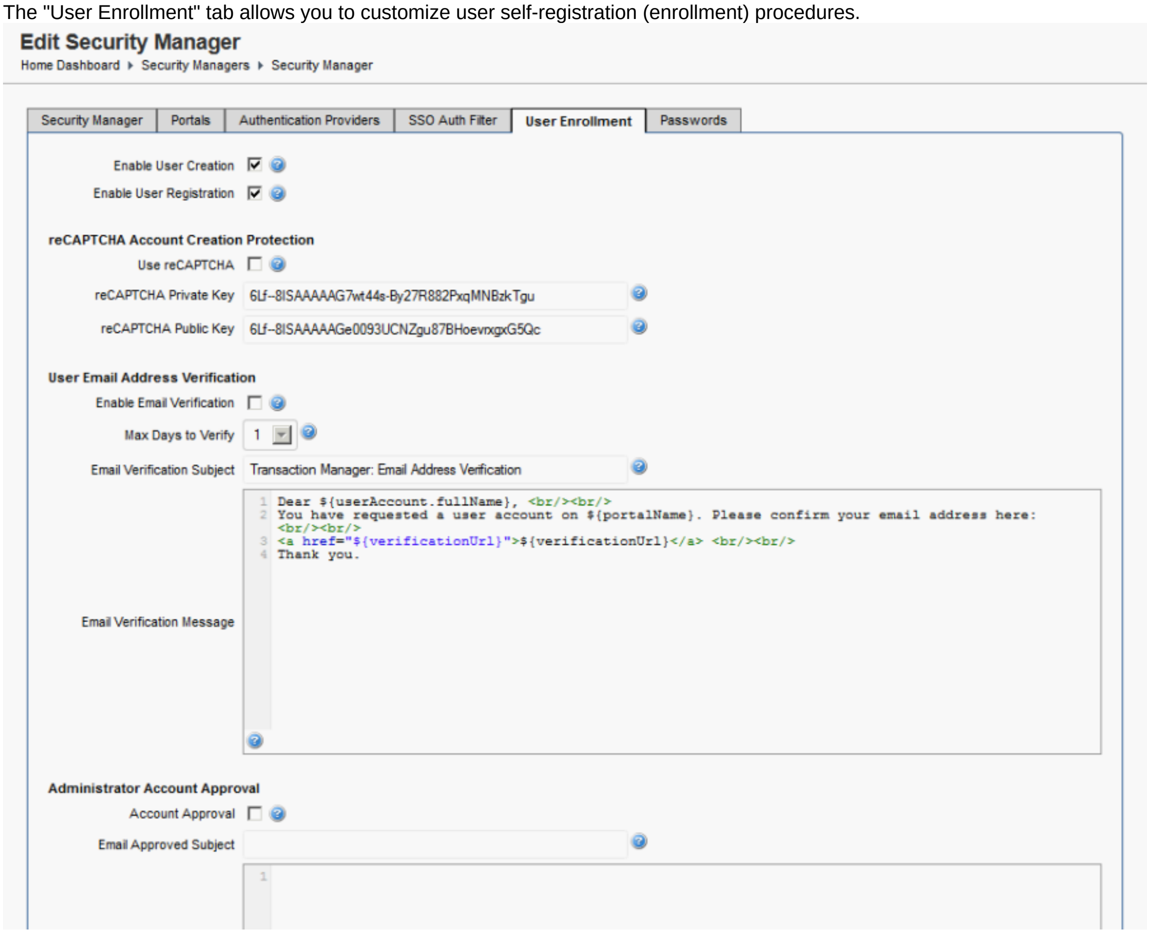
Customizing User Enrollment

User accounts are generally created in one of two ways: An administrator creates the user account via the management console, or the user can apply for an account on a space. User self-registration is a convenient feature but it might not be supported for all spaces (for example, the management console does not offer self-registration for security reasons).
Users who are self-registering can be asked to pass email verification (by clicking on a link in an email sent to their nominated email address). In addition, it is possible to enable pending user enrollment, which means that new accounts must be approved by an administrator.
The general steps for a user to register an account using pending user enrollment are as follows (many of the steps are customizable as discussed later in this section):

1. User registers an account with a Transaction Manager space
   If pending user enrollment is enabled, the user will not be logged on right away. Instead, they are shown a page notifying them that their account will need to be approved.
2. Transaction Manager sends an email to all members of the group "Receive User Account Creation Alerts", alerting them of the account creation request.

3. An administrator logs on to the management console and reviews the new user account. They can then choose to either activate or reject the user.
4. Once the user account has been updated, Transaction Manager sends an appropriate email to the user. If the user was approved, they can now log on to the space they registered the account on.

The "User Enrollment" tab allows you to customize user self-registration (enrollment) procedures.



Customizing user enrollment

The "Enable User Creation" checkbox controls whether new users can be created at all for spaces using the current security manager (whether by self-registration or creation by an administrator).
The "Enable User Registration" checkbox controls whether users can self-register new accounts (note: depending on further configuration, users may be asked to pass email verification and/or require administrator approval prior to being able to use their account).
The "reCAPTCHA Account Creation Protection" section allows you to configure a reCAPTCHA challenge (http://www.google.com/recaptcha) that users will need to pass during self-registration.
If you would like to enable reCAPTCHA, tick the "Use reCAPTCHA" checkbox. If you have your own account with reCAPTCHA, enter the public and private reCAPTCHA keys into the corresponding fields.
The "User Email Address Verification" section allows you to configure email verification.

You can use the "Enable Email Verification" checkbox to turn email verification on or off.

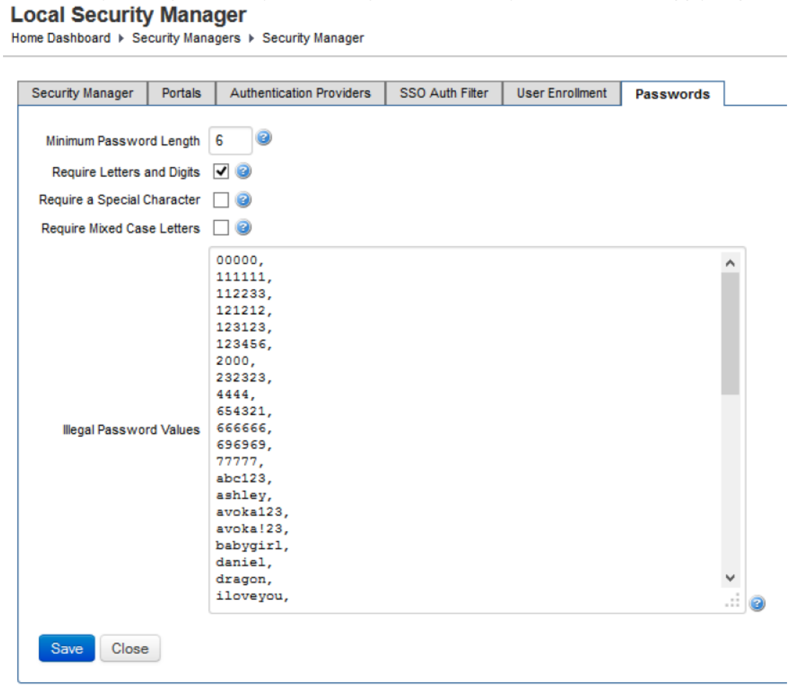
The "Max Days to Verify" dropdown lets you specify the time in days that users have to complete email verification. If a user does not verify their email address in time, their account may be purged automatically (this avoids abandoned accounts polluting the system).
You can also customize the email sent to the user during email verification by editing the "Email Verification Subject" and "Email Verification Message". Note: Ensure the email always contains the URL the user needs to access to complete email verification (the placeholder is "${verificationUrl}").
The "Administrator Account Approval" section allows you to configure pending user enrollment as described at the start of this section. To turn it on, tick the "Account Approval" checkbox.
If you would like TM to send email to administrators belonging to the "Receive User Account Creation Alerts" when a user self-registers during pending user enrollment, tick the "Notify Administrators on Change" checkbox and enter an email subject and message into the "Email Pending Subject" and "Email Pending Message" fields.

If you would like TM to send email to users who have registered an account during pending user enrollment and whose account has been either approved or rejected by an administrator, tick the "Notify User on Change" checkbox and enter email subjects and messages into the "Email Approved Subject", "Email Approved Message", "Email Rejected Subject" and "Email Rejected Message" fields.

Once you save your changes, they will take effect immediately but will not affect user enrollment that is already in progress.

Password Security

The security manager allows you to configure the criteria you would like to apply to passwords chosen by your users on the "Passwords" tab.



Configuring password strength

Tick the "Require Letters and Digits" checkbox to require passwords to require at least one letter and one digit character.

Tick the "Require a Special Character" checkbox to require at least one non-alphanumeric character (e.g. !@#$%) in the password.

Tick the "Require Mixed Case Letters" checkbox to require at least one upper case and one lower case character in the password.

Set the "Minimum Password Length" field to the minimum number of characters you want to allow for passwords.

Add any passwords you want to specifically forbid to the "Illegal Password Values" field (comma-separated).

Once you save your changes, they will apply to new passwords. Passwords currently existing in the system will not be affected.

Permissions (Manager v5.0)

The following is the list of permissions in the TM management console. As a general note, permissions ending in "View." give read-only access to entities, permissions ending in "Edit" allow modification of an entity and permissions ending in "Remove" allow deletion of entities. These permissions are separate in the sense that "Edit" and "Remove" do not include or permit "View" access, but rather "View" is needed in order for "Edit" or "Remove" permissions to apply. So for example, if you would like a role to include edit access to document types, you would add the permissions "Document Type View" as well as "Document Type Edit" to the account.

The list below identifies and describes all of the permissions used in the TM management console.

- Admin Directory - This is a special permission that is required to access the management console. If a user does not have this permission, they will not be able to use the management console.

- Audit View - This permission provides access to the system audit log and user login history.

- Collaboration Job View - This permission provides read-only access to collaboration job information, including viewing details about a collaboration job and its steps, actions and events.

- Collaboration Job Edit - This permission allows users to manipulate collaboration jobs (e.g. pausing/triggering job processing, editing job properties and action details).

- Collaboration Job Remove - This permission allows users to delete collaboration jobs. Note that only canceled jobs can be deleted.

- Core Service Edit - This permission is needed to edit advanced service settings and some job-related services.

- Data Retention View - This permission allows users to view global data retention settings.

- Data Retention Edit - This permission allows users to modify and apply global data retention settings.
- Db Version Update View - This permission allows users to view the database version history (listing automatic changes made to the database schema during upgrades).

- Delivery Details View - This permission allows users to view the set of delivery channels (part of the organization configuration) and manually trigger delivery actions for submissions.

- Delivery Details Edit - This permission allows users to create and edit delivery channels (part of the organization configuration). It is also needed to delete delivery channels.

- Delivery Details Remove - This permission is currently unused. This will be addressed in a future TM version. For now, please use "Delivery Details Edit".

- Deployment Property View - This permission allows users to view the list of system deployment properties and their values ("System > Deployment Properties").

- Deployment Property Edit - This permission allows users to edit deployment property values and access and modify email settings ("System > Email Settings").

- Document Type View - This permission allows users to view the list of document types and details about them ("Forms > Document Types").

- Document Type Edit - This permission allows users to create and edit document types ("Forms > Document Types").

- Document Type Remove - This permission allows users to delete document types ("Forms > Document Types").

- Email Delivery Acknowledge - This permission is related to delivery of type "Email Secure", where administrators receive an email notifying them of new submissions, asking them to acknowledge delivery by accessing a page on TM and confirming. This permission controls access to the page in question and should, therefore, be assigned to any administrators involved in processing secure email delivery.

- Email Queue View - This permission allows users to access the system email queue ("System > Email Queue"). It does not include viewing details about each email entry ("Email Queue Edit" is required for this).

- Email Queue Edit - This permission allows users to create, modify and access details about email queue items ("System > Email Queue").

- Error Log View - This permission provides access to the system error log ("System > Error Log") and error details. It also enables links to error log entries on other pages.

- Event Log View - This permission provides access to the system event log ("System > Event Log") and event details. It also enables links to event log entries on other pages.

- Export Application Package - This permission allows users to export application packages (bundles of forms and services) for organizations they have access to.

- Export Form - This permission allows users to export a form and all its versions ("Forms > Forms").

- Export Form Version - This permission allows users to export a single form version.

- Export Groups - This permission allows users to export one or more groups ("Security > Groups").

- Export Metadata - This permission allows users to export the set of global metadata tags ("Forms > Metadata Tag Definitions").

- Export Organization - This permission allows users to export an organization ("Forms > Organizations").

- Export Portal - This permission allows users to export a form space ("Forms > Form Spaces").

- Export Roles - This permission allows users to export one or more roles ("Security > Roles").

- Export Services - This permission allows users to export one or more service definitions ("System > Service Definitions", "System > Job Services").

- Export Security Manager - This permission allows users to export one or more security managers ("Security > Security Managers").
- Finder View - This permission allows users to view the list of finder categories ("Forms > Finder Categories").
- Finder Edit - This permission allows users to create, modify and view details of finder categories ("Forms > Finder Categories").
- Finder Remove - This permission allows users to delete finder categories ("Forms > Finder Categories").
- Form View - This permission allows access to the list of forms ("Forms > Forms") and a read-only view of the majority of form configuration. It also enables links to forms on other pages. Note that if you need to view form version or form data configuration details, you need the "Form Edit" permission as the "Form View" permission does not include this.
- Form Edit - This permission allows users to view all details about a form ("Forms > Forms") as well as create and edit forms. It includes editing form versions and form data configuration.
- Form Service Edit - This permission is a less powerful version of "Service Edit", which allows administrators to manage service definitions related to form development.
- Form Test - This permission exposes links to access forms on a space for testing. It also allows users to access the PDF receipt test harness (link in the "Form URLs" section of the form edit page).
- Form Data Retention - This permission allows users to edit form specific data retention settings (section headed "Transaction Form Data Retention" on the "Details" tab of the form edit page).
- Form Remove - This permission allows users to delete forms, form versions, form promotion entries and form data extract mappings.
- Groovy Console - This permission allows access to the Groovy console ("System > Groovy Script Console").
- Groovy Log View - This permission allows access to the Groovy log ("System > Groovy Service Log").
- Group View - This permission allows users to view the list of groups ("Security > Groups") though it does not allow access to detail information (use "Group Edit" if needed).
- Group Edit - This permission allows users to create, edit and view details about groups ("Security > Groups").
- Group Remove - This permission allows users to delete groups ("Security > Groups").
- Group Assign - This permission allows users to assign users to a group. It is also necessary to access the "Forms" panel on the group edit page.
- Import Action View - This permission allows access to the import log ("System > Import Log"). It does not include read only access to import action details.
- Import Action Edit - This permission allows users to edit and view details about import actions ("System > Import Log"). It is also currently necessary to import form and space archives exported from another TM server, along with the "Import ..." permission corresponding to the entity to be imported.
- Import Action Remove - This permission allows users to delete import actions ("System > Import Log"). Only completed import actions are eligible for deletion.
- Import Application Package - This permission allows users to import application packages (bundles of forms and services) for organizations they have access to.
- Import Composer Form - This permission allows users to import forms published by Transact Composer and view details about past imports ("Forms > Composer Package Imports").
- Import Form - This permission allows users to import form version archives created by exporting a form version from another TM server. Note that to import form version archives successfully, users also currently need the "Import Action Edit" permission.
- Import Groups - This permission allows users to import group archives created by exporting one or more groups from another TM server.
- Import Metadata - This permission allows users to import metadata tag archives created by exporting one or more metadata tags from another TM server. In addition, this permission is needed to upload files with values for metadata tags of types "List" and "List Hierarchy".
- Import Organization - This permission allows users to import an organization archive (containing organization configuration but no forms) exported from another TM server. Note that only administrators with access to more than one organization will be able to see the import button
.
- Import Portal - This permission allows users to import a space archive (containing space configuration but not the backing WAR file) exported from another TM server as well as space WAR files. Note that to import space archives successfully, users also currently need the "Import Action Edit" permission.
- Import Roles - This permission allows users to import role archives (containing roles and associated permissions) exported from another TM server.
- Import Security Manager - This permission allows users to import security manager archives exported from another TM server.
- Import Services - This permission allows users to import service archives (containing service definitions and their service connections) exported from another TM server.
- Licensing Edit - This permission allows users to edit system licensing information and configure how licensing report data will be published to Avoka.
- Metadata Tag View - This permission allows users to view metadata tags and their configuration details ("Forms > Metadata Tag Definitions").
- Metadata Tag Edit - This permission allows users to create and edit metadata tags ("Forms > Metadata Tag Definitions").
- Metadata Tag Remove - This permission allows users to delete metadata tags ("Forms > Metadata Tag Definitions").
- Metadata Value View - This permission allows users to view values configured for metadata tags (on the organization or form version level).

- Metadata Value Edit - This permission allows users to create and edit values configured for metadata tags (on the organization or form version level).
- Metadata Value Remove - This permission allows users to delete values configured for metadata tags (on the organization or form version level).
- Notification View - This permission allows users to view the list of notifications set up to appear on TM spaces ("System > Notification Messages").
- Notification Edit - This permission allows users to create and edit notifications set up to appear on TM spaces ("System > Notification Messages").
- Notification Remove - This permission allows users to delete notifications set up to appear on TM spaces ("System > Notification Messages").
- Offline Sync Log View - This permission allows users to view the T.Field related offline sync and client logs ("System > T.Field Device Log", "System > T.Field Sync Log").
- Operation Chart View - This permission allows users to view a number of charts set up in TM (e.g. "Operations > Forms Activity", "System > Error Log Trend" and the entries in "Analytics" except "History Analytics").
- Operation Export Report View - This permission is currently unused.
- Operation Form Events View - This permission is currently unused.
- Organization View - This permission allows users to view information about an organization ("Forms > Organizations") and part of its configuration. It also enables links to organizations on other pages.
- Organization Edit - This permission allows users to create and edit organizations ("Forms > Organizations") and associated configuration (categories, application packages.
- Organization Remove - This permission allows users to delete organizations ("Forms > Organizations"). Because deleting organizations will also delete all associated forms and submissions, this is a permission that should be assigned sparingly.
- Organization Copy - This permission allows users to create a copy of an existing organization ("Forms > Organizations").
- Organization Portal Assign - This permission allows users to assign spaces to an organization for form hosting purposes ("Forms > Organizations").
- Organization Payment Edit - This permission allows users to view and modify an organization's payment configuration, including merchant account information for the configured payment gateway.
- Organization Property View - This permission allows users to view the set of organization property values (tab on the organization edit page).
- Organization Property Edit - This permission allows users to edit and delete organization property values (tab on the organization edit page).
- Organization Property Remove - This permission is currently not used (the edit permission also confers deletion privileges). This may be addressed in future versions.
- Organization Storage Edit - This permission allows users to edit organization data storage and retention settings (tab on the organization edit page).
- Organization Transaction Audit Log View - This permission allows users to view organization transaction reports ("Reports > Org. Transaction Log").
- Payment Log View - This permission allows users to view details of payments made by users ("Operations > Payment Transactions").
- Portal View - This permission allows users to view details of spaces ("Forms > Form Spaces", "System > Modules").
- Portal Edit - This permission allows users to edit spaces and all associated configuration ("Forms > Form Spaces", "System > Modules").
- Portal Content Edit - This permission allows users to edit space content (all portal pages and resources).
- Portal Style Edit - This permission allows users to edit some space content related to styling (the index page, childportal.css and all resources of type .jpg).
- Portal Remove - This permission allows users to delete spaces ("Forms > Form Spaces", "System > Modules").
- Promotion View - This permission allows users to view the promotion log ("Forms > Form Version Promotions" and a tab on the form edit page).
- Promotion 1 Development, Promotion 2 Ready For Test, Promotion 3 Test Failed, Promotion 4 Test Passed, Promotion 5 Ready For Production, Promotion 6 Not Ready For Production, Promotion 7 Production - These permissions allow users to promote form versions to the promotion level indicated in the permission ("Forms > Form Version Promotions" and a tab on the form edit page).
- Property Type View - This permission allows users to view property types and their configuration ("Forms > Property Types").
- Property Type Edit - This permission allows users to create and edit property types ("Forms > Property Types").
- Property Type Remove - This permission allows users to delete property types ("Forms > Property Types").
- Report View - This permission allows users to view reports ("Reports > Report Configurations"). It does not give read-only access to report configuration details (use "Report Edit" if needed).
- Report Edit - This permission allows users to create and edit reports ("Reports > Report Configurations").
- Report Remove - This permission allows users to delete reports ("Reports > Report Configurations").
- Report Render - This permission allows users to render reports on-screen ("Reports > Report Configurations". "Reports > System Report Schedules", report schedules tab on the client edit page).
- Report Schedule View - This permission allows users to view report schedules on the organization level (report schedules tab on the client edit page).

- Report Schedule Edit - This permission allows users to create and edit report schedules on the organization level (report schedules tab on the client edit page).
- Report Schedule Remove - This permission allows users to delete report schedules on the organization level (report schedules tab on the client edit page).
- Report Schedule Trigger - This permission allows users to trigger report schedules on the organization level, causing an email to be sent out for the most recent reporting period.
- Request Log View - This permission allows users to view the request log and detailed information about each request ("Operations > Form Requests"). It also provides access to the form rendering metrics ("Operations > Form Rendering Metrics").
- Rest Delivery Service API - This permission allows users to invoke the REST Delivery API provided in TM.
- Rest Delivery Service Invoke - This permission allows users to invoke REST submission delivery by accessing the REST delivery service.
- Rest Service Definitions API - This permission allows users to invoke the REST Service Definitions API provided in TM.
- Rest Transactions API - This permission allows users to invoke the REST Transactions API provided in TM.
- Role View - This permission allows users to view the set of roles ("Security > Roles"). It does not give read-only access to role configuration details (use "Role Edit" if needed).
- Role Edit - This permission allows users to create, edit and view all details for roles ("Security > Roles").
- Role Remove - This permission allows users to delete roles ("Security > Roles").
- Role Assign - This permission allows users to assign user accounts to a role ("Security > Roles", "Security > User Accounts"). The set of roles that a user can assign is further restricted based on whether the user is a global administrator.
- Security Manager View - This permission allows users to view the set of security managers ("Security > Security Managers"). It does not confer read-only access to security manager details (use "Security Manager Edit" if needed).
- Security Manager Edit - This permission allows users to create, edit and view all details of security managers ("Security > Security Managers"), including managing authentication providers.Note: "Service Edit" currently controls access to some content (such as creating security managers and authentication providers and parameters). This may be addressed in a future release.
- Security Manager Remove - This permission allows users to delete security managers ("Security > Security Managers").
- Service View - This permission allows users to view the full set of service definitions ("System > Service Definitions"), service connections ("System > Service Connections"), collaboration job service definitions ("System > Job Services"), server nodes ("System > Server Nodes") and scheduled jobs ("System > Scheduled Jobs"). It does not give read-only access to the configuration details (use "Service Edit" if needed).
- Service Edit - This permission allows users to create, edit and view configuration details of service definitions ("System > Service Definitions"), service connections ("System > Service Connections"), collaboration job service definitions ("System > Job Services"), server nodes ("System > Server Nodes"), scheduled jobs ("System > Scheduled Jobs") and system data retention settings ("System > Data Retention Management"). If you would like a form developer to be able to modify their own form-related services without giving them access to all service types, you can use "Form Service Edit" instead.
- Service Remove - This permission allows users to delete service definitions, service connections, collaboration job service definitions, server nodes and scheduled jobs.
- Submission View - This permission allows users to view details for submissions in TM ("Operations > Form Transactions", "Operations > Transactions History", "Operations > Transaction Delivery", "Operations > Saved Submissions", "Operations > Abandoned Submissions"). It also enables links to submissions on other pages.Note that access to the actual content submitted by users is not granted by this permission. If needed, use "Submission Data View" and "Submission XML Data View".
- Submission Edit - This permission allows users to modify submission statuses and retry delivery.
- Submission Data View - This permission allows users to view non-XML submission content (receipt and attachment content but not the submitted XML) submitted by the user. It also provides access to the set of submission data extract values, including the form submission data page ("Operations > Form Submission Data").As this permission allows access to user-entered data, it may need to be handled carefully.
- Submission XML Data View - This permission allows users to view XML submission content, i.e. the data entered into the form by the user.As this permission allows access to user-entered data, it may need to be handled carefully.
- Submission XML Data Edit - This permission allows users to edit XML submission content and delete submission data extract values.As this permission allows write access to user-entered data, it may need to be handled very carefully.
- Submission Save Edit - This permission allows administrators to open and fill in saved submissions ("Operations > Form Transactions", "Operations > Saved Transactions"). This is useful in support cases where a user asks for help in completing a submission.As this permission allows access to user-entered data, it may need to be handled carefully.
- System Info View - This permission allows users to access some pages that provide information on the TM system and its performance ("System > System Info", "Operations > TM Server Performance", "Operations > LiveCycle Performance").
- System Report Schedule View - This permission allows users to view report schedules for global reports ("Reports > System Report Schedules").
- System Report Schedule Edit - This permission allows users to create and edit report schedules for global reports ("Reports > System Report Schedules").
- System Report Schedule Remove - This permission allows users to delete report schedules for global reports ("Reports > System Report Schedules").Note: Currently TM also requires "Report Schedule Edit" for this to work. This will be addressed in future releases.
- System Report Schedule Trigger - This permission allows users to trigger report schedules for global reports, causing an email to be sent out for the most recent reporting period.

- Task View - This permission allows users to view the list of task submissions ("Operations > Assigned Tasks").
- Task Edit - This permission allows users to create task submissions and edit task-related data ("Operations > Assigned Tasks").
- Transaction Audit Log View - This permission allows users to view system transaction reports ("Reports > Transaction Log").
- User View - This permission allows administrators to view the list of user accounts ("Security > User Accounts"). It does not include a read-only view of user details (use "User Edit" if needed).
- User Edit - This permission allows administrators to create, edit and view details of user accounts ("Security > User Accounts").
- User Remove - This permission allows administrators to delete user accounts ("Security > User Accounts").
- User Bulk Import - This permission allows administrators to bulk import user accounts defined in an Excel spreadsheet ("Security > User Accounts").
- User Profile Edit - This permission allows administrators to create and edit profiles for a user account. It also allows administrators to mark a profile as the default for a user.
- User Profile Remove - This permission allows administrators to delete user profiles.
- User Property View - This permission is currently unused.
- User Property Edit - This permission is currently unused.
- User Property Remove - This permission is currently unused.
- Web Service Invoke - This permission allows users to invoke TM web services.

User Accounts (Manager v5.0)

A user account is a login to a Transact Manager space. A user account is characterized by personal data, their credentials, user type ("Local", "LDAP" or "SSO"), one or more user profiles, the spaces they can log on to, the organizations they can access and the roles and groups they belong to.

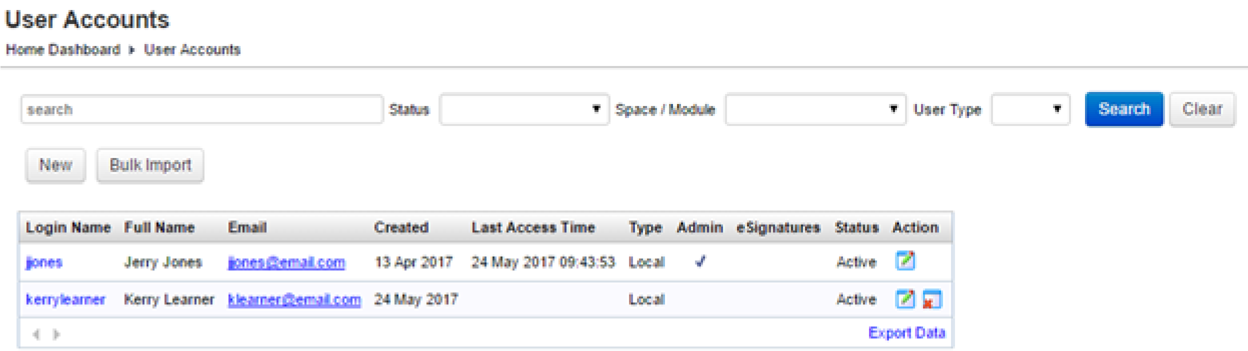
Browsing the List of User Accounts

To view the list of user accounts that are currently set up in your instance of Transact Manager, follow the steps below.

1. Select the *Security menu.*
   Select Security will display the security drop down menu.

2. Click *User Accounts* from the menu.
   Selecting User Accounts will display a list of user accounts currently created in your instance of Transact Manager.

The screenshot below displays a list of User Accounts created in an instance of Transact Manager.



You can use the search box to find a specific user account based on the login name. You can also filter the list by status ("Active", "Inactive", "Locked", "Pending", "Rejected"), and user type. In addition, you can choose to show only user accounts associated with a specific space.

You can export the search results to Excel by clicking the "Export Data" link below the table.

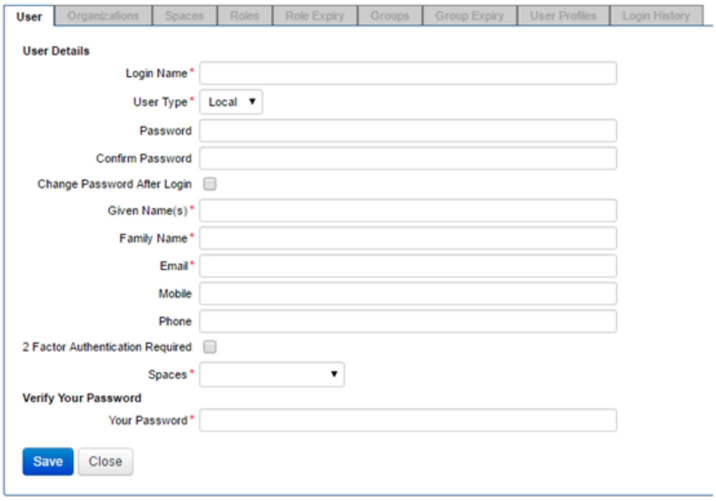
From this page, you can add, edit (, also allows you to assign spaces, roles and groups to the user) and delete ( ) user accounts.

Creating a New User Account

New user accounts can be created as needed. Follow the steps below to create a new user account.

1. Navigate to the User Accounts screen (Security > User Accounts).

2. Click the *New* button.
   Clicking New will display a blank form and direct you to the User tab. The screenshot below displays this blank form.



3. Provide a Login Name for the new user
   Login Name is what the user will enter when logging into a Transact product (Maestro, Manager, Composer). Please be aware that the Login Name must be the same as the Login Name used for the Maestro cloud instance. If the Login Name is not the same users will not be able to update TM form versions built in Maestro.

4. Select the User Type
   For users managed by Transaction Manager, use "Local" (selected by default). If you would like to manually set up an LDAP user, choose "LDAP" (note that you will get a restricted field set as the user is managed in the LDAP directory in this scenario - see section "Authentication Providers" for more on user authentication); similarly, choose "SSO" for an SSO user account (only applicable if you are using an SSO security manager).

5. Enter a password for the user account
   Password and Confirm Password must be the same value. By default, the password must be at least twelve, but at most 20 characters long and contain at least one special character and mixed case letters (the exact requirements depend on the security manager configuration for the chosen space).

   If "Change Password After Login" is ticked, the new user will be forced to change their initial password when logging on, and will be unable to access any restricted content until this has been done.

6. Fill in the personal details (given and family names, email address and phone numbers).

7. Choose a space from the "Spaces" dropdown.
   This will be the space (either the management console or one of the user spaces) that the user will be associated with initially. More spaces can be added later from the Spaces tab.

8. For security reasons, you must enter your own password to create the user.

9. Once you have filled in all the data, save the new user account.
   Once the user account has been saved you will be directed to the Roles tab. Note that the user will not have access permissions until they have been assigned a role. When creating a new user, Transact Manager will automatically open the "Roles" tab for you so you can assign roles as needed.

   The following sections describe how to assign one or more roles and spaces to a user account and set up organization access. These steps are essential for setting up functional administrator users.

Assigning Roles to User Account

There are several scenarios where you might want to change the roles assigned to a user account:

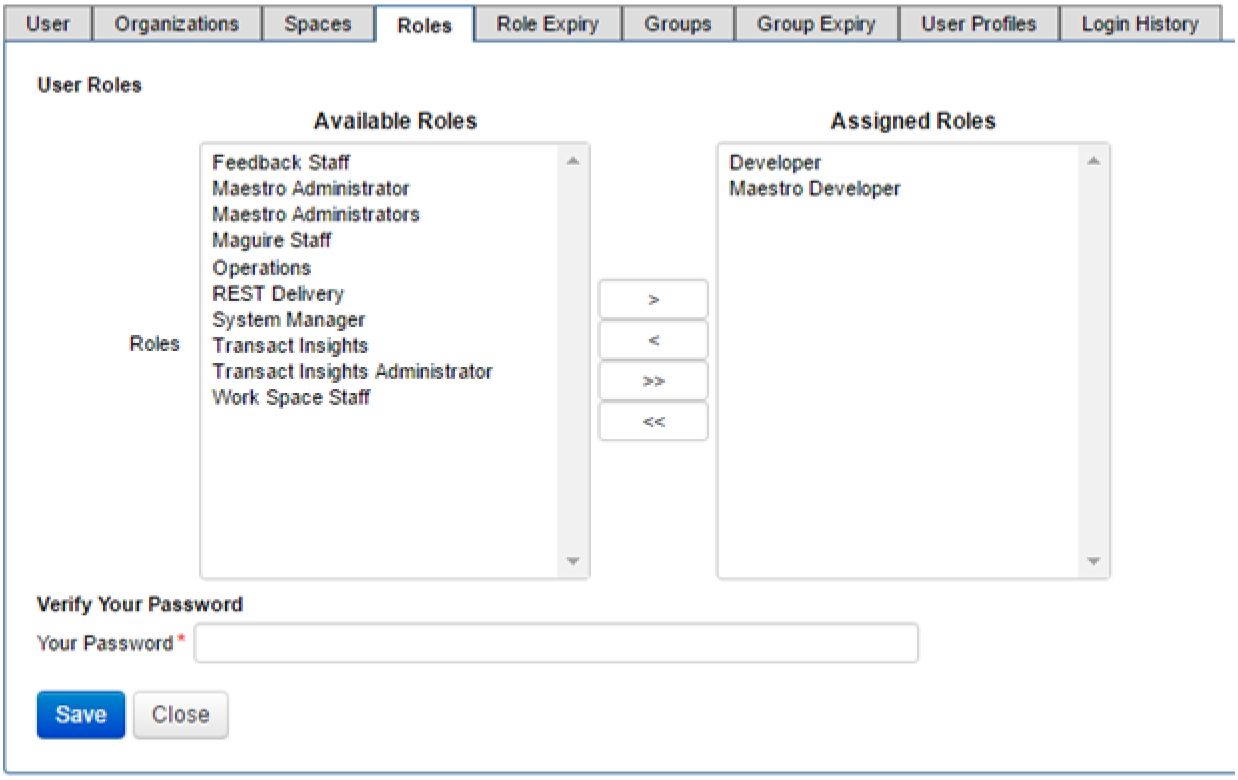
- A new user account needs to be set up with its initial set of roles, and
- existing user accounts may need further access rights.

If you suspect that a user account may have been compromised, you can remove all their roles and spaces and make the user account inactive as well, making it unable to log on or perform any restricted actions.

Follow the steps below to assign roles to the user account. Remember, if you are creating a new user account you will be automatically directed to the Roles tab once you save the details on the User tab.

1. Navigate to the User Account details if needed (Security > User Accounts)

2. Click the "Edit" icon () next to the user account and switch to the "Roles" tab
   The screenshot below displays the Roles tab of the selected user account.



Please note that there are precautions built into the Transact Manager management console that prevent organization administrators from assigning themselves inappropriate roles. If an organization administrator edits the role assignments for a user account, they will see only roles for which "Organization Assignable" is ticked. All other roles will be hidden. Global administrators, on the other hand, can see and assign all roles to other user accounts (if their own roles contain the permissions around role assignment).

3. To change the roles assigned to a user account, select roles (on Windows, use the Shift and Ctrl keys to select multiple roles) and use the ">" and "<" button to assign and remove them.
   You can also add all available roles to a user account by using the ">>" button; likewise, the "<<" button will remove all roles, leaving the user account without any privileges.

4. For security reasons, you may have to enter your own password to save changes to the roles.

5. Once all roles have been assigned appropriately, save the user account.
   The roles assigned to a user account are loaded when the user logs in, so changes will be visible to the user after logging out of the space and logging in again.

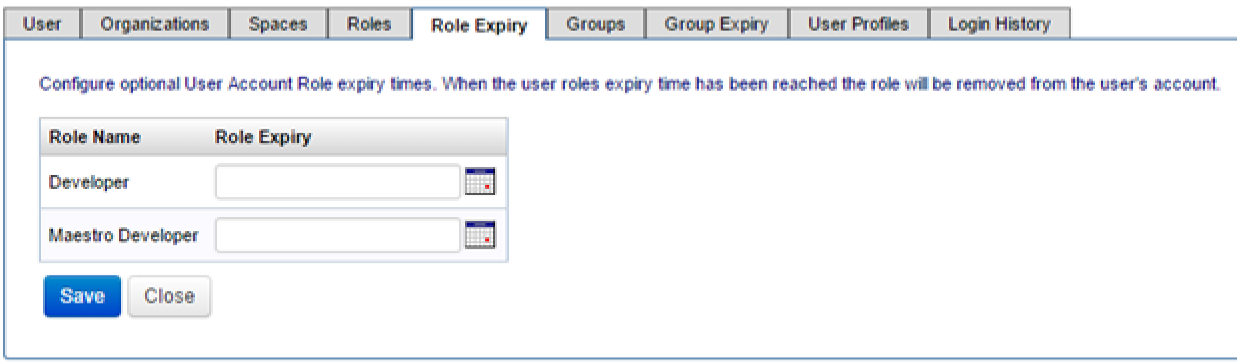
Managing Role Expiry

Sometimes it makes sense to assign a role to a user for a limited amount of time (e.g. to allow Avoka support to help debug a problem). TM allows you to configure an expiry time for user roles so you don't have to remember to remove the roles yourself.

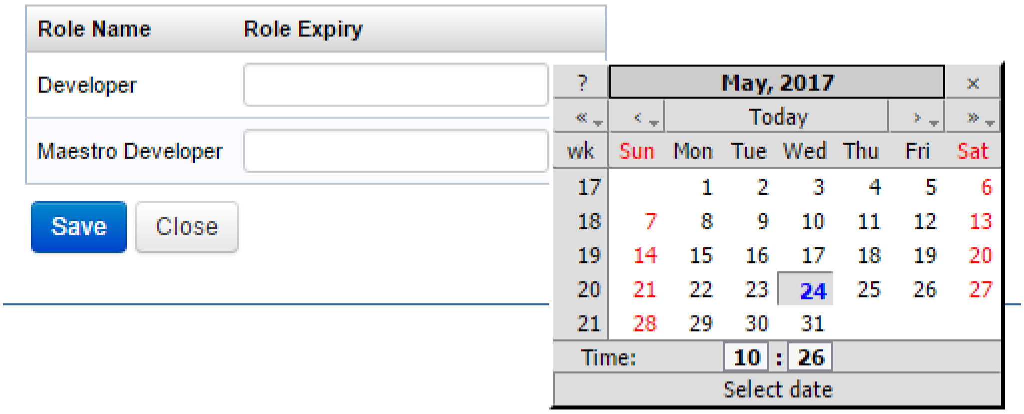
Follow the steps below to manage role expiry times for a user.

1. Navigate to the User Account details if needed (Security > User Accounts)

2. Click the "Edit" icon ( ) next to the user account and switch to the "Roles Expiry" tab.
   Each role assigned to the user will display on the Role Expiry tab. You can enter an expiry time for each role or remove expiry times if needed



3. Select the Calendar icon next to the role.
   This will display a calendar pop up.



4. Select a date and time for role expiry.
   You should select the time first as the pop up will close on selection of a date.

5. Save your changes.
   When the expiry time arrives, the background job "Security Policy Manager" (see chapter "Scheduled Jobs", section "The Security Policy Manager Job") will remove the role from the user account.

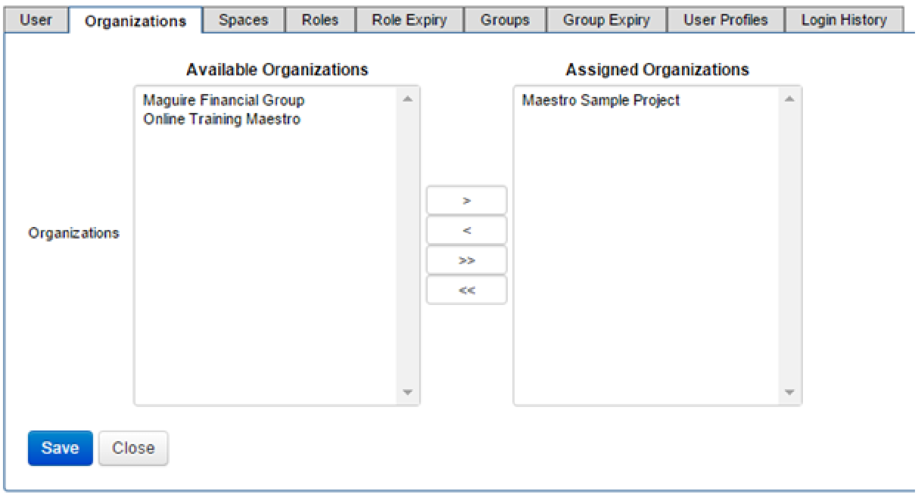
Assigning Organizations to User Account

Users associated with the TM management console can be given global access or restricted to one or more organizations. Organization administrators have restricted access privileges; in particular, whenever there is data associated with organizations, they will see only the subset relevant to their own organization (s) (e.g. on the "Forms" page, they will see only their own forms). Organization administrators can also assign only organization assignable roles to other administrators (effectively preventing them from promoting themselves or others to global administrators).

To change the organizations a user account can access, following the steps below.

1. Navigate to the User Account details if needed (Security > User Accounts)

2. Click the "Edit" icon ( ) next to the user account and switch to the "Organizations" tab.



3. Assign the user to the necessary Organization(s).
   To change the Organizations assigned to a user account, select Organizations (on Windows, use the Shift and Ctrl keys to select multiple spaces) and use the ">" and "<" button to assign and remove them.

   You can also add all available Organizations to a user account by using the ">>" button; likewise, the "<<" button will remove all Organizations

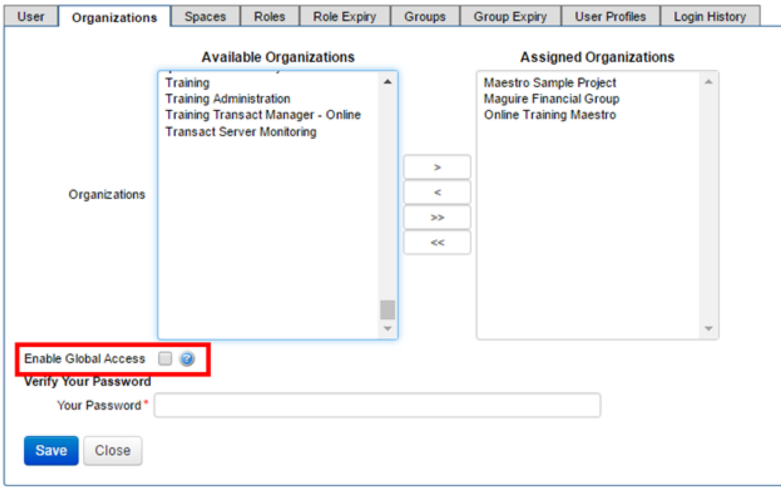
4. For security reasons, you have to enter your own password to save changes to the organizations.

5. Once all Organizations have been assigned appropriately, save the user account.
   Note: Only global administrators can assign all Organizations to user accounts. Administrators restricted to one or several organizations can assign only the spaces they themselves have been assigned.

Enable Global Access

The "Enable Global Access" checkbox controls whether a user is a global administrator or an organization administrator. If ticked, the user automatically has access to all organizations. Otherwise individual organizations must be assigned to the user.

The screenshot below displays the “Enable Global Access” checkbox.



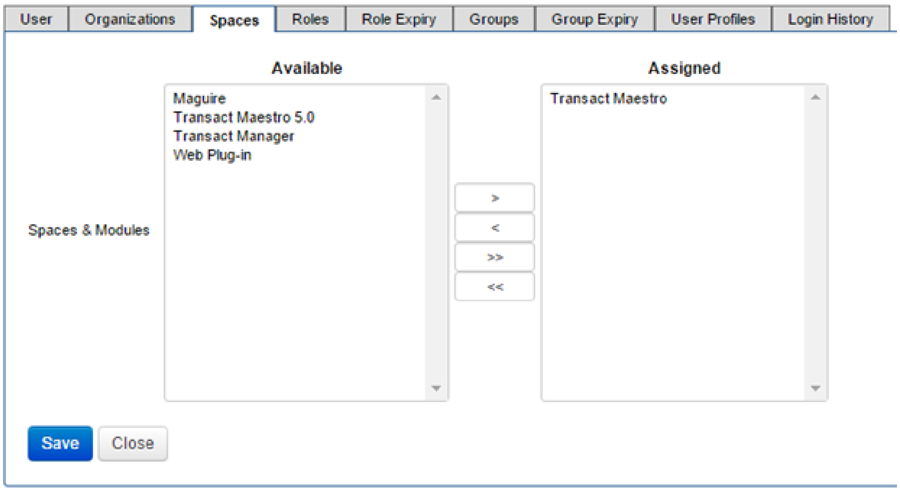
Changing the Spaces Assigned to a User Account

Before a user can log in to the management console or a user space, they need to be associated with the corresponding space record. An existing user can register for a space account if the user space supports this; the management console does not support self-registration.

To change the Spaces a user account can access, following the steps below.

1. Navigate to the User Account details if needed (Security > User Accounts)

2. Click the "Edit" icon ( ) next to the user account and switch to the "Spaces" tab.



3. Assign the user to the necessary Space(s).
   Users requiring access to Maestro will need to be given the Transact Maestro and Transact Maestro 5.0 spaces. Users requiring access to Transact Manager will need to be given the Transact Manager space.

   To change the spaces assigned to a user account, select spaces (on Windows, use the Shift and Ctrl keys to select multiple spaces) and use the ">" and "<" button to assign and remove them.
   You can also add all available spaces to a user account by using the ">>" button; likewise, the "<<" button will remove all spaces.

4. Once all spaces have been assigned appropriately, save the user account.

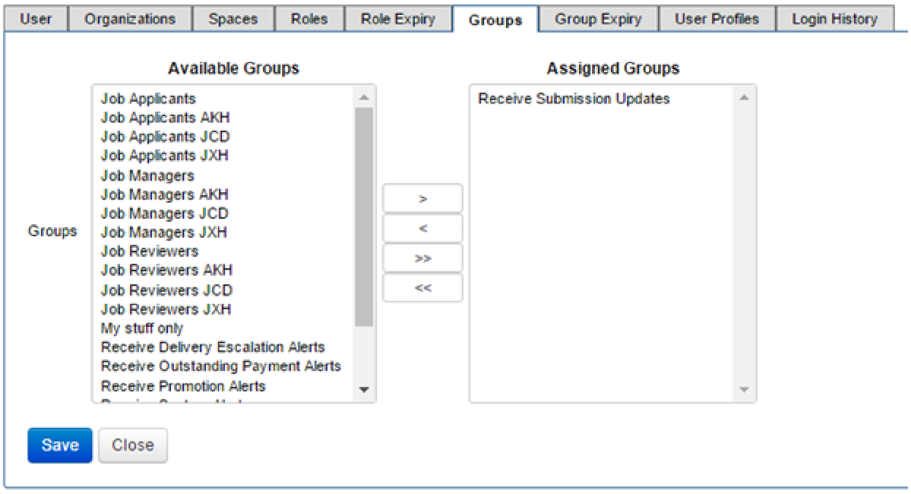
> Note: Only global administrators can always assign all spaces to user accounts. Administrators restricted to one or several organizations can assign only the spaces they themselves have been assigned.

Changing a User's Group Associations

To change the Groups a user account can access, following the steps below.

1. Navigate to the User Account details if needed (Security > User Accounts)

2. Click the "Edit" icon ( ) next to the user account and switch to the "Groups" tab.



3. Assign the user to the necessary Group(s).
   To change the groups assigned to a user account, select groups (on Windows, use the Shift and Ctrl keys to select multiple groups) and use the ">" and "<" buttons to assign and remove them.

   You can also add the user to all available groups by using the ">>" button; likewise, the "<<" button will remove the user from all groups.

4. Once all groups have been assigned appropriately, save the user account.

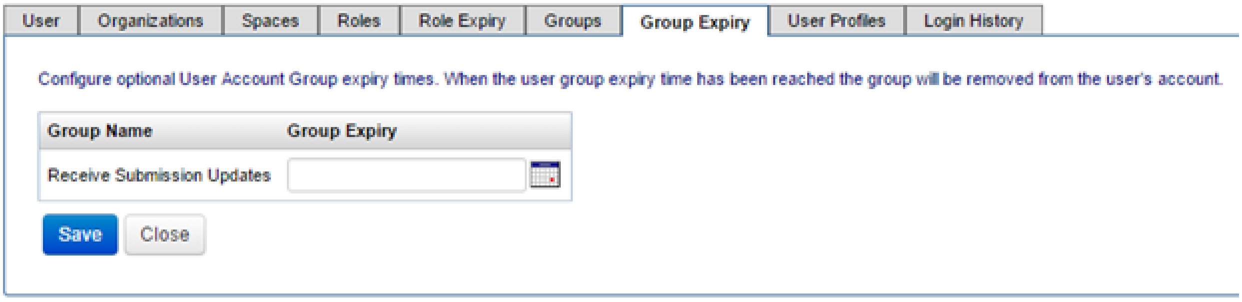
Managing Group Expiry

Sometimes it makes sense to assign a user to a group for a limited amount of time. TM allows you to configure an expiry time for user groups so you don't have to remember to remove the user from the groups yourself.

To manage group expiry times for a user, follow the steps below.

1. Navigate to the User Account details if needed (Security > User Accounts)

2. Click the "Edit" icon ( ) next to the user account and switch to the "Group Expiry" tab.
   Each role assigned to the user will display on the Role Expiry tab. You can enter an expiry time for each role or remove expiry times if needed.



3. Use the calendar icon to set the group expiry details.
   You should select the time first as the pop up will close on selection of a date.

4. Save your changes.
   Save your changes. When the expiry time arrives, the background job "Security Policy Manager" (see chapter "Scheduled Jobs", section "The Security Policy Manager Job") will remove the user account from the group.

User Profiles

Users can set up one or more profiles that can be used to pre-populate forms if the user space supports this. Each profile has the same fields, and each field corresponds to a user property (for more on properties and how to set them up, see chapter "Organization Configuration", section "Property Types"). When you create a user via the management console, a default profile is created automatically.

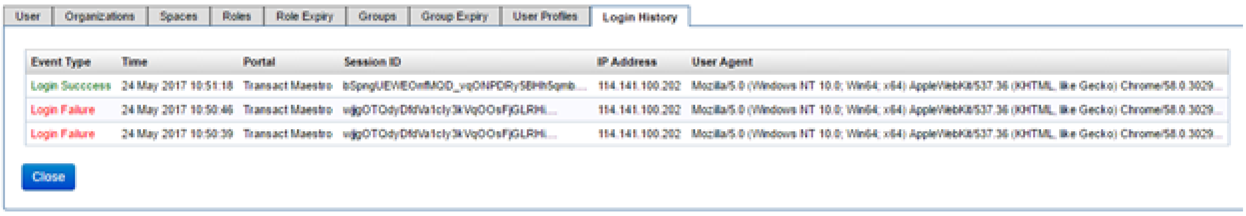
Examples for user profile fields are first name, last name and phone number. Transact Manager allows you to define whichever fields you need and map them to locations in a form data configuration so they can be pre-populated into the form.

Login History

The Login History tab displays a list of login attempts for the user. To view the login history for a user account follow the steps below.

1. Navigate to the User Account details if needed (Security > User Accounts)

2. Click the "Edit" icon ( ) next to the user account and switch to the "Login History" tab.
   The Login History tab gives you information on the event type, date and time as well as what space (portal) was accessed.



Removing a User Account

If you wish to disable a user account, you have two options:

1. You can deactivate the account, which will make them unable to log on but preserve the user record in Transaction Manager, or
2. You can delete the user account, which irrevocably removes all data associated with the user account.

Note that submissions made by a user will be preserved even after their user account is deleted; however, these submissions will not be associated with a user any more.

Deactivating a User Account

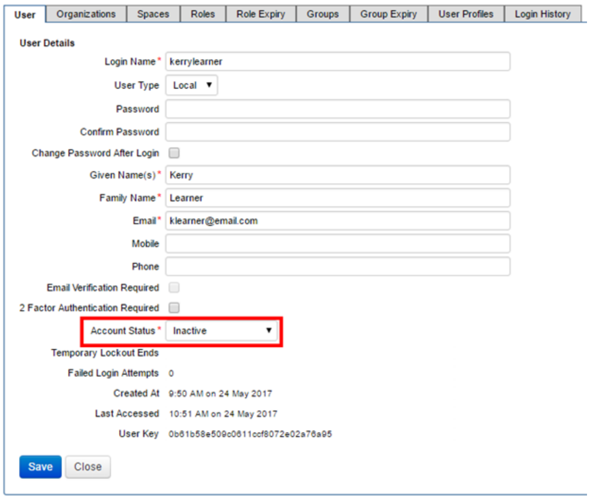
To deactivate a user account, follow the steps below.

1. Navigate to the User Account details if needed (Security > User Accounts)

2. Click the "Edit" icon ( ) next to the user account.

3. Choose "Inactive" from the "Account Status" dropdown

4. Save your changes.
   The user account record still exists, but the user can no longer log on or access restricted content.
   To reactivate the user, simply edit the user account and choose "Active" from the "Account Status" dropdown, then save.

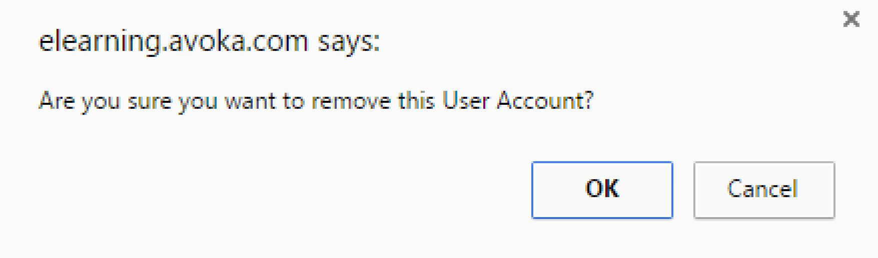


Deleting a User Account

To delete a user account, follow the steps below.

1. Navigate to the User Account details if needed (Security > User Accounts)

2. Click the "Delete" icon ( ) next to the user account.
   Since this operation will permanently remove the user record, you will be asked to confirm your decision.



   Note that you cannot delete the user you are currently logged on as, nor can the system-generated account "Root Administrator" be deleted.

   Even after a user account has been deleted, the changes the user made to the system are still visible in the audit log (see section "Tracking Changes").

Organization Configuration (Manager v5.0)

Transaction Manager is designed to accommodate anything from a single organization with a single form to hosting multiple organizations with a number of forms, and multiple form versions.

Search

Concepts

The following are the core concepts relevant to this chapter:

User: Users are people who are using Transaction Manager. There are two categories of users: Administrators use the Transaction Manager management console, whereas space users use a user space to fill in forms.

Organization: An organization is an entity that owns forms. Separate Transaction Manager organizations are usually set up for different organizations, different regional branches of an organization, or different products (with different brands). Each organization has properties which will change the way forms appear and operate. Properties can be used to control things like colors, logos, addresses, signatures, form options, or almost any part of form behavior.

Form: Forms are requested, completed and submitted by space users. A form is owned by an organization, and can be configured in many ways. Forms are rendered as either HTML or PDF to the user.

Form version:
Each form can have multiple versions. A form version contains the set of versioned information for a form. To allow flexible switching between versions, almost all data related to a form is versioned. Form versions contain the actual form template file (e.g. XDP or PDF), the form data configuration, property and metadata values, attachments and other information.

Form data configuration: A form data configuration defines how prefill data are mapped into the form's XML schema and identifies special sections in the form. Form data configurations are crucial for pre-populating forms with property values such as a logo, attachments and contact details.

Property: Properties are the attributes of organizations, users and forms that are used to define how a form appears and functions. Each property consists of a property type describing the attribute and a set of property values (which are the concrete attribute values assigned to the entities that have this attribute).

Metadata: Metadata values are the attributes of organizations and forms that are used to define externally visible information about the form. Other applications can use web services to query the metadata set up in Transaction Manager.

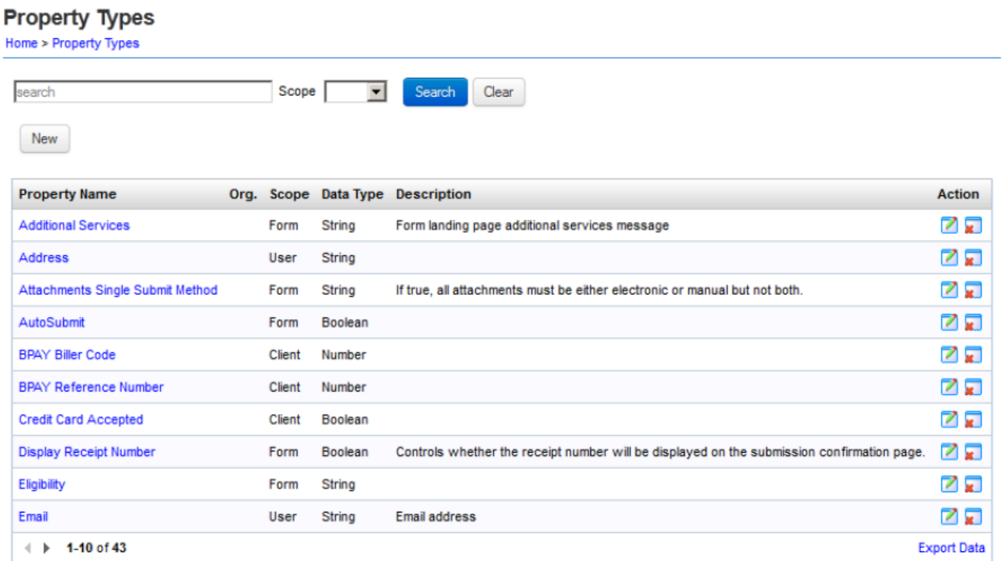
XML prefill: An XML prefill definition can be used to allow the client organization to pass in an XML file at form render time. The data contained in the XML file can then be used to pre-populate the form.

Property Types

Properties are crucial for pre-population of data into the form and are also used by TM directly. Using properties, you can configure all sorts of attributes for your organizations and forms that will be passed into the form when the user requests it, for example branding, links to related resources, contact details and help text.
Properties are always based on a property type defined for all entities of a certain scope (organization, form or user) which can then be assigned a value for each entity.
This section tells you how to manage property types and how to assign property values to your entities.

Browsing the List of Property Types

To see the list of properties that have been set up in Transaction Manager, go to "Forms > Property Types".



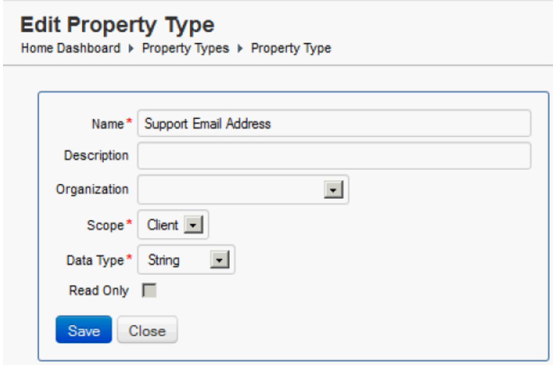
Property Name	Org.	Scope	Data Type	Description	Action
Additional Services		Form	String	Form landing page additional services message	
Address		User	String		
Attachments Single Submit Method		Form	String	If true, all attachments must be either electronic or manual but not both.	
AutoSubmit		Form	Boolean		
BPAY Biller Code		Client	Number		
BPAY Reference Number		Client	Number		
Credit Card Accepted		Client	Boolean		
Display Receipt Number		Form	Boolean	Controls whether the receipt number will be displayed on the submission confirmation page.	
Eligibility		Form	String		
Email		User	String	Email address	

Browsing the list of property types
You can filter the list using a substring of the property name, by scope ("Organization", "Form" or "User") and data type (see the following section), and

export the search results to Excel by clicking the "Export Data" link below the table.
This page allows you to create new property types and to edit ( ) and delete ( ) existing ones.

Adding a Property Type

To add a new property type, go to "Forms > Property Types" and click "New". You will be directed to the "Edit Property Type" page.



Creating a property type

Adding the Basic Data

Fill in a unique name and (optionally) a description for the property type.
If you want the property to belong to a specific organization, select it from the "Organization" dropdown. This means that organization administrators for other organizations will not see this property.
Set the "Scope" field to either "Organization", "Form" or "User" depending on what entity the property is associated with.
Finally, the "Data Type" field can be used to define the domain of expected property values (see the following section).
The "Sequence" field allows you to introduce a custom order for your property values. On pages that display property values (i.e. "Edit Organization", "Edit Form", "Edit User"), properties will be sorted by their sequence number. If no sequence is defined, properties are sorted by property name.
If the "Read Only" flag is ticked, the property cannot be modified. This is generally only the case for properties created by Transaction Manager on startup (e.g. the user property "Email").
Save the data to create the property type. You can now assign values to it, but until you map the property to a location in a form data configuration, these values will not be used (see Form Version Configuration (Manager v5.0)#_sec_setuppropertyprefillmappings).

Possible Data Types for a Property

To help restrict property values that a user can set up, properties can have one of several different data types.
The property data types that Transaction Manager offers are:

- BooleanAccepts only the values "true" and "false".Example: Signature Required
- NumberAccepts only numeric characters (0-9, +, -, .).Example: ABN
- StringAccepts an arbitrary string.Example: Country
- ListList property values are used to populate lists in the form. This means that list properties can have several property values assigned to them, all of which will be passed into the form.Example: Agency Contact Name
- ImageAccepts images of type PNG, GIF and JPG. Images can be uploaded to the server when assigning a value to the property.Example: Logo
- Long TextAccepts a text file with HTML-style formatting, with no length restrictions. Long text files can be uploaded to the server when assigning a value to the property.Example: Terms and Conditions

If you choose a more restrictive data type, administrators will be prevented from entering unacceptable values. This helps avoid errors caused by invalid data being passed to a form.

Assigning a Value to a Property

To assign a value to an organization property, please follow the instructions in _sec_addorganizationproperties. Please note that when TM resolves property values, it always uses the values stored for the form version. When a new form is created, the current organization property values are copied to the form version to make up the initial set of property values for that form version.Changing organization property values will not have any effect on the set of property values maintained by existing form versions. This is intentional and preserves existing versioned information.
To change the value for a versioned property value, please follow the instructions in Form Version Configuration (Manager v5.0) #_sec_addingformversionproperties.

Deleting a Property Type

To delete a property type, go to "Forms > Property Types", locate the property type and click the "Delete" icon ( ). You will be asked to confirm this action before the property type and any organization and user property values associated with it are irrevocably deleted.
Note: Property values maintained on the form version level are not deleted along with the property type. These property values store the key attributes associated with the property type, and so can exist without being associated with a property type.

Metadata Configuration

Metadata in Transaction Manager is data describing the forms for classification and searchability purposes. Rather than being used internally to render the forms like properties are, metadata is exposed to external systems via a web service interface that Transaction Manager provides.
This allows you to offer your users various ways of searching for and filtering forms while maintaining the metadata along with your form definition in Transaction Manager. If you need a new search criterion, you can simply add a metadata tag to Transaction Manager and then define appropriate metadata values for your forms.

Concepts

Metadata is organized into *metadata tag definitions*, which define a particular attribute you wish to expose (e.g. form type, creator or target audience) and *metadata tag values*, which are concrete values for a metadata tag and a specific form.
To make it as easy as possible to set up metadata for all forms, metadata tags can have different *scopes*: either "Client" or "Form". Client metadata tags are configured once per organization and new forms for a particular organization automatically inherit that organization's metadata tag values.

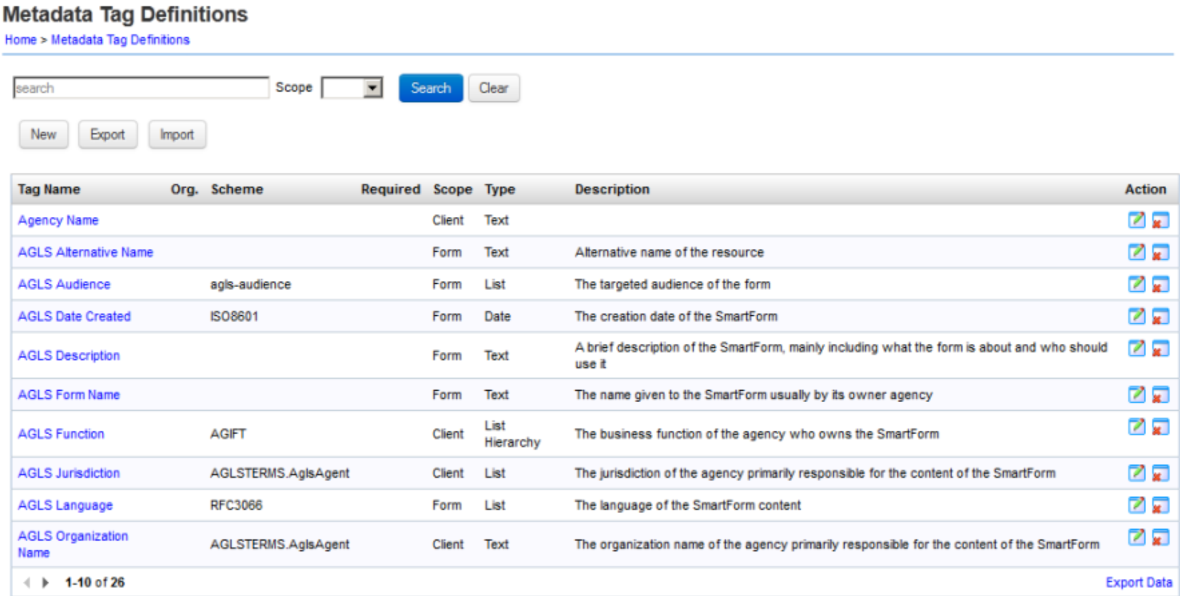
Metadata Tag Types

Transaction Manager allows you to declare types for your metadata tags. This can be useful if you wish to constrain metadata values to a certain domain. The following metadata types can be used:

- StringA generic type accepting any string.
- DateA type accepting only date values. When assigning values to a date metadata tag, a date picker will be used.
- ListA type defining a list of string values. When assigning values to a list metadata tag, a dropdown with all list values will be used.
- List HierarchyA type defining a hierarchical set of lists (with a maximum depth of three). When assigning values to a list hierarchy metadata tag, a series of three dropdowns will be used, with the selected item on one level affecting the choices shown in the next dropdown.

Browsing Metadata Tags

To view a list of all metadata tags set up in Transaction Manager, go to "Forms > Metadata Tag Definitions". You can filter the list by organization, using a substring of the tag name or scheme, or by scope, and export the search results to Excel by clicking the "Export Data" link below the table.



Metadata Tag Definitions

Home > Metadata Tag Definitions

search Scope Search Clear

New Export Import

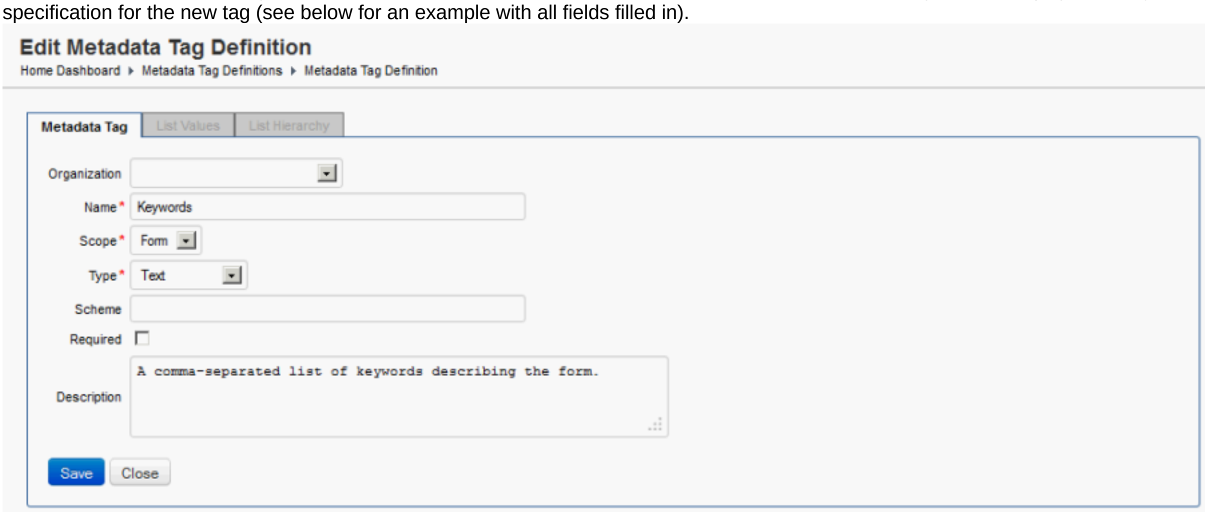
Tag Name	Org.	Scheme	Required	Scope	Type	Description	Action
Agency Name				Client	Text		
AGLS Alternative Name				Form	Text	Alternative name of the resource	
AGLS Audience		agls-audience		Form	List	The targeted audience of the form	
AGLS Date Created		ISO8601		Form	Date	The creation date of the SmartForm	
AGLS Description				Form	Text	A brief description of the SmartForm, mainly including what the form is about and who should use it	
AGLS Form Name				Form	Text	The name given to the SmartForm usually by its owner agency	
AGLS Function		AGIFT		Client	List Hierarchy	The business function of the agency who owns the SmartForm	
AGLS Jurisdiction		AGLSTERMS.AglsAgent		Client	List	The jurisdiction of the agency primarily responsible for the content of the SmartForm	
AGLS Language		RFC3066		Form	List	The language of the SmartForm content	
AGLS Organization Name		AGLSTERMS.AglsAgent		Client	Text	The organization name of the agency primarily responsible for the content of the SmartForm	

1-10 of 26 Export Data

Browsing the list of metadata tags
You can also create, edit and delete metadata tag definitions from this page, as well as export or import the set of metadata tag definitions. Metadata tag values, however, are maintained elsewhere (see _sec_settingmetadatatagvalue for details).

Adding a Metadata Tag Definition

If you need to expose another attribute of your forms to external systems, you will need to add a new metadata tag definition.
Go to "Forms > Metadata Tag Definitions" and click "New". You will be directed to the "Edit Metadata Tag Definition" page where you fill in the specification for the new tag (see below for an example with all fields filled in).



Creating a metadata tag
If the metadata tag is specific to an organization, choose it from the "Organization" dropdown. On the other hand, if the tag should be available to all organizations, leave the dropdown blank.
Give your metadata tag a unique name and, optionally, a description.
Choose the most appropriate scope, i.e. "Client" or "Form". Generally, if you can use "Client" because all forms of that organization will have the same

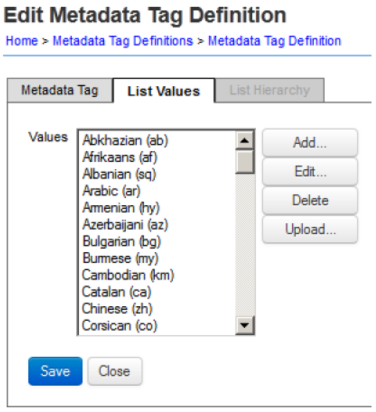
metadata value, do so instead of using "Form" as the scope. If you cannot determine a metadata value that is constant for each organization, you will need to use "Form".
Choose the metadata type from the "Type" dropdown. If you are creating a list or list hierarchy tag, you will need to add list values to the metadata tag after saving it.
Optionally, enter the tag's scheme name into the "Scheme" field.
The "Required" checkbox can be ticked if a value for this metadata tag must be assigned. If a default service definition of type "Form Meta Data" is set up in TM, a form can be promoted only if it defines values for all mandatory metadata tags (see Form Configuration (Manager v5.0)#_sec_formpromotion for more on form promotion).
Once you have saved the new tag, it will appear in the list.

Adding or Modifying List Values for Metadata Tags

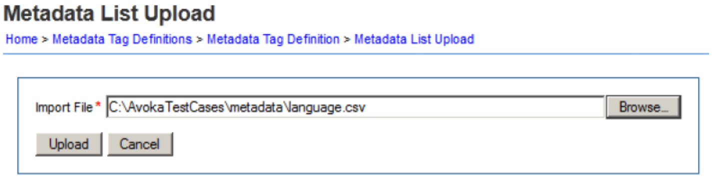
For metadata tags of type "List" or "List Hierarchy", you need to define a set of list values that can be assigned to these tags. To view or edit list values for a metadata tag, go to "Forms > Metadata Tag Definitions" and edit the tag.

Editing Values for a List Metadata Tag

If the tag is of type "List", switch to the "List Values" tab. You will see a list of the currently defined values as shown below.



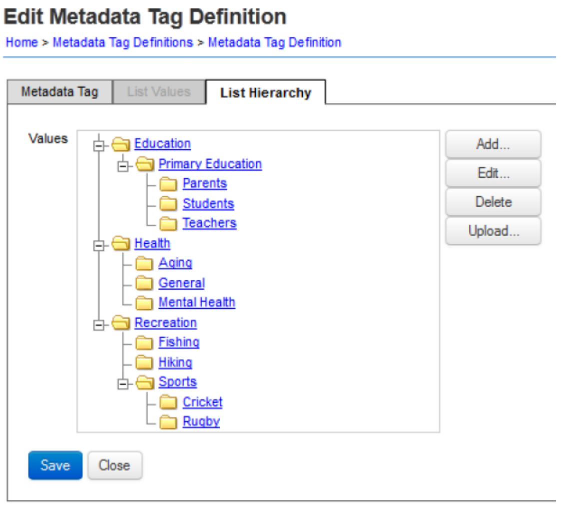
Values for a list metadata tag
You can either modify individual values by using the "Add", "Edit" and "Delete" buttons, or upload a new set of values.
If you are modifying values individually, note that all metadata list values have a value as well as a sequence. This sequence is used to sort values when the list is presented during metadata value assignment. If you wish to use sequences, please make sure sequence numbers are consistent and unique.
If you want to upload a new set of values, click the "Upload" button. You will then be directed to the "Metadata List Upload" page.



Uploading a file containing list values
Browse to a CSV file containing the set of list values (one per line, no separators apart from line breaks) and click "Upload". When you upload new values, all old values will be deleted. The CSV file will be parsed, and the values contained in it will be added to the list in the same order.

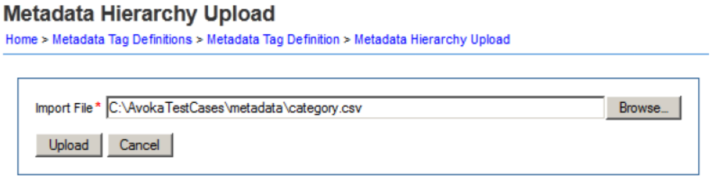
Editing Values for a List Hierarchy Metadata Tag

If the tag is of type "List Hierarchy", switch to the "List Hierarchy" tag. You will be able to create a hierarchical list of up to three levels, with values on the top two levels containing an arbitrary number of children.



Values for a list hierarchy metadata tag
You can either modify individual entries by using the "Add", "Edit" and "Delete" buttons, or upload a new set of values.
If you are modifying values individually, please note that all metadata list values have a value as well as a sequence. This sequence is used to sort values when the list is presented during metadata value assignment. If you wish to use sequences, please make sure sequence numbers are consistent and unique. Sequences are used to sort list values on the same hierarchy level with the same parent, so you can re-use the same sequence numbers for groups of values on different hierarchy levels or with different parents.

If you want to upload a new set of values, click the "Upload" button. You will then be directed to the "Metadata Hierarchy Upload" page.



Uploading a file containing list hierarchy values

Browse to a CSV file containing the set of list values as follows: Each line contains three values separated by commas. These values define the values at the three hierarchy levels. You can leave the last or last two values blank.An example for a valid hierarchy file is shown below.

```
Education,,
Education,Primary Education,
Education,Primary Education,Parents
Education,Primary Education,Students
Education,Primary Education,Teachers
Health,,
Health,Aging,
Health,General,
Health,Mental Health,
Recreation,,
Recreation,Fishing,
Recreation,Hiking,
Recreation,Sports,
Recreation,Sports,Cricket
Recreation,Sports,Rugby
```

An example CSV file containing list hierarchy values

Click "Upload" to load the metadata list values from the file. When you upload new values, all old values will be deleted. The CSV file will be parsed, and the values in it will be added to the list hierarchy in the same order.

Setting a Metadata Tag Value

To assign a value to an organization metadata tag, please follow the instructions in _sec_setuporganizationmetadata. Please note that when TM resolves metadata values, it always uses the values stored for the form version. When a new form is created, the current organization metadata values are copied to the form version to make up the initial set of metadata values for that form version. Changing organization metadata values will not have any effect on the set of metadata values maintained by existing form versions. This is intentional and preserves existing versioned information.
To modify a versioned metadata value and view the set of metadata values for a form version, please follow the instructions in Form Version Configuration (Manager v5.0)#_sec_setupformmetadata.

Deleting a Metadata Tag Definition

If you want to permanently remove a metadata tag, you can delete it. Please note that this operation cannot be undone and will also remove organization metadata values defined for the tag, so use it with caution.

To delete a metadata tag, go to "Forms > Metadata Tag Definitions", find the metadata tag you want to remove, and click the "Delete" icon ( ). You will be asked to confirm this action; once you do so, the tag and all organization metadata values will be irrevocably removed.
Note: Metadata values maintained on the form version level are not deleted along with the metadata tag. These metadata values store the key attributes associated with the metadata tag, and so can exist without being associated with a metadata tag.

Importing or Exporting Metadata Tag Definitions

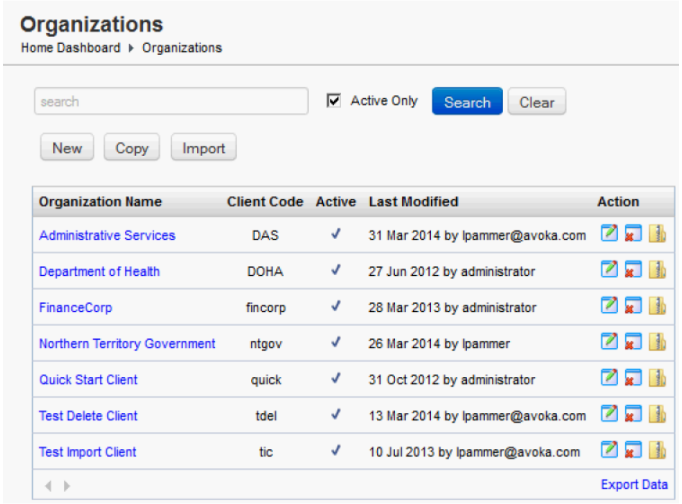
Please refer to Multi-Environment Support (Manager v5.0)#_sec_migratingmetadatatags on how to export and import metadata between servers.

Organization Configuration

In Transaction Manager, forms are owned by an organization, and organizations have properties and metadata values that are used by the forms.
You will need at least one organization configured in Transaction Manager. Setting up multiple organizations allows you to customize branding and other organization related configuration, as well as allowing you to set up administrators with access rights to only data associated with a single organization.
Each organization has a set of attributes associated with it. Some of these attributes are required (e.g. name, description), while others are optional. The set of optional attributes can be extended by using properties to match the requirements of your forms (see _sec_propertytypes).

Browsing the List of Organizations

To view the list of organizations set up in Transaction Manager, go to "Forms > Organizations". See the screenshot below for an example of what you should be seeing.



Browsing the list of organizations

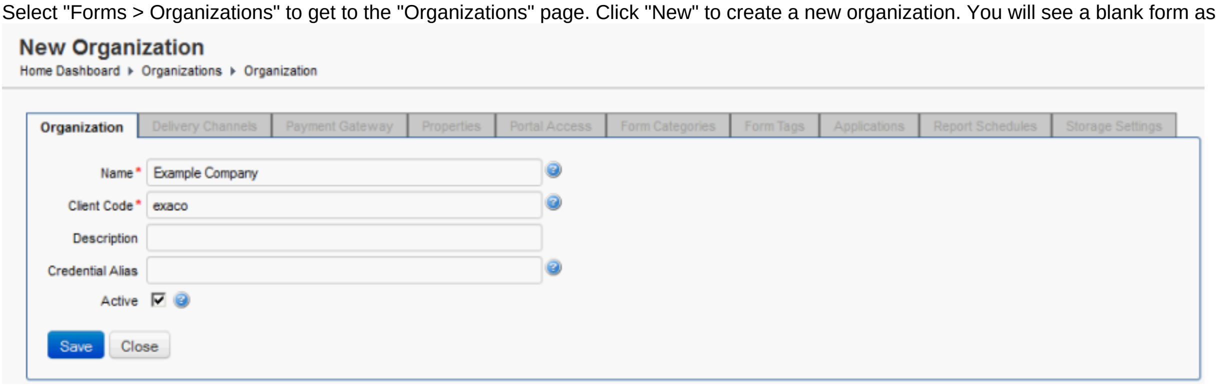
You can filter this list using a substring of the organization name, as well as the client code and client key, and filter out organizations that do not have the "Active" flag by ticking the "Active Only" checkbox. You can also export the search results to Excel by clicking the "Export Data" link below the table.

This page allows you to create new organizations, create a copy of an organization and edit ( ) or delete ( ) existing organizations. These operations are described in more detail in the following sections.

You can also export ( ) or import organizations. This is described in detail in Multi-Environment Support (Manager v5.0)#_sec_migratingorganization.

Creating an Organization

Select "Forms > Organizations" to get to the "Organizations" page. Click "New" to create a new organization. You will see a blank form as shown below.



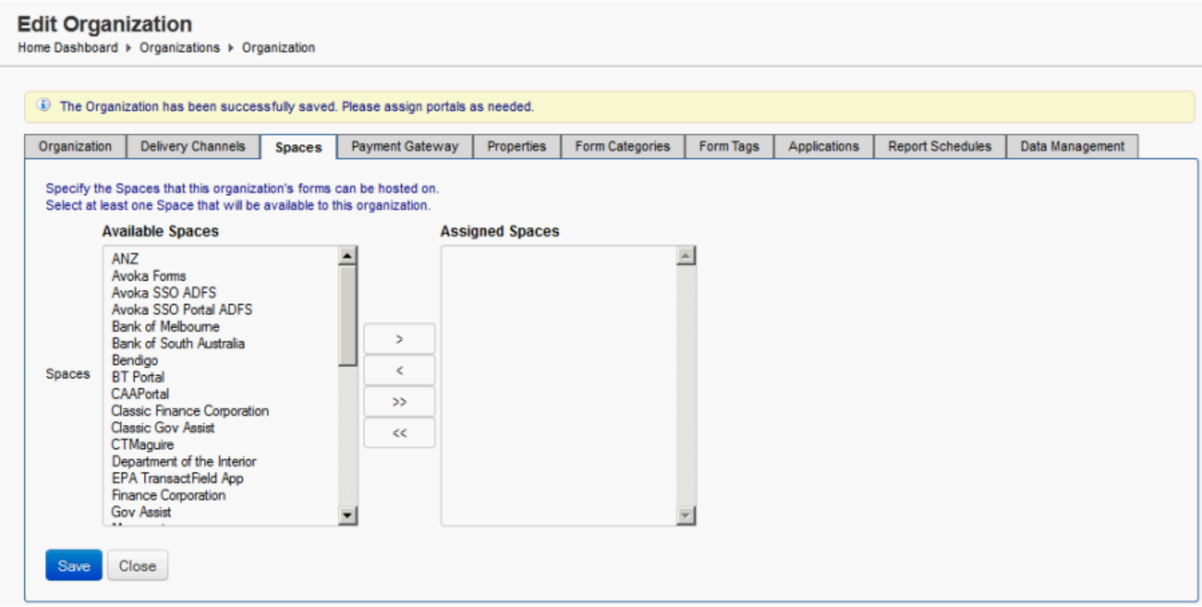
Creating an organization

Setting Up the Organization's Essentials

First, complete the mandatory properties shown in the "Organization" tab.
Set the "Name" field to the name of the organization / department / product etc. that the organization represents. If the name is very long, use an abbreviation or shortened version.
The "Client Code" will be used internally to uniquely identify the organization when requesting forms. It should be set to an alphabetic string, for example "exaco" for an organization called "Example Company".
Set the "Description" field to the organization's full name and / or a descriptive text to help identify the organization.
The "Credential Alias" can be used to specify the name of a custom credential set up in Adobe LiveCycle to be used during receipt rendering.
The "Active" flag indicates that the organization is active. Inactive organizations will often not be included in reports, and the Self Service Space does not show forms for inactive organizations.
Click "Save" to create a new organization record. All the other tabs should become active at this point, allowing you to further configure the organization. A read-only client key will also be automatically generated for the organization.
To complete setting up the new organization, please follow the instructions in the sections below.

Configuring Space Access

Once you save your new organization, you will automatically be directed to the "Spaces" tab. You will need to decide which user spaces the organization shall be able to host forms on.
Note: If you do not assign any spaces to the organization, no forms can be created for the organization as there are no spaces available to host these forms. Therefore, be sure to assign at least one space to the organization.



Assigning spaces to a new organization

Select the set of spaces for the organization and click ">" to assign them. The "<" button can be used to unassign selected spaces. When you are done, click "Save".

Configuring Delivery Channels

Delivery channels are discussed in detail in chapter "Delivery Configuration". Please refer to Delivery Configuration (Manager v5.0) #_sec_createdeliverychannel in particular on how to set up a delivery channel for an organization.

Configuring Payment Gateway Details

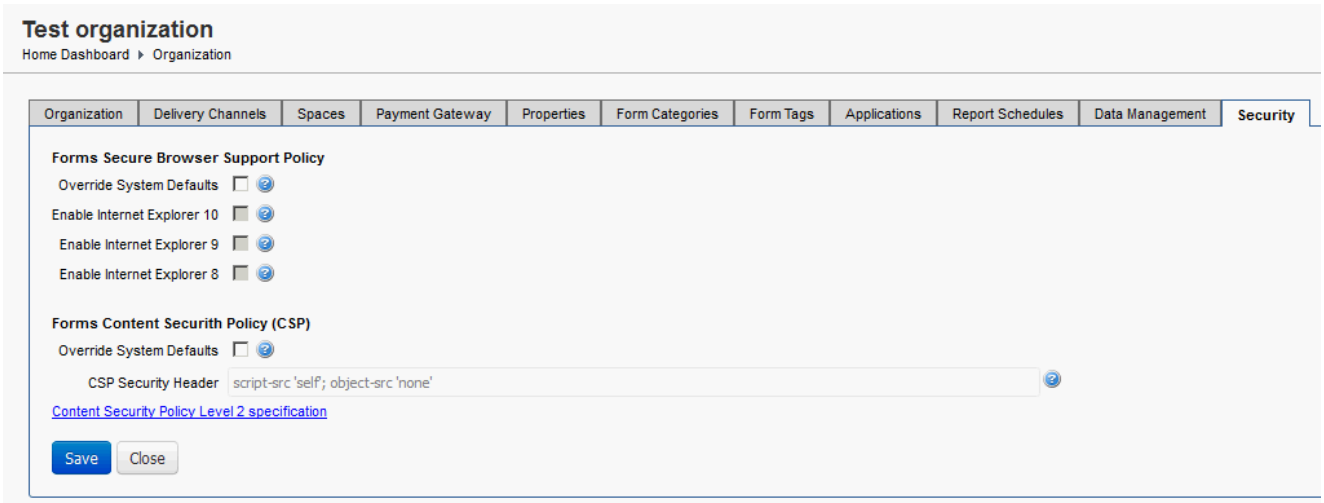
The "Payment Gateway" tab contains settings related to the payment gateway configuration used by the organization to enable credit card payments by users. Please refer to Online Payment (Manager v5.0) for more information about online payment in Transaction Manager.

Configuring Data Retention and Storage Settings

The "Storage Settings" tab contains settings related to data storage security and data retention. Please see Operations (Manager v5.0) #_sec_dataretention for general information, and its Operations (Manager v5.0)#_sec_dataretentionorganizationlevel for settings available on the organization level.

Configuring Security Settings

The "Security" tab allows you to customize some settings related to form rendering.



Editing organization security settings

You can override browser support by modifying the settings in the "Forms Secure Browser Support Policy" section.

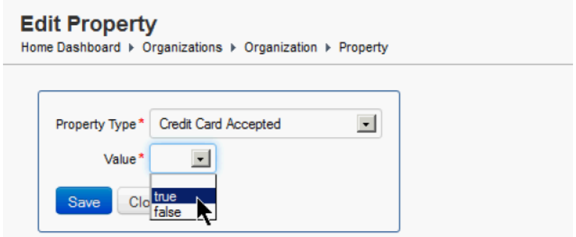
In addition, you can specify a Content Security Policy header that will be used by all forms in the "Forms Content Security Policy (CSP)". .

In both cases, the system settings are defined in the default "Form Submission Access Controller" service. See Services (Manager v5.0) for more on service definitions.

Adding Organization Properties

Before you add forms to the organization, you can define the organization further by creating organization property types as required and assigning values to them. Organization properties can be used to pre-populate your forms or modify their behavior (see _sec_propertytypes for more on property types and how to create them).

To edit organization properties, open your new organization by clicking the "Edit" icon ( ) on the Organizations page and go to the "Properties" tab. You can either create a new property value by clicking "New" or change an existing property by clicking the "Edit" icon next to it. Either way, you will see the "Edit Property" screen.

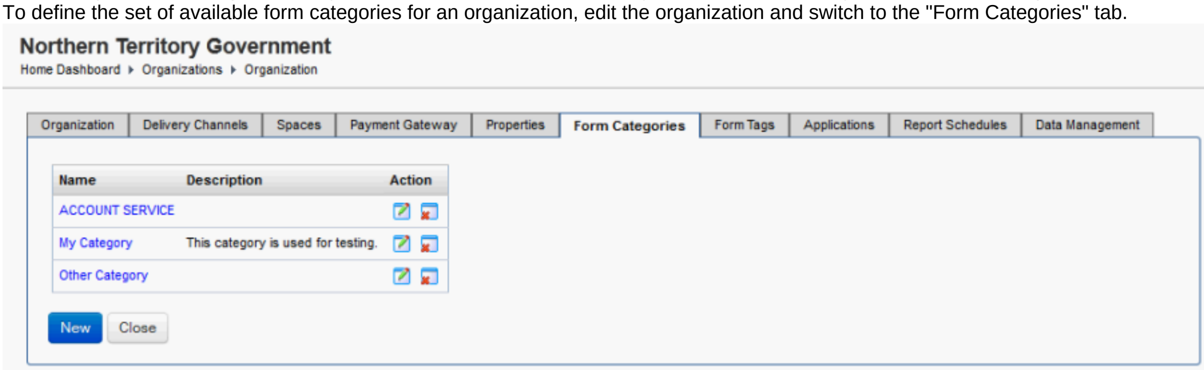


Editing an organization property

Enter the new value for the property (depending on the property type, values may have to follow a specific format; see _sec_propertytypes) and save. Verify that the new value for the property is displayed on the "Properties" tab.
Please note: Property values are versioned on a form version level. Adding or modifying an organization property value will not affect existing form versions (so as to preserve versioned data). However, new forms will receive a copy of all current organization property values. In addition, you can edit a form version and synchronize organization property values.

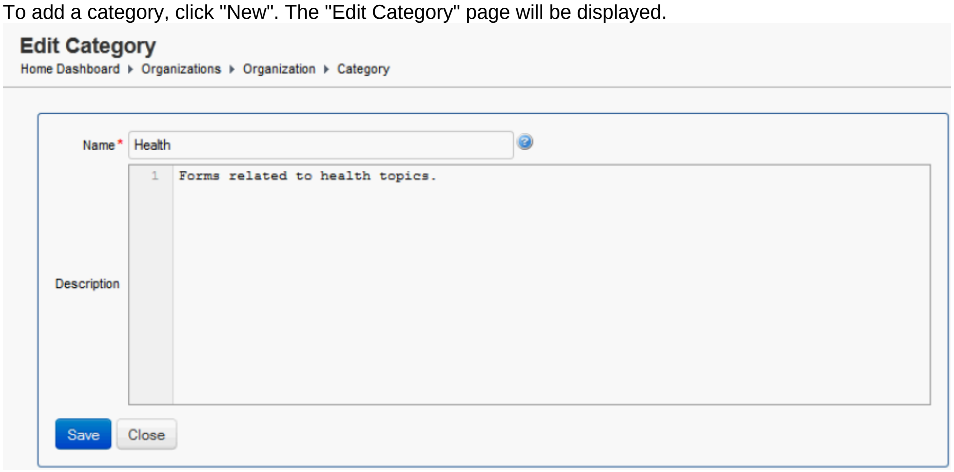
Setting Up Form Categories

Forms can be organized into categories. Space users can then filter by category when searching for forms, if the space supports this. This can help users locate forms, especially if a large number of forms are hosted on a space.
Form categories are set up on the organization level. Forms can be assigned one or more categories and will show up under all these categories.
To define the set of available form categories for an organization, edit the organization and switch to the "Form Categories" tab.



The list of form categories for an organization

You can add, edit and delete categories.
To add a category, click "New". The "Edit Category" page will be displayed.



Creating a form category

Enter a unique category name and an optional description and save the category. Now you can assign forms to the new category as described in Form Version Configuration (Manager v5.0)#_sec_configureformcategories.

Setting Up Organization Metadata

Metadata can be used to run queries on the set of forms defined in Transaction Manager using a Web Service API (see _sec_metadataconfiguration on more details about metadata, including how to create metadata tags).
If you want to define metadata for an organization, you can set values for these metadata tags on the "Edit Organization" page. All forms will inherit organization metadata when they are created.
To add a metadata value, edit the organization and go to the "Form Tags" tab. Click "New" to bring up the "Edit Organization Form Tag" page (the screenshot below shows the page with all values filled in).

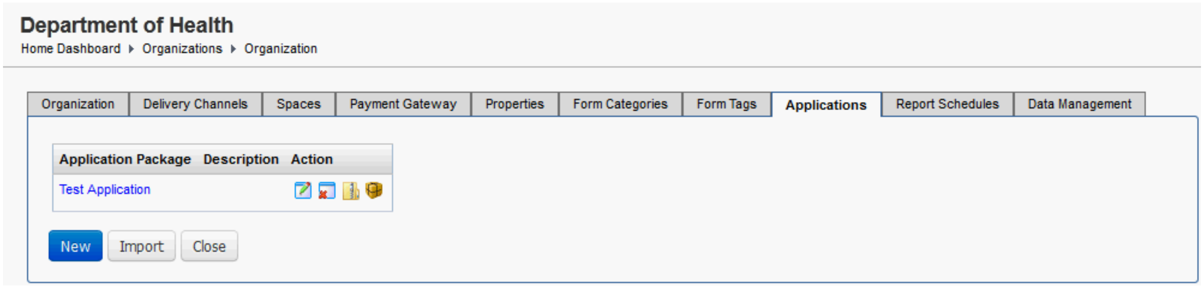
Adding an organization-based metadata value

Choose the metadata tag to assign a value to from the "Tag Name" dropdown (only metadata tags with a scope of "Client" will be displayed).
Set "Tag Value" to the metadata value for this organization and save. You will see that the metadata value has been added to the "Form Tags" tab. All forms that are set up for this organization will inherit this metadata value.

Setting Up Application Packages

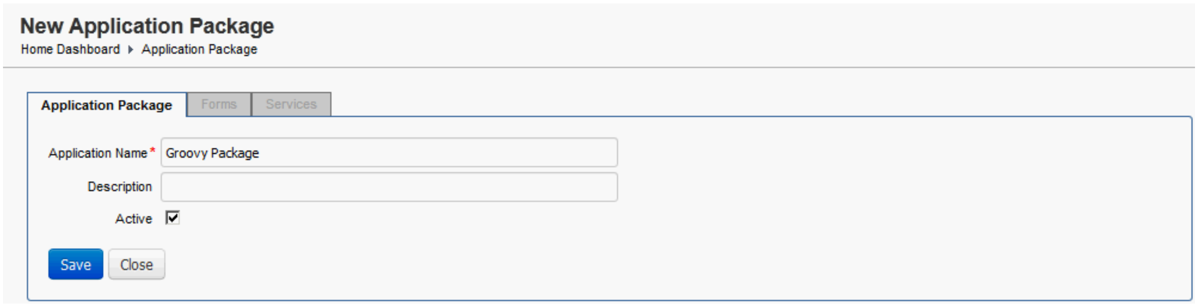
Application packages can be used to group organization specific services (see Services (Manager v5.0)) and forms so that they can be migrated between TM servers more easily.

To manage application packages, edit the organization and switch to the "Applications" tab (see below).



Managing application packages

You can edit and delete application packages, or create new ones by clicking the "New" button (the screenshot below shows the "New Application Package" page).

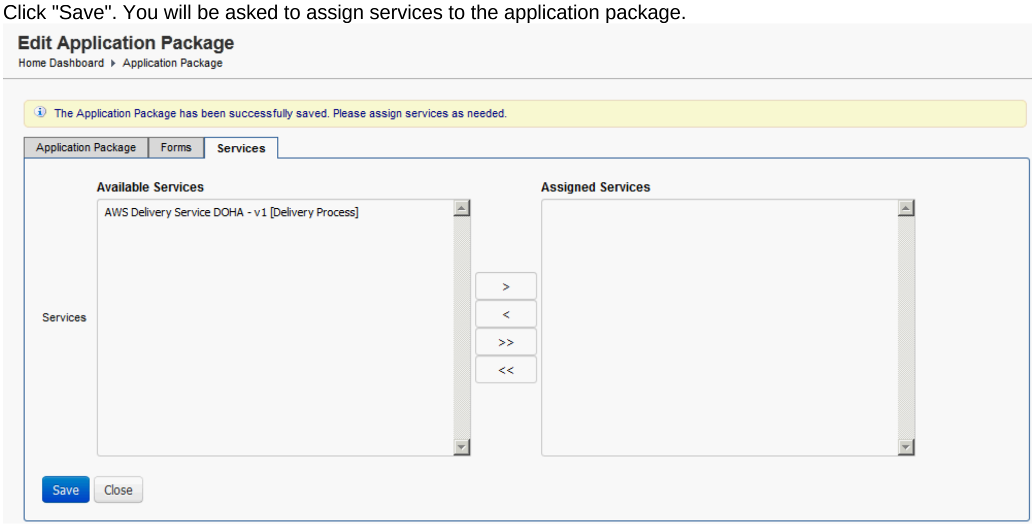


Creating an application package

Enter a name and an optional description for the application package.

Leave the "Active" checkbox ticked if you would like the application package to be available for use.

Click "Save". You will be asked to assign services to the application package.

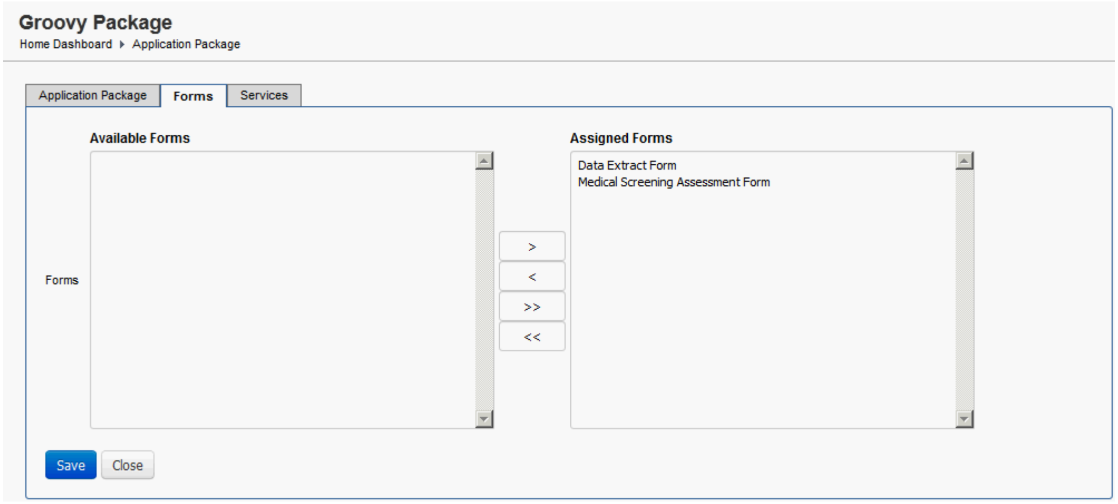


Assigning services to an application package

You will be offered all service definitions belonging to the organization (global services will not be offered).

Select one or more services and add them to the application package by using the ">" button. Remove services from the application package by using the "<" button.
Save your changes when you have assigned the appropriate services.

You can also assign forms to the application package. To do so, switch to the "Forms" tab.



Assigning forms to an application package

You will be offered all form belonging to the organization.

Select one or more forms and add them to the application package by using the ">" button. Remove forms from the application package by using the "<" button.

Save your changes when you have assigned the appropriate forms

You can export and import application packages as a single unit. Please see Multi-Environment Support (Manager v5.0)#_sec_Migrating_an_Application_Package for more information.

Setting Up Forms for the Organization

Transaction Manager allows you to quickly create a completely new form and associated entities by using the "New Form" wizard (see Form Configuration (Manager v5.0)#_sec_newformwizard).

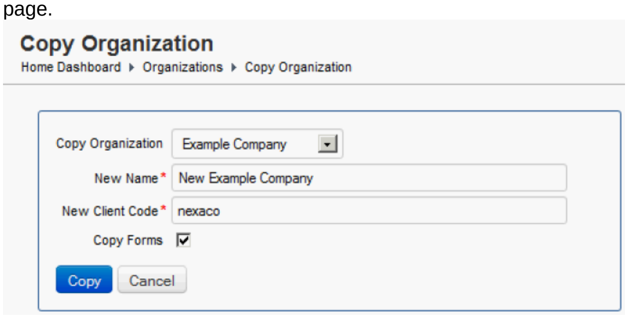
Deleting an Organization

You can delete an organization and all its dependent entities (e.g. forms, submissions) from Transaction Manager. Do consider this carefully as this cannot be undone and will irrevocably remove all forms, submissions and users for the organization. If possible, organizations should be made inactive instead of deleting them, especially if there have been submissions for any of their forms.
To inactivate an organization, go to "Forms > Organizations" and edit the organization. On the "Organization" tab, untick the "Active" checkbox and save.

To delete an organization, go to "Forms > Organizations" and click the "Delete" icon ( ) next to it. You will be asked to confirm this action before the organization and all entities associated with it are irrevocably removed.

Copying an Organization

Transaction Manager allows you to quickly make a copy of an organization. This can be useful for testing purposes (setting up multiple copies of the same organization) or setting up new organizations from a common "template".
The new organization will receive the same configuration as the parent organization (with a few exceptions). You can also choose to include forms and their configuration in the copy operation.
To create a copy of an existing organization, go to "Forms > Organizations" and click the "Copy" button. You will be directed to the "Copy Organization" page.



Creating a copy of an organization
Choose the organization you want to copy from the "From Client" dropdown.
Enter the name and client code for the new client into the "Target Client Name" and "Target Client Code" fields. If an organization with the same name or code already exists, Transaction Manager will not allow you to create the copy - you need to choose unique values for both values.
If you want to copy forms and related configuration, tick the "Copy Forms" checkbox; otherwise, leave it blank.
Click the "Copy" button and wait for the operation to complete. Transaction Manager will create copies of all organization-specific entities and add references to global entities where necessary. Once the copy operation has completed, you can start using and modifying the new organization.
Note that while most aspects of organization and form configuration are copied, some entities are specifically excluded and will not be copied to the new

organization. These entities are: Payment configuration, report schedules, delivery details and form manager references. Also, the newly created organization always has the "Active" flag set initially.

Form Configuration (Manager v5.0)

Forms are the central entity in Transaction Manager. Forms can be configured to a great extent, and most settings are versioned. The configuration includes form properties (for example contact details), attachments, metadata values and delivery details. Some information such as property values can be inherited from the organization.

Search

Concepts

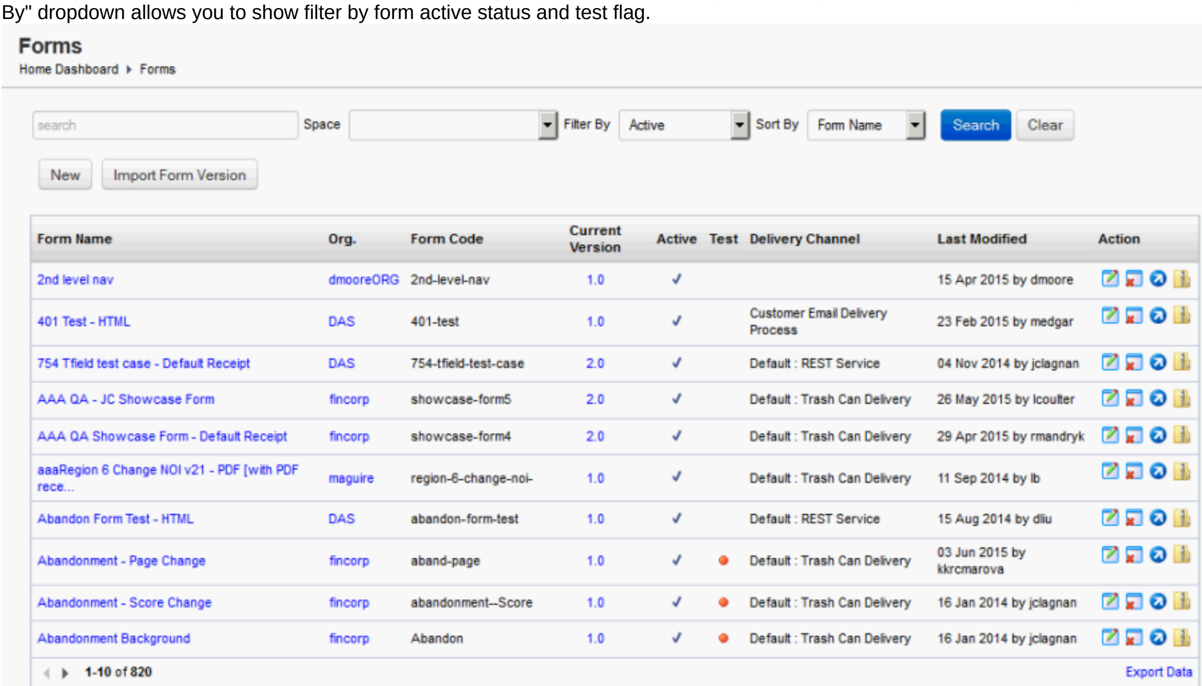
Forms are exposed to users by being hosted on a *user space*. Each form has at least one *form version*, an entity that wraps the set of form-related data to be versioned. If multiple form versions exist, one of them is distinguished as the *current form version*, and is used when rendering the form. There is some data shared by all form versions, but most settings are maintained separately for each form version.
Transaction Manager recognizes forms of different *form types*, such as dynamic PDF forms or Composer SmartForms. Each form version has a form type and the artifacts needed to render the form. The *form template file* is used to render the form and possibly the form receipt. The *receipt template file* can be used to render the receipt instead. The *form data configuration file* defines the data structure underlying the form, and is vital for dynamic SmartForms hosted on Transaction Manager.
Other versioned information includes *property values*, *metadata values* and *attachments* (supporting documentation that will be requested after form submission).

Form Types

There are four different form types supported by Transaction Manager: Dynamic PDF forms, Composer SmartForms, static PDF forms and other documents.
Dynamic PDF forms are XDP or PDF forms that are filled in and then submitted electronically to Transaction Manager. The organization may still require the form to be printed and sent in, but a record of form submission and the submitted data is also created on Transaction Manager.
Composer SmartForms are based on an XDP form but rendered as HTML in the user's browser (as opposed to being rendered as PDF in the user's PDF viewer plugin). These forms can also be rendered to devices (e.g. tablets or smartphones) where no PDF viewer plugin is available. Transact Composer offers extensive capabilities to create HTML forms that render on a variety of devices and screen resolutions.
Static PDF forms are simple forms that will be used for "fill and print", i.e. they will not be submitted to Transaction Manager after being filled in.
Other documents can be defined in Transaction Manager using the "Other Document" form type. The file extension of the uploaded document is used to determine the MIME type, and the document itself is streamed back to the user's browser.
When creating a form, you will have to specify which form type you will be using, and provide a file in the appropriate format (which will be described in the sections below).

Browsing Forms

To browse the list of all forms, go to "Forms > Forms". You will see a list of all forms that have been defined. You can filter the list by organization, a substring of the form name, the form code, the form template file name and / or by the user space on which the form is exposed. Additionally, the "Filter By" dropdown allows you to show filter by form active status and test flag.


Forms
Home Dashboard ▸ Forms

search | Space | Filter By: Active | Sort By: Form Name | Search | Clear

New | Import Form Version

Form Name	Org.	Form Code	Current Version	Active	Test	Delivery Channel	Last Modified	Action
2nd level nav	dmooreORG	2nd-level-nav	1.0	✓			15 Apr 2015 by dmoore	
401 Test - HTML	DAS	401-test	1.0	✓		Customer Email Delivery Process	23 Feb 2015 by medgar	
754 Tfield test case - Default Receipt	DAS	754-tfield-test-case	2.0	✓		Default : REST Service	04 Nov 2014 by jclagnan	
AAA QA - JC Showcase Form	fincorp	showcase-form5	2.0	✓		Default : Trash Can Delivery	26 May 2015 by lcoulter	
AAA QA Showcase Form - Default Receipt	fincorp	showcase-form4	2.0	✓		Default : Trash Can Delivery	29 Apr 2015 by rmandryk	
aaaRegion 6 Change NOI v21 - PDF [with PDF rece...	maguire	region-6-change-noi-	1.0	✓		Default : Trash Can Delivery	11 Sep 2014 by lb	
Abandon Form Test - HTML	DAS	abandon-form-test	1.0	✓		Default : REST Service	15 Aug 2014 by dliu	
Abandonment - Page Change	fincorp	aband-page	1.0	✓	●	Default : Trash Can Delivery	03 Jun 2015 by kkrcmarova	
Abandonment - Score Change	fincorp	abandonment--Score	1.0	✓	●	Default : Trash Can Delivery	16 Jan 2014 by jclagnan	
Abandonment Background	fincorp	Abandon	1.0	✓	●	Default : Trash Can Delivery	16 Jan 2014 by jclagnan	

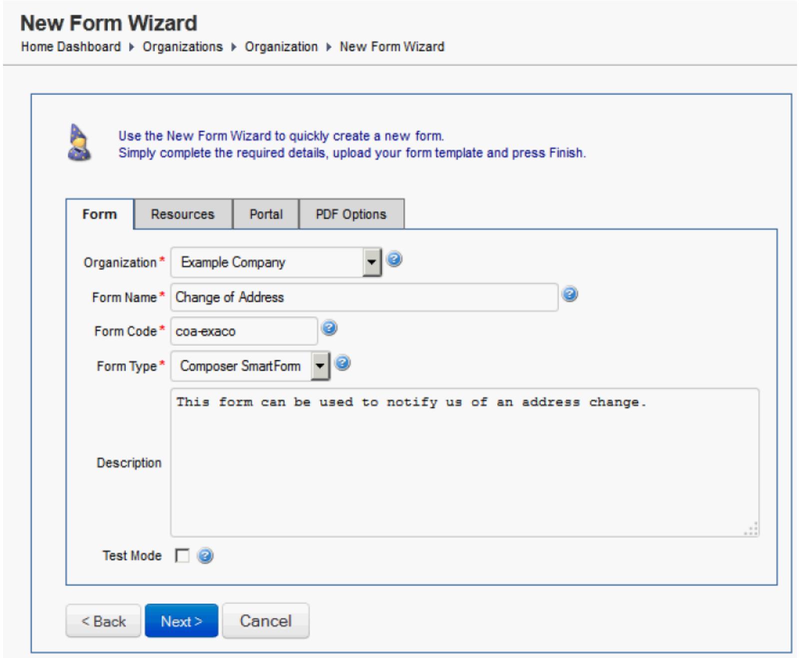
1-10 of 820 | Export Data

Browsing the list of forms
This page also allows you to invoke the "New Form" wizard and import form archives exported from another Transaction Manager instance.

You can also edit existing forms ( ), delete forms ( ) and export ( ) forms. Please see Multi-Environment Support (Manager v5.0) #_sec_migratingform for exporting and importing form archives.

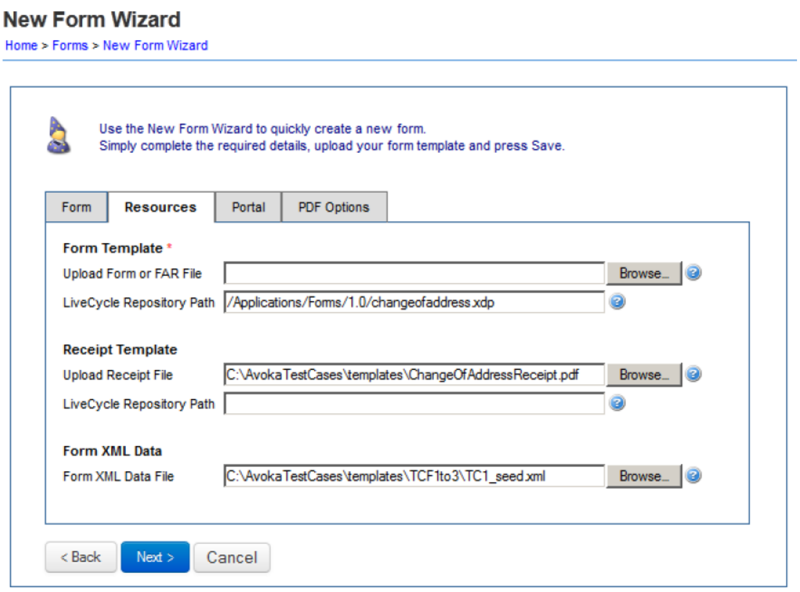
Creating a Form Using the "New Form" Wizard

The "New Form" wizard is a tool that allows you to create new forms with a minimum of effort. The wizard asks for some basic information and then creates a new form, with a form version and form data configuration.
To use the "New Form" wizard, you need to have the organization owning the form set up first (see Organization Configuration (Manager v5.0)#_sec_createorganization on how to do this). Next, go to "Forms > New Form Wizard" (or click the "New" button on the "Forms" page) to open the wizard.



The "New Form" wizard

Choose the organization for which the form will be set up from the "Organization" dropdown.
Enter the form name into the "Form Name" field. Assign the form a unique form code, which can be any string (e.g. "cofa-examc") containing letters, numbers, dashes, underscores and spaces.
Choose the form type (described in _sec_formtypes) from the "Type of Form" dropdown.
Enter an optional form description, which will be displayed to space users.
By default, forms start out in production mode (making them visible on the example spaces and T.Field). If you would like to set your form into test mode, tick the "Test Mode" button.
Next, switch to the "Resources" tab.



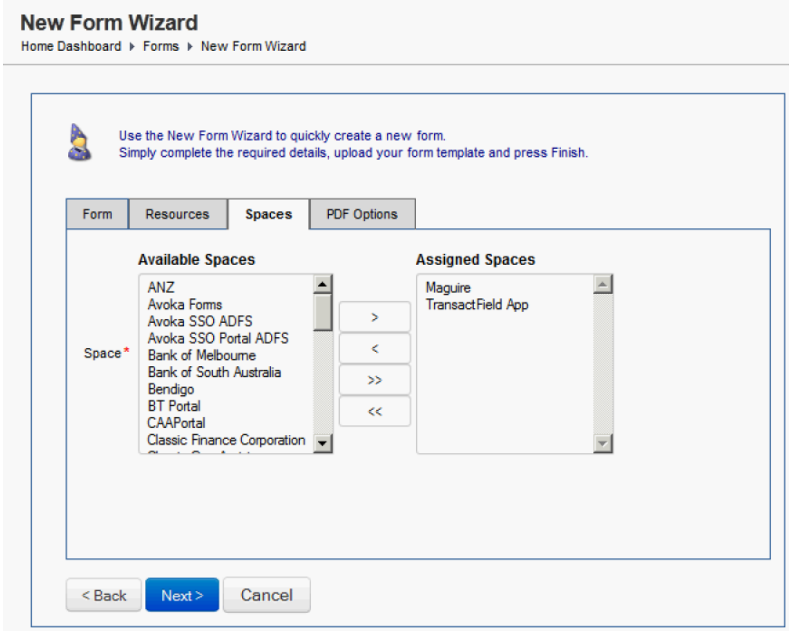
Defining the resources for a form

You need to specify a template file, which contains the actual form as a PDF or XDP template. Please refer to the table below for the file formats supported by the various form types.

Form type	Expected file type
Dynamic PDF form	XDP or PDF
Composer SmartForm	Transact Composer FAR file
Static PDF form	PDF
Other document	Common document formats such as Word documents

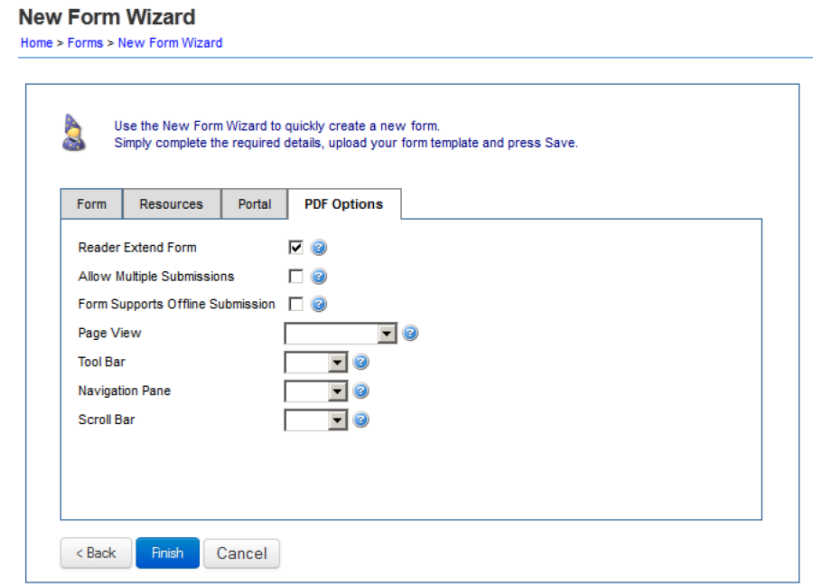
Instead of uploading a form, you can also enter the URL to the form in the Adobe LiveCycle repository into the "LiveCycle Repository Path".

If you would like the receipt to be based on a different form template, you can upload a file or enter a path in the Adobe LiveCycle repository into the "Receipt Template" section. If a receipt template is defined, Transaction Manager will use it to render the receipt, passing in the submitted form XML.
If you would like to use prefill for the form, you also need to upload an XML seed file in the "Form XML Data" section.
Transaction Manager will automatically create a form version and form data configuration based on these files.
Next, switch to the "Space" tab (shown below) and choose the user space(s) that will be used to host your form. The form will be available on all selected spaces. Note: Only spaces available to the selected organization will be shown. For information on how to view or modify the spaces for an organization, please refer to Organization Configuration (Manager v5.0)#_sec_organizationspaceaccess.



Assigning a new form to user spaces

The "PDF Options" tab allows you to specify some options on how the form will be displayed in Adobe Reader, such as which tool bars will be shown and how the form will be fitted to the available space (see also Form Version Configuration (Manager v5.0)#_sec_versionpdfoptions).

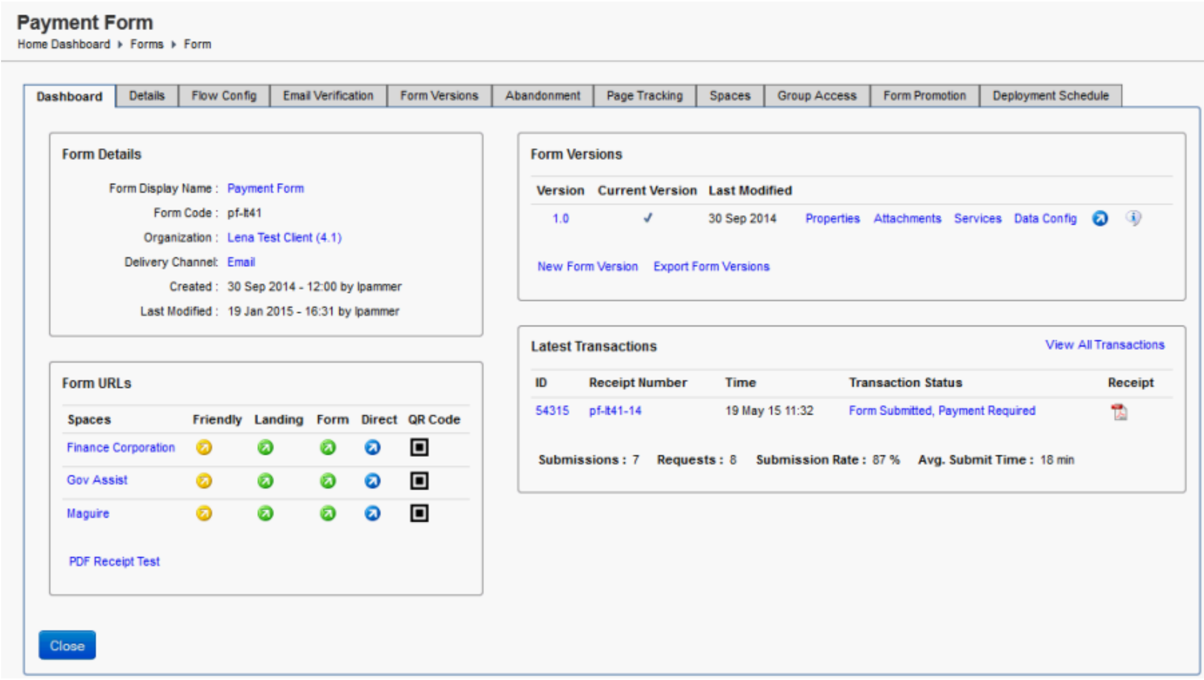


Setting PDF render options for a new form

By default, the "Reader Extend Form" checkbox is ticked, causing the form to be reader-extended at render time. For this to work, you will need to set up an Adobe Reader Extensions certificate on LiveCycle as well (see the Transaction Manager Installation Guide for detailed instructions).
If your form needs to support multiple submissions based on the same form request, tick the "Allow Multiple Submissions" checkbox. This will allow a user to save a form and submit it multiple times.
Save the form wizard to create your new form. The form wizard will assume default values for some fields, but will basically generate a renderable form in test mode. The form is ready to be customized (e.g. form properties, attachments, delivery details) but can also be tested right away.
After saving the form, you will be taken to the form detail page. The following sections describe the page in detail.

Editing a Form

To edit a form, go to "Forms > Forms", find your form and click the "Edit" icon next to it ( ). You will see the form dashboard, which shows an overview of the form configuration as well as the latest transactions for the form.
The data shown depends on the permissions assigned to the administrator. The screenshot below shows an example of the form dashboard viewed by a user who has all applicable permissions.
Note that all configuration that can be done via the dashboard can also be accessed via other paths; the purpose of the dashboard is to provide an overview and convenient shortcuts for administrators.



The form dashboard

The form dashboard consists of four sections:

The "Form Details" section shows key information about the form, such as the form name, organization, test mode and last modified date. Some items contain links to other entities (e.g. organization), which you can use to edit these entities directly. However, keep in mind that these entities may be used by many other forms, so be careful when making changes.

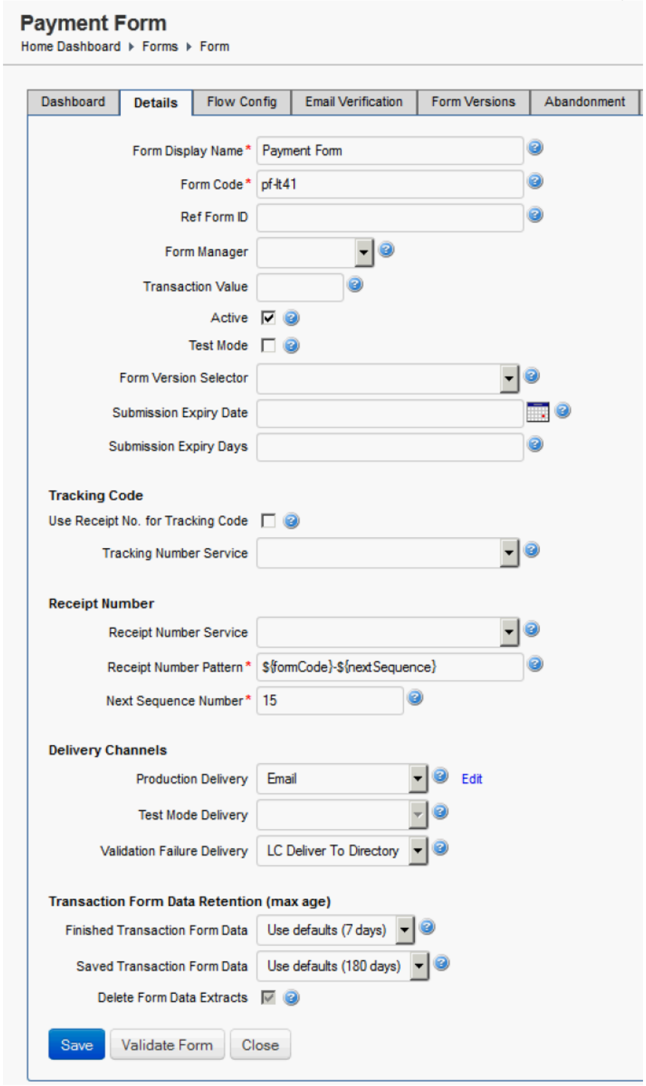
The "Form Versions" section shows the list of versions set up for the form. You can jump directly into various configuration pages (e.g. form version properties) for a version by using the links next to each version.

The "Form URLs" section shows what spaces the form is currently hosted on, and provides the relevant links to the space pages where the form can be accessed.The "Landing Page" link points to the form's landing page, where users can review a form description and log in, if applicable.The "Friendly URL" link points to the form's landing page, but uses a "user-friendly" URL format.The "Form Page" link points to the space's form page, which will render the form.The "Direct Render" link points directly to the space's form render servlet.

Finally, the "Latest Transactions" section shows the most recent submissions made for the form along with their status. In addition, some statistics about the set of completed form submissions are displayed. It may take a few minutes for the statistics to update.

Form Details

The "Details" tab contains basic form attributes. Note that these settings are not versioned.



Editing form details

The "Form Display Name" contains the form name, which needs to be unique within the organization. The form name is displayed to space users and identifies the form.

The "Form Code" uniquely identifies the form across all organizations. It is an alphanumeric short string. The form code is not generally displayed to space users.

If there is a form identifier that is used internally, you can enter it into the "Ref Form ID" field. Transaction Manager will pass this value into web service calls to external systems (e.g. to obtain prefill data or create receipt numbers), if these calls are set up.

You can also set a dedicated form manager by choosing an existing administrator user from the "Form Manager" dropdown. This user will be notified via email whenever the form's promotion status changes (see _sec_formpromotion for more on form promotion).

The "Active" checkbox should be ticked initially to indicate the form is in use. If it is unticked, the form is treated as soft-deleted and will not be rendered by Transaction Manager. Untick the checkbox to retire a form without deleting it.

The "Test Mode" flag indicates whether the form is in test mode. If it is unticked, the form is in production mode and submissions for the form will be marked as live submissions. Note that the Example Self-Service Space and T.Field do not display test forms in the list of forms.

The "Production Delivery" and "Test Mode Delivery" fields control the way form submissions are delivered to the organization. There are multiple supported delivery methods, including web service delivery, delivery to an Adobe LiveCycle process and email delivery. Please refer to Delivery Configuration (Manager v5.0)#_sec_deliverymethods for more information. Note that if the form is in test mode, the "Test Mode Delivery" setting will be used; otherwise the "Production Delivery" setting will be used to deliver submissions.

The "Validation Failure Delivery" setting will be used to deliver submissions for which a service of type "Submission Data Validator" was defined and found validation errors when the user submitted the form (see Services (Manager v5.0)#_sec_submissiondatavalidatorservice).

There are two fields that control form expiry. You can have a form expire *n* days after the user first opened it by setting "Submission Expiry Days" to a positive integer *n*. This prevents a user from saving a form and then submitting it much later, potentially even after a form version change. Alternatively, you can set a fixed expiry date by entering it into "Submission Expiry Date". This can be useful if your form has to be submitted before a fixed deadline.

The "Form Version Selector" dropdown allows you to configure a specific form version selector service for this form; otherwise the system default service will be used. Please refer to Services (Manager v5.0)#_sec_formversionselectorservice and the Groovy Services Guide in the management console for more about form version selector services.

The fields in the "Transaction Form Data Retention (max age)" section concern form specific data retention policies and will be discussed in Operations (Manager v5.0)#_sec_dataretentionformlevel.

The fields in the "Tracking Code" and "Receipt Number" sections concern the generation of tracking and receipt numbers for each submission of the form and will be discussed in detail in the following section.
Once you are done editing the form, click "Save".

Tracking Code and Receipt Number

Transaction Manager provides multiple ways to generate unique tracking numbers for form submissions: A unique alphanumeric tracking code is generated by default when a submission is created, and a customizable receipt number is assigned when the user saves or submits. By default, the tracking code is used throughout (though you can configure a form to use the receipt number in its place), and the receipt number is merely written into the receipt where it can be displayed.

Configuring the Tracking Code

To configure the tracking code for a form, edit the form and switch to the "Details" tab. The "Tracking Code" section contains the relevant settings (_img_formdetails).
Ensure that "Use Receipt No. for Tracking Code" is not ticked (otherwise the receipt number will be used throughout).
The "Tracking Number Service" dropdown can be set to explicitly specify the service that will generated the tracking number. If left blank, the system default service for service type "Tracking Number" will be used.

Configuring Receipt Numbers

To configure the receipt number for a form, edit the form and switch to the "Details" tab. The "Receipt Number" section contains the relevant settings.
Note that by default, the receipt number is not displayed to the user (though the form can elect to display it on the receipt).
If you would like to use the receipt number in place of the tracking code, tick the "Use Receipt No. for Tracking Code" checkbox.
You can configure multiple ways to generate the receipt number for a submission. The following sections list the various options built into TM.

Using a Pattern to Generate Receipt Numbers

To use a pattern as the basis of receipt number generation, go to "Forms > Forms", edit the form and switch to the "Details" tab. The "Receipt Number" section contains fields relevant to receipt number generation (_img_formdetails).
Select "Sequence Pattern Receipt Number" from the "Receipt Number Service" dropdown to use the default pattern receipt number generation service. If you leave the value blank, the system default service will be used.
The "Receipt Number Pattern" field contains the current receipt pattern. This pattern defines the string template that Transaction Manager will base the receipt numbers on. You can use arbitrary strings, interspersed with the following placeholders (omit the quotes when using them):

- "${formCode}"This placeholder will be replaced with the client form code.
- "${submissionId}"This placeholder will be replaced with the globally unique numeric submission OID.
- "${submitKey}"This placeholder will be replaced with the alphanumeric UUID created for the submission.
- "${submissionTime}"This placeholder will be replaced with the date/time when the form was first submitted or saved to Transaction Manager. The date/time can be further formatted by using "${submissionTime.format(<patternString>)}", where <patternString> is a Java SimpleDateFormat pattern string (for further information, see http://docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html).
- "${nextSequence}"This placeholder will be replaced with the sequence number that is maintained for the current form.
- ${format.decimal(...)}This notation allows you to format numbers to a fixed number of digits using leading zeroes. The format object is of type org.apache.click.util.Format.Example: ${format.decimal($submissionId, "00000000")}

To ensure your receipt numbers are globally unique, make sure the pattern contains either the submission ID or the submit key, which are globally unique, or both the client form code and the sequence number.
The "Next Sequence Number" field contains the sequence number that the next submission will be assigned (replacing "${nextSequence}"). If needed, this field can be increased; however, to avoid creating non-unique values, you will not be able to change this field to a value already assigned to a submission. We recommend leaving the "Next Sequence Number" field unchanged.
Pattern Examples:

- Sequence with date: "${formCode}${nextSequence}-${submissionTime.format("dd/MM/yy")}"
  This will generate receipt numbers such as "coa17-05/07/2012". Uniqueness is assured because the client form code as well as the sequence number are used.
- PIN: "${submissionId}-${submitKey.substring(0,4)}"
  This will generate receipt numbers such as "423-f94e". These short receipt numbers are useful for forms that can be saved to Transaction Manager anonymously, as the user must provide the receipt number (and possible pass additional challenges) before to reaccess their saved form. While you could also use the submission OID by itself, this is not as secure since submission OIDs are generally simply ascending numbers and so could easily be guessed. Combining the OID with part of the randomly generated submit key ensures both uniqueness and obscurity.
- Padded receipt number: ${formCode}-$format.decimal($nextSequence, "00000000")
  This will generate receipt numbers such as "coa-00017530" (assuming nextSequence is 17530).

Using a Web Service to Provide Receipt Numbers

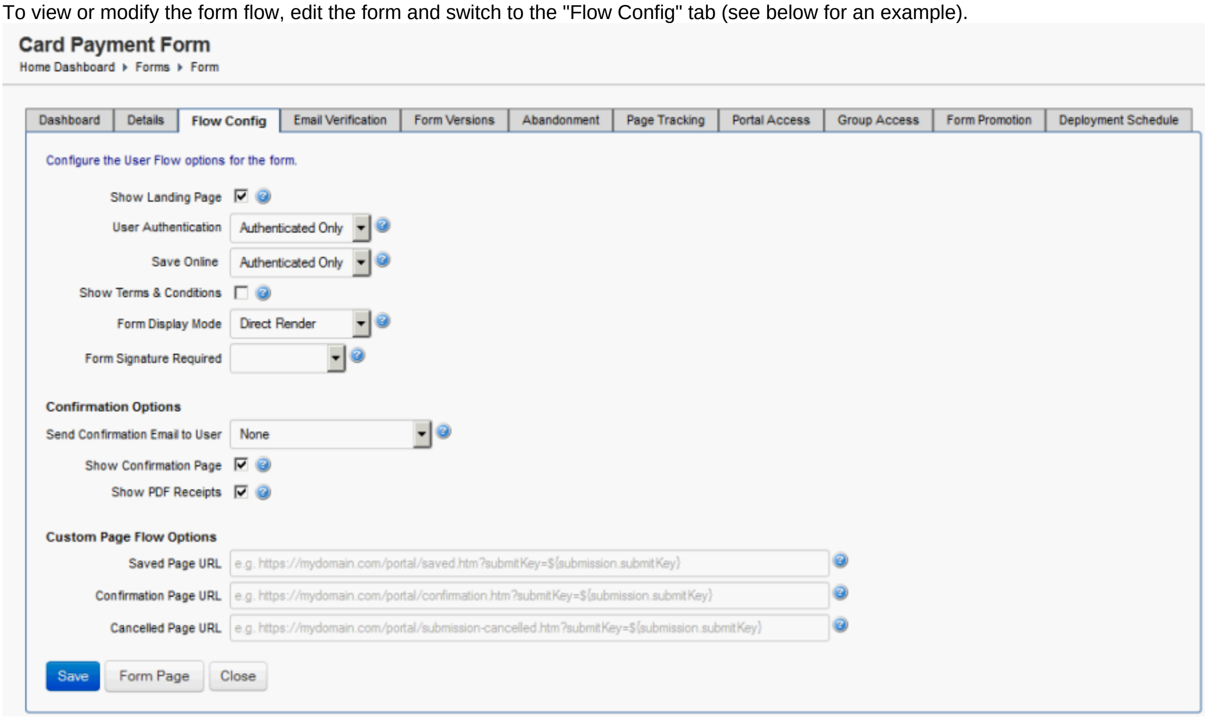
If you wish to use a web service to provide receipt numbers, please refer to the Transaction Manager Web Service API document for details about the web service API.
Once you have provided the web service, you will need to edit the "Web Service Receipt Number" service definition and adjust the "webServiceAddress" service parameter to point to the endpoint of the web service. Please refer to Services (Manager v5.0)#_sec_managingservicedefinitions for more on service definitions and parameters.
Now you can configure your forms to use the receipt number web service. To do this, go to "Forms > Forms", edit the form and ensure you are on the "Form" tab.
Select "Web Service Receipt Number" from the "Receipt Number Service" dropdown.
Leave the other fields unchanged and save.
From that point onwards, Transaction Manager will call the web service whenever a form is submitted to obtain the receipt number.

Using a Groovy Script to Create Receipt Numbers

Transaction Manager allows you to specify a Groovy script that calculates the receipt number for a submission. Given the form, submission and submitted XML, you can implement your own logic to provide a unique receipt number. The Groovy script simply needs to return the receipt number.
To edit the Groovy script, you will need to edit the "Groovy Receipt Number" service definition and adjust the "groovyScript" parameter. Please refer to Services (Manager v5.0)#_sec_managingservicedefinitions for more on service definitions and parameters, and contact your Avoka contact assistance with the Groovy script, if desired.
Once you are done, you can configure your forms to use the Groovy receipt number service. To do this, go to "Forms > Forms", edit the form and ensure you are on the "Form" tab.
Select "Groovy Receipt Number" from the "Receipt Number Service" dropdown.
Leave the other fields unchanged and save.

Flow Configuration

Depending on a form's configuration, the steps required of and options available to an applicant filling in the form can vary to a significant degree. For example, a simple form may launch from a landing page and show a confirmation page after submission. More complex forms may add additional steps and options, both prior to fill-in (e.g. terms & conditions page, allowing the user to authenticate) as well as after (e.g. saving and returning to form drafts, payment, attachments, signatures, availability of a receipt).
Transaction Manager allows you to customize this "form flow" by customizing the steps an application for a specific form will go through. The form flow is configured on a form level and is therefore not versioned.
To view or modify the form flow, edit the form and switch to the "Flow Config" tab (see below for an example).



Editing the flow configuration of a form

The options on this page are listed in chronological order from a user's perspective when going through the application process.
Note: If a custom user space is being used to host this form, not all form flow options may be supported. This section assumes that a fully features user space is deployed.
You can choose to show or hide the form landing page. This page generally contains content describing the form, and allows users to log in or continue to the form anonymously.
The "User Authentication" dropdown can be used to restrict access to the form. If left blank, anyone can access the form; if set to "Authenticated", only users logged on as an TM space user can access the form; and if set to "Anonymous", anyone can access the form, but the submission is always treated as anonymous (i.e. no user is recorded against the submission even when the user is logged in). Additionally, if "Authenticated" is selected, users are also prompted to re-authenticate at form submission time if necessary (e.g. browser session timeout, offline submission), and submissions will always be associated with a TM user account.The options available to the user on the landing page will reflect the choice you make.
The "Save Online" dropdown controls whether the form can be saved as a draft to TM, and by whom. The options are "Disabled", "Authenticated" (authenticated users only) and "Anonymous" (both anonymous and authenticated users). Note: If a form was saved anonymously, it can still be accessed by authenticated users if they can pass the save challenge(s). This is useful when users share submissions or ask an administrator for help in completing their submission.
The "Show Terms & Conditions" checkbox controls whether a page is shown in between the landing page and the actual form. The content of this page is typically a list of terms and conditions the user must accept and can be customized via the "Form Terms And Conditions" form property (it can also be defined as a space property for an entire user space, though the form property takes precedence).If you do not need to show a separate terms and conditions page to your users, leave the checkbox unticked.
The "Form Display Mode" dropdown allows you to control how the form is rendered. It can be rendered directly, or embedded in a wrapping page either inside a div or an iFrame. Do not change this setting unless you specifically need to as your form may not render as intended if the wrong setting is chosen.
The "Form Signature Required" dropdown controls whether TM will ask the users to sign their submission once other work has been completed. You can enable two types of signatures: "Wet" for physical wet ink signatures and eSignatures using a digital certificate. Please refer to Signatures (Manager v5.0) for detailed information.

Confirmation Options

The "Confirmation Options" section allows you to customize what happens when the user completes a submission for this form.
The "Send Confirmation Email To User" dropdown allows you to configure automatic emails to the user submitting the form when they complete the submission (including all user steps such as payment and attachments). Choose "None" to disable automatic confirmation emails (the example space allows the user to send an email if they so choose), "Confirmation" to send just a notification email without attachments and "Confirmation and PDF Receipt" to send an email that includes the PDF receipt.
You can customize the email subject and message templates by defining values for the "Email Form Receipt Subject" and "Email Form Receipt Message" on the form version, organization and/or space.

The "Show Confirmation Page" checkbox controls whether a summary page shall be shown to the user after they have successfully completed a submission. The confirmation page can be used to display a submission message to the user, display the receipt number, allow the user to access a receipt and provide further actions to the user (e.g. review their submission history, fill in another form etc.).If unchecked, the receipt for the submission will be displayed to the user instead.
Finally, the "Show PDF Receipts" checkbox controls whether space users should be offered a receipt when they submit this form. If unticked, they will not have access to a receipt (whether from the confirmation page or their submission history).Note that if "Show Confirmation Page" is unticked, the receipt will still be shown to the user on submission completion.

Custom Page Flow

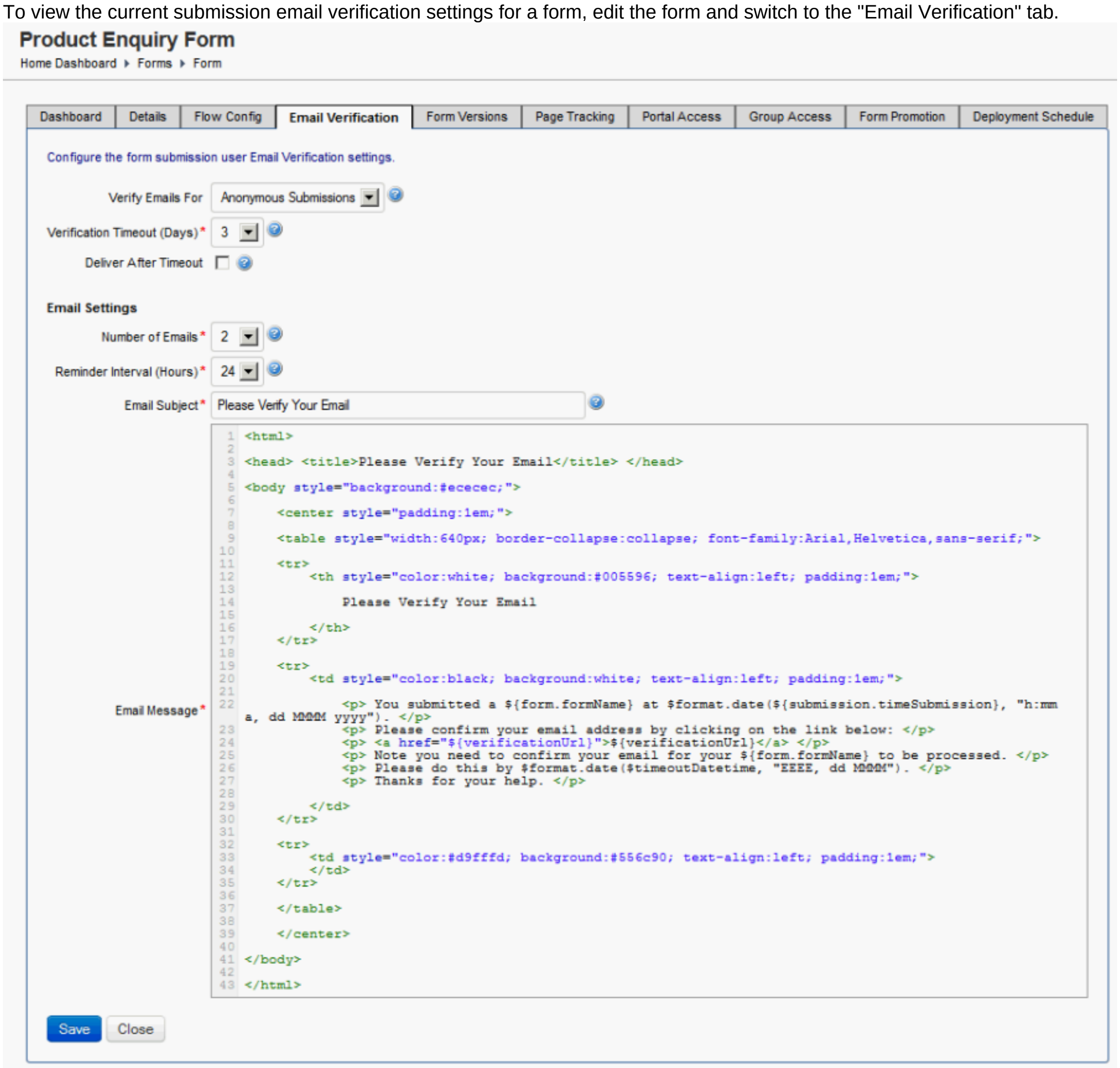
Normally, TM takes care of the form flow, including all post submission steps such as showing a confirmation page. If you would like to provide custom pages instead, you can configure custom URLs to redirect to when a submission for this form is saved, completed or cancelled.
These URLs can be entered in the "Custom Page Flow Options" section. You can use the following Velocity variables in the URLs:

- $submissionThe submission object. For example, ${submission.submitKey} would be substituted with the submit key (a GUID).
- $formDataMapThe map of submission data extract values. For example, ${formDataMap.firstName} would be substituted by the value in the map for the key "firstName".
- $requestThe HTTP servlet request.

Save your changes. If you have entered custom page URLs, TM will use them instead of redirecting to the corresponding space page.

Submission Email Verification Settings

Transaction Manager supports two forms of user email verification: At account registration time, users can be asked to verify their email address by clicking on a link in an email sent to them. Additionally, you can ask users who have just submitted a form to verify their email address (= submission email verification). This allows you to ensure that user-entered email addresses are valid and belong to the user even for anonymous users. This section describes how to configure submission email verification (for email verification during user enrollment, see Security (Manager v5.0)#_sec_customizinguserenrolment).
Note: For submission email verification to work, a contact email address must have been stored for the submission. This can be achieved by configuring the "Contact Email XPath" configuration mapping in the form data configuration set up for the current form version and ensuring in the form that the email address cannot be blank at submission time (note that this works the same way even for authenticated users – the email address stored against their user account is not used for submission email verification). Please refer to Form Version Configuration (Manager v5.0)#_sec_configurationmappings to learn about configuration mappings.
To view the current submission email verification settings for a form, edit the form and switch to the "Email Verification" tab.



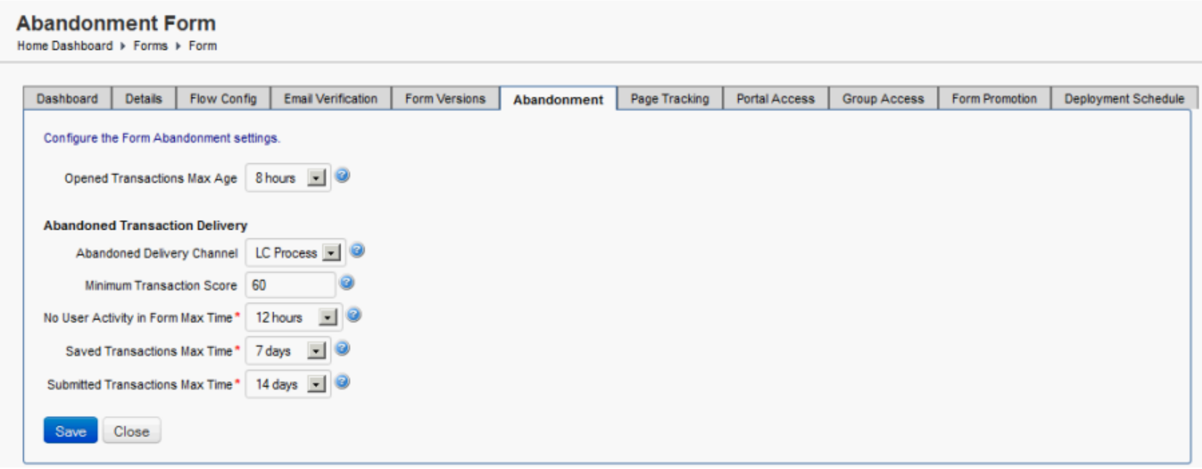
Configuring submission email verification for a form

To turn on email verification, set the "Verify Emails For" dropdown to "Anonymous Submissions" or "All Submissions".
The "Verification Timeout (Days)" field allows you to specify how much time you want to give the user to complete email verification. Note that if this time passes without user response, the submission is then either considered abandoned or delivered anyway (see below).

Tick "Deliver after Timeout" if you consider submission email verification to be non-mandatory, and the submission should be delivered even if the user does not complete email verification. Also note that this means that delivery for such submissions will occur only after email verification has timed out; therefore, it is important to set the verification timeout in such a way that timed out submissions are delivered within an acceptable timeframe.
As to the verification email itself, you can customize the content, number of times it will be sent if the user has not completed email verification and the interval between emails. The first email will be sent when the user submits the form.
Choose the maximum number of verification emails that shall be sent to the user from the "Number of Emails" dropdown (TM stops sending emails if the user completes email verification, or if the verification timeout is reached).
Choose the interval between emails from the "Reminder Interval (Hours)" dropdown.
Customize the subject and message sent to the user by adjusting the "Email Subject" and "Email Message" fields. Note: Ensure the email always contains the URL the user needs to access to complete submission email verification (the placeholder is "${verificationUrl}").
Once you save your changes, they will apply to all subsequent submissions for the form in question.

Abandonment Settings

You can customize when and how submissions for a form will be automatically marked as abandoned, and what will happen to abandoned submissions. To do so, edit the form and switch to the "Abandonment" tab.

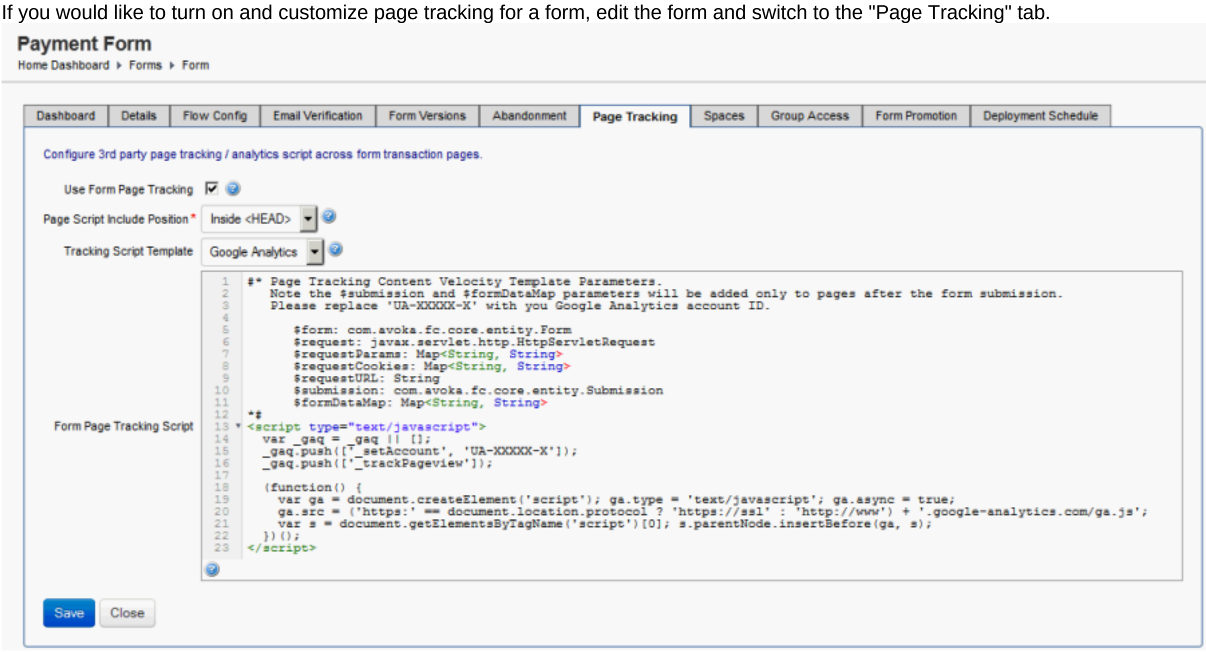


Abandonment settings

The "Opened Transactions Max Age" setting determines how soon a submission in form status "Opened" (= opened by the user, but not submitted or saved to TM) will be automatically set to "Abandoned" by the system.
All other settings are related to abandonment delivery. This feature allows you to mark submissions in other form statuses as abandoned when they reach a certain age, and deliver these abandoned submissions to a separate delivery channel. The organization can then review these submissions and decide if any of them should be followed up to help the user complete their application.
To enable abandoned transaction delivery, select a delivery channel from the "Abandoned Delivery Channel" dropdown. This will enable the other fields in this section.
If you would like only submissions that exceed a certain transaction score (a numerical value calculated by the form, generally based on how much of the form was filled in) to be delivered, you can enter the minimum score into the "Minimum Transaction Score" field.
You can customize the timeout for opened forms for which a background save was performed by setting the "No User Activity in Form Max Time" field.
The maximum age for saved and submitted (but incomplete, e.g. due to missing attachments) transactions can be controlled via the "Saved Transactions Max Time" and "Submitted Transactions Max Time" fields.
When you are satisfied with the settings, save your changes.

Page Tracking Settings

To provide support for external analytics and page tracking (e.g. Google DoubleClick), TM allows you to specify custom script that will be appended to all space pages relating to the form flow.
If you would like to turn on and customize page tracking for a form, edit the form and switch to the "Page Tracking" tab.



Configuring page tracking for a form

The "Use Form Page Tracking" checkbox controls whether page tracking is enabled for this form.
To enable page tracking, tick this checkbox and customize the page tracking settings (TM provides several script templates including Google DoubleClick).
Save your changes.

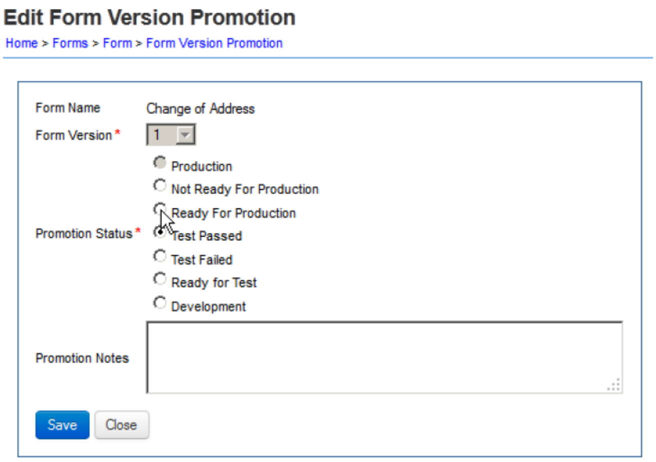
Promotion

The form version promotion functionality allows the organization to progress their forms from development to production, with the ability to add notes and set up different users with different permissions (e.g. developers may be allowed to send a version to test, but not promote it any further than that).
Promotion applies to a particular form version. If a form has multiple versions, they are promoted independently of each other.

Promoting and Demoting a Form Version

To promote a form version along its life cycle, go to "Forms > Forms", edit the form and switch to the "Promotion" tab. You will see a list of promotion entries for various form versions.

If there is not yet a promotion record for a form version, click the "New" button. Otherwise, click the "Edit" icon ( ) next to the record. You will be directed to the "Edit Form Version Promotion Log" page (the example shows a form version being promoted to "Ready For Production").

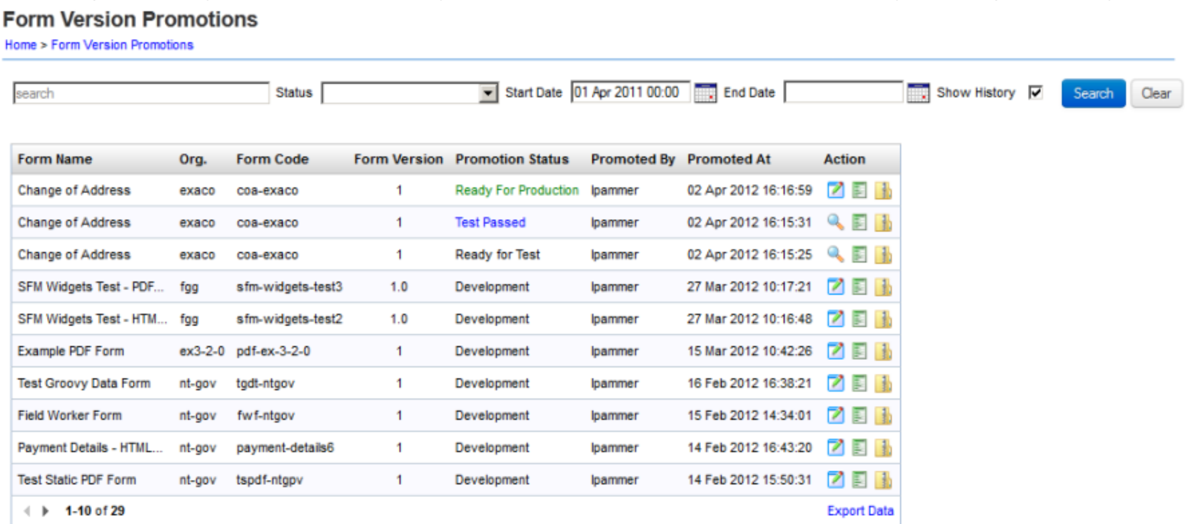


Promoting a form version

Choose the form version you want to promote from the "Form Version" dropdown. You can also add promotion notes.
To promote the form version, choose one of the available radio buttons in "Promotion Status". You can promote only a single step at a time, which helps set up a permission system with several interacting roles such as developer, tester and project manager.
Save the promotion log and verify that it shows in the list of promotion log entries for the form.
After a form version has been promoted, notification emails will be automatically sent out to all organization and global administrators who have the permissions to promote the form to the next level. For example, if a form is promoted to "Test Passed", all administrators who can promote this form to "Ready For Production" will be sent an email. Requirements for receiving promotion emails are being a member of the "Receive Promotion Alerts" group (see Security (Manager v5.0)#_sec_predefinedgroups) as well as having an email address set up in the administrator details.
Please note that you can also set a dedicated form manager for each form. This administrator will receive a notification whenever the promotion status changes (see _sec_editform on how to set the form manager).
You can also demote a form version by setting it back to a lower level. After saving, the promotion level will be updated, and only the next level up is available as the next promotion step.

Viewing the Promotion Log

To view the promotion log entries across all forms, go to "Forms > Form Version Promotions". You will get a list of promotion log entries.



Form Name	Org.	Form Code	Form Version	Promotion Status	Promoted By	Promoted At	Action
Change of Address	exaco	coa-exaco	1	Ready For Production	lpammer	02 Apr 2012 16:16:59	
Change of Address	exaco	coa-exaco	1	Test Passed	lpammer	02 Apr 2012 16:15:31	
Change of Address	exaco	coa-exaco	1	Ready for Test	lpammer	02 Apr 2012 16:15:25	
SFM Widgets Test - PDF...	fgg	sfm-widgets-test3	1.0	Development	lpammer	27 Mar 2012 10:17:21	
SFM Widgets Test - HTM...	fgg	sfm-widgets-test2	1.0	Development	lpammer	27 Mar 2012 10:16:48	
Example PDF Form	ex3-2-0	pdf-ex-3-2-0	1	Development	lpammer	15 Mar 2012 10:42:26	
Test Groovy Data Form	nt-gov	tgdt-ntgov	1	Development	lpammer	16 Feb 2012 16:38:21	
Field Worker Form	nt-gov	fwf-ntgov	1	Development	lpammer	15 Feb 2012 14:34:01	
Payment Details - HTML...	nt-gov	payment-details6	1	Development	lpammer	14 Feb 2012 16:43:20	
Test Static PDF Form	nt-gov	tspdf-ntgpv	1	Development	lpammer	14 Feb 2012 15:50:31	

Viewing the promotion log

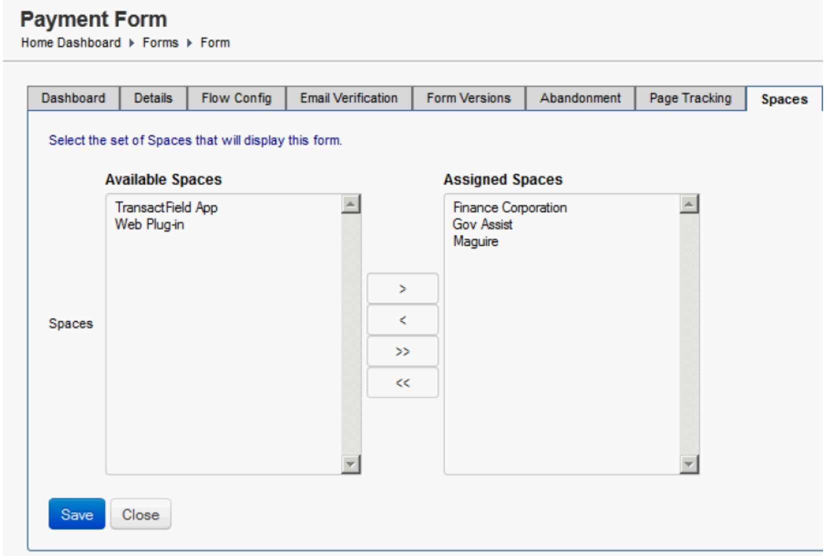
This page shows the latest log entry for each form version by default; if you are interested in seeing previous promotions as well, tick the "Show History" checkbox.
You can also filter by promotion status and a substring of the form name, and export the search results to Excel by clicking the "Export Data" link below

the table.

To promote or demote a form version, you can click the edit icon ( ) to open the "Edit Promotion Log" page as described in the previous section. Note that historic entries that have been superseded by a more recent promotion log entry for the same form version cannot be edited.

Form Space Access

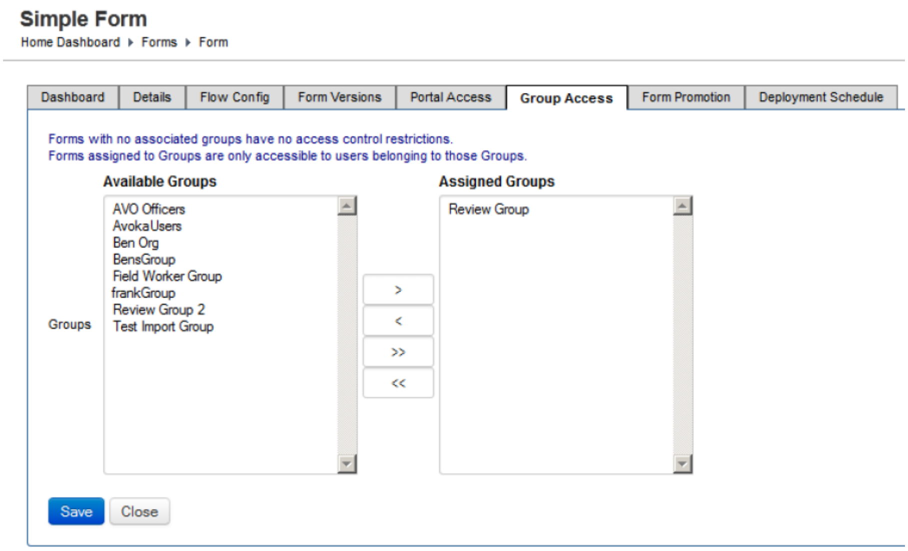
For your users to be able to access a form, it must be hosted on one or more user spaces (see Spaces (Manager v5.0) for the various types of spaces supported in Transaction Manager).
For each form, you can choose which of the available spaces it will be hosted on. To do so, edit the form and switch to the "Spaces" tab.



Changing the spaces a form is hosted on
Note: Only spaces available to the organization owning the form will be shown. For information on how to view or modify the spaces for an organization, please refer to Organization Configuration (Manager v5.0)#_sec_organizationspaceaccess.
Add your form to the desired spaces to the form by selecting them and clicking ">"; similarly, remove your form from a space by selecting it and clicking "<".
When you are done, click "Save".

Form Group Access

By default, new forms can be accessed by any space user. To restrict access rights to a certain set of users, forms can be assigned to a number of form work groups. If a form is assigned to at least one form work group, only space users associated with one of the same form work groups can access the form.
You can create a suitable set of form work groups (see Security (Manager v5.0)#_sec_addgroup). In addition, if LDAP authentication is used, the user's LDAP groups can be loaded and used as the set of that user's form groups (the matching between LDAP groups and form groups is done by name). Please refer to Security (Manager v5.0)#_sec_securitymanagers for more details.
To view or modify a form's group associations, go to "Forms > Forms", edit the form and switch to the "Group Access" tab.



Changing a form's group associations
The "Assigned Groups" field shows all form work groups that the form is currently associated with. To make the form accessible to all users of the hosting user space, ensure that no form work groups are selected.
You can assign the form to or remove it from existing form work groups by using the ">" and "<" buttons. You can also use the ">>" button to assign the form to all form work groups, or the "<<" button to remove it from all form work groups.
Once you have completed form work group assignment, save your changes by clicking "Save".

Removing a Form

Deleting a form is a dangerous operation as it affects many other related entities. If possible, forms should be made inactive rather than be deleted, especially if there have been submissions for the form.
To make a form inactive, go to "Forms > Forms" and edit the form. Untick the "Active" checkbox and save the form.
If you wish to delete a form, be aware that this will delete all form versions and submissions as well as related entities for the form.

To delete a form, go to "Forms > Forms", find the form you want to remove and click the "Delete" icon (? Unknown Attachment). You will be asked to confirm this action before the form and all related entities are irrevocably deleted.

Rendering a Form

You can render a form quickly using Transaction Manager. The form will appear according to its configuration (organization and form properties, attachments etc.) but no prefill based on the user will occur.
You normally render a form to quickly view its appearance. If you want to test its behavior, it may be better to use the "Test" function (described in the next section), which gives you a form fill-in experience identical to that of a user space user.

To render a form, go to "Forms > Forms" and click the "Render" icon ( ) next to the form. The form will be loaded in a new window and can be filled in normally.

Testing a Form

Testing a form is similar to rendering it, however the form request will be treated as if a Transaction Manager user had requested a new form, i.e. you will be directed to the landing page, where you may get additional options depending on the user space (such as logging on). The landing page will take you to the form, which you can then fill in, submit and make payments or attach supporting documentation as needed.
You would test a form via the Transaction Manager management console if you wanted to validate its behavior. You can also test forms by finding and opening them on the user space; however, this may show you only active forms in production mode (depending on how the space is implemented). Therefore, if you need to test forms that are in test mode, you should use the method described below.

To test a form, edit the form, ensure you are on the "Dashboard" tab and click the "Landing Page" link ( ) next to the space you want to test on. The landing page for the form will be loaded in a new window.

Testing a Form Using Friendly URLs

When testing a form as described in the previous section, the URL to the landing page will look something like: http://www.host.com/tmportal/landing.htm?formCode=lf-exp
Transaction Manager also provides "friendly" URLs as an alternative way to access the same page. These URLs look like: http://www.host.com/tmportal/exp/local-form/
which uses the client code ("exp") and the form name ("local-form") to form the URL. When creating the friendly URL, Transaction Manager will convert the client code and form name to lower case and replace spaces with dashes ("-"), and then remove duplicate dashes. In addition, ampersands ("&") are converted to "and", and all other non-alphanumeric characters are stripped out.

To test a form via the friendly URL, edit the form, ensure you are on the "Dashboard" tab and click the "Friendly URL" link ( ) next to the space you want to test on.

Testing Form Receipting

Receipts for a form often need to be tested separately, especially if a separate template file is used to render receipts. The PDF receipt test harness can be used to test receipting.
To test form receipts using the prefill test harness, edit the form, ensure you are on the "Dashboard" tab and click the "PDF Receipt Test" link in the "Spaces" section. This will direct you to the "PDF Receipt Test Harness" page with the form name and an example seed file already loaded.

Testing form receipting
You can enter additional data into the seed file shown in the "XML Receipt Data" field.
You can also enter a receipt number into the "Receipt Number" field, which will also be populated into the receipt.
Once you have entered all the data, click "Test Receipt". The receipt will be rendered in a new window. No record of the receipt requests made via the PDF receipt test harness will be kept in the request log.

Exporting a Form

To export a form and all its versions, go to "Forms > Forms" and click the "Export Archive" link ( ) next to the form. For more information on exporting and importing forms, please see Multi-Environment Support (Manager v5.0)#_sec_migratingform.

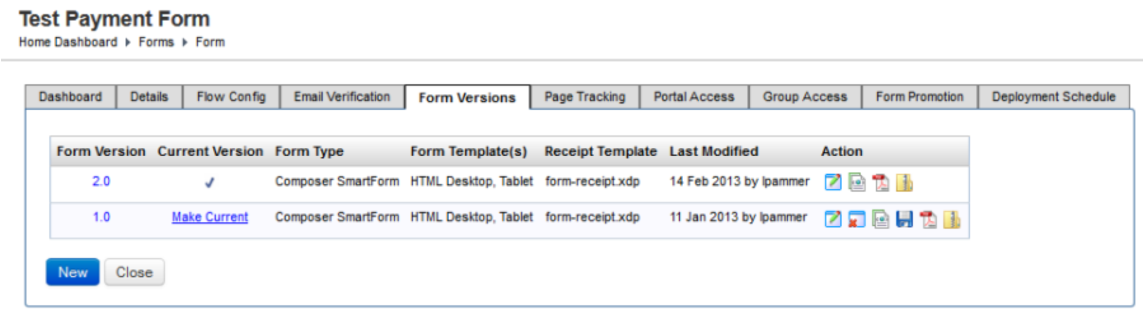
Form Version Configuration (Manager v5.0)

Each form can have multiple versions. A form version contains the set of versioned information for a form. To allow flexible switching between versions, almost all data related to a form is versioned. This allows controlled releases of new versions at any time you choose as well as the ability to roll back a release if needed.
Form versions may contain the actual form template and receipt files, the form data configuration, property and metadata values, attachments and other information.

Search

Viewing the List of Form Versions

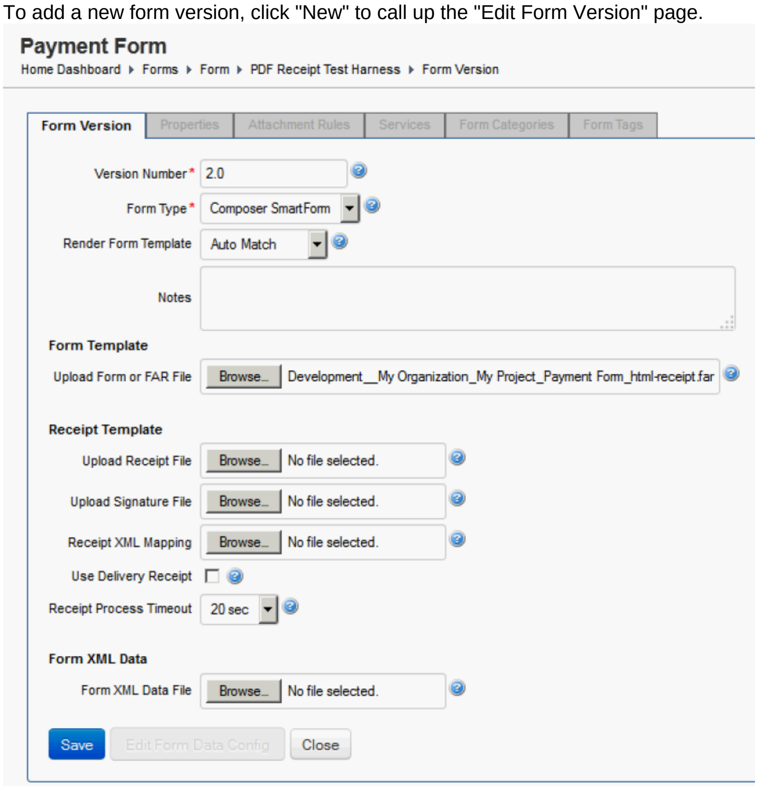
To view the list of all form versions that have been set up for a particular form, go to "Forms > Forms", edit the form and switch to the "Form Versions" tab.



The list of form versions
You will see a list of all form versions along with the information when the version was last modified, and by which user. You can create a new version, edit ( ) an existing version, delete ( ) versions that are not currently in use, edit the form data configuration ( ), see _sec_formdataconfigsetup for more on form data configurations), download the form template file ( ), download the receipt template file if configured ( ) or export the form version ( ).
Please note that the current form version is distinguished in the "Current Version" column and cannot be deleted. You can immediately promote another form version to be the current one by clicking the "Make Current" link in the "Current Version" column. In addition, _sec_changecurrentformversion describes how to schedule version changes for a future date and time.

Creating a Form Version

To add a new form version, click "New" to call up the "Edit Form Version" page.

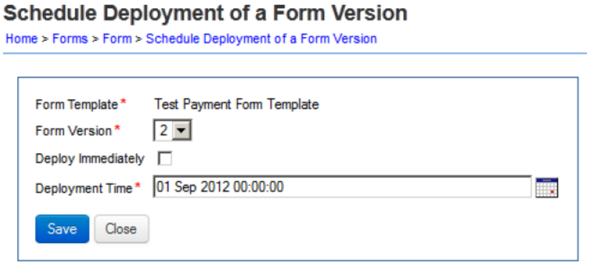


Creating a form version
The "Version Number" field should be set to a unique version number that is only used internally. You would usually start with version number "1" and then increase it sequentially as new versions are uploaded.

Enter notes about the version into the "Notes" field (optional). These notes are not exposed to space users.
Select the appropriate form type from the "Form Type" dropdown. Refer to Form Configuration (Manager v5.0)#_sec_formtypes for details about the various form types.
Upload the PDF, XDP, FAR or ZIP file using the "Browse" button next to the "Upload Form File" section or specify the URL to the form in the LiveCycle repository in the "Form Template" section.
If you would like Transaction Manager to use a different file when rendering the receipt, upload a file or enter the URL to the LiveCycle repository in the "Receipt Template" section.
If your form is asking the user to sign the receipt, you can specify a file that will be used when generating the receipt for the user to sign. If not specified, the receipt template (if set) or the form template file will be used in the signature step.
The checkbox "Use Delivery Receipt" should be enabled only if the form has been specifically designed to make use of delivery receipts. If ticked, TM will create two different kinds of receipt for each submission instead of only one: One receipt to be presented to the user after submission (which has the node representing "Display Mode" set to "Receipt" as usual) and another receipt that will be stored in the TM database and used for delivery and administrators viewing receipts in TM (with the node representing "ReceiptMode" set to "Delivery"). This can allow your form to present different information in a receipt meant for the end user vs. the receipt that is delivered on to the backend, for example enabling office-use only sections in the latter.
The "Receipt Process Timeout" setting is used to control the maximum time allocated to render a receipt for this form when using PhantomJS. This can be useful for complex forms for which the receipt can take a long time to render.
Finally, you can upload a new form data configuration file using the "Upload Form XML Data File" field. Note: While Transaction Manager supports forms that do not use form data configurations, many features such as prefill, managed attachments and payment will not be available.
Save the form version once you are done.
When a new form version is created, it inherits existing data. If the new version is the only existing version for the current form, the new version inherits all organization property and metadata values. Otherwise, the new version will inherit attachments, properties and metadata values from the existing form version with the highest version number.All inherited values can be adjusted in Transaction Manager.

Changing the Current Form Version

You can change the current form version at any time, whether to roll out a new release or roll back to an older version. The current form version is used whenever a user requests the form.
To change the current form version, go to "Forms > Forms", edit the form and switch to the "Version Deployment Schedule" tab.
You will see the current schedule for the form, showing all deployments that are scheduled for the future. If there are any entries in the schedules, please ensure that your changes will not conflict with the schedule.
To schedule a form version change, click "Schedule Version Deployment". You will see the "Schedule Deployment of a Form Version" page.



Scheduling a form version change
On this page, choose the new form version to be deployed from the "Form Version" dropdown.
If you would like to change to the new version immediately, tick "Deploy Immediately". Otherwise, enter the desired changeover date and time into the "Deployment Time" field.Note: Do not schedule multiple deployments on the same form for the exact same time (down to the minute). This is not supported.
Save your changes.
If you chose to make the change immediately, the form will be configured to use the new form version straight away. No schedule record needs to be created.
If you chose a future deployment date, a schedule record is created. The change will be made by a scheduled job (named "Template Version Deployment"). Please ensure this job is configured correctly and not paused; otherwise scheduled form version changes will not be processed.

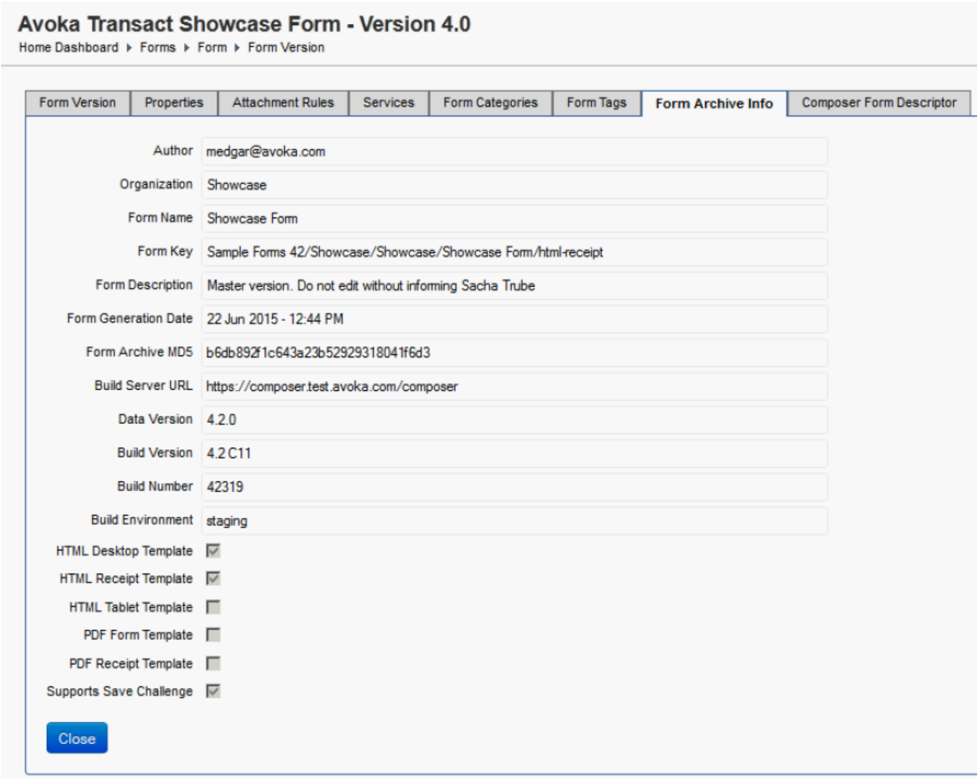
Editing a Form Version

To edit an existing form version, go to "Forms > Forms", edit the form and switch to the "Form Versions" tab. Click the "Edit" icon ( ) next to the version you want to edit.

Note: If you want to edit the current form version, simply go to "Forms > Forms" and click the "Current Form Version" icon ( ) next to the form.
The basic form version settings are described in _sec_createformversion. All the other tabs are described in the sections below.

Composer Form Information

If the form version is of form type "Composer SmartForm", the "Form Archive Info" tab is shown. It contains information about the Composer environment used to create the form as well as Composer metadata describing the form.

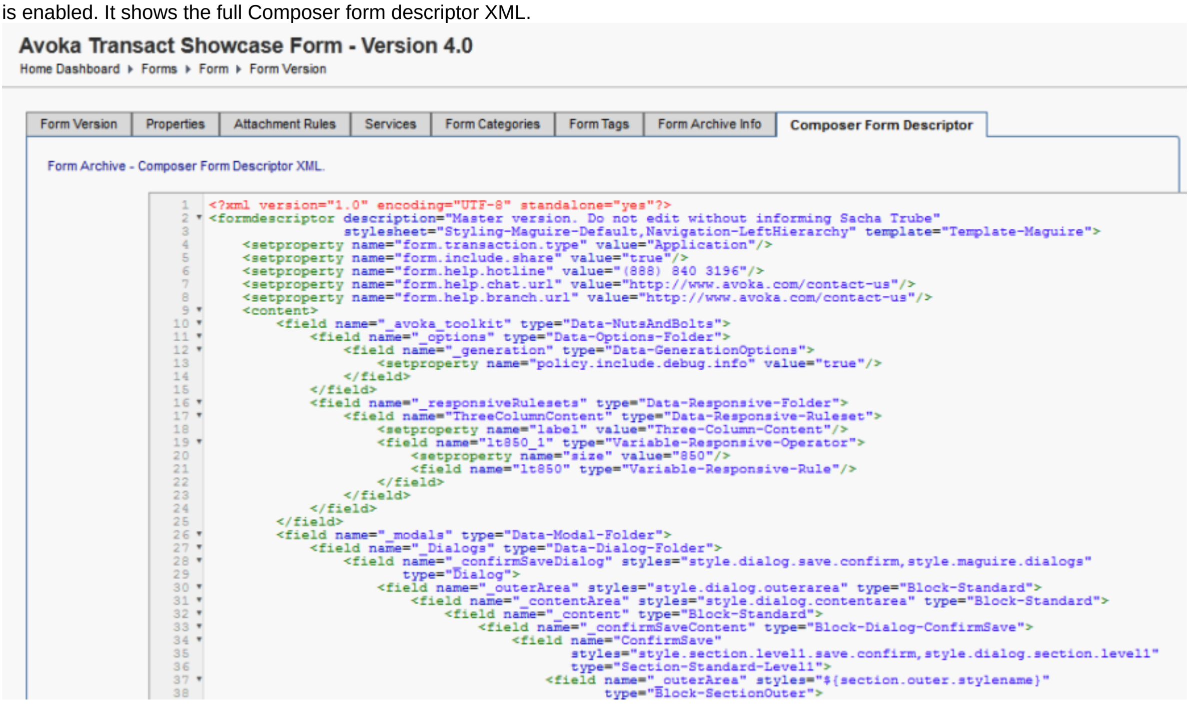


Composer form archive information for a form version

The information shown helps you find the original form in Composer as well as find out when the form was generated and by whom.

Composer Form Descriptor

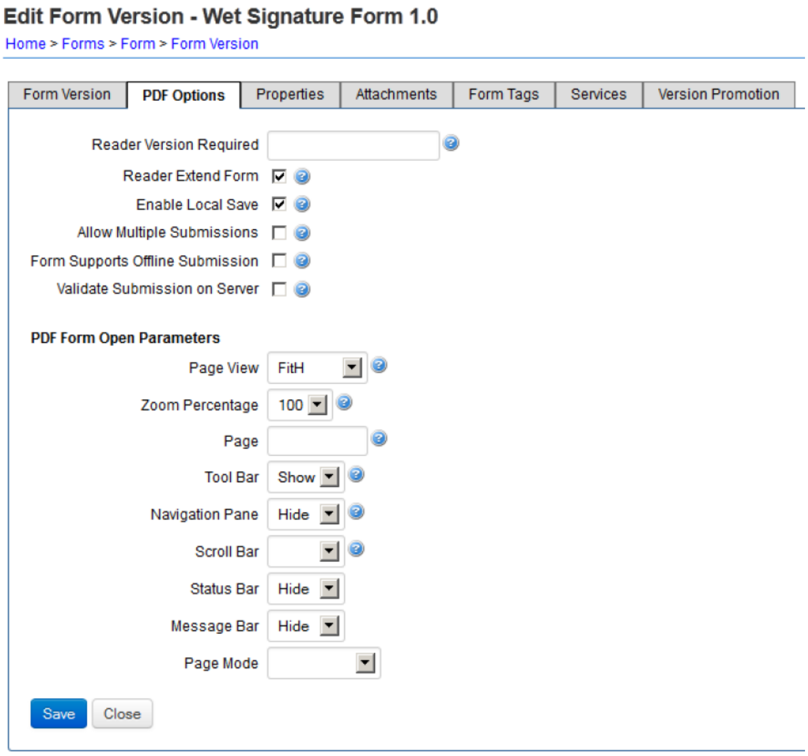
If the form version is of form type "Composer SmartForm" and the FAR file includes the Composer form descriptor, the "Composer Form Descriptor" tab is enabled. It shows the full Composer form descriptor XML.



The Composer form descriptor for a form version

Configuring PDF Options

Transaction Manager allows you to control how your forms will be displayed in Adobe Reader or Acrobat by specifying a set of so-called PDF open parameters (for a list of these parameters, refer to http://www.adobe.com/devnet/acrobat/pdfs/pdf_open_parameters.pdf).
Note that these parameters apply only to forms rendered in Adobe Reader or Acrobat (including the respective browser plugins).
To view or change the PDF options for a form version, edit the form version and switch to the "PDF Options" tab.



PDF options for a form version

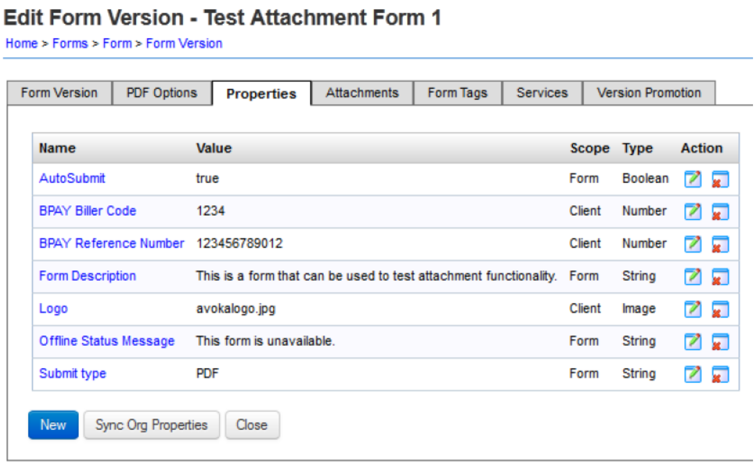
In addition to the PDF open parameters, you can also edit several other settings on this page:
If your form requires a minimum version of Adobe Reader, you can enter this minimum version into the "Reader Version Required" field (e.g. "9.0.0" to specify a minimum version of Adobe Reader 9.0.0. The user space can then attempt to detect the user's Reader version and ask users to upgrade if needed (this is the behavior of the Example Self Service Space).
If you are uploading a non-reader-extended form file that will have to be reader-extended whenever the form is rendered, tick the "Reader Extend Form" checkbox. You will need to reader extend forms if they submit as PDF (as opposed to XML or XDP) or if the user should be able to save them to their own hard disk and later continue to fill in and submit the form.
The "Enable Local Save" checkbox can be ticked to allow the user to save the form offline (i.e. to their own computer) for later completion. Note that Transaction Manager cannot control whether or not the user saves the form offline (saving offline is done in Adobe Reader); instead, the flag is passed into the form, which can then use it in its internal logic. In addition, if a form is submitted after having been saved offline and the "Enable Local Save" flag is not ticked, Transaction Manager will reject the submission.
The "Allow Multiple Submissions" checkbox can be ticked to allow the user to submit a form multiple times, typically after saving a form offline for reuse. If ticked, Transaction Manager will accept submissions, even if the same requested form was already submitted previously. Otherwise, Transaction Manager will reject duplicate submissions.
If you would like the form data to be validated by LiveCycle when the user submits the form, tick the "Validate Submission on Server" checkbox. This validation is done in addition to the validation rules applied by your form. Complex forms may cause validation errors on the server even though the submitted data is valid from the form's perspective; if this is the case, try unticking this checkbox.
If your form has been designed to render the offline submission response, tick the "Form Supports Offline Submission". Refer to Offline Submission (Manager v5.0) for more information on offline submission.
Change the settings as desired and save.

Setting Up Form Version Attachments

Attachments can be set up to instruct Transaction Manager to ask the user to attach supporting documentation after submitting the form. To view or modify attachments for a form version, edit the form version and switch to the "Attachments" tab.
For detailed information on how to set up attachments, see Managing Form Attachments (Manager v5.0)#_sec_addattachmenttoversion.

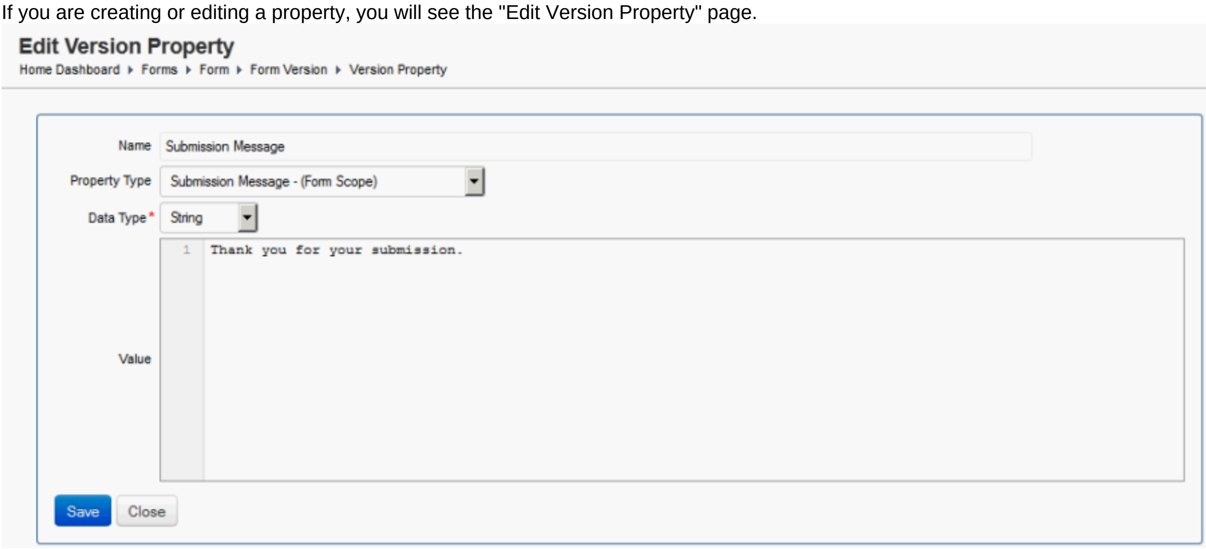
Adding Form Version Properties

Properties can be used to pre-populate your forms or modify their behavior (see Organization Configuration (Manager v5.0)#_sec_propertytypes for more on properties and property types). Each form version maintains its own set of organization and form properties independently.
If you wish to view or modify property values for a form version, edit the form version and switch to the "Properties" tab.



The list of properties for a form version

You can either create a new property value by clicking "New" or change a property by clicking the "Edit" icon ( ) next to it. You can also synchronize organization level properties with the properties currently set up for the organization by clicking "Sync Org Properties".
If you are creating or editing a property, you will see the "Edit Version Property" page.



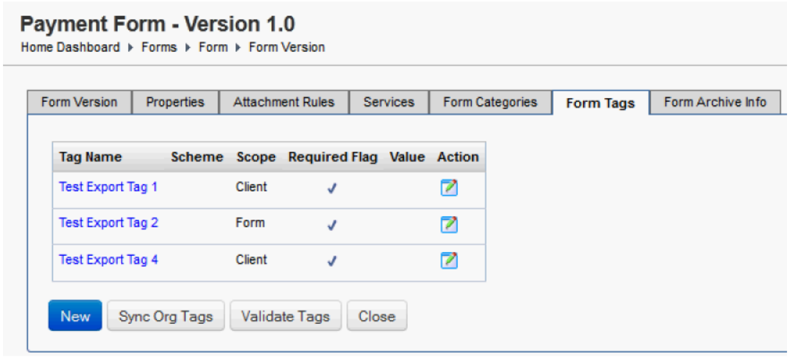
Creating / Editing a form version property
If you are creating a new property, you can choose the appropriate property type from the "Property Type" dropdown. If no suitable property type exists in the system, you can either create one first, or leave the "Property Type" dropdown blank and fill in appropriate values for the "Name", "Data Type" and "Scope" fields.
Enter the new value for the property (depending on the property type, values may have to follow a specific format; see Organization Configuration (Manager v5.0)#_sec_propertytypes), using the "Browse" button to upload a file for properties of type "Image" or "Long Text".
If the property type is "List", you can enter a label for the form property.
To sort list properties, you can also assign them a sequence number.
Save the property and verify that it now appears in the list of version properties on the "Properties" tab.

Adding a Form Description

Form descriptions can be used to present your users with additional information about a particular form to help them determine whether the form meets their needs.
To add a description to a form version, edit the form version and add a property of type "Form Description".
In the Example Self Service Space included in Transaction Manager, this description will be displayed on the form's landing page.

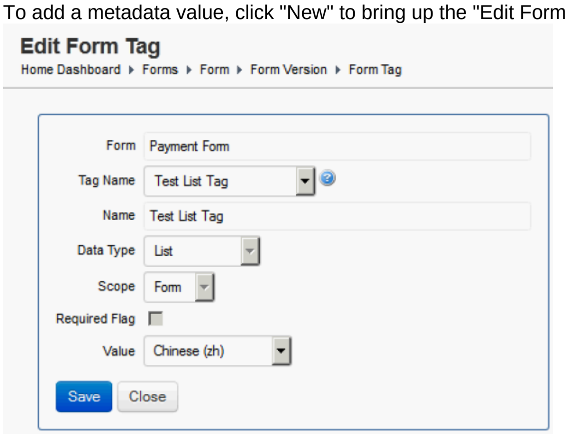
Setting Up Form Version Metadata

Metadata can be used to run queries on the set of forms defined in Transaction Manager using a Web Service API (see Organization Configuration (Manager v5.0)#_sec_metadataconfiguration on more details about metadata, including how to create metadata tags). Each form version maintains its own set of organization and form metadata independently.
If you wish to view or modify metadata values for a form version, edit the form version and switch to the "Form Tags" tab.



The list of metadata values for a form version

You can create or edit ( ) a metadata value, delete existing values ( ) and validate the set of metadata values. You can also synchronize organization metadata values with the current values stored for the organization by clicking "Sync Org Tags".
To add a metadata value, click "New" to bring up the "Edit Form Tag" page (the screenshot below shows the page with all values filled in).

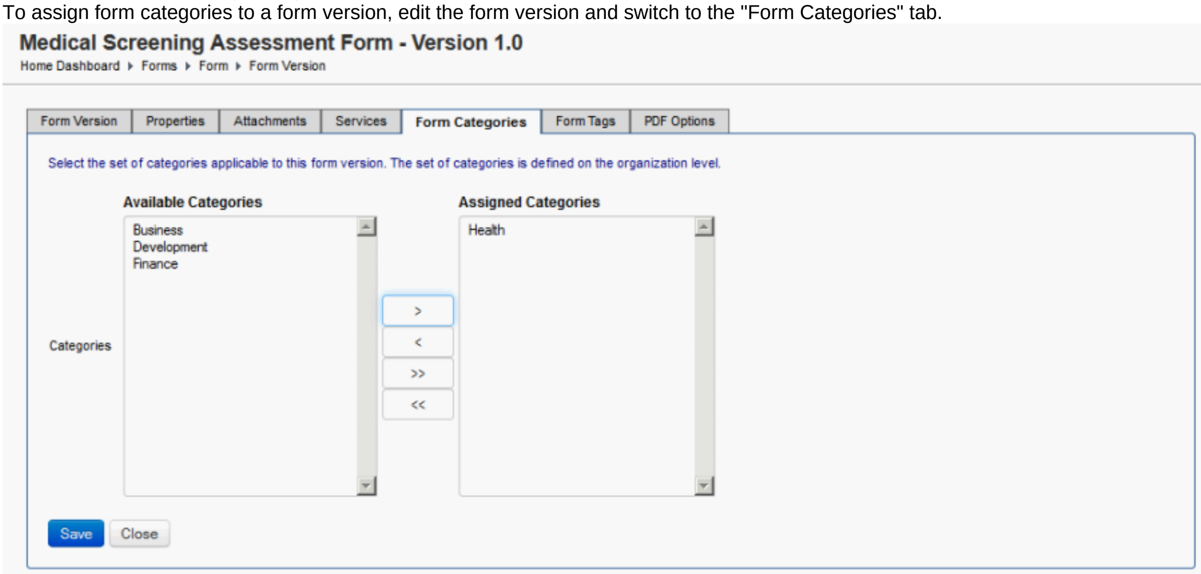


Adding a metadata value to a form version

Choose a metadata tag to assign a value to from the "Tag Name" dropdown. If no suitable metadata tag exists in the system, you can either create one first, or leave the "Metadata Tag" dropdown blank and fill in appropriate values for the "Name", "Data Type" and "Scope" fields.
Set "Value" to the metadata value for this particular form version and save. You will see that the metadata value has been added to the "Form Tags" tab.
You can also click "Validate Tags" to validate the metadata values for the form version. Any errors (such as missing values for required metadata tags) will be reported so you can correct them.

Configuring Form Categories

Forms can be assigned one or more form categories (e.g. "Business") to enable space users to search forms by category. This can be helpful especially if a large set of forms is being hosted.
Form categories are defined on the organization level (see Organization Configuration (Manager v5.0)#_sec_setupformcategories). Once you have created a set of form categories, you can assign them to form versions.
To assign form categories to a form version, edit the form version and switch to the "Form Categories" tab.



Assigning form categories to a form version

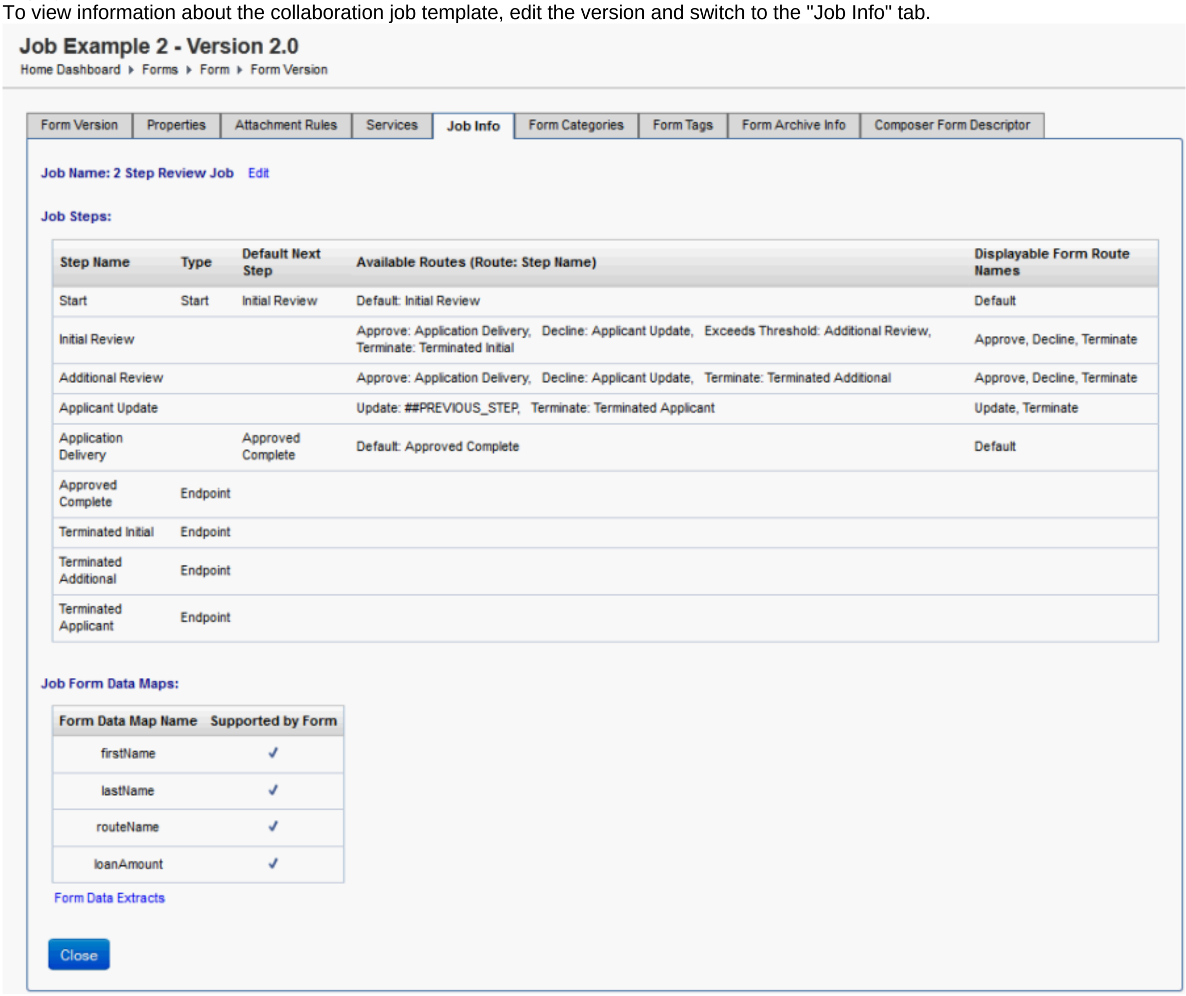
All available form categories set up on the organization level will be shown. Use the ">" and "<" buttons to assign and unassign individual form categories.
To assign or remove all form categories, use the ">>" and "<<" buttons.
Save your changes. If your space supports form categories and you have defined a set of categories for forms hosted on the space, a category selector will automatically be included on the form search page.

Configuring Custom Services

Transaction Manager uses configurable services to perform key processes such as form rendering (see Services (Manager v5.0) for more on services). Some services relevant to form processing can be configured on a form version level. This allows you to override some behavior for individual form versions, while other form versions simply use the default services configured in Transaction Manager.
To view or edit a form version's service configuration, edit it and switch to the "Services" tab.
Please note that selecting the blank entry from one of the dropdowns means that Transaction Manager will use the default service for this form version. It is generally preferable to leave the dropdowns blank where possible, and customize only those services for which a different behavior is desired.
By default, all forms will be rendered using the default form rendering service (in the standard Transaction Manager build, this uses Adobe LiveCycle Output on the Adobe LiveCycle server that does the rendering). If you want to use a different form rendering service for a form version, choose the service from the "Form Render Service" (rendering forms for fill-in) or "Receipt Render Service" dropdown.
Similarly, if you would like to use a specific service to retrieve the XML prefill data, choose a service from the "Form Prefill Service" dropdown.
If you would like to generate receipts electronically signed by the organization for this form, you can either rely on the default service set up in Transaction Manager (if any) or specify a custom service in the "eSignature Render Service" dropdown.
If you would like to use a specific service to monitor, modify and possibly even reject certain submissions for this form version, select a service from the "Submission Preprocessor Service" dropdown.
If you would like to use a service to validate submission XML data at submission time (not when the user saves or abandons), select a service from the "Submission Data Validator" dropdown. Submission data validator services can be developed to detect issues in the submitted data, such as users attempting to manipulate the XML data. If errors are detected, the submission is stored but not treated as a successful submission (it will be delivered using a specialized delivery channel if configured for the form, see Form Configuration (Manager v5.0)#_sec_formdetails).
If you would like to use a service that can execute custom logic when the submission is completed by the user, select a service from the "Submission Completed Processor" dropdown. Transaction Manager provides a default service out of the box, which will execute a customizable Groovy script. For example, it could be used to assign new tasks to the user or forward the submission on for review.

Collaboration Job Information

If the form version is configured to use a service of type "Job Controller", an additional tab showing information about the collaboration job will be displayed.
To view information about the collaboration job template, edit the version and switch to the "Job Info" tab.



Step Name	Type	Default Next Step	Available Routes (Route: Step Name)	Displayable Form Route Names
Start	Start	Initial Review	Default: Initial Review	Default
Initial Review			Approve: Application Delivery, Decline: Applicant Update, Exceeds Threshold: Additional Review, Terminate: Terminated Initial	Approve, Decline, Terminate
Additional Review			Approve: Application Delivery, Decline: Applicant Update, Terminate: Terminated Additional	Approve, Decline, Terminate
Applicant Update			Update: ##PREVIOUS_STEP, Terminate: Terminated Applicant	Update, Terminate
Application Delivery		Approved Complete	Default: Approved Complete	Default
Approved Complete	Endpoint			
Terminated Initial	Endpoint			
Terminated Additional	Endpoint			
Terminated Applicant	Endpoint			

Form Data Map Name	Supported by Form
firstName	✓
lastName	✓
routeName	✓
loanAmount	✓

Job information for a form version
The panel displays the name of the job controller service, the set of job steps it defines and the set of job form data mappings.
This information can also be viewed when editing a service definition of type "Job Controller".

Deleting a Form Version

Deleting a form version is a dangerous operation as it affects many other related entities, including submissions for the form version.
If you wish to keep submission data, you can simply promote a different form version or, if the form is to be retired, deactivate the form as described in Form Configuration (Manager v5.0)#_sec_removeform.
To delete a form version, go to "Forms > Forms", edit the form and switch to the "Form Versions" tab. Find the form version you want to remove and click the "Delete" icon ( ) next to it. You will be asked to confirm this action before the form version and all related entities are irrevocably deleted.
Note that you cannot delete the currently deployed form version.

Exporting a Form Version

To export a specific form version, go to "Forms > Forms", edit the form and switch to the "Form Versions" tab. Click the "Export Archive" link ( ) next to the version you want to export. For more information on exporting and importing form versions, please see Multi-Environment Support (Manager v5.0) #_sec_exportsingleformversion.

Form Data Configuration Setup

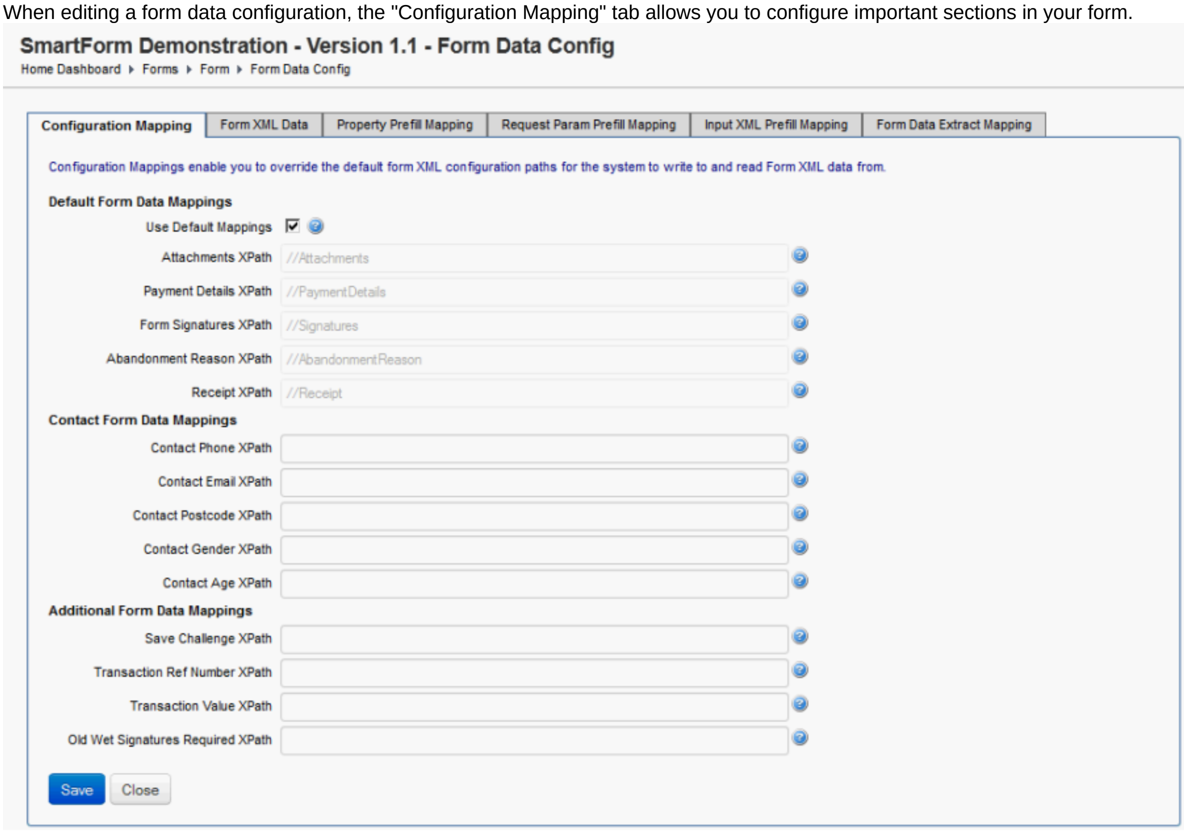
Transaction Manager form data configurations define the XML data structure being passed into and out of a form. If a form data configuration is set up for a form version, a variety of features such as prefill, managed attachments and payments becomes available.
A form data configuration can contain three kinds of mappings from prefill data into this data structure: Property prefill mappings, where the source of prefill data are the form version properties set up in Transaction Manager; XML prefill mappings, where an external XML file containing prefill data is obtained at render time; and parameter prefill mappings, where prefill data is passed through as HTTP request parameters, session attributes or cookies. For improved submission handling, form data configurations can include submission data extract mappings, which specify values that should be parsed out of the submitted XML and stored separately.
To use some features built into Transaction Manager such as attachments and receipting you also need to set up configuration mappings for your form data configuration (see _sec_setuppropertyprefillmappings).

Viewing the Form Data Configuration for a Form Version

Each form version is set up with its own form data configuration (or possibly none at all for simple forms). To edit a form data configuration, go to "Forms > Forms", edit the form and switch to the "Form Versions" tab. Click on the "Form Data Configuration" icon ( ) next to the form version.
You can also access the form data configuration of the current form version by going to "Forms > Forms" and clicking the "Current Form Data Config" icon ( ) next to the form.
If a form version does not have a form data configuration set up, these icons will not be present. You can upload the form data configuration file by editing the form version and uploading the XML file using the "Upload Form XML Data File" field (see _sec_createformversion).

Configuration Mappings

When editing a form data configuration, the "Configuration Mapping" tab allows you to configure important sections in your form.



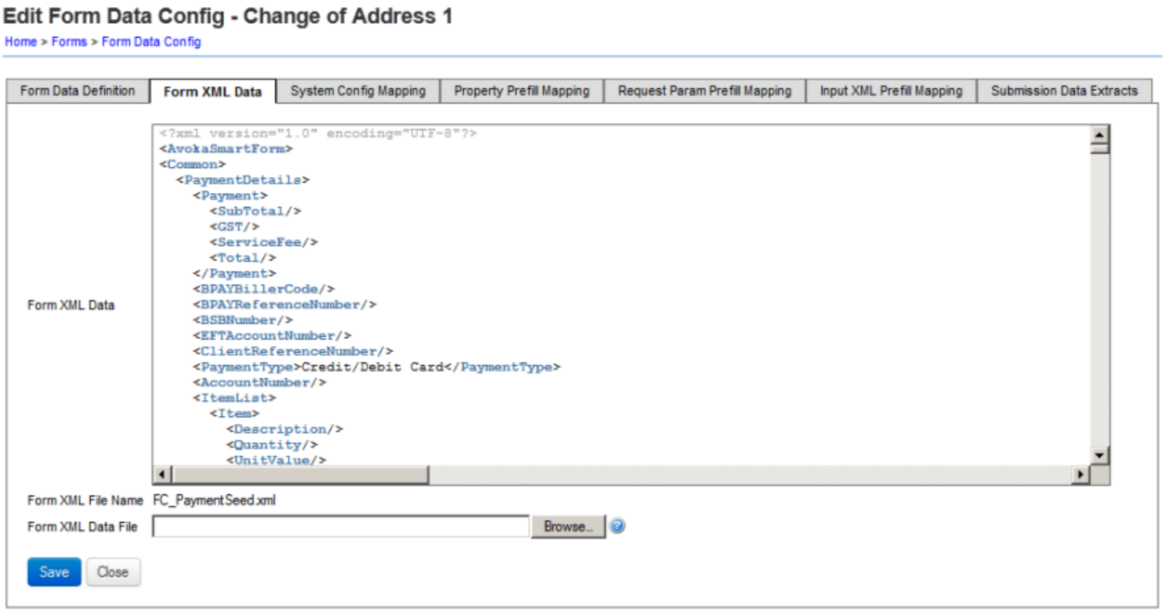
Configuration mappings for a form data configuration

The sections in your form you can identify on this page are needed to support certain features TM provides. For example, if you want TM to request payment for your form, your form must contain a section of a particular structure (normally named PaymentDetails).
By default, the most important sections (related to form attachments, payments and signatures) have pre-defined mappings ("Use Default Mappings" is ticked and the default XPath pointing to the section appear as a placeholder in the respective fields).
You can also define additional mappings to sections for which no default mapping exists:
"Contact Phone XPath" should contain the XPath to an element containing the user's contact phone number.
"Contact Email XPath" should contain the XPath to an element containing the user's contact email address. This value is necessary for post-submission email verification and anonymous online save.
"Contact Postcode XPath" should contain the XPath to an element containing the user's (or the designated contact's) postcode.
"Contact Gender XPath" should contain the XPath to an element containing the user's (or the designated contact's) gender.
"Contact Age XPath" should contain the XPath to an element containing the user's (or the designated contact's) age.
"Save Challenge XPath" should contain the XPath to an element containing the answer to the challenge question users will need to answer to return to an anonymously saved form. The element in question must not be blank when the form is saved; please refer to the Transact Integration Guide, chapter "Form Saving and Retrieval" for detailed information on anonymous online save.
"Transaction Ref Number XPath" should contain the XPath to an element containing a transaction reference number that will be stored against the

submission. The transaction reference number can be used to identify multiple TM submissions that are part of a larger transaction (while the TM receipt number always identifies an individual submission).
"Transaction Value XPath" should contain the the XPath to an element containing a number describing the value of this transaction as defined by the form developer. If set, this value is stored against the submission and can be used for reporting or in Groovy scripts.
"Signatures Required XPath" is a setting related to signatures that is being maintained for compatibility reasons. For new forms that require signatures, please use the mapping defined in "Signature XPath" instead.

Modifying the XML Seed File

To upload a new XML seed file or edit the seed file contents, edit the form data configuration you want to modify and switch to the "Form XML Data" tab.

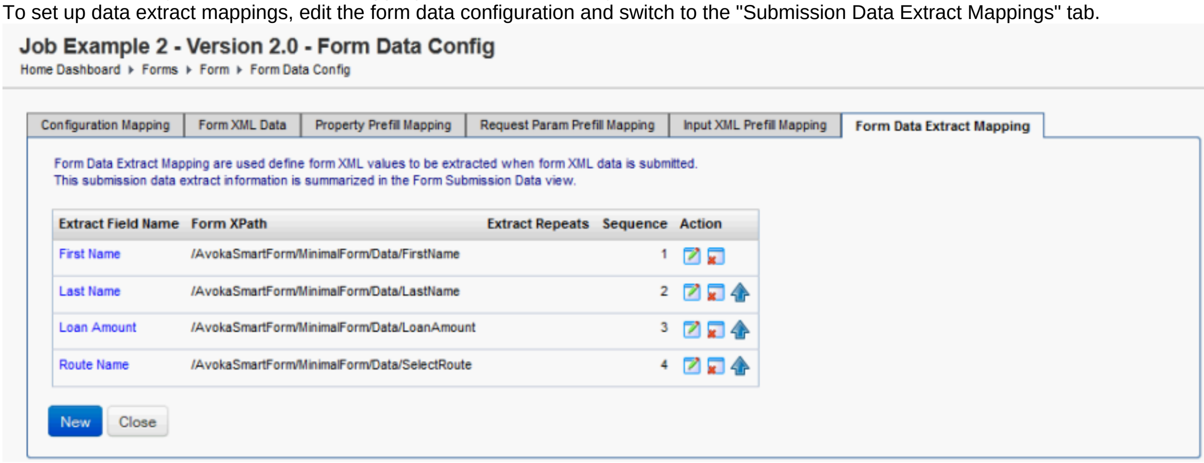


The seed XML for a form data configuration

You can upload a new seed file by clicking the "Browse" button and selecting the new file (which must be an XML file) and saving. The XML file will be loaded, and its contents will be displayed in the "Form XML Data" field.
To change the seed XML, you can either upload another file or edit the "Form XML Data" field directly. To avoid syntax errors, the XML will be validated when you save.

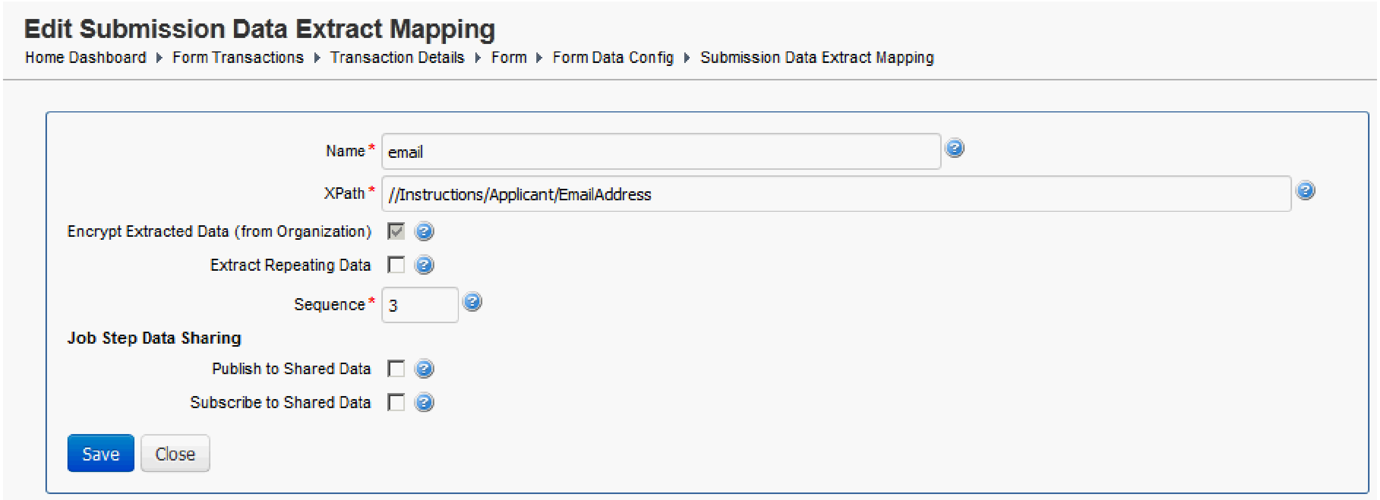
Setting Up Data Extract Mappings

Data extract mappings can be used to read data items from arbitrary locations in the submitted XML, which are then stored separately in the Transaction Manager database. Each form data configuration can have its own set of data extract mappings.
Please note: For Composer SmartForms, data extract mappings can be easily defined in Transact Composer. When these forms are published to Transaction Manager, the data extract mappings will be loaded along with the form.
To set up data extract mappings, edit the form data configuration and switch to the "Submission Data Extract Mappings" tab.



Viewing the list of data extract mappings

You can use this page to set up new data extract mappings or edit ( ) and delete ( ) existing mappings. You can reorder mappings by using the "Move Up" ( ) action.
To create a new mapping, click "New". You will see the "Edit Submission Data Extract Mapping" page as shown below.



Creating a data extract mapping

Enter the name you want to give to this data item into the "Name" field. This is the name under which the extracted data will be stored.

Enter the XPath to the location in the submitted XML into the "XPath" field.
If the XPath points to a repeating element and you want a data extract value to be created for each instance, tick the "Extract Repeating Data" checkbox.

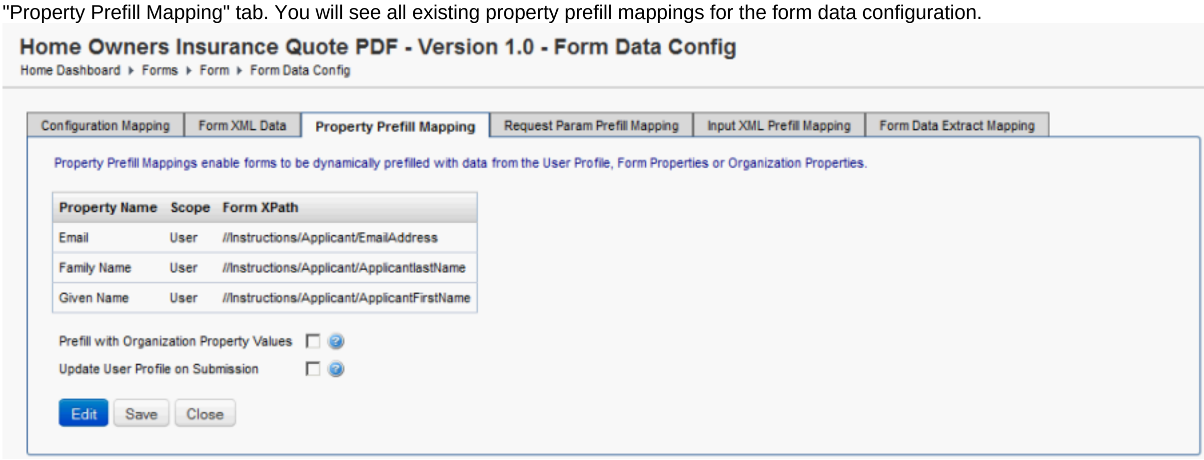
Values extracted from the submission can be encrypted for additional security. To encrypt values for this data extract mapping only, tick the "Extract Encrypted Data" checkbox. Note that encryption can also be enabled on the organization level, in which case it cannot be turned off for an individual data extracy mapping (see the screenshot above for an example). You can change this setting by editing the organization (see chapter "Operations", section "Data Retention and Storage Configuration (per Organization)").

The "Sequence" field is used for ordering the extract values. You can adjust the values as needed. Note that the sequence number must be unique within the form data configuration.
The "Job Step Data Sharing" section allows you to specify whether values for this data extract mapping will be shared between submissions associated with the same job. If "Publish to Shared Data" is ticked, the value will be shared with all submissions who themselves have a data extract mapping with the same name and "Subscribe to Shared Data" ticked.

Save your changes to create the mapping. Whenever a form using this form data configuration is submitted, Transaction Manager looks up the value at the specified XPath and writes it to a data extract element with the specified name (see Operations (Manager v5.0)#_sec_submissiondataextractvalues on how to view extracted data for a submission).

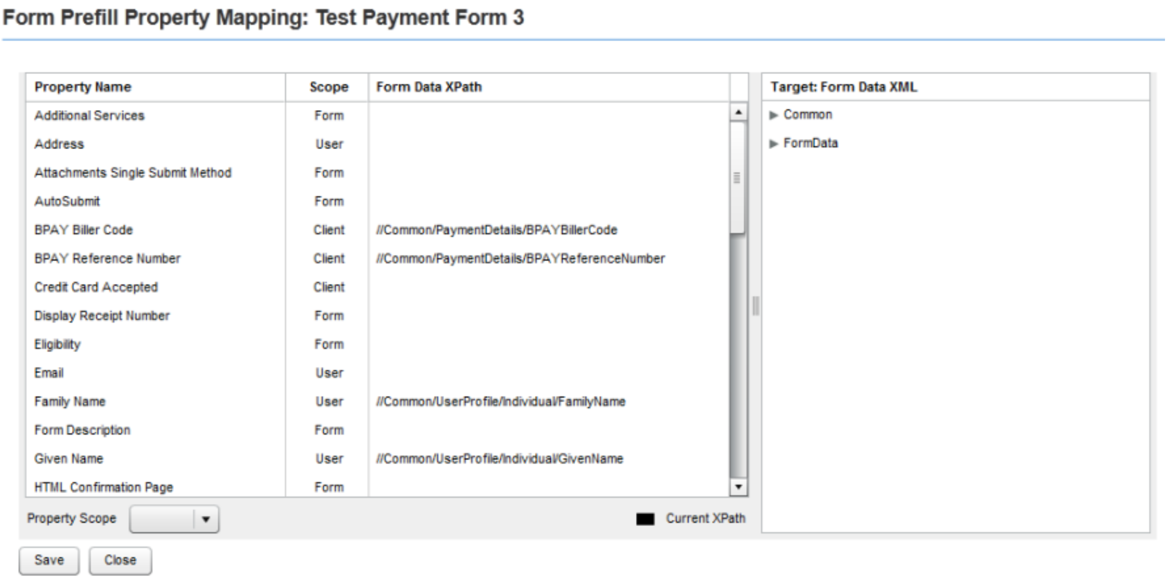
Setting Up Property Prefill Mappings

One of the benefits of setting up a form data configuration in Transaction Manager is the ability to map properties into arbitrary form XML data elements, causing Transaction Manager to populate the corresponding property values when a form is rendered. Using properties and property prefill mappings, you can customize and adapt your form without having to upload a new version of the form file.
To set up property prefill mappings for an existing form data configuration, edit the form data configuration you want to modify and switch to the "Property Prefill Mapping" tab. You will see all existing property prefill mappings for the form data configuration.



Viewing the list of property prefill mappings

In addition to using properties maintained in the form version (scopes "Form" and "Client"), you can additionally also instruct TM to fall back to properties defined for the organization by ticking the "Prefill with Organization Property Values" checkbox. This is not the default behavior in order to preserve data encapsulation in form versions.
If you are planning on mapping user profile properties into the form, you can also choose to populate data entered by the user back into the user profile at submission time by checking the "Update User Profile on Submission" checkbox and saving. If you do this, all property prefill mappings of scope "User" will be applied in reverse at submission time, i.e. data from the form will be written to the user profile.
To modify property prefill mappings, click "Edit". You will see the "Form Prefill Property Mapping" page.

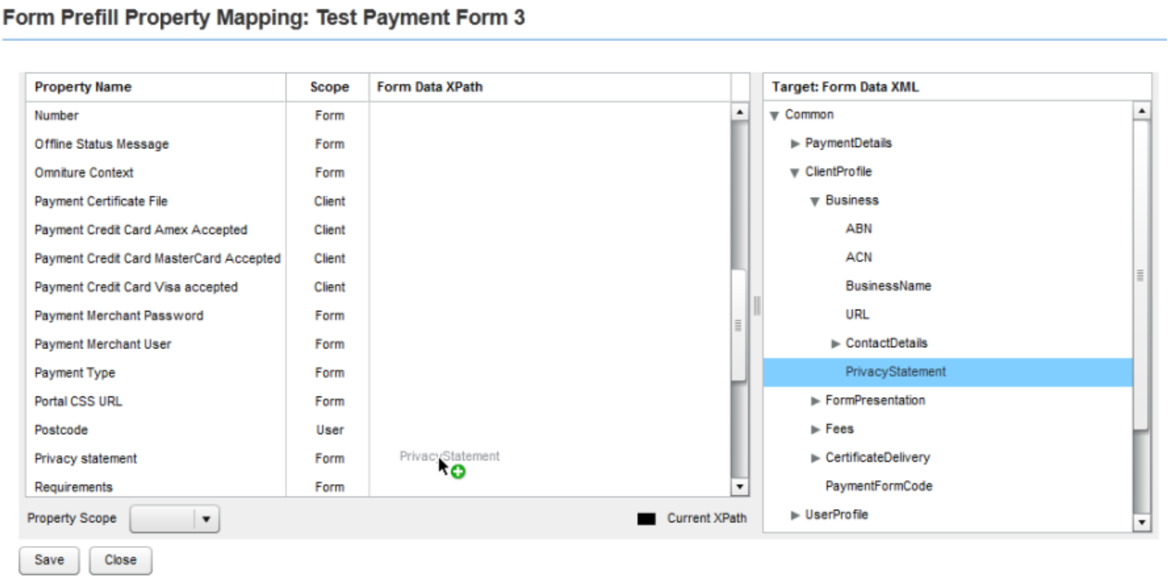


Editing property prefill mappings for a form data configuration

The page is divided into two areas: The table on the left shows the set of available properties, which comprises all available property types as well as all version property values set up for the form version that the form data configuration belongs to. For each entry, the XPath it is currently mapped to is shown next to it. You can filter the properties by their scope ("Client", "Form" or "User") using the "Scope" dropdown.

The tree in the "Target: Form Data XML" section represents the form data XML document and contains all its elements. Nodes with children can be expanded or collapsed.

To define a property prefill mapping, find the form data element you want to map a property to, then simply drag and drop the element into the "Form Data XPath" column next to the property. The full XPath will be populated into the cell.



Adding a new property prefill mapping

When you are done assigning property prefill mappings, save your changes.

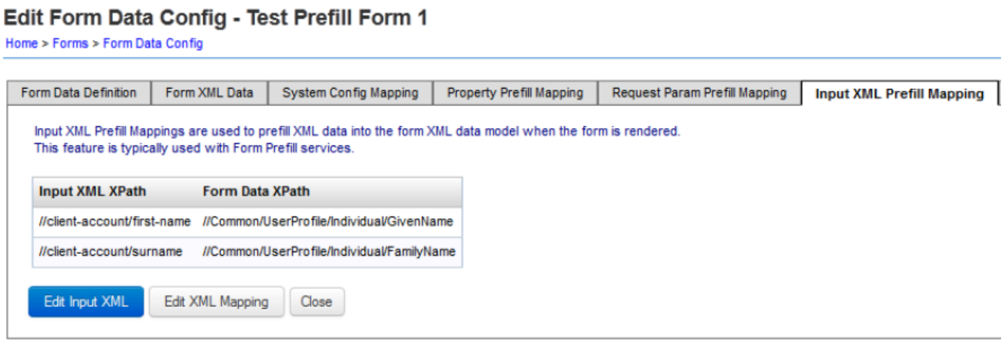
Setting up XML Prefill Mappings

When forms are rendered, they are pre-populated with various data: Prefill data based on properties is mapped into the form data document as discussed previously. In addition, Transaction Manager allows you to pass data to the form at render time using an XML prefill document. The data elements are copied to the form's seed XML data using previously defined mappings (similar to property prefill mappings).

This method of pre-population allows the organization to maintain the data used for pre-population outside of Transaction Manager. It can replace or complement property prefill.

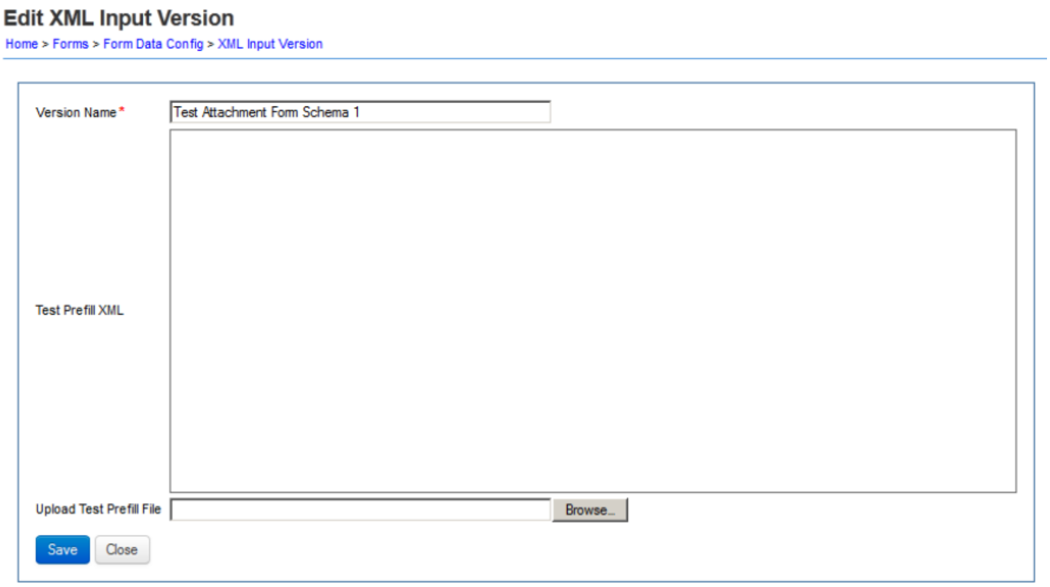
Note: For security reasons, XML prefill is disabled by default. It can be enabled by setting the space property "Allow XML Request Param Form Prefill" to "true" (see Spaces (Manager v5.0)#_configurespaceproperties for more on space properties).

Each form data configuration can have a set of XML prefill mappings defined. To view a list of all XML prefill mappings set up for a form data configuration, edit it and switch to the "Input XML Prefill Mappings" tab.



Viewing the list of XML prefill mappings for a form data configuration

If XML prefill has not yet been set up for the form data configuration, click "New Input XML" to upload an XML file showing the structure of the prefill data (this is generally a blank example of the prefill data). The screenshot below shows the page that allows you to upload and later edit the prefill XML structure.



Uploading a sample XML prefill file

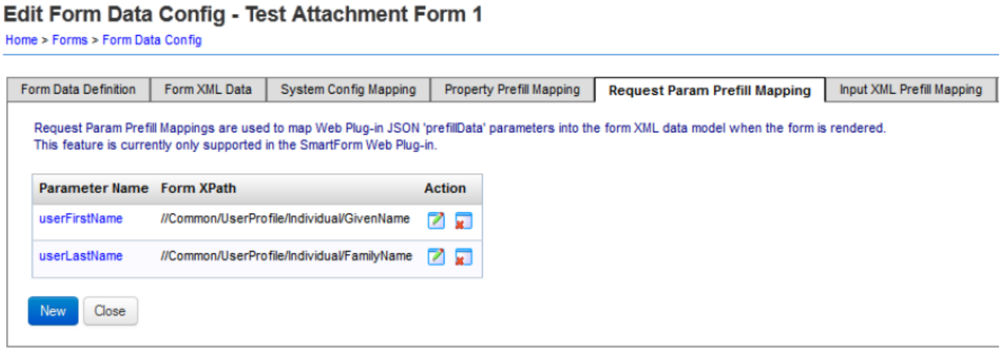
Simply use the "Upload Test Prefill File" button to upload the sample XML prefill file, and then save.
Once the prefill XML file has been defined, click "Edit XML Mapping" to access the "Form Prefill Data XML Mapping" page, where you can define how the prefill data shall be mapped into the form data configuration file.

The XML prefill mapping tool

The mapping tool consists of three sections: the XML prefill document structure on the left, the form data configuration XML structure on the right and the mappings in the center. To create a new mapping, drag an element from the left into the middle, and drag the schema seed element you want to map it to next to it. Transaction Manager will automatically fill in the full XPath of the elements you drop onto the mapping area.
To delete a mapping, simply delete both its "Prefill Data XPath" and "Form Data XPath" entries.
When you are done, click the "Save" button.
Once XML prefill is set up for a form version, you can use XML prefill when rendering your forms. To enable XML prefill, edit the form and tick the "Require Form Prefill" checkbox, then save. If this checkbox is ticked, Transaction Manager will attempt to acquire XML prefill data at render time, using the "Form Prefill" service for the form version (see Services (Manager v5.0)#_sec_formprefillservice).

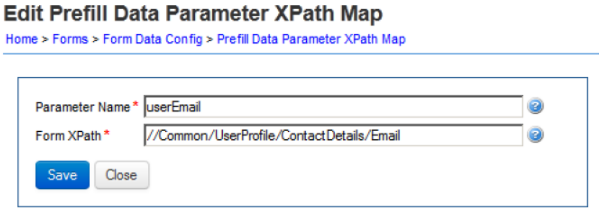
Setting up Request Parameter Prefill

In addition to property and XML prefill, you can also choose to prefill your form with data obtained from the HTTP request when the form is rendered. Prefill data can be passed in via request parameters, session attributes or cookies.
To view and modify the set of request parameter prefill mappings, edit the form data configuration and switch to the "Request Param Prefill Mapping" tab.



Viewing the list of request parameter prefill mappings for a form data configuration

You can edit and delete existing mappings, or create a new mapping by clicking "New" (the screenshot below shows the "Edit Prefill Data Parameter XPath Map" page).



Creating a request parameter prefill mapping

Enter the request parameter name into the "Parameter Name" field. This name is used at render time to look up matching request parameters, session attributes and cookie values.
Enter the XPath in the form data configuration into the "Form XPath" field.
When a form is rendered in the Web Plugin module, the HTTP and JSON request parameters, HTTP session attributes and cookies will be searched for entries with matching names, and their values will be mapped into the form data configuration according to the mappings.

Importing Forms from Transact Composer (Manager v5.0)

Transaction Manager is designed to host forms created in Avoka Transact Composer. In addition, you can easily publish forms from Composer to Transaction Manager. This section describes how to publish forms created in Composer.

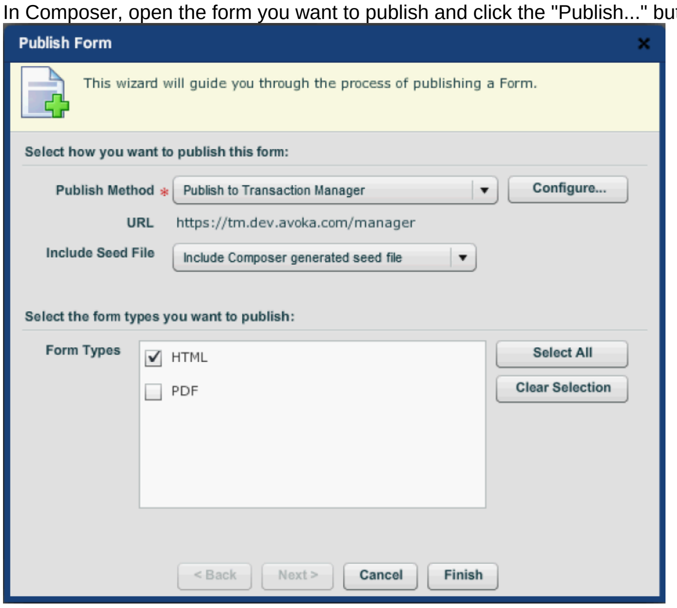
Search

Concepts

Transact Composer is a web application that allows for rapid design of sophisticated HTML and PDF SmartForms, and is part of Avoka Transact. Composer can publish form *packages* to Transaction Manager, either directly or using ZIP file exports.

Publishing a Form Using Composer

Please note: Before you can publish forms to Transaction Manager, your instance of Composer must have the deployment property "External URL" set to the URL of Composer as visible to Transaction Manager. If this is not possible, you will need to publish forms by exporting a ZIP file and importing it into Transaction Manager.
In Composer, open the form you want to publish and click the "Publish..." button. You will see the "Publish Form" dialog.

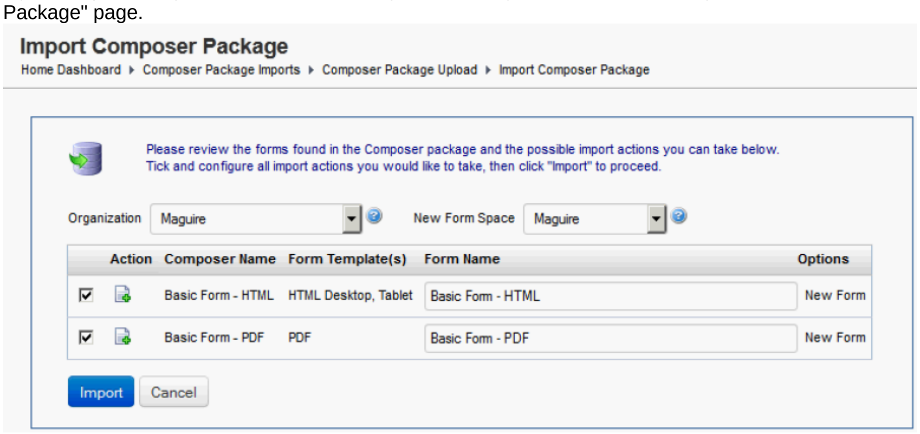


Publishing a form in Transact Composer

You can either publish the form directly to a Transaction Manager instance, or export it as a zip file and import it to Transaction Manager at your convenience.
To modify the URL of the Transaction Manager instance that will be published to, click "Configure..." and edit the entry associated with publishing to Transaction Manager (alternatively, you can add a new entry with a different URL). Adjust the URL and save your changes.
Once you are satisfied with your publishing target, select the form types you want to include and click "Finish".
If you are publishing directly to Transaction Manager, please read on. If you are publishing to a ZIP file, save the file and import it to Transaction Manager by accessing the "Composer Import Summary" page as described in _sec_viewcomposerimportlog.

Importing a Form in Transaction Manager

If you are publishing to a Transaction Manager instance, you will be asked to log in to Transaction Manager and then shown the "Import Composer Package" page.



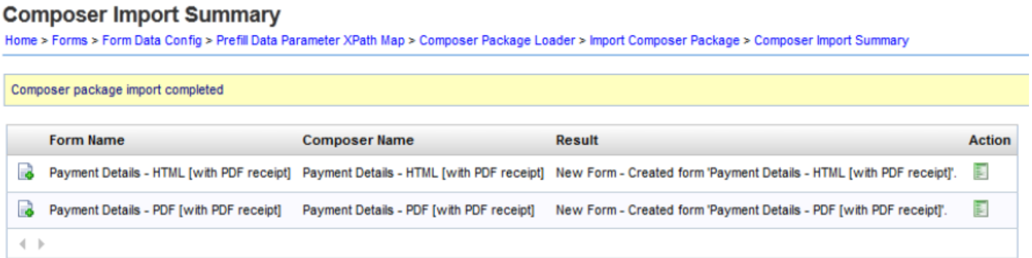
Importing a Composer form package to Transaction Manager

You will be offered to import all forms included in the package published by Composer. Please note that all form types will be listed separately, and a

Transaction Manager form will be created for each.
Make sure to select the Transaction Manager organization you want to import the forms to from the "Organization" dropdown. All forms will be imported to this organization. If no organizations exist on your Transaction Manager instance, you need to create one before you can import Composer forms.
In addition, select a space from the "New Form Space" dropdown. All new forms created during the import will be associated with this space (if necessary, this can be manually adjusted later for individual forms). Note: Only spaces available to the selected organization will be shown. For information on how to view or modify the spaces for an organization, please refer to Organization Configuration (Manager v5.0)#_sec_organizationspaceaccess.
You can choose to import some or all of the forms by ticking the checkboxes next to the entries. For each Composer form in the package, two actions may be available:

- Create a new form
  This action is always available. A new Transaction Manager form with the name specified in the "Form Name" field will be created.
- Update an existing form
  If a Transaction Manager form derived from the same Composer form was found, you can create a new version of the existing form or update the current version (use with caution). This is the default action if a matching form is found.

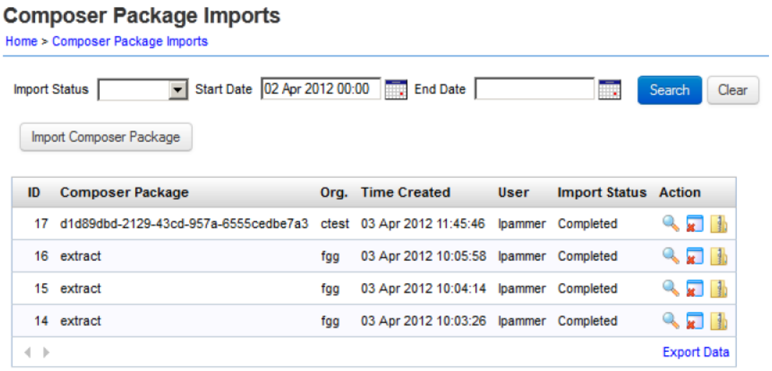
Make your selection and then click "Import". The forms will be imported, and you will be shown a summary of the outcome. Please review the result to check that all forms were imported successfully.



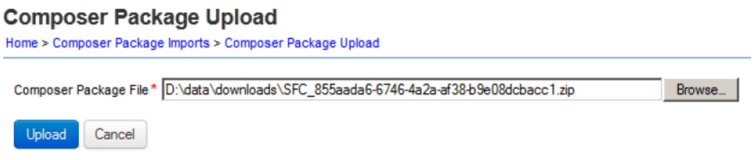
The page summarizing Composer import results

Viewing the Composer Import Log

Every package that is imported from Composer is recorded in Transaction Manager. This allows you to monitor past imports, manually import Composer package ZIP files and complete previously abandoned imports.
To view the Composer import log in Transaction Manager, go to "Forms > Composer Package Imports". You will see the list of previous Composer package imports and their status.



The list of Composer package imports
You can filter the list by organization, import status ("Ready" or "Completed") and import date.
You can view details for an import item ( ), delete import records ( ) and download the ZIP file for an import item ( ).
You can also import ZIP files created by Composer by clicking the "Import Composer Package" button. You will be asked to upload the ZIP file as shown below.



Uploading a Composer package file to Transaction Manager
Follow the instructions in the previous section to import the package.

Delivery Configuration (Manager v5.0)

Once a form is submitted and ready for further processing, the organization needs to be notified so they can take appropriate action. Transaction Manager allows you to set up automatic email, web service or LiveCycle orchestration delivery on a per-form basis, with separate settings for test and production mode. A delivery channel is such a configuration defining the delivery process, and is a reusable entity associated with an organization.

Search

Delivery Types

Each form can be set up with two delivery channels: production and test delivery. The production delivery channel is used when the form is in production mode (i.e. its "Test Mode" flag is turned off), otherwise the test delivery channel is used.
Submission delivery is initiated when the user completes all aspects of the form on Transaction Manager and other tasks (such as attachment virus scanning and receipt generation) have also been completed. For example, this means that forms that have been submitted, but for which required attachments are still outstanding, will not be delivered automatically until the user has completed everything.In addition, submission delivery can occur after a submission has been abandoned, but it contains enough information that the organization may want to follow up with the user.
For more details on how to tell whether a submission is ready for delivery or has already been delivered, see Operations (Manager v5.0) #_tracksubmissions.

Delivery Methods

Transaction Manager currently supports four different delivery methods.

Email Delivery

Email delivery sends out an email to a set of email addresses. The email can contain the submission data as well as a link to the attachments uploaded by the user. Delivery is marked as completed as soon as the email has been sent.

Secure Email Delivery

Secure email delivery is similar to the standard email delivery in that it sends out an email to notify the organization of a submission. However, the email itself does not contain any submission data; instead, it contains a link to a secure page on Transaction Manager which provides the submission data for download.
In addition, delivery completes only after an administrator user has accessed this page and confirmed that the submission has been received.
If delivery has not been acknowledged within a certain time period (by default 24 hours), reminders are sent out to the recipients defined in the delivery channel every day. If the delivery channel's "Email Addresses" field is blank, the reminder is sent to all global and organization administrators in the "Receive Delivery Escalation Alerts" group.
If delivery remains unacknowledged (by default 72 hours after the form was submitted), escalation emails are sent out to all global and matching organization administrators in the "Receive Delivery Escalations" group.

Web Service Delivery

Web service delivery notifies the organization by making a web service call, using a pre-defined API (see the Transaction Manager Web Service API document).
Organizations can choose to either implement these web services themselves or install the Transact Integration Agent (please refer to the TIA documentation for more information).
Delivery is marked as complete once the web service client confirms that it has received the submission.
If delivery has not been completed (by default 72 hours after the form was submitted), escalation emails are sent out to all global and organization administrators in the "Receive Delivery Escalation Alerts" group.

REST Delivery

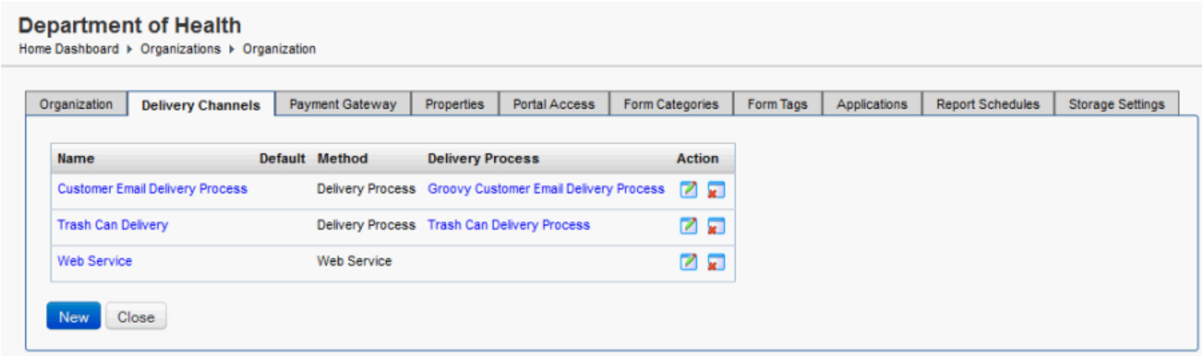
TM provides a REST endpoint that can be used to retrieve submission data. For more information on REST delivery, please see the Groovy services guide included in the TM management console ("System > Transact Services Guide", section "REST Services").

Process Delivery

Process delivery invokes a service definition in TM to perform delivery. TM contains multiple service implementations, including a service that calls an orchestration on an Adobe LiveCycle server and passes in the submission data as well as the attachments, as well a customizable Groovy delivery service. Delivery is marked as completed if the service indicates that delivery was successful.
If delivery has not been completed (by default 72 hours after the form was submitted), escalation emails are sent out to all global and organization administrators in the "Receive Delivery Escalation Alerts" group.
The sample LiveCycle delivery orchestration included in Transaction Manager writes the files out to C:\formcenter\processdelivery\<submitKey>\, where *submitKey* is the submission's submit key.

Browsing Delivery Channels

To browse the list of available delivery channels for an organization, go to "Forms > Organizations", edit the organization and switch to the "Delivery Channels" tab.

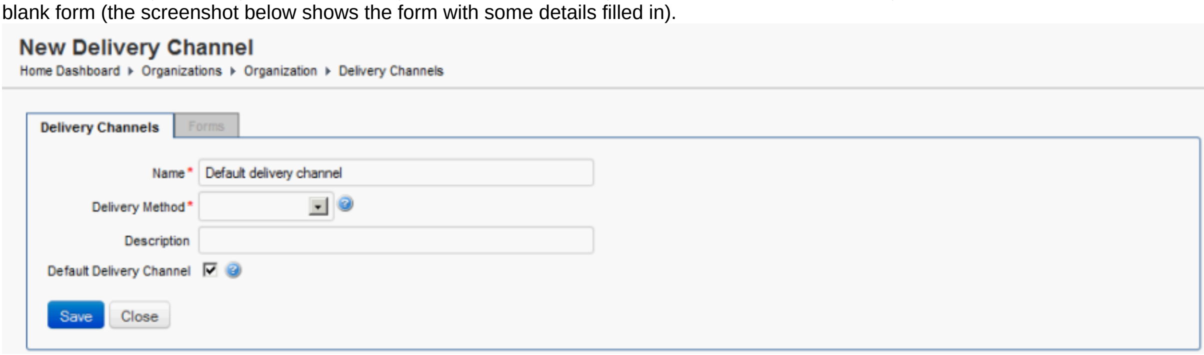


Browsing the list of delivery channels

This page allows you to create, edit ( ) or delete ( ) delivery channels – see the sections below for details.

Creating a Delivery Channel

Delivery channels can be re-used for several forms. This makes it easier to later e.g. change the email addresses in one place without having to update each form individually.
To create delivery details, go to "Forms > Organizations", edit the organization, switch to the "Delivery Channels" tab and click "New". You will see a blank form (the screenshot below shows the form with some details filled in).



Creating a delivery channel

Give the delivery channel a name (unique for all delivery channels available to the chosen organization) and, optionally, a description.
If this delivery channel shall be the default channel for the organization, tick the "Default Delivery Channel" checkbox. Submissions not explicitly associated with a delivery channel will be delivered using the default delivery channel.
Now you need to choose how you want the submissions to be delivered: by email (standard or secure delivery), via a web service, via a REST call or to a LiveCycle orchestration. Choose the appropriate value from the "Delivery Method" dropdown.
For more information on the Transact Integration Agent and web service delivery, see the separately delivered TIA Administration Guide.

Creating an Email Delivery Channel

Choose "Email" from the "Delivery Method" dropdown.

The "Email Addresses" and "Cc Addresses" fields are the email addresses submissions will be delivered to. You can fill in multiple email addresses by separating them with commas.

You can customize the email subject and message (the default values are stored as service parameters of the default "Submission Delivery Controller" service).

Note that you can also reference data extract values stored for the submission in the email subject or message. Data extract values generally represent data entered by the user filling in the form, and can be used in delivery emails to display data about the user and/or their choices.To add a data extract value to the email message or subject, use the Velocity variable "${submissionExtractData.extractname}", replacing "extractname" with the submission data extract name and omitting the quote characters. However, if you are unsure whether a given extract value is present, you should add a conditional around its usage.

For example, the following would safely add a reference to the contact phone number provided by the user (assumed to be stored in a data extract value with the name "UserContactNumber"):

Please refer to the Velocity Template Language documentation for more information on Velocity notation (http://velocity.apache.org/engine/devel/vtl-reference-guide.html) and to Operations (Manager v5.0)#_sec_submissiondataextractvalues for more about submission data extract values.

To control what data you wish to include in the delivery email as attachments, you can use the "Deliver PDF" and "Deliver XML" flags:

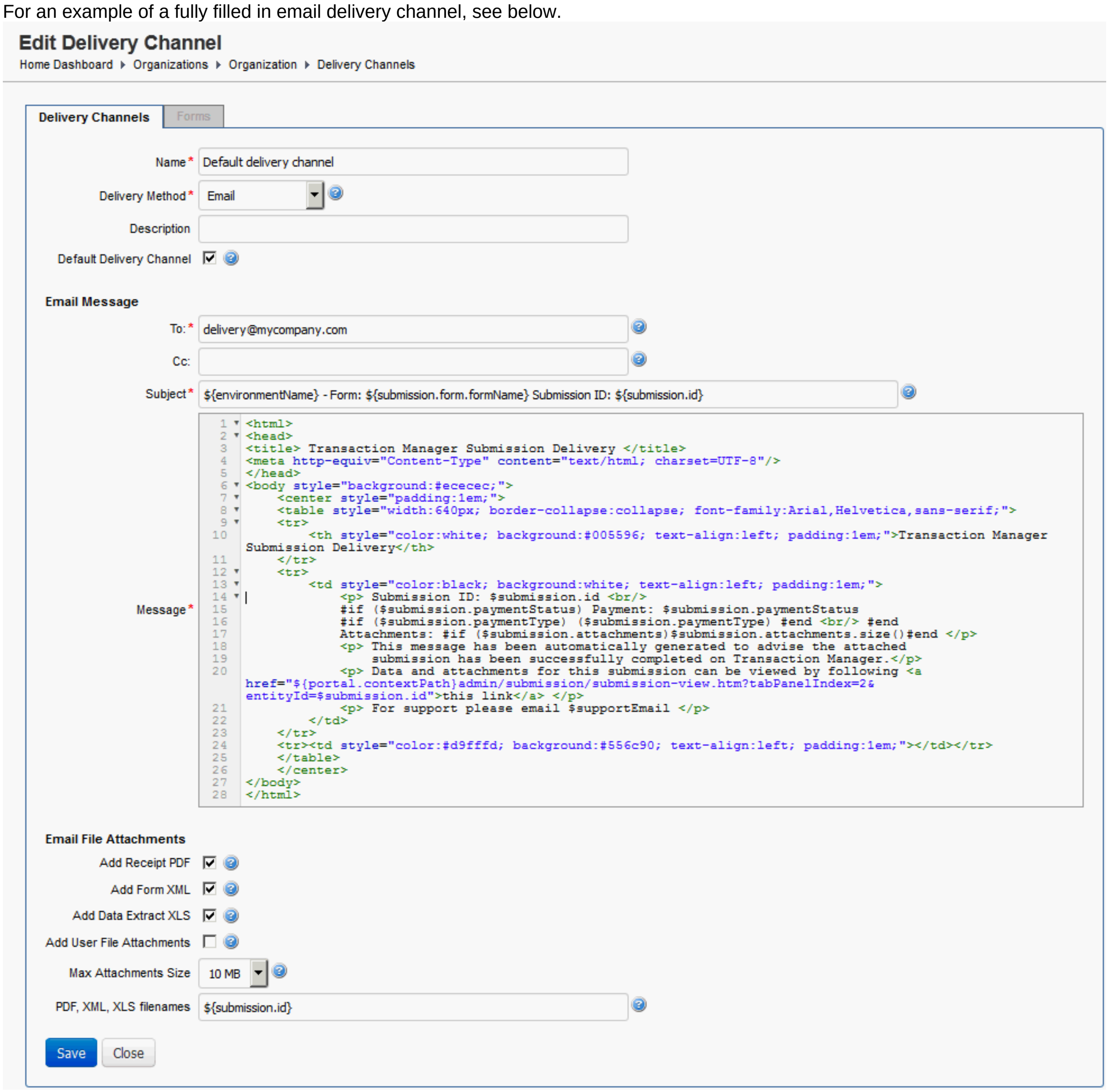
- If you want the PDF receipt to be included in delivery, tick the "Deliver PDF Receipt" checkbox.
- If you want the XML-based form data to be included in delivery, tick the "Deliver XML" checkbox.
- If you would like to include an Excel file containing submission extract data values (see Operations (Manager v5.0)#_sec_submissiondataextractvalues), tick the "Deliver Excel Receipt" checkbox.

The above attachments will be assigned file names by TM, by default using the submission ID. If you would like to change the name template, edit the "Attachment Filename Template" field.

You can include attachments uploaded by the submitting user by checking "Add User File Attachments".

If your email server has size limitations, you can specify the maximum total size of attachments with the "Max Attachments Size" dropdown. If the total attachment size exceeds the limit, TM will not attach any files but instead provide links in the email.

For an example of a fully filled in email delivery channel, see below.



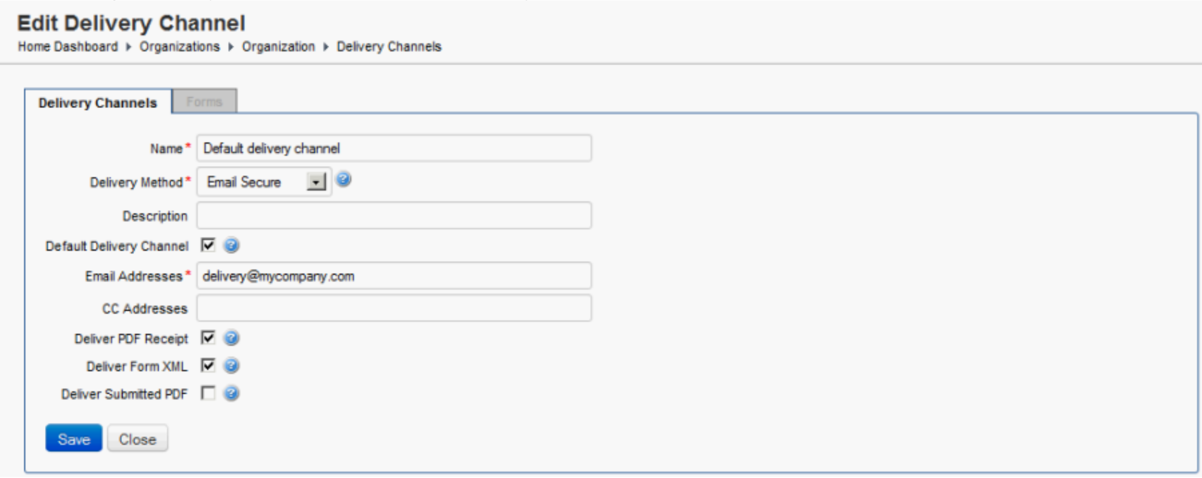
Setting up an email delivery channel

Creating a Secure Email Delivery Channel

Choose "Email Secure" from the "Delivery Method" dropdown.
The "Email Addresses" and "Cc Addresses" fields are the email addresses submissions will be delivered to. You can fill in multiple email addresses by separating them with commas.
To control what you wish to include in the delivery, you can use the "Deliver PDF", "Deliver XML" and "Deliver Submitted PDF" flags:

- If you want the PDF receipt to be included in delivery, tick the "Deliver PDF Receipt" checkbox.
- If you want the XML-based form data to be included in delivery, tick the "Deliver XML" checkbox.
- If you want the submitted PDF in editable form to be included in delivery, tick the "Deliver Submitted PDF" checkbox.

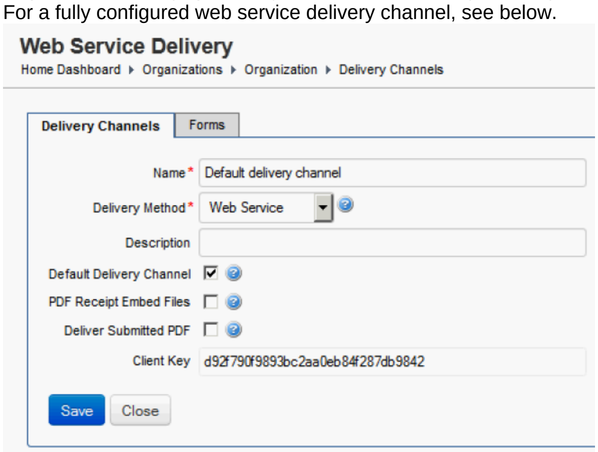
For an example of a fully filled in set of secure email delivery details, see below.



Setting up a secure email delivery channel

Creating a Web Service Delivery Channel

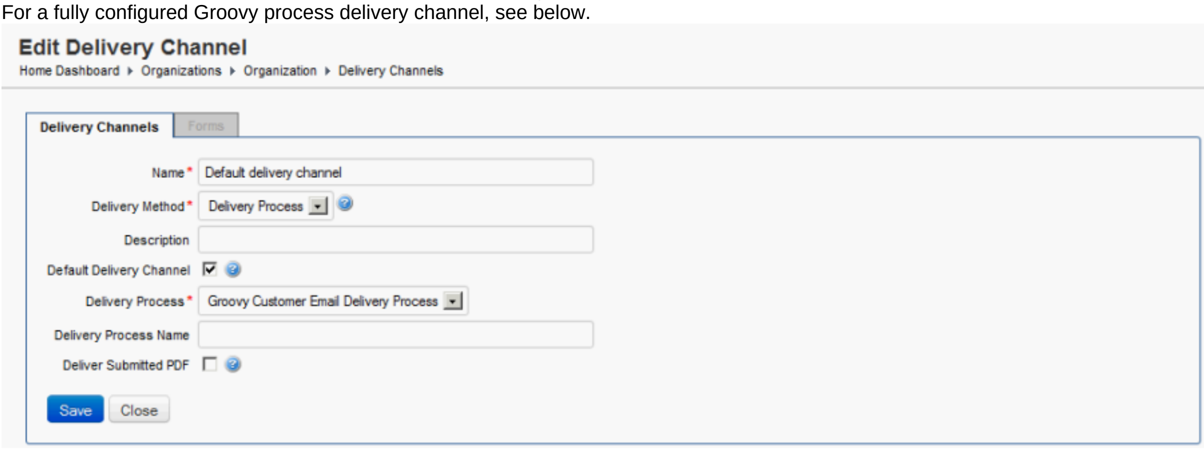
Choose "Web Service" from the "Delivery Method" dropdown.
Tick "PDF Receipt Embed Files" to cause the submitted XML and submission attachments to be embedded into the receipt during delivery.
If you want the submitted PDF in editable form to be included in delivery, tick the "Deliver Submitted PDF" checkbox.
The "Client Key" displays the client key for the organization to which the delivery channel belongs. The client key is used during delivery and can be helpful when setting up delivery on the Transact Integration Agent.
For a fully configured web service delivery channel, see below.



Setting up a web service delivery channel

Creating a Delivery Process Delivery Channel

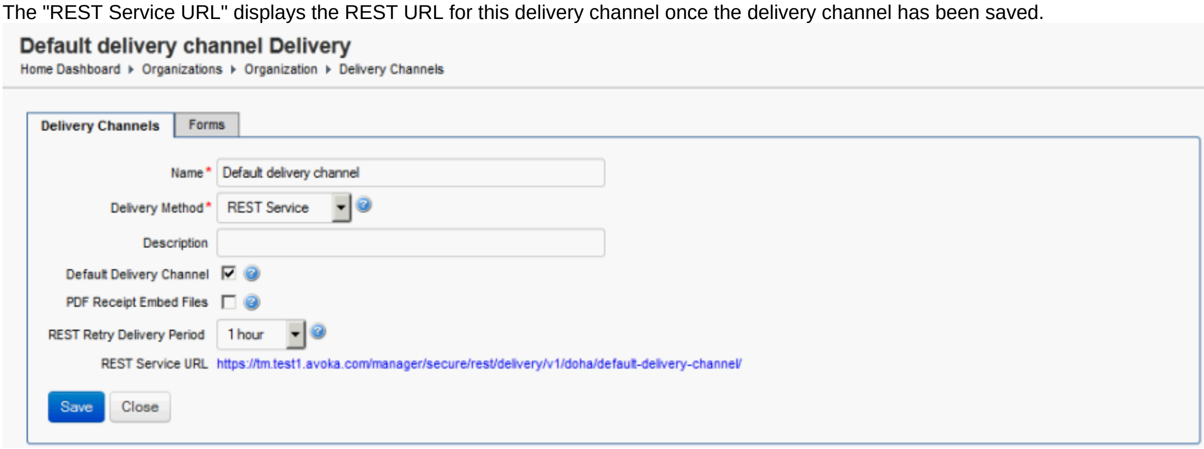
Choose "Delivery Process" from the "Delivery Method" dropdown. Choose the delivery process you want to use. Delivery processes are set up as service definitions (see Services (Manager v5.0) for more information on service definitions and service connections).
The "Delivery Process Name" field specifies which orchestration will be invoked (for LiveCycle delivery processes).
If you want the submitted PDF in editable form to be included in delivery, tick the "Deliver Submitted PDF" checkbox.
Once a submission is complete, the delivery process service passes the submission delivery service, including the submitted form XML, the receipt in PDF format, all attachments as well as an XML file containing metadata about the attachments.
For a fully configured Groovy process delivery channel, see below.



Setting up a process delivery channel

Creating a REST Delivery Channel

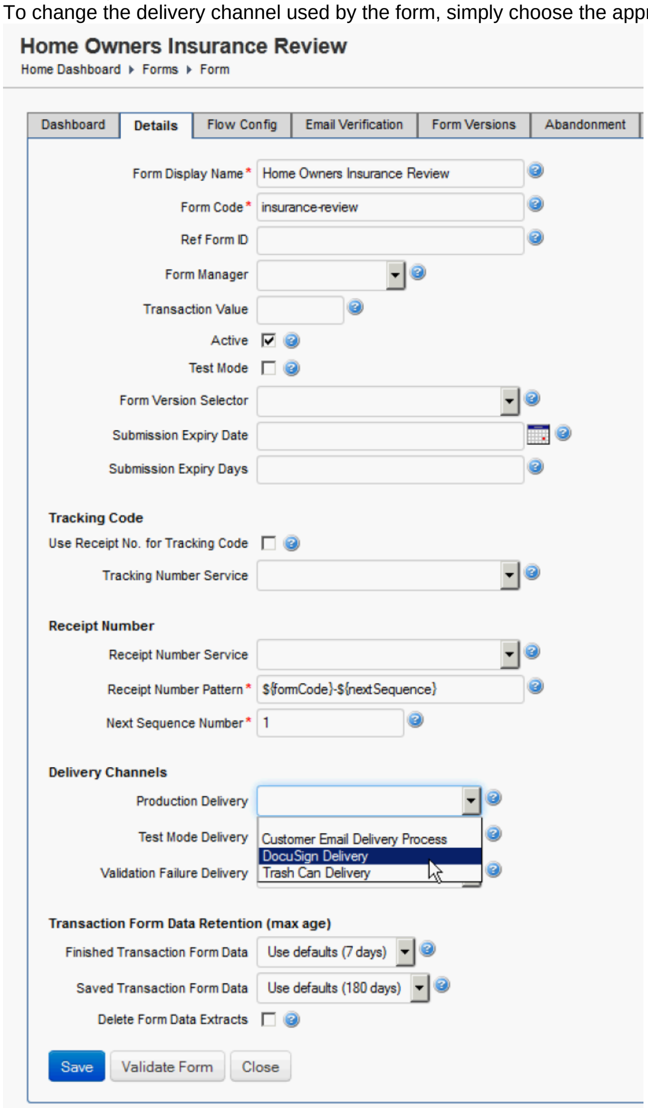
Choose "REST Service" from the "Delivery Method" dropdown.
Tick "PDF Receipt Embed Files" to cause the submitted XML and submission attachments to be embedded into the receipt during delivery.
The "REST Retry Delivery Period" determines after how long a submission is reset to delivery status "Ready" after having been in status "In Progress".
The "REST Service URL" displays the REST URL for this delivery channel once the delivery channel has been saved.



Setting up a REST delivery channel

Setting the Delivery Channel for a Form

To change the delivery channel used by a form, go to "Forms > Forms" and edit the form. The "Details" tab has two dropdown fields that control delivery channels: "Production Delivery" and "Test Delivery". Both offer all delivery channels set up for the form's organization. The production delivery details will be used if the form is submitted while in production mode ("Test Mode" checkbox unticked); otherwise the test delivery details apply.
If no delivery channel is set, submissions will be delivered via the default delivery channel for the organization, if one is defined.
To change the delivery channel used by the form, simply choose the appropriate entry from the dropdown lists and save.

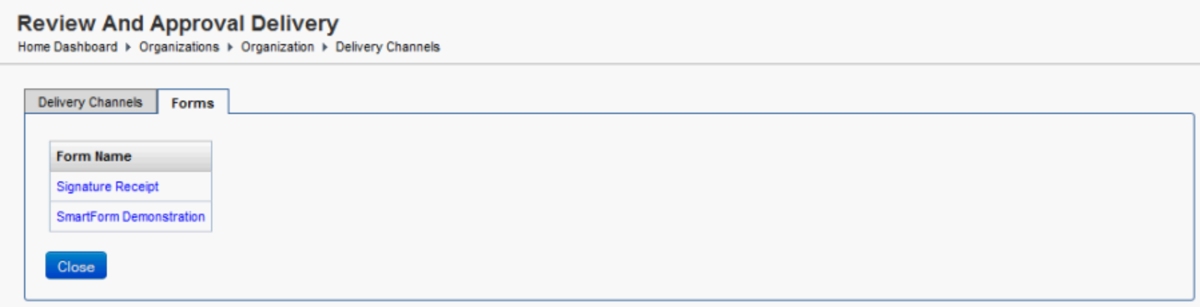


Changing the delivery channel for a form

All submissions made for this form after you have saved it will use the new delivery channels.
To set a delivery channel for abandoned forms, change to the "Abandonment" tab and select a delivery channel from the "Abandoned Delivery Channel" dropdown, then save.

Viewing Forms Associated with a Delivery Channel

When modifying or deleting a delivery channel, it can be helpful to know which forms are using it to avoid unwanted side effects. To see the list of forms using a given delivery channel, edit it and switch to the "Forms" tab.



The list of forms using a delivery channel

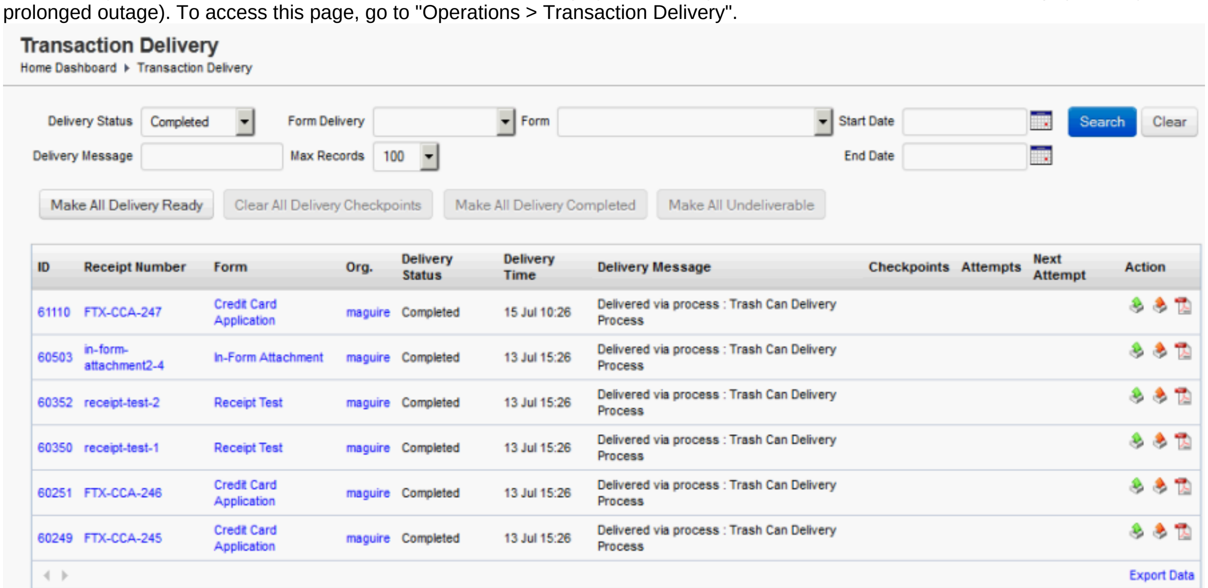
Note that this list includes only forms explicitly associated with the delivery channel. If a delivery channel is marked as the default for the current organization, it will be used for all forms not explicitly set to use other delivery channels.

Deleting a Delivery Channel

Delivery channels can be deleted; however, this will leave forms that had been using them without a delivery channel. To avoid delays in delivery, you can check the organization's forms and reassign delivery channels first, where necessary.
Once all forms have been reconfigured, go to "Forms > Organizations", edit the organization and switch to the "Delivery Channels" tab. Find the delivery channel that you want to remove and click the "Delete" icon. Because this action cannot be undone, you will be asked for confirmation.

Monitoring Submission Delivery

Transaction Manager automatically delivers submissions according to the delivery channel currently configured for the form (and the organization default delivery channel if the form is not assiocated with a delivery channel) and will automatically retry delivery if failures occur (depending on the delivery method used).
In addition, you can monitor submission delivery on a dedicated page and change the status for a set of submissions (e.g. to retry delivery after a prolonged outage). To access this page, go to "Operations > Transaction Delivery".



Monitoring submission delivery

You will see the list of submissions with a certain delivery status. You can apply additional filters such as the organization associated with the submission, the delivery channel, the form, a substring of the delivery message and the delivery date and time.
You can view details for each submission, which also allows you to change the delivery status for an individual submission (see Operations (Manager v5.0)#_sec_submissionstatusvalues). To apply a bulk delivery status change to all submissions in the current search result, you can use the "Make All Delivery Ready" (reset delivery status to "Ready" so delivery will be retried), "Clear All Delivery Checkpoints" (relevant only if the delivery service supports delivery checkpoints), "Make All Delivery Completed" (set delivery status to "Completed" without retrying delivery) and "Make All Undeliverable" (set delivery status to "Undeliverable", meaning that delivery will not be reattempted) button.

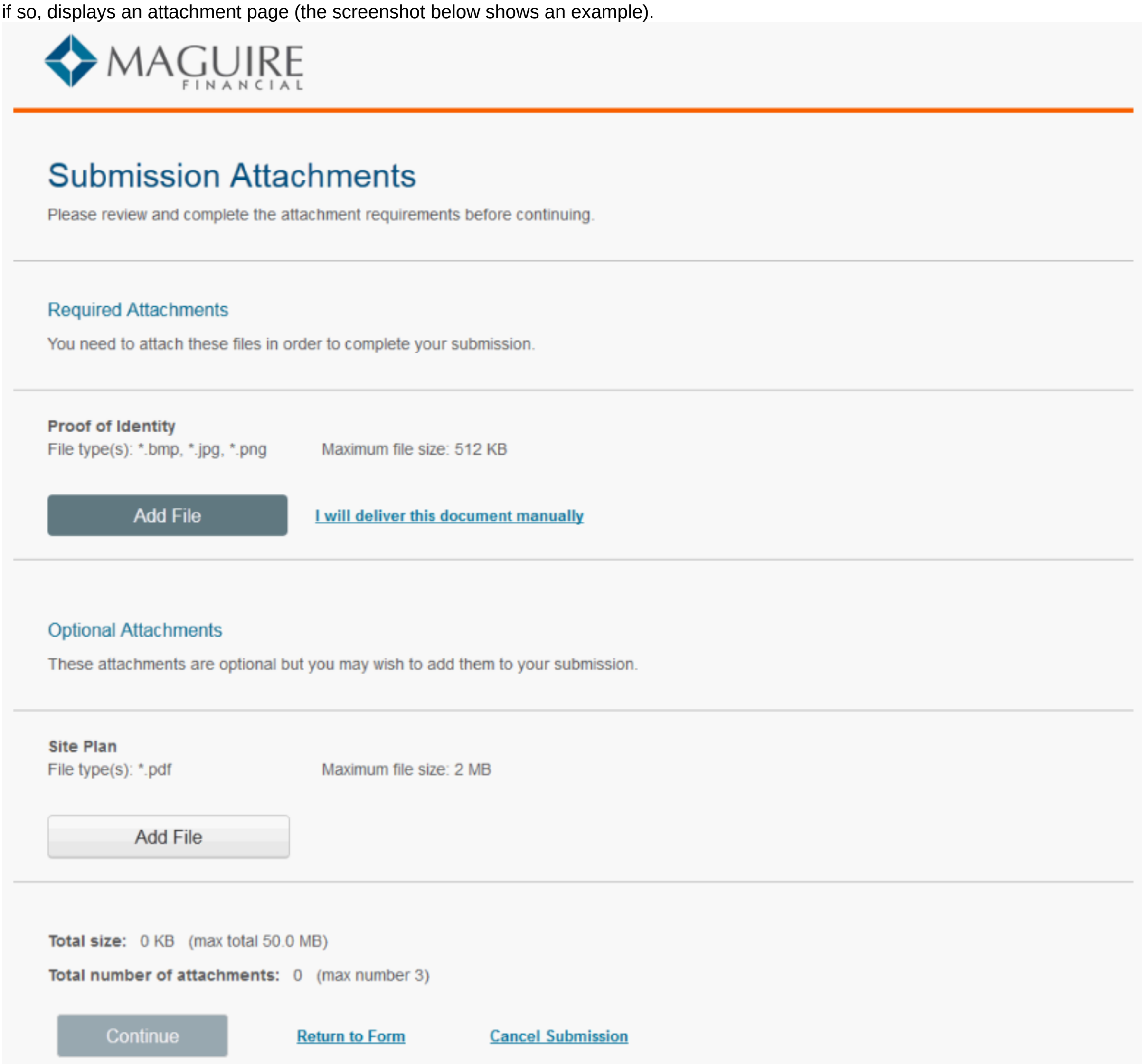
Managing Form Attachments (Manager v5.0)

(Forms are very often not stand-alone documents but include other aspects such as supporting documentation. This chapter details how to set up supporting documentation ("attachments") in Transaction Manager.

Search

Concepts

Attachments are separate entities (files or physical documents) that have to be submitted alongside the form. Forms may have multiple attachments, some of which may be required to complete the submission, while others may be optional.
Attachments can be submitted in one of two ways: Either the user attaches them to their Transaction Manager submission by uploading files (*electronic submission*), or the user submits them in any other way, e.g. by mail or fax (*manual submission*). You can choose to allow one or both methods when defining an attachment.
Please note that Transaction Manager considers a submission complete only if all required attachments with electronic submission have been attached by the user. Transaction Manager cannot check attachments that the user submits manually (by definition, such attachments are submitted independently of Transaction Manager).
Attachments are defined for each form version. When a form is submitted, Transaction Manager checks whether attachments have been set up for it and if so, displays an attachment page (the screenshot below shows an example).



TM attachments page for a submission

As you can see, attachments are split into required and optional attachments. The user can decide how they want to submit attachments if they were set up to allow both electronic and manual submission. A user can also upload files on the same page and gets a list of all files uploaded so far. Multiple files can be attached for a single attachment.

Certain types of forms can also contain *embedded attachments*, which the user uploads using the PDF viewer. Transaction Manager can store these attachments when the form is submitted, and they are also visible to administrators and are included in delivery.

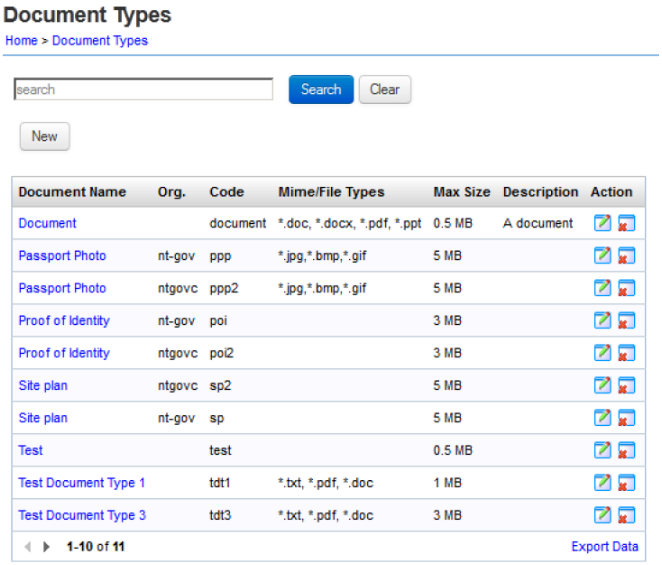
The attachments submitted electronically by the user can be viewed and downloaded via Transaction Manager directly.

Attachment Types

Every attachment managed by Transaction Manager can be associated with a document type, e.g. "Detailed Specification". This allows you to define attachment categories you can then code against if needed.

Browsing Document Types

To view all document types currently defined in Transaction Manager, go to "Forms > Document Types". You can filter the list by organization or using a substring of the document type's name, code or description, and export the search results to Excel by clicking the "Export Data" link below the table.



Document Name	Org.	Code	Mime/File Types	Max Size	Description	Action
Document		document	*.doc, *.docx, *.pdf, *.ppt	0.5 MB	A document	
Passport Photo	nt-gov	ppp	*.jpg,*.bmp,*.gif	5 MB		
Passport Photo	ntgovc	ppp2	*.jpg,*.bmp,*.gif	5 MB		
Proof of Identity	nt-gov	poi		3 MB		
Proof of Identity	ntgovc	poi2		3 MB		
Site plan	ntgovc	sp2		5 MB		
Site plan	nt-gov	sp		5 MB		
Test		test		0.5 MB		
Test Document Type 1		tdt1	*.txt, *.pdf, *.doc	1 MB		
Test Document Type 3		tdt3	*.txt, *.pdf, *.doc	3 MB		

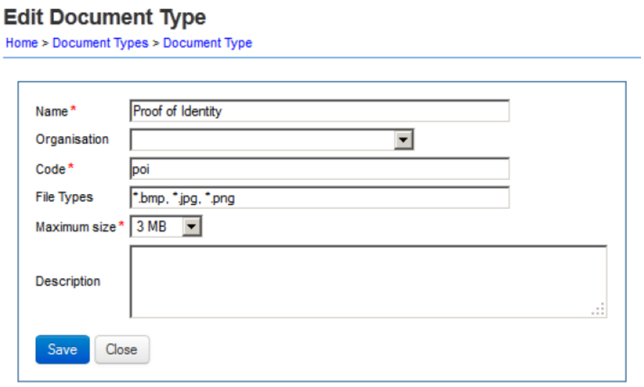
1-10 of 11 Export Data

Browsing the list of document types

You can add, edit ( ) and delete ( ) document types from this page.

Adding a Document Type

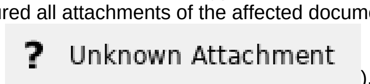
You can define new document types as needed. Go to "Forms > Document Types" and click "New". You will be directed to the "Edit Document Type" page where you can fill in the details for the new document type.



Creating a document type

Enter the name and, optionally, a description of the document type. This description is used only to characterize document types internally and will never be displayed to space users.
If the document type is specific to a particular organization, select it from the "Organization" dropdown. If you do not set an organization, the document type will be globally available.
Enter a unique document type code into the "Code" field. This can be any string.
If you want to restrict the document type to certain file types, you can enter the supported file extensions into "File Types" field. Please enter a string such as "*.jpg, *.gif, *.png" (i.e. extensions in the format *.<*extension*> separated by commas).
Choose the maximum size for attachments of this document from the "Maximum Size" dropdown.
After you save the document type, it can be assigned to your attachments. Note that while these attachments will default to the document type's allowed file types and maximum size, each attachment can be modified to use different values.

Deleting Document Types

If you want to permanently remove a document type, you can delete it as long as it is not currently in use. This means that you need to review your attachments to make sure no attachments of that document type exist.
Once you have removed or reconfigured all attachments of the affected document type, go to "Forms > Document Types", find the document type to be removed and click the "Delete" icon (

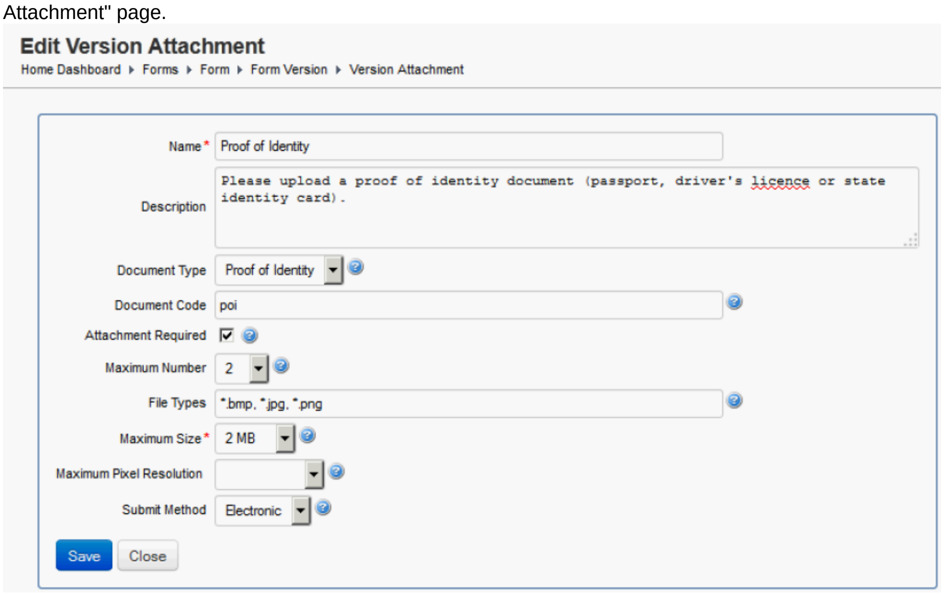
). You will be asked to confirm your decision before the document type is deleted.

Configuring a Form to Support Attachments

Transaction Manager populates the form with the list of specified attachments and later reads the list of required attachments from the form when it is submitted. While this gives the form the opportunity to manipulate the list of attachments, it also means that the form must contain a certain data structure that holds the specified attachments.
To enable Transaction Manager to read and write the form's attachment section, you may need to add a configuration mapping to the form data configuration. Please refer to Form Version Configuration (Manager v5.0)#_sec_configurationmappings on how to do this; the mapping you need is from "Attachment" to the XPath of the attachment section in your form.

Adding an Attachment to a Form Version

To add an attachment to a form version, edit the form version and switch to the "Attachments" tab. Click the "New" button. You will see the "Edit Version Attachment" page.

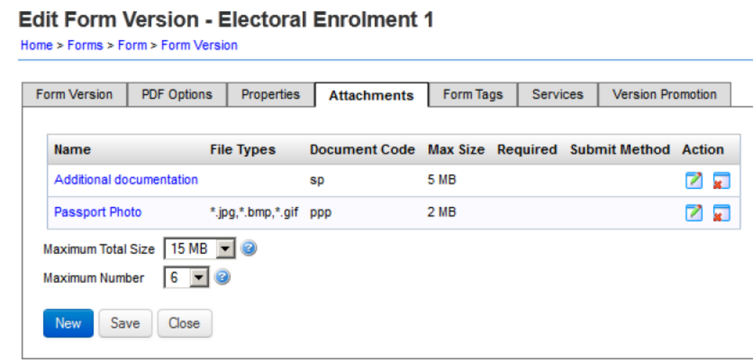


Creating an attachment for a form version

Fill in the attachment name as you want it to appear to the space user. Add a description to help the user determine what to attach and if more than one copy is required for manual submission.
Choose the most appropriate document type from the "Document Type" dropdown. This will initialize the document code, file types and maximum size with the values stored in the document type. You can also create an attachment that is not based on a document type, and fill in the "Document Code", "File Types" and "Maximum Size" fields yourself.
Please note that the user space will be able to enforce that only files matching the allowed file types and maximum size can be uploaded. The Example Self Service Space rejects files that do not match these criteria.
The "Attachment Required" checkbox controls whether the user will have to provide this attachment for the submission to complete. If left unticked, the attachment is treated as optional.
The "Maximum Number" dropdown lets you control the maximum number of files that the user can attach for this attachment. This restriction applies in addition to global restrictions to the number of overall attachments.
The "Maximum Pixel Resolution" is used on the TransactField App only, to restrict image files to a maximum resolution.
Finally, the "Submit Method" determines how users can submit their attachments. The choices are electronic (attach directly via the Transaction Manager attachments page), manual (any other way) or both. Please note that if you choose manual submission only, the organization will be responsible for monitoring and managing attachments. Please also indicate to the user what manual submission means for them – either in the form or the attachment description.
When you save the data, the new attachment will be created.

Size Limits

As you have seen, the size of each individual attachment can be limited. While this is useful, it applies only to a single file uploaded by a user. It can be helpful to restrict the total number and size of all attachments combined. Transaction Manager allows you to do this for each form version. This will affect all attachments made for a submission, including embedded attachments.
To limit the total attachment size, edit the form version and switch to the "Attachments" tab.



Viewing the list of attachments and setting size limits

Select the desired total attachment size from the "Maximum Total Size" dropdown.
If you would like to limit the number of attachments that are submitted, choose the maximum number from the "Maximum Number" dropdown.
Save your changes.

Deleting Attachments

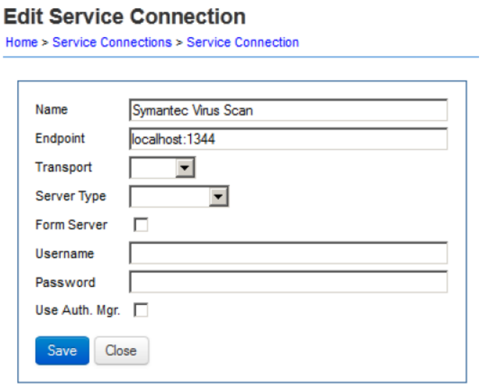
If you want to delete an attachment so Transaction Manager no longer asks for it to be submitted, edit the form version and switch to the "Attachments" tab. Find the attachment you want to delete and click the "Delete" icon ( ).
After you confirm your decision, the attachment will be removed. Previous submissions with attachments remain unaffected by this; Transaction Manager will simply no longer request the attachment in the future.

Virus Scanning

To avoid users uploading files infected with viruses, Transaction Manager includes support for virus scanning. If a virus scanning service is configured, Transaction Manager will maintain a virus status for each uploaded file.
If a virus scanning service is configured, delivery will be initiated only once all attachments have been scanned and are virus-free. Files that are found to contain a virus will be deleted to avoid contamination. The Example Self Service Space will scan files as they are uploaded and will give immediate feedback to the user in case a virus was found.
Transaction Manager includes support for two virus scan engines: Symantec Scan Engine 5.2 and ClamAV.

Setting Up the Symantec Virus Scanning Service

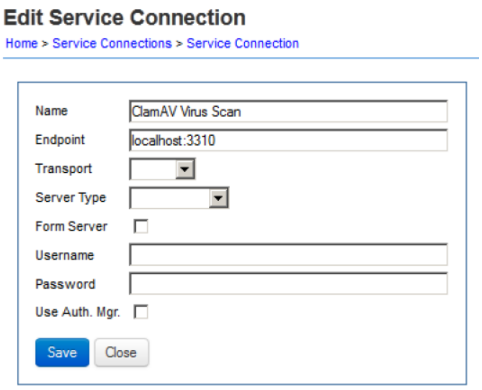
Transaction Manager includes a virus scanning service that can connect to the Symantec Scan Engine 5.2 (http://www.symantec.com/business/scan-engine). Make sure you have set up Symantec Scan Engine before you activate the service.
If Symantec Scan Engine is running on a different server or a port other than the default (1344), you will need to edit the service connection. Go to "System > Service Connections" and edit the service connection named "Symantec Virus Scan". Adjust the "Endpoint Value" field as needed and save.



The service connection for the Symantec virus scan service
Activate the Symantec virus scan service by following the instructions in _sec_activatevirusscanservice. The relevant service definition is named "Virus Scan".

Setting Up the ClamAV Virus Scanning Service

Transaction Manager also contains a virus scanning service that can connect to ClamAV (http://www.clamav.net/). Make sure that ClamAV is set up before you activate the Transaction Manager service.
If ClamAV is running on a different server or a port other than the default (3310), you will need to edit the service connection. Go to "System > Service Connections" and edit the service connection named "ClamAV Virus Scan". Adjust the "Endpoint Value" field as needed and save.



The service connection for the ClamAV virus scan service
Activate the ClamAV virus scan service by following the instructions in _sec_activatevirusscanservice. The relevant service definition is named "ClamAV Virus Scan".

Activating a Virus Scanning Service

To activate a virus scanning service, go to "System > Service Definitions", find the appropriate service of type "Virus Scan" and edit it. Tick the "Service Type Default" checkbox and make sure the "Service Connection" dropdown is set correctly. Save the service definition.
To deactivate the virus scanning service, simply edit it again, untick the "Service Type Default" checkbox and save.
Once you have turned on the virus scanning service, attachments will be scanned automatically. Transaction Manager can handle temporary outages of the virus scan engine and will simply re-attempt to have unscanned attachments scanned periodically.

Embedded Attachments

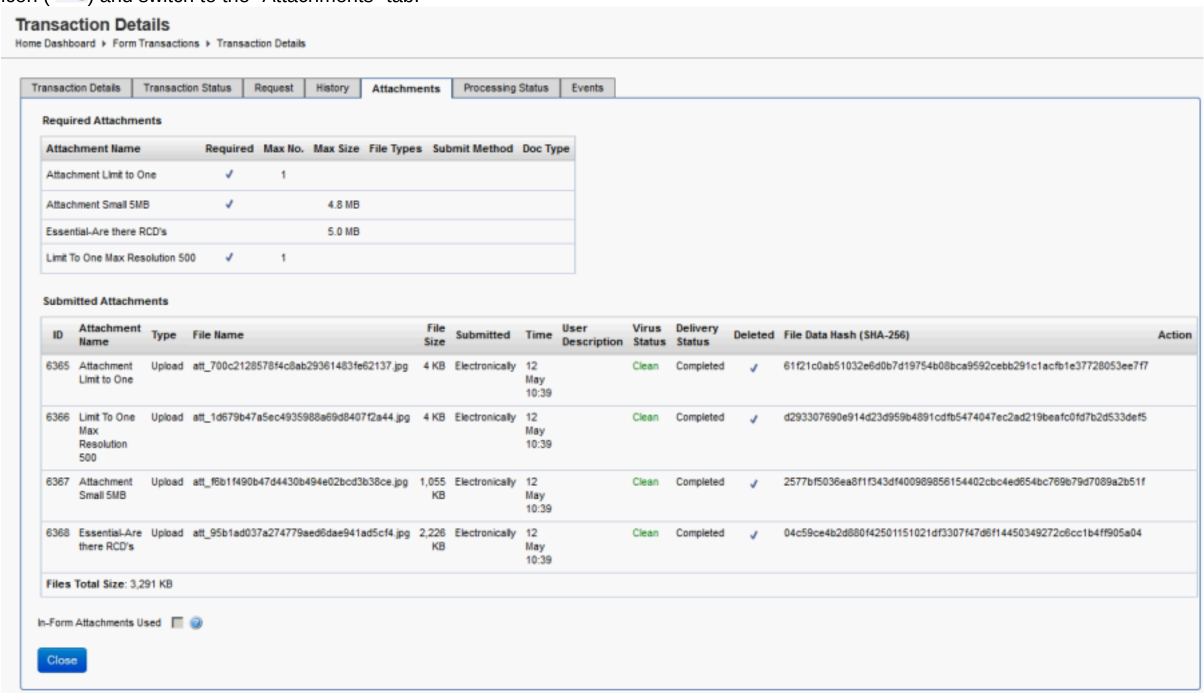
Embedded attachments are files that are uploaded to the PDF form directly, using the PDF viewer (as opposed to attachments defined in and uploaded via Transaction Manager).
These attachments can be used only with reader-extended dynamic PDF forms that submit as PDF (as opposed to XML or XDP). When the form is submitted, Transaction Manager will extract the attached files and add them to the submission.
Administrators alongside other attachments can view these embedded attachments and will be virus scanned and included in delivery. However, embedded attachments are not managed by Transaction Manager, and hence are not individually constrained in their maximum size or file type. Note that the form can be designed to enforce constraints on these attachments, and they are also subject to the total attachment size configured for the form.

Managing Submitted Attachments

This section describes how attachments that are submitted via Transaction Manager (including embedded attachments) can be viewed and managed.

Viewing Attachments for a Submission

To view all attachments for a submission, go to "Operations > Form Transactions". Find the submission you are interested in, click the "View Details" icon ( ) and switch to the "Attachments" tab.



Transaction Details

Home Dashboard ▸ Form Transactions ▸ Transaction Details

Transaction Details | Transaction Status | Request | History | Attachments | Processing Status | Events

Required Attachments

Attachment Name	Required	Max No.	Max Size	File Types	Submit Method	Doc Type
Attachment Limit to One	✓	1				
Attachment Small 5MB	✓		4.8 MB			
Essential-Are there RCD's			5.0 MB			
Limit To One Max Resolution 500	✓	1				

Submitted Attachments

ID	Attachment Name	Type	File Name	File Size	Submitted	Time	User Description	Virus Status	Delivery Status	Deleted	File Data Hash (SHA-256)	Action
6365	Attachment Limit to One	Upload	att_700c2128578f4c8ab29361483fe62137.jpg	4 KB	Electronically	12 May 10:39		Clean	Completed	✓	61f21c0ab51032e6d0b7d19754b08bca9592cebb291c1acfb1e37728053ee7f7	
6366	Limit To One Max Resolution 500	Upload	att_1d679b47a5ec4935988a69d8407f2a44.jpg	4 KB	Electronically	12 May 10:39		Clean	Completed	✓	d293307690e914d23d959b4891cdfb5474047ec2ad219beafc0fd7b2d533def5	
6367	Attachment Small 5MB	Upload	att_f6b1f490b47d4430b494e02bcd3b38ce.jpg	1,055 KB	Electronically	12 May 10:39		Clean	Completed	✓	2577bf5036ea8f1f343df400989856154402cbc4ed654bc769b79d7089a2b51f	
6368	Essential-Are there RCD's	Upload	att_95b1ad037a274779aed6dae941ad5cf4.jpg	2,226 KB	Electronically	12 May 10:39		Clean	Completed	✓	04c59ce4b2d880f42501151021df3307f47d6f14450349272c6cc1b4ff905a04	

Files Total Size: 3,291 KB

In-Form Attachments Used

Close

The attachments for a submitted form

The "Attachment Type" column contains the type of attachment. This can be one of the following values: "Upload" for a managed attachment uploaded by the user; "Embedded" for an embedded attachment; "Signed Receipt" for a scanned-in signed form receipt uploaded for a form requiring wet signatures; and "Submitted Form" form for PDF forms submitted as PDF, where the submitted form is stored as a separate attachment.
The "Attachment Name" column contains the name of the corresponding attachment set up in the form. For embedded attachment, this column is blank.
The "Doc Type" column contains the name of the document type of the corresponding version attachment. For embedded attachments, this column is blank.
The "Required" column shows whether the attachment was mandatory or optional.
The "Submitted" column shows how the user chose to submit the attachment and will be set to either "Electronically" or "Manually".
The "File Name" and "File Size (KB)" columns contain the file name and size in KB of the file that was uploaded. This column is blank if the attachment was submitted manually.
The "Virus Status" column contains the virus scan status for the uploaded file. This column is blank if the attachment was submitted manually or no virus scan service has been configured. Otherwise, it may also contain the values "Clean" (scanned successfully; no virus found), "Virus" (scanned successfully; virus found) and "Error" (error during virus scan; repeat scan pending).
The "Delivery Status" column contains the delivery status for the attachment. The column will change to "Completed" once the attachment has been delivered successfully.

Click the "Download File" icon ( ) to save the file to disk. This will download the file using your browser's normal download routines.
Please note that Transaction Manager provides an automated purge mechanism for submissions, attachments and many other entities (see Operations (Manager v5.0)#_sec_dataretention).

Online Payment (Manager v5.0)

A submission often requires payment to be processed further. Transaction Manager includes a payment module to manage online credit card payments. This chapter explains how to set up organizations and forms to use the payment module.

Search

Concepts

Forms in Transaction Manager can request payment once they have been completed. The total payment consists of one or more *payment line items*. Payment can either be requested via credit card, BPAY or using external payment services such as PayPal.
If BPAY is requested, the form should display the biller code and reference number to the user. Transaction Manager cannot process BPAY payments, so it does not provide a separate payment step for submissions using BPAY.

If you would like to process payment directly via Transaction Manager, there are two different payment modes you can choose from: 2-party services, where the payment is processed via a payment page in the Transaction Manager user space, with Transaction Manager calling the payment provider directly; or hosted (3-party) services, where the user is redirected to the payment provider's website, makes payment there, and is then redirected back to Transaction Manager.

Transaction Manager integrates with a number of 2-party payment providers as well as several hosted payment services (including PayPal). The 2-party payment services will ask for payment via credit/debit card, whereas hosted payment services can be very flexible (since the payment provider's website is used). For example, with the PayPal payment service, the user can choose to use their PayPal account, or simply enter credit card details instead.

If a submitted form indicates that payment should be performed via the 2-party model, Transaction Manager reads out the payment amount from the payment section in the form and requests credit card payment from the user by displaying a payment page.
Once the user makes payment via credit card, the payment details are forwarded to the *payment gateway* set up for the organization. A payment gateway processes online credit card payments and forwards them to the organization's bank account. Transaction Manager comes with adapters for several payment gateways. Generally, an account must be set up with the payment gateway for each organization that wishes to use it.

If, on the other hand, a submitted form indicates that payment should be performed using a hosted payment gateway, the user is directed to a page in Transaction Manager that contains information about the payment as well as a link to the payment provider's website. The user can click this link, make their payment and then get redirected back to Transaction Manager to continue the submission process.

Once payment is complete, the submission can progress and eventually be delivered. For security reasons, a user's credit card details are not stored in Transaction Manager. Instead, records on successful and failed payment attempts are kept, which can be used to reconcile payments made in Transaction Manager with payments actually received by the organization.

For payment reconciliation, payment gateways commonly report on credit card payments for a *batch*. A batch consists of payments made during 24 hours, but does not normally coincide with a calendar day. Instead, a batch changes when the organization's bank stops processing payments for the day, e.g. at 5:30 pm. Payments made after this time (the *batch cutoff time*) are processed during the next calendar day. For accurate reconciliation reporting, it is important to know the batch cutoff time that is appropriate for the organization's bank.

Payment Gateways

To process a payment (whether using credit card details entered by the user in the 2-party model, or redirecting to an external website in the hosted model), Transaction Manager needs to call a payment gateway that either confirms or rejects the payment. Payment gateways are set up as service definitions in Transaction Manager (see Services (Manager v5.0)). By default, Transaction Manager provides a number of payment services as described below.
Payment gateway configuration can be on one of two levels: The global level of the payment service definition (usually containing parameters such as the payment gateway URL) and the organization level (containing information about the organization's account with the payment gateway such as user name).

Two-Party Payment Gateways

This section lists the two-party payment gateways supported by Transaction Manager, and how to configure them.

Demo Card Payment

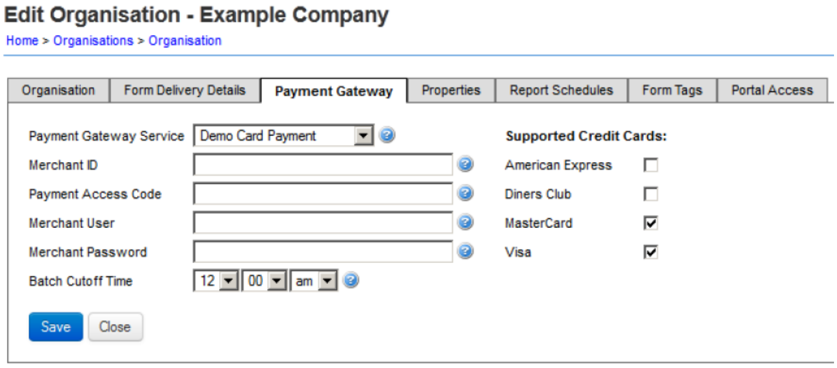
This payment service can be used for testing and demonstration purposes. It is not linked to a gateway; instead, it accepts all credit card payments without actually processing them.

Global Configuration

No special configuration is needed for this service.

Organization Configuration

To configure an organization to use the Demo Card Payment service, go to "Forms > Organizations", edit the organization and switch to the "Payment Gateway" tab.
Choose "Demo Card Payment" from the "Payment Gateway Service" dropdown. No other values are required.
Save your changes.



An example configuration for the Demo Card Payment service

NAB Card Payment

This payment service connects to the NAB payment gateway (https://transact.nab.com.au). To use it, the organization needs to set up an account with NAB and enter its merchant details into Transaction Manager.

Global Configuration

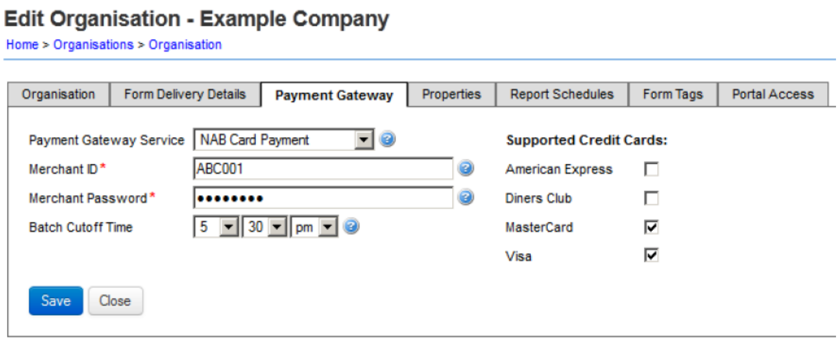
The NAB card payment service has two global configuration parameters: The URL to the payment gateway server and the transaction timeout value.
To configure these parameters, go to "System > Service Definitions", find the NAB card payment service and edit it. Switch to the "Service Parameters" tab.
To change the URL to the payment gateway server, edit the "paymentUrl" parameter and enter one of the payment URLs below (unless provided with different URLs by NAB):

Test	https://transact.nab.com.au/test/xmlapi/payment
Live	https://transact.nab.com.au/live/xmlapi/payment

To change the transaction timeout value, edit the "timeout" parameter and enter the desired timeout value in minutes.

Organization Configuration

To configure an organization to use the NAB card payment service, ensure the organization has obtained a merchant account with NAB first.
Go to "Forms > Organizations", edit the organization and switch to the "Payment Gateway" tab (see below).
Choose "NAB Card Payment" from the "Payment Gateway Service" dropdown.
Enter the organization's merchant ID into the "Merchant ID" field.
Enter the merchant password into the "Merchant Password" field.
Enter a batch cutoff time consistent with the organization's bank into the "Batch Cutoff Time" field.
Save your changes.



An example configuration for the NAB Card Payment service

TNS Card Payment and CommWeb Card Payment

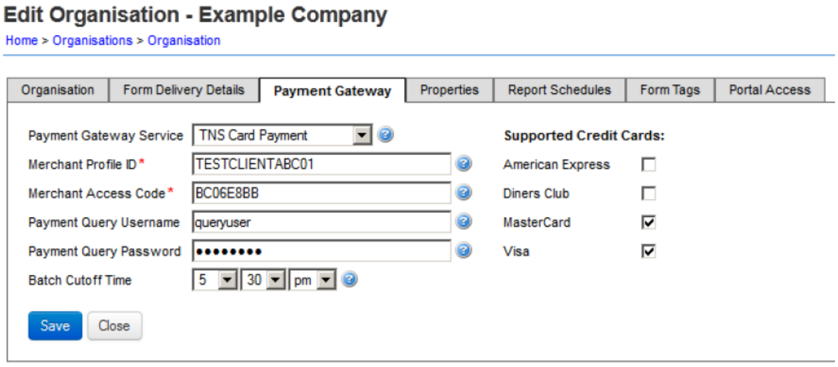
These payment services both connect to the TNS payment gateway (http://www.tnsi.com/). To use it, the organization needs to set up an account with TNS and enter its merchant details into Transaction Manager.

Global Configuration

The TNS and CommWeb card payment services have two global configuration parameters relevant to making payments: The URL to the payment gateway server and the transaction timeout value.
To configure these parameters, go to "System > Service Definitions", find the payment service and edit it. Switch to the "Service Parameters" tab.
Set the "paymentUrl" to https://migs.mastercard.com.au/vpcdps unless instructed otherwise by the gateway provider.
To change the transaction timeout value, edit the "timeout" parameter and enter the desired timeout value in minutes.

Organization Configuration

To configure an organization to use the TNS or Commweb card payment service, ensure the organization has obtained a merchant account on the TNS or Commweb payment gateway first. In addition, a payment query user needs to be set up on the web administration console associated with the payment gateway. This user needs to have the permission to query past payments (permission "Enable advanced Merchant Administration features"). Please contact your payment gateway provider for assistance if required.
Go to "Forms > Organizations", edit the organization and switch to the "Payment Gateway" tab (see below).
Choose "TNS Card Payment" or "CommWeb Card Payment" from the "Payment Gateway Service" dropdown.
Enter the organization's Merchant Profile ID into the "Merchant Profile ID" field. Note that each organization is normally provided with two profiles, one for testing and one for production use.
Enter the merchant access code into the "Merchant Access Code" field.
Enter the user name of the payment query user into the "Payment Query Username" field.
Enter the password of the payment query user into the "Payment Query Password" field.
For later reconciliation of payments received via Transaction Manager, enter the batch cutoff time (specific to the bank used by the organization) into the "Payment Batch Cutoff Time" field.
Tick the checkboxes for the credit cards that are enabled for your merchant account and save your changes.



An example configuration for the TNS Payment Gateway service

Westpac Card Payment

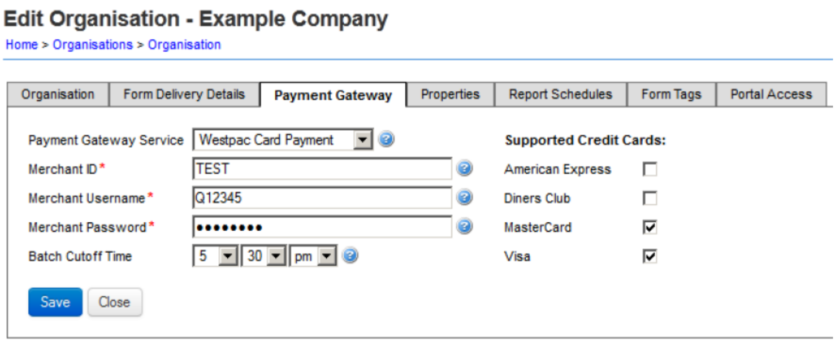
This payment service connects to the Westpac payment gateway (https://www.payway.com.au). To use it, the organization needs to set up an account with Westpac and enter its merchant details into Transaction Manager.

Global Configuration

No special configuration is needed for this service.

Organization Configuration

To configure an organization to use the Westpac card payment service, ensure the organization has obtained a merchant account with Westpac first.
The organization will also receive a certificate file from Westpac, which must be stored to a directory accessible to Transaction Manager.
Go to "Forms > Organizations", edit the organization and switch to the "Payment Gateway" tab (see below).
Choose "Westpac Card Payment" from the "Payment Gateway Service" dropdown.
Enter the organization's merchant ID into the "Merchant ID" field. Please note each organization is usually provided with two IDs, one for testing and one for production use.
Enter the merchant username into the "Merchant Username" field.
Enter the merchant password into the "Merchant Password" field.
Set the "Batch Cutoff Time" field to 6:00 pm unless advised otherwise by Westpac.
Tick the checkboxes for the credit cards that are enabled for your merchant account.



An example configuration for the Westpac Card Payment service

Save your changes and switch to the "Properties" tab.
Create a new organization property of type "Payment Certificate File" and set it to the full path to the certificate file obtained from Westpac for the organization (e.g. C:\paywayCert\ccapi.q0).

Bizgate Card Payment

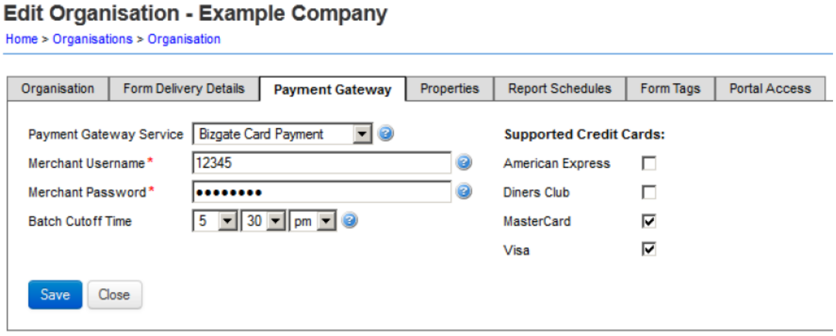
This payment service connects to the Bizgate payment gateway (http://www.bizgate.sa.gov.au/). To use it, the organization needs to set up an account with Bizgate and enter its merchant details into Transaction Manager.

Global Configuration

The Bizgate card payment service allows you to configure the URL to the payment gateway server.
To configure the URL, go to "System > Service Definitions", find the Bizgate card payment service and edit it. Switch to the "Service Parameters" tab.
Edit the "bizgateUrl" parameter and enter one of the payment URLs provided by Bizgate.

Organization Configuration

To configure an organization to use the Bizgate card payment service, ensure the organization has obtained a merchant account with Bizgate first.
Go to "Forms > Organizations", edit the organization and switch to the "Payment Gateway" tab (see below).
Choose "Bizgate Card Payment" from the "Payment Gateway Service" dropdown.
Enter the merchant username into the "Merchant Username" field.
Enter the merchant password into the "Merchant Password" field.
For later reconciliation of payments received via Transaction Manager, enter the batch cutoff time (specific to the bank used by the organization) into the "Payment Batch Cutoff Time" field.
Tick the checkboxes for the credit cards that are enabled for your merchant account and save your changes.



An example configuration for the Bizgate Card Payment service

SecurePay Card Payment

This payment service connects to the SecurePay payment gateway (http://www.securepay.com.au). To use it, the organization needs to set up an account with SecurePay and enter its merchant details into Transaction Manager.

Global Configuration

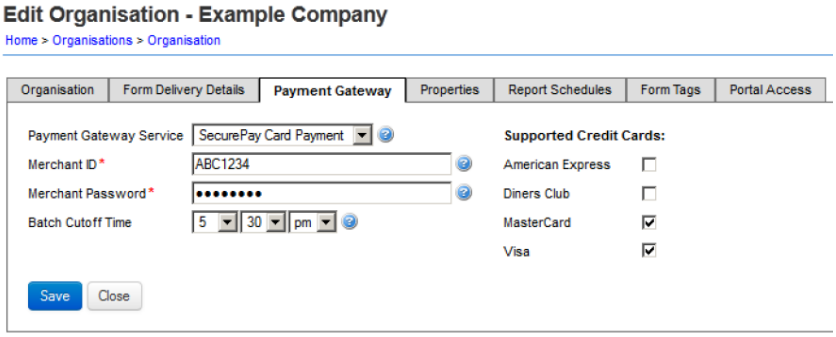
The SecurePay card payment service has two global configuration parameters: The URL to the payment gateway server and the transaction timeout value.
To configure these parameters, go to "System > Service Definitions", find the SecurePay card payment service and edit it. Switch to the "Service Parameters" tab.
To change the URL to the payment gateway server, edit the "paymentUrl" parameter and enter one of the payment URLs below (unless provided with different URLs by SecurePay):

Test	http://test.securepay.com.au/xmlapi/payment
Live	https://api.securepay.com.au/xmlapi/payment

To change the transaction timeout value, edit the "timeout" parameter and enter the desired timeout value in minutes.

Organization Configuration

To configure an organization to use the SecurePay card payment service, ensure the organization has obtained a merchant account with SecurePay first.
Go to "Forms > Organizations", edit the organization and switch to the "Payment Gateway" tab (see below).
Choose "SecurePay Card Payment" from the "Payment Gateway Service" dropdown.
Enter the organization's merchant ID into the "Merchant ID" field.
Enter the merchant password into the "Merchant Password" field.
Enter a batch cutoff time consistent with the organization's bank into the "Batch Cutoff Time" field.
Tick the checkboxes for the credit cards that are enabled for your merchant account and save your changes.



An example configuration for the SecurePay Card Payment service

BPoint Payment Gateway

This payment service connects to the BPoint payment gateway (https://www.bpoint.com.au/bpoint/).
To use it, the organization needs a BPoint merchant account.

Global Configuration

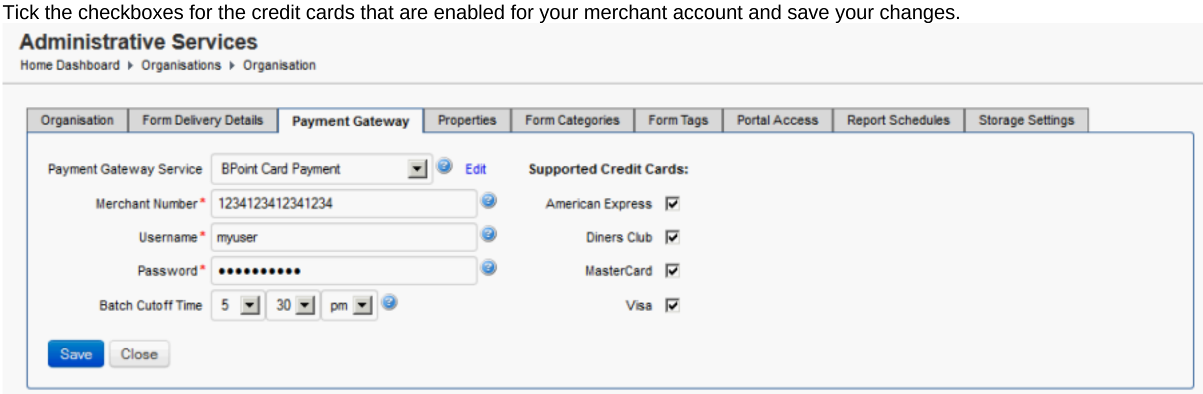
The BPoint card payment service has two global configuration parameters: The URL to the payment gateway server and an expiry date value for use during testing.
To configure these parameters, go to "System > Service Definitions", find the BPoint card payment service and edit it. Switch to the "Service Parameters" tab.
To change the URL to the payment gateway server, edit the "paymentUrl" parameter and enter one of the payment URLs below (unless provided with different URLs by BPoint):

Test	https://evolve-uat.premier.com.au/evolve/service.asmx?WSDL
Live	https://www.bpoint.com.au/evolve/service.asmx?WSDL

The second parameter (named "testExpiryDate") MUST NOT be set when the payment service is in production mode. An empty string value is acceptable.
While the payment service is in test mode, BPoint uses the credit card expiry date to control the returned result. For a test payment to be successful, the expiry date must be "9900" or "9908" (using a format of MMYY). As this does not correspond to a valid date and TM does not allow users to enter such expiry dates, it would be impossible to make successful BPoint test payments. Therefore, the BPoint payment service provides a way to set a hard-coded expiry date that, if set, will be passed to BPoint in place of the expiry date entered by the user.
In short, set the test expiry date to a blank string while in production mode, or set it to "9908" or "9900" for successful payments in test mode, or another value to produce payment errors in test mode (the two digits for the year will be returned as the response code).

Organization Configuration

To configure an organization to use the BPoint card payment service, ensure the organization has obtained a BPoint merchant account first.
Go to "Forms > Organizations", edit the organization and switch to the "Payment Gateway" tab (see below).
Choose "BPoint Card Payment" from the "Payment Gateway Service" dropdown.
Enter the organization's merchant ID into the "Merchant Number" field.
Enter the merchant username and password into the "Username" and "Password" fields.
Enter the batch cutoff time provided by BPoint into the "Batch Cutoff Time" field.
Tick the checkboxes for the credit cards that are enabled for your merchant account and save your changes.



An example configuration for the BPoint card payment service

Hosted Payment Gateways

This section lists the hosted (three-party) payment gateways supported by Transaction Manager, and how to configure them.

PayPal Hosted Payment Gateway

This payment service uses the PayPal Express Checkout service to process payments. Users are offered to pay using their PayPal account, or using a credit card.

Global Configuration

The PayPal Hosted Payment service has three relevant parameters: The URL of the payment website, the name of the PayPal environment that will be used, and the currency.
To configure these parameters, go to "System > Service Definitions", find the PayPal hosted payment service and edit it. Switch to the "Service Parameters" tab.
To change the URL to the payment website, edit the "paymentUrl" parameter and enter one of the payment URLs below (unless provided with different URLs by PayPal):

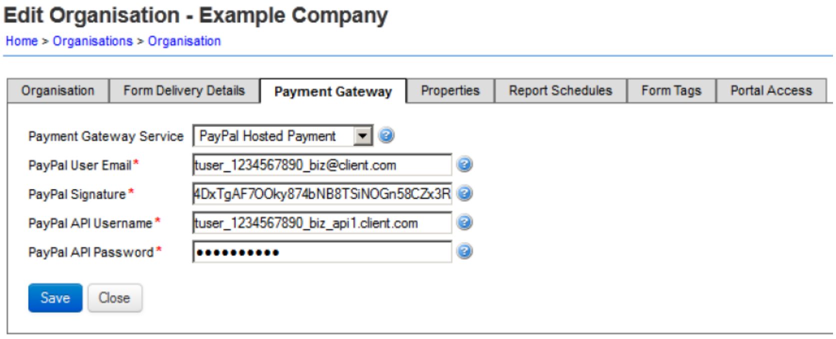
Test	https://www.sandbox.paypal.com/cgi-bin/webscr?cmd=express-checkout&token=
Live	https://www.paypal.com/cgi-bin/webscr?cmd=express-checkout&token=

Also ensure the "environment" parameter is set correctly: It should be "sandbox" while using the sandbox test environment, and "live" while in production mode.
To change the currency, edit the "currency" parameter and enter the desired currency code.

Organization Configuration

To configure an organization to use the PayPal hosted payment service, ensure the organization has obtained a merchant account with PayPal first.
Go to "Forms > Organizations", edit the organization and switch to the "Payment Gateway" tab (see below).
Choose "PayPal Hosted Payment" from the "Payment Gateway Service" dropdown.
Enter the email address associated with the organization's merchant account into the "PayPal User Email" field.
Enter the PayPal signature (an alphanumeric string) into the "PayPal Signature" field.

Enter the organization's PayPal API username into the "PayPal API Username" field.
Enter the organization's PayPal API password into the "PayPal API Password" field.



An example configuration for the PayPal Hosted Payment service

Westpac QuickWeb Hosted Payment

This payment service connects to the QuickWeb payment gateway (https://www.quickweb.com.au). To use it, the organization needs to set up an account with QuickWeb and enter its merchant details into Transaction Manager.
The QuickWeb payment gateway is a hosted payment gateway where the user's credit card information is actually captured by the Westpac QuickWeb gateway. The Transaction Manager Space redirects the user to the QuickWeb website where the payment details are captured and processed. The results of the transaction are passed back to Transaction Manager through an asynchronous server to server HTTP post handled by the Space. After payment, the user is redirected back to the confirmation page.
Submission payment is marked as completed when the asynchronous call back is received. If the callback is not received the submission will be marked as "Payment Verification Pending" and will not be delivered until the server to server callback is received.

Prerequisites

To enable this service the following actions need to be performed:

-
    - QuickWeb needs to build a payment portal for the client and set the gateway up to support a token request.
    - QuickWeb/merchant need to provided Community Code, Business Code, Username and Password.
    - Make sure the TM SSL Certificate is not Self-Signed as this is not supported by QuickWeb.
    - The TM server's IP address needs to be white-listed by QuickWeb.
    - Make sure the self-service space is available externally otherwise QuickWeb will not be able to send the payment result callback.
    - The QuickWeb portal needs to be setup to allow a callback to a URL of the following format: https://<domainname>/<portalcontext>/servlet/hostedpayment/callback.

Global Configuration

The Westpac hosted payment service has global configuration parameters as described in the following table:

Parameter	Description
currency	The currency of the transaction
paymentURL	The URL that the user will be redirected to. The following is an example for the test system:- https://quickweb.support.qvalent.com/OnlinePaymentServlet
paymentRequestUrlParametersTemplate	Velocity template to control the parameters passed to QuickWeb when the user is redirected to the payment gateway. Please see section "QuickWeb Outgoing URL Configuration" below.
tokenURL	The URL that is used to obtain a token for the transaction. The following is an example for the test system:- https://quickweb.support.qvalent.com/CommunityTokenRequestServlet
requestTokenUrlParametersTemplate	Velocity template to control the parameters passed to QuickWeb when the system requests a token from the payment gateway. Please see section "QuickWeb Outgoing URL Configuration" below.
currency	The payment currency (default AUD)

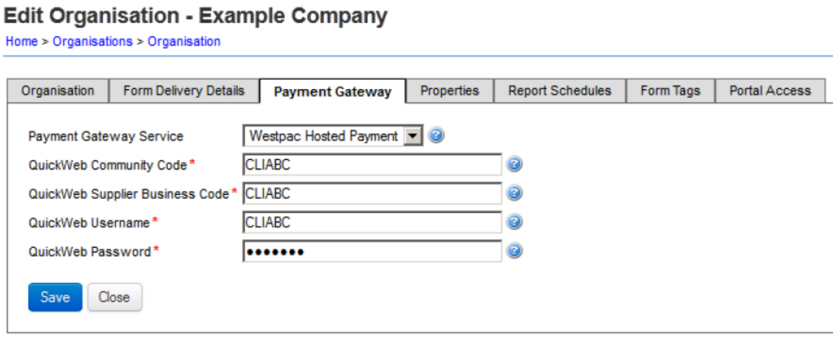
customParameterMap	JSON Map of XPath references of values to pull out of the form xml for use with the customParameterTemplate string. The following is an example mapping.
	{
	"cpParam1": "//AvokaSmartForm/SFMData/SystemProfile/TrackingCode"
	}

callbackParametersMapping	Provides a mapping to the values in the QuickWeb callback used to process the result of the transactions (Default QuickWeb3)
passbackParametersMapping	Provides a mapping to the values in the QuickWeb user return browser redirection used to process the result of the transactions (Default QuickWeb3)
keepReconciliationFileForDays	The number of days that the reconciliation files will be kept in the database.
paymentTransactionLifespanMinutes	The session time for a payment transaction in the provides payment gateway
reconciliationConfiguration	JSON structure used to identify the columns within the reconciliation file. This should have an empty value unless reconciliation is being utilised.

To configure these parameters, go to "System > Service Definitions", find the Westpac hosted payment service and edit it. Switch to the "Service Parameters" tab.

Organization Configuration

To configure an organization to use the Westpac hosted payment service, ensure the organization has obtained a merchant account with Westpac first. Westpac will also provide a number of parameters that need to be entered in the organization configuration section.
Go to "Forms > Organizations", edit the organization and switch to the "Payment Gateway" tab (see below).
Choose "Westpac Hosted Payment" from the "Payment Gateway Service" dropdown.
Enter the organization's QuickWeb community code into the "QuickWeb Community Code" field.
Enter the QuickWeb supplied business code into the "QuickWeb Supplier Business Code" field.
Enter the QuickWeb username into the "QuickWeb Username" field.
Enter the QuickWeb password into the "QuickWeb Password" field.



An example configuration for the Westpac Hosted Payment service

SSL Certificates

HTTPS is required on all communications with Westpac this requires Westpac's QuickWeb SSL certificates to be trusted by the application server and also for Westpac to trust the Transaction Manager certificates.

QuickWeb Outgoing URL Configuration

There are two URL's used to initiate a payment transaction with QuickWeb (tokenURL and paymentURL) and they differ between older versions of QuickWeb and QuickWeb3. To support this the URL strings that are used are configurable using Velocity Templates in the payment service definition settings paymentRequestUrlParametersTemplate and requestTokenUrlParametersTemplate. The following table contains a description of the values available for use in these templates. The default configuration is for QuickWeb 3.

Name	Description
paymentLog	The Transact Manager PaymentLog entity. This can provide access to a variety of information from the form transaction. Please see the API documentation for more information about this object.
amPayment	The payment amount as defined by form transaction.
returnURL	The system return URL used when the user is returned from the QuickWeb payment gateway.
serviceReturnURL	The servlet that QuickWeb will use to post the results of the payment transaction.
cancelURL	The system cancel URL used when the user is returned from the QuickWeb payment gateway after they have cancelled the payment operation.
errorURL	The system error URL used when the user is returned from the QuickWeb payment gateway after there has been a system error.
userName	The QuickWeb User Name from the Organisation Payment Configuration.
password	The QuickWeb Password from the Organisation Payment Configuration
communityCode	The QuickWeb Community Code from the Organisation Payment Configuration
supplierBusinessCode	The QuickWeb Supplier Business Code from the Organisation Payment Configuration
currencyCode*	The currency code from the payment gateway service definition.
serviceReturnEmail*	The QuickWeb service return email address from the payment gateway service definition.

These parameters can be overridden using on the form, client or space properties.
In addition to the above any parameters defined in the customParameterMap payment gateway service definition parameter map are also available to these Velocity templates.

WorldPay Hosted Payment

This payment service uses the WorldPay payment gateway (http://www.worldpay.com/) to process payments. Users are redirected to the WorldPay website and can pay using their credit card.

Global Configuration

The WorldPay Hosted Payment service has two relevant parameters: The URL of the payment website and the currency.
To configure these parameters, go to "System > Service Definitions", find the WorldPay hosted payment service and edit it. Switch to the "Service Parameters" tab.
To change the URL to the payment website, edit the "paymentUrl" parameter and enter one of the payment URLs below (unless provided with different URLs by PayPal):

Test	https://secure-test.worldpay.com/jsp/merchant/xml/paymentService.jsp
Live	https://secure.worldpay.com/jsp/merchant/xml/paymentService.jsp

To change the currency, edit the "currency" parameter and enter the desired currency code.
Do not modify the "configUiMapping" service parameter.

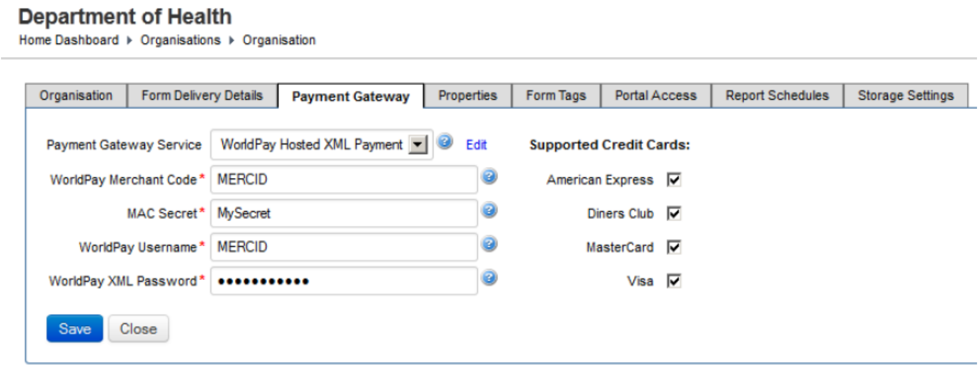
Prerequisites

To enable this service the following actions need to be performed:

- The TM server's IP address needs to be white-listed by the service provider
- The service provider/merchant needs to provide a Merchant Code, MAC Secret Key, Username and Xml Password

Organization Configuration

To configure an organization to use the WorldPay hosted payment service, ensure the organization has obtained a merchant account with WorldPay first.
Go to "Forms > Organizations", edit the organization and switch to the "Payment Gateway" tab (see below).
Choose "WorldPay Hosted XML Payment" from the "Payment Gateway Service" dropdown.
Enter the organization's WorldPay merchant code into the "WorldPay Merchant Code".
Enter the MAC secret configured via the WorldPay merchant interface into the "MAC Secret" field.
Enter the organization's WorldPay username and XML password into the "Merchant Username" and "Merchant XML Password" fields.
Tick the credit cards you would like to support (you must enable at least one credit card).
Save your changes.



An example configuration for the WorldPay Hosted Payment service

BPoint Hosted Payment

This payment service connects to the BPoint payment gateway (https://www.bpoint.com.au). To use it, the organization needs to set up an account with BPoint and enter its merchant details into Transaction Manager.
The BPoint payment gateway is a hosted payment gateway where the user's credit card information is actually captured by the Commonwealth Bank BPoint gateway. The Transaction Manager space redirects the user to the BPoint website where the payment details are captured and processed. After payment, the user is redirected back to the Transaction Manager space. Transaction Manager verifies payment via a HTTP POST to BPoint, after which the user is redirected to either (dependent upon payment status) the confirmation page, or back to the initial hosted payment page.
When the payment process is first initiated, the payment is marked as "Pending". The submission payment is marked as completed upon a successful result from the verification POST. If the verification call is not completed for some reason, a scheduled job will attempt to reconcile the payment via BPoint's SOAP API. If the SearchTransactions SOAP call indicates the payment has been completed, the submission payment status is accordingly updated.

Prerequisites

To enable this service the following actions need to be performed:

- BPoint need to create/nominate a Merchant account to be used
- BPoint need to provide a Merchant Number, API Username and password
- BPoint need to build a hosted payment page and link this to a biller code
- BPoint need to provide the URL to the hosted payment page
- BPoint need to provide the biller code

Global Configuration

The BPoint hosted payment service has three global configuration parameters as described in the following table:

authURL	The URL for the authentication HTTP Post, which returns a token sent to BPoint as part of the initial redirect away from TM: Examples: Prod: https://www.bpoint.com.au/payconnect/auth.aspx

	Test: https://bpoint-uat.premier.com.au/payconnect/auth.aspx
verifyURL	The URL for the verification HTTP Post, which accepts a token and several other parameters received in the post-payment redirect back to TM: Examples: Prod: https://www.bpoint.com.au/payconnect/verify.aspx Test: https://bpoint-uat.premier.com.au/payconnect/verify.aspx
soapURL	The URL for the SOAP endpoint, used to reconcile payments that could not be verified by the verification POST. Examples: Prod: https://www.bpoint.com.au/evolve/service.asmx?WSDL Test: https://evolve-uat.premier.com.au/evolve/service.asmx?WSDL

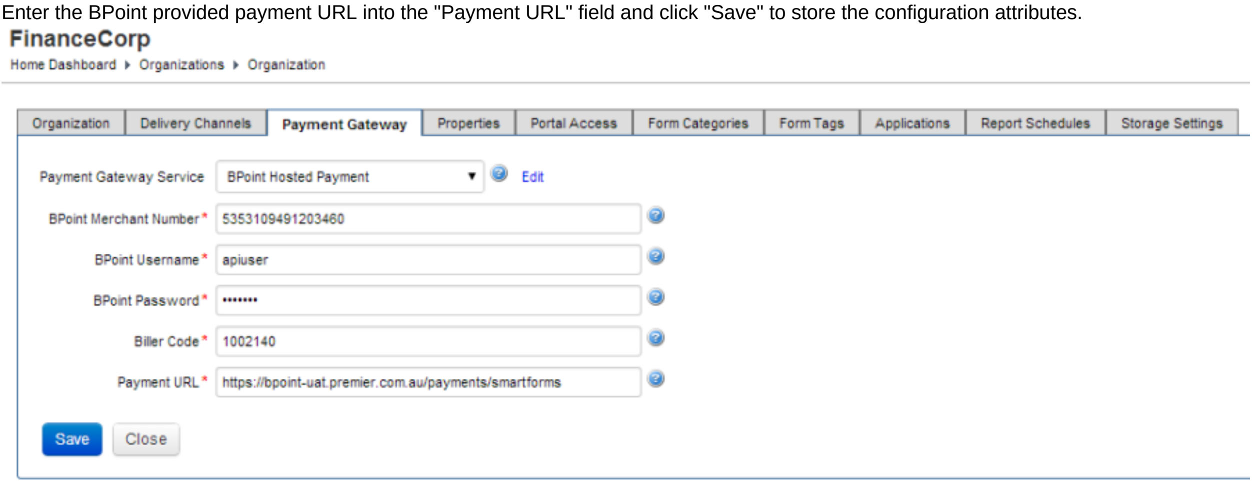
customParameterMap	JSON Map of XPath references of values to pull out of the form xml and populates CRN2 and CRN3 in the BPoint request.
	{
	"in_crn2":"/AvokaSmartForm/Common/UserProfile/Individual/GivenName",
	"in_crn3":"/AvokaSmartForm/Common/UserProfile/Individual/FamilyName"
	}

paymentTransactionLifespan Minutes	The session time for a payment transaction in the provides payment gateway

To configure these parameters, go to "System > Service Definitions", find the BPoint hosted payment service and edit it. Switch to the "Service Parameters" tab.

Organization Configuration

To configure an organization to use the BPoint hosted payment service, ensure the organization has obtained a merchant account with BPoint first. BPoint will also provide a number of parameters that need to be entered in the organization configuration section.
Go to "Forms > Organizations", edit the organization and switch to the "Payment Gateway" tab (see below).
Choose "BPoint Hosted Payment" from the "Payment Gateway Service" dropdown.
Enter the organization's BPoint merchant number into the "BPoint Merchant Number" field.
Enter the BPoint supplied username into the "BPoint Username" field.
Enter the BPoint supplied password into the "BPoint Password" field.
Enter the BPoint supplied biller code into the "Biller Code" field.
Enter the BPoint provided payment URL into the "Payment URL" field and click "Save" to store the configuration attributes.



An example credentials configuration for the BPoint Hosted Payment service

Form Configuration

The BPoint payment gateway allows the form to override the biller code that is used with the payment transaction. To override the biller code for a form, the form developer is required to set up a data extract mapping in TM called "billercode" and populate this with the required biller code. If a form does not provide a "billercode" or the data extract content is blank, the default organization biller code will be used.

Bizgate Hosted Payment

This payment service uses the Bizgate payment gateway to process payments. Users are redirected to the Bizgate website and can pay using their credit card.
The Bizgate payment gateway is a hosted payment gateway where the user's credit card information is captured by the Bizgate Remote Payment system and forwarded to the relevant bank. The Transaction Manager space redirects the user to the Bizgate website where the payment details are captured and processed.
The results of the payment transaction is passed back via a synchronous web page redirect to the Transaction Manager space's hosted-payment servlet and processed accordingly. If the transaction was successful the space redirects the user to the confirmation page. Otherwise the user is redirected back to the space payment page with a description of the last unsuccessful payment transaction.

Data transmission security and SSL certificates

Bizgate payment gateway service recommends applying a digital signature to the Request XML which is sent via a secure HTTP POST. A digital signature is a standard security measure for confirming the validity and integrity of data sent over the Internet.
To create a payment the remote payment service consumes requests with a digitally signed XML parameter [xml]. Any response from this service back to Transaction Manager will be also digitally signed using the same method. SSL certificates and keys which are used to encrypt/decrypt the XML requests and responses are read from a "JKS" key-store stored to a directory accessible to Transaction Manager.

Prerequisites

To enable this service the following actions need to be performed:

- Bizgate needs to create/nominate a Merchant account to be used
- Bizgate needs to provide a Merchant ID and password
- A valid "JKS" key-store with certificates for XML signature needs to be created and it contains:
    - a valid private key to sign XML payment requests
    - a valid public key to verify XML payment responses provided by the service provider (Bizgate)
- The service provider needs to receive a public key to verify signed XML payment requests from TM.

Global Configuration

The Bizgate Hosted Payment service has a couple of relevant parameters. To configure these parameters, go to "System > Service Definitions", find the Bizgate hosted payment service and edit it. Switch to the "Service Parameters" tab.
Do not modify the "configUiMapping" service parameter.

keystoreFilePath	The path to the "JKS" key-store stored to a directory accessible to Transaction Manager.
keystorePassword	Password to the provided "JKS" key-store. The same password will be used to read the private key stored under the "requestXmlSignKeyAlias" alias.
paymentRequestTemplate	Template used to generatethe SOAP request. Do not modify unless the payment gateway service provider requests.
paymentTransactionLifespanMinutes	The transaction Lifespan in minutes for which a user session in the payment gateway is valid.
paymentURL	The Bizgate URL to request a redirect for payment
requestXmlSignKeyAlias	Alias to the private key used to sign the XML request. The same "keystorePassword" password will be used to access the key from the key-store. Note that only one key pair is used across all organizations.
responseXmlSignKeyAlias	Alias to the Bizgate public key to decrypt the XML response. The public key should be imported into the key-store as "trustcacert" and does not require a password.
transactionStatusURL	The Bizgate URL Transaction Manager uses to request a transaction status for a payment

Organization Configuration

To configure an organization to use the Bizgate hosted payment service, ensure the organization has obtained a merchant account with Bizgate first. The service provider will also provide a number of parameters that need to be entered in the organization configuration section.
Go to "Forms > Organizations", edit the organization and switch to the "Payment Gateway". Choose "Bizgate Hosted Payment" from the "Payment Gateway Service" dropdown.
Enter the merchant username into the "Merchant ID" field.
Enter the merchant password into the "Merchant Password" field and save the changes.

SecurePay DirectPost Payment

This payment service uses the DirectPost Payment gateway (https://api.securepay.com.au/live/directpost/authorise) to process payments.
The SecurePay payment gateway is a hosted payment gateway where the user's credit card information is processed by the DirectPost Payment gateway. In this type of gateway the page that captures the card details is a customizable HTML form configured in the payment gateways global configuration property "htmlForm". The page submits the details directly to the payment gateway service configured in the "paymentURL" property. Transaction manager does not store or log any of these card details.
The result of the payment transaction is passed back via a synchronous web page redirection to the Transaction Manager space. This gateway also utilizes a server to server callback for added reliability. The results of the transaction is passed to the hosted-payment servlet and processed accordingly. In the case of a successful transaction the space redirects to the confirmation page. Otherwise, the user is redirected back to the space payment page with a description of the last unsuccessful payment transaction.

Prerequisites

To enable this service the following actions need to be performed:

- SecurePay need to provide Merchant ID and Password
- Make sure the self-service space is available externally otherwise SecurePay will not be able to send the payment result callback.

Global Configuration

The SecurePay Hosted Payment service has a couple of relevant parameters. To configure these parameters, go to "System > Service Definitions", find the SecurePay DirecPost payment service and edit it. Switch to the "Service Parameters" tab.
Do not modify the "configUiMapping" service parameter.

htmlForm	The html form content.

keepReconciliationFilesForDays	Number of days to keep reconciliation files.
paymentTransactionLifespanMinutes	The transaction Lifespan in minutes for which a user session in the payment gateway is valid.
paymentURL	The URL to use when processing a payment
reconciliationConfiguration	Allows the overriding of the names of the columns and timestamp formats in the reconciliation file. (JSON). This should have an empty value unless reconciliation is being utilised.

Organization Configuration

To configure an organization to use the SecurePay DirectPost payment service, ensure the organization has obtained a merchant account with SecurePay first. The service provider will also provide a number of parameters that need to be entered in the organization configuration section.
Go to "Forms > Organizations", edit the organization and switch to the "Payment Gateway". Choose "SecurePay DirectPost Payment" from the "Payment Gateway Service" dropdown.
Enter the merchant username into the "Merchant ID" field.
Enter the merchant password into the "Merchant Password" field.
For later reconciliation of payments received via Transaction Manager, enter the batch cutoff time (specific to the bank used by the organization) into the "Payment Batch Cutoff Time" field.
Tick the checkboxes for the credit cards that are enabled for your merchant account and save your changes.

NAB DirectPost Payment

This payment service uses the NAB DirectPost Payment gateway to process payments.
The NAB DirectPost payment gateway is a hosted payment gateway where the user's credit card information is processed by the DirectPost Payment gateway. In this type of gateway the page that captures the card details is a customizable HTML form configured in the payment gateways global configuration property "htmlForm". The page submits the details directly to the payment gateway service configured in the "paymentURL" property. Transaction manager does not store or log any of these card details.
The result of the payment transaction is passed back via a synchronous web page redirection to the Transaction Manager space. This gateway also utilizes a server to server callback for added reliability. The results of the transaction is passed to the hosted-payment servlet and processed accordingly. In the case of a successful transaction the space redirects to the confirmation page. Otherwise, the user is redirected back to the space payment page with a description of the last unsuccessful payment transaction.

Prerequisites

To enable this service the following actions need to be performed:

- NAB need to provide Merchant ID and Password
- Make sure the self-service space is available externally otherwise NAB will not be able to send the payment result callback.

Global Configuration

The NAB DirectPost Payment service has a couple of relevant parameters. To configure these parameters, go to "System > Service Definitions", find the NAB DirecPost Payment service and edit it. Switch to the "Service Parameters" tab.
Do not modify the "configUiMapping" service parameter.

htmlForm	The html form content.
keepReconciliationFilesForDays	Number of days to keep reconciliation files.
paymentTransactionLifespanMinutes	The transaction Lifespan in minutes for which a user session in the payment gateway is valid.
paymentURL	The URL to use when processing a payment
reconciliationConfiguration	Allows the overriding of the names of the columns and timestamp formats in the reconciliation file. (JSON). This should have an empty value unless reconciliation is being utilised.

Organization Configuration

To configure an organization to use the NAB DirectPost Payment service, ensure the organization has obtained a merchant account with NAB first. The service provider will also provide a number of parameters that need to be entered in the organization configuration section.
Go to "Forms > Organizations", edit the organization and switch to the "Payment Gateway". Choose "NAB DirectPost Payment" from the "Payment Gateway Service" dropdown.
Enter the merchant username into the "Merchant ID" field.
Enter the merchant password into the "Merchant Password" field.
For later reconciliation of payments received via Transaction Manager, enter the batch cutoff time (specific to the bank used by the organization) into the "Payment Batch Cutoff Time" field.
Tick the checkboxes for the credit cards that are enabled for your merchant account and save your changes.

TNS Hosted Payment

This payment service uses the TNS Hosted Payment gateway to process payments. Users are redirected to the TNS website and can pay using their credit card.

Prerequisites

To enable this service the following actions need to be performed:

- TNS need to provide a Merchant Profile ID, Access Code, Username and Password
- Make sure the self-service space is available externally otherwise TNS will not be able to send the payment result callback.

Global Configuration

The TNS Hosted Payment service has a few relevant parameters. To configure these parameters, go to "System > Service Definitions", find the TNS hosted payment service and edit it. Switch to the "Service Parameters" tab.
To change the URL to the payment website, edit the "paymentUrl" parameter and enter one of the payment URLs below (unless provided with different URLs by TNS):

apiURL	https://secure.ap.tnspayments.com/api/nvp/version/17
paymentURL	https://secure.ap.tnspayments.com/page/pay

To change the transaction Lifespan for which a user session in the payment gateway is valid, edit the "paymentTransactionLifespanMinutes" parameter and enter the desired period in minutes.
To change the currency, edit the "currency" parameter and enter the desired currency code.
Do not modify the "configUiMapping" service parameter.

Organization Configuration

To configure an organization to use the TNS hosted payment service, ensure the organization has obtained a merchant account and all related details with TNS first. The service provider will also provide a number of parameters that need to be entered in the organization configuration section.
Go to "Forms > Organizations", edit the organization and switch to the "Payment Gateway". Choose "TNS Hosted Payment" from the "Payment Gateway Service" dropdown.
Enter the organization's TNS Merchant ID into the "TNS Merchant ID" field.
Enter the provided TNS Merchant Name into the "TNS Merchant Name" field.
Enter the provided TNS Password into the "TNS Password" field and save your changes.

SmartPay Hosted Payment

This payment service uses a service offered by Optus through the RBA bank to process payments. Users are redirected to the SmartPay website and can pay using their credit card.

Prerequisites

To enable this service the following actions need to be performed:

- SmartPay needs to build a portal and provide a URL to the portal
- Encryption of the payment requests has been agreed with the service provider
- The encryption secret key and any required encryption URLs have been provided by the service provider
- SmartPay has obtained the following URLs for each space and organization
  - Approved: [space context path]/servlet/hostedpayment/response/[Organization code]
  - Cancelled: [space context path]/servlet/hostedpayment/cancel/[Organization code]
  - Problem: [space context path]/servlet/hostedpayment/error/[Organization code]
- SmartPay configured up the source URL that will be passed in the "Referrer Request Header" which will be [space context path]/hosted-payment.htm
- Make sure the self-service space is available externally otherwise SmartPay will not be able to send the payment result callback.

[Space context path] – The portal's context path configured in "Forms > Form Spaces", column "URL".
[Organization code] – The organization's code configured in "Forms > Organizations", client code column / field.

Global Configuration

The SmartPay Hosted Payment service has a set of relevant parameters. To configure these parameters, go to "System > Service Definitions", find the SmartPay Hosted payment service and edit it. Switch to the "Service Parameters" tab.
Do not modify the "configUiMapping" service parameter.

paymentURL	The URL to use when processing a payment.
reconciliationConfiguration	Allows the overriding of the names of the columns and timestamp formats in the reconciliation file. (JSON). This should be blank if reconciliation is not supported.

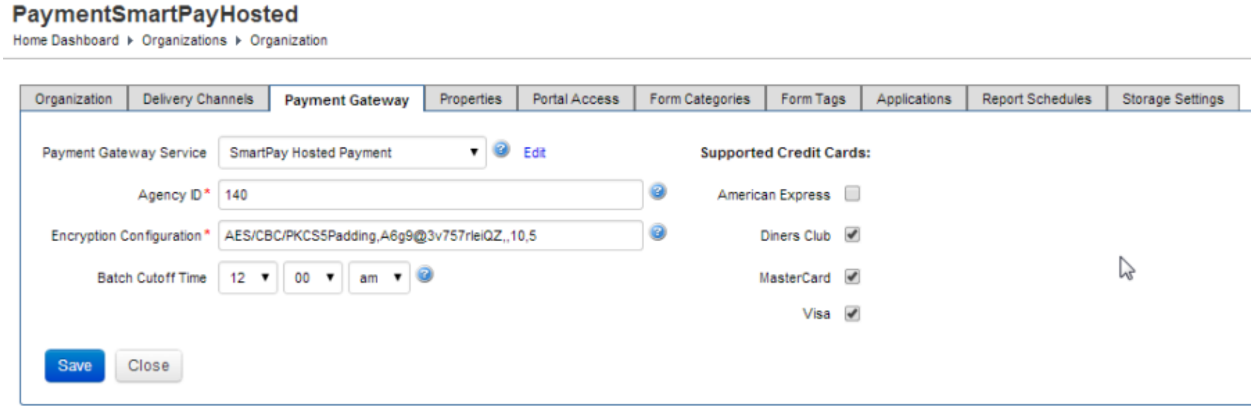
Organization Configuration

To configure an organization to use the SmartPay Hosted Payment service, ensure the organization has obtained a merchant account and all related details with SmartPay payment service provider first. The service provider will also provide a number of parameters that need to be entered in the organization configuration section.
Go to "Forms > Organizations", edit the organization and switch to the "Payment Gateway". Choose "SmartPay Hosted Payment" from the "Payment Gateway Service" dropdown.
Enter the organization's Agency ID into the "Agency ID" field.
Enter the Encryption Configuration into the "Encryption Configuration" field. The encryption configuration is a CSV formatted string containing the following values in the same order as described in the table:

Encryption Algorithm	needs to be one of the following:

	• AES/CBC/PKCS5Padding (128) • AES/ECB/PKCS5Padding (128) • DES/CBC/PKCS5Padding (56) • DES/ECB/PKCS5Padding (56) • DESede/CBC/PKCS5Padding (168) • DESede/ECB/PKCS5Padding (168)
Encryption key	Provided by the service provider when creating an account
Encryption password	Negotiated with the service provider when creating an account
Encryption padding interval	Negotiated with the service provider when creating an account
Encryption padding char count	Negotiated with the service provider when creating an account

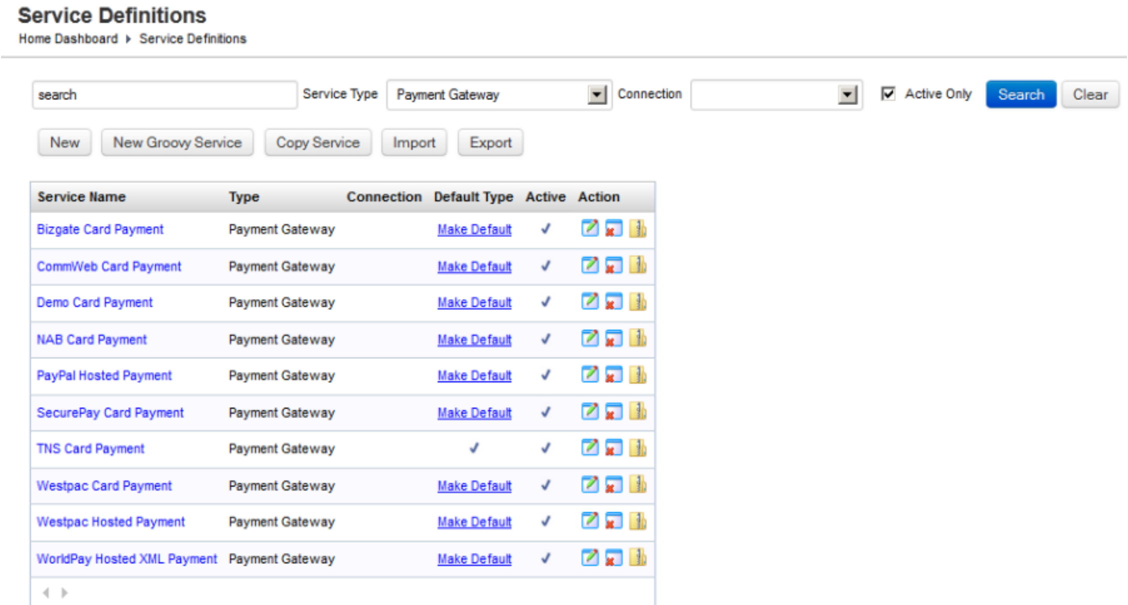
For later reconciliation of payments received via Transaction Manager, enter the batch cutoff time (specific to the bank used by the organization) into the "Payment Batch Cutoff Time" field.
Tick the checkboxes for the credit cards that are enabled for your merchant account and save your changes.



An example credentials configuration for the SmartPay Hosted Payment service

Browsing the List of Payment Services

To view the list of payment services and change the default service, go to "System > Service Definitions", choose "Payment Gateway" from the "Service Type" dropdown and click "Search" (the screenshot below shows an example).



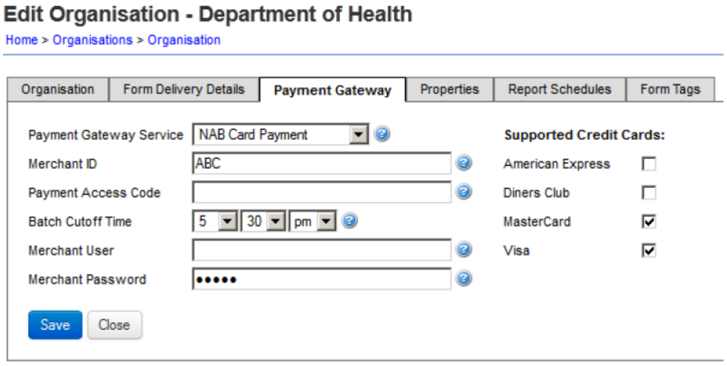
Searching for payment services

TM includes a set of service templates which can be used to create the payment services you need. See Services (Manager v5.0) #_sec_createservicefromtemplate for information on how to create a service based on a template.

You can change the default payment service by clicking the "Make Default" link in the "Default Type" column. Generally, it is best to configure the payment service explicitly for each organization. If you wish to use a "real" payment service as the default, be aware that each organization that wants to use payments needs to be set up with its account details (or configured to use a different service); otherwise payments will not be accepted by the gateway.

Configuring an Organization for Payment

For an organization to be able to use online payment, they first need to set up an account with one of the payment gateways supported by Transaction Manager. Once this is done, the payment configuration can be done.
To view an organization's payment details, go to "Forms > Organizations" and edit the organization. Switch to the "Payment Gateway" tab. The screenshot below shows an example configuration using the NAB payment gateway.



Configuring the payment service for an organization

Generally, the fields have the following meaning:
The "Payment Gateway Service" determines which payment service will be used to process credit card transactions for this organization.
The "Merchant ID" normally identifies the organization to the payment gateway. It is often assigned by the payment gateway provider and corresponds to a bank account that the payments will be credited to. For this reason, it is vitally important to ensure that the correct merchant ID is used.
The "Payment Access Code" is an additional identifier that might be assigned by the payment gateway provider. It is used in conjunction with the merchant ID.
The "Payment Batch Cutoff Time" field should be populated with the batch cutoff time (specific to the bank used by the organization). This is currently used by the payment reconciliation report provided along with Transaction Manager (see Reports (Manager v5.0)#_sec_paymentreconciliationreport), but is generally not required for the payment process itself.
The "Merchant User" can be used to further distinguish payments using the same merchant ID. A payment gateway provider might issue two user names for a single merchant ID, where one is to be used during testing (where transactions can be tested without money being actually transferred) and one to be used in production.
The "Merchant Password" is generally used when the payment gateway requires a password to be provided.
There are also four checkboxes in the "Supported Credit Card" area that can be ticked to indicate which credit cards will be accepted by the payment gateway. This is often specific to the merchant ID. On the Example Self Service Space, this value is used only to give an indication to the user as to what credit cards are accepted, but the credit card number a user enters is not validated against these settings.
The descriptions above are intended to give a general impression what the fields do. Please refer to the section describing the payment gateway that will be used for specific configuration instructions.

Configuring a Form for Payment

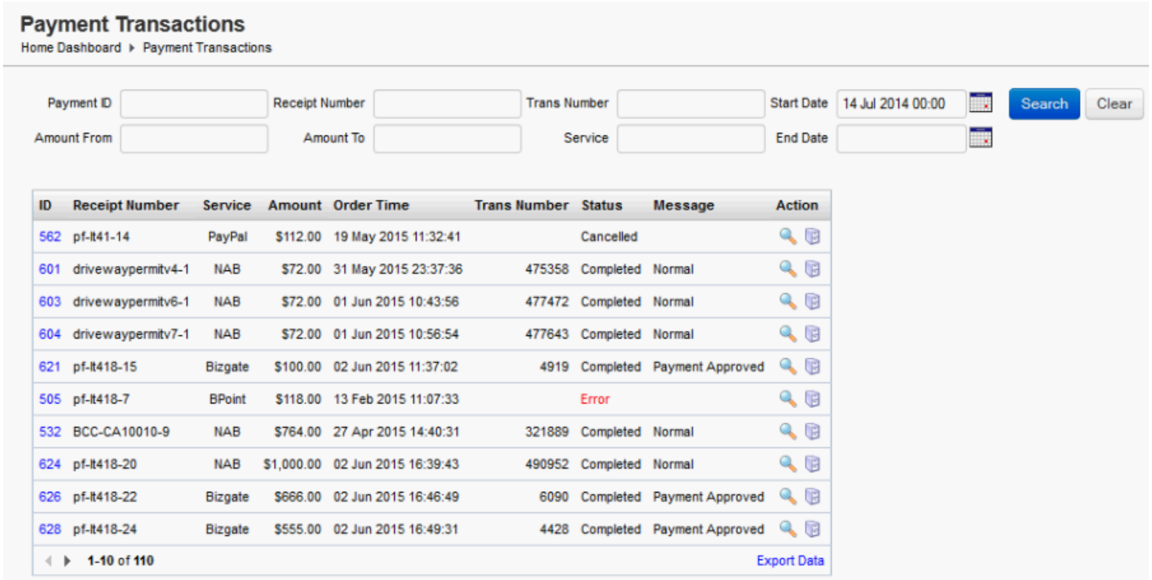
Transaction Manager reads the payment information from the form when it is being submitted. This means that the form needs to pass through the payment data in the correct format. Transact Composer contains widgets to help capture payment information.
Given a form that follows these guidelines, there is only one thing you need to do to enable payment for the form: You need to set up a configuration mapping for the form data configuration.
To do this, go to "Forms > Form Data Configurations", edit the form data configuration that your form is using and switch to the "Config Mappings" tab. You can either enable default mappings for your form by ticking "Use Default Mappings", or add a custom configuration mapping from "PaymentDetails" to the appropriate element in your schema seed file (the one containing the payment information subtree).
If you wish to have Transaction Manager update the form's receipt section, also add a mapping for "Receipt".
Once these mappings have been added, the form is set up for payments. Please also make sure the organization's payment details have been configured, as described in the previous section.
Note: If the form indicates that payment shall be made using a payment service, TM will use the payment service configured for the organization (even if the payment type in the form does not match the payment service, e.g. form specifies "Hosted" as the payment type, but the payment service is a 2 party credit card payment service).

The Payment Log

All credit card payments and payment attempts made via Transaction Manager are logged to the payment log. Note that Transaction Manager does not store the user's credit card details. However, the payment log can be helpful when monitoring and tracking down problems that occurred during payment.

Browsing the Payment Log

To view the payment log, go to "Operations > Payment Transactions". You will see the "Payment Transactions" page. You can filter this page by payment ID, the receipt number, the payment service, the transaction number assigned by the payment gateway and the date and time of payment.



ID	Receipt Number	Service	Amount	Order Time	Trans Number	Status	Message	Action
562	pf-lt41-14	PayPal	$112.00	19 May 2015 11:32:41		Cancelled		
601	drivewaypermitv4-1	NAB	$72.00	31 May 2015 23:37:36	475358	Completed	Normal	
603	drivewaypermitv6-1	NAB	$72.00	01 Jun 2015 10:43:56	477472	Completed	Normal	
604	drivewaypermitv7-1	NAB	$72.00	01 Jun 2015 10:56:54	477643	Completed	Normal	
621	pf-lt418-15	Bizgate	$100.00	02 Jun 2015 11:37:02	4919	Completed	Payment Approved	
505	pf-lt418-7	BPoint	$118.00	13 Feb 2015 11:07:33		Error		
532	BCC-CA10010-9	NAB	$764.00	27 Apr 2015 14:40:31	321889	Completed	Normal	
624	pf-lt418-20	NAB	$1,000.00	02 Jun 2015 16:39:43	490952	Completed	Normal	
626	pf-lt418-22	Bizgate	$666.00	02 Jun 2015 16:46:49	6090	Completed	Payment Approved	
628	pf-lt418-24	Bizgate	$555.00	02 Jun 2015 16:49:31	4428	Completed	Payment Approved	

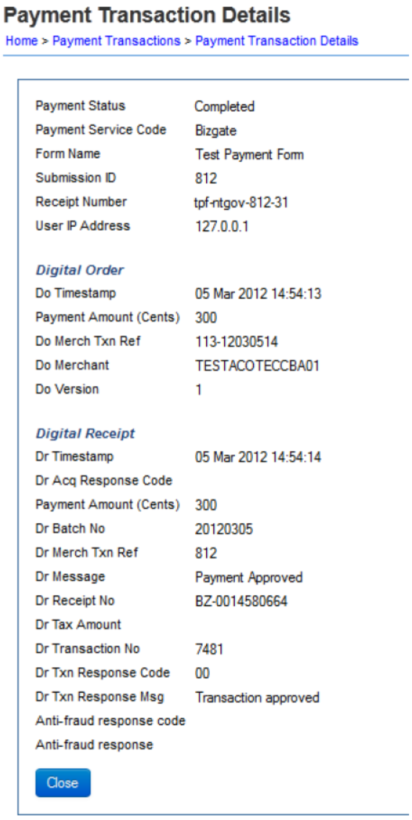
1-10 of 110 Export Data

Browsing the payment log

You can view a payment log entry ( ), the details for the corresponding submission ( ) and re-query the payment gateway about the status of the payment ( ), if the payment gateway supports this.

Viewing a Payment Log Entry

To view a payment log entry, click the "View Details" icon ( ) next to it. You will be directed to the "View Payment Transaction" page (an example for a payment via the Westpac payment gateway is shown below).
Please note that field semantics may vary between payment gateways, and not all fields may be populated.



Viewing a payment log entry

Important fields include:
Payment Status: Indicates whether the payment was successful or not
Payment Service Code: The payment service that was used to make this payment.
User IP Address: The IP address of the user making the payment.
Do Amount: The payable amount (usually in cents) passed in by Transaction Manager
Do Merchant: The Merchant ID that was passed in
Dr Timestamp: The date and time of the response from the payment gateway
Dr Batch No: The batch in which the payment was processed, as obtained by the payment gateway
Dr Message, Dr Txn Response Code, Dr Txn Response Msg, Dr Acq Response Code: Fields containing information about the response type (Success /Error) and diagnostic information
Dr Transaction No: Transaction identifier assigned by the payment gateway

Signatures (Manager v5.0)

Like paper forms, SmartForms often need to request the user's signature to confirm their submission. Transaction Manager can provide multiple ways to enable user signatures.
This chapter lists the various signature methods that Transaction Manager supports, and tells you how to check on the signature status for individual submissions.

Page Contents

Search

Concepts

In Transaction Manager, there are three supported signature modes: Wet signatures, where the user prints and signs the signatures conventionally; digital signatures, where the user signs the form with a digital certificate; and eSignatures, where the form receipt is signed by a certificate associated with the organization, and then presented to the user to confirm or cancel their submission.
Forms can control the signature method by defining an XML block for signatures; for more information, please refer to the Transact Integration Guide, chapter "Signatures". Alternatively, wet signatures and eSignatures can be turned on for a form using Transaction Manager.

Configuring Wet Signatures

To enable wet signatures for a form, edit it, select "Wet Signature" from the "Signature Type" dropdown and save.
When the Example Self Service Space encounters a form submission for which wet signatures are enabled, they will ask the user to first download the receipt, then either print, sign and upload the receipt straight away, or send in the signed receipt manually (e.g. via mail or fax).
In addition, the user is prevented from completing wet signatures before they have made credit card payment for the submission (if applicable). This ensures that the receipt has been finalized before it is printed and signed.

Configuring Digital Signatures

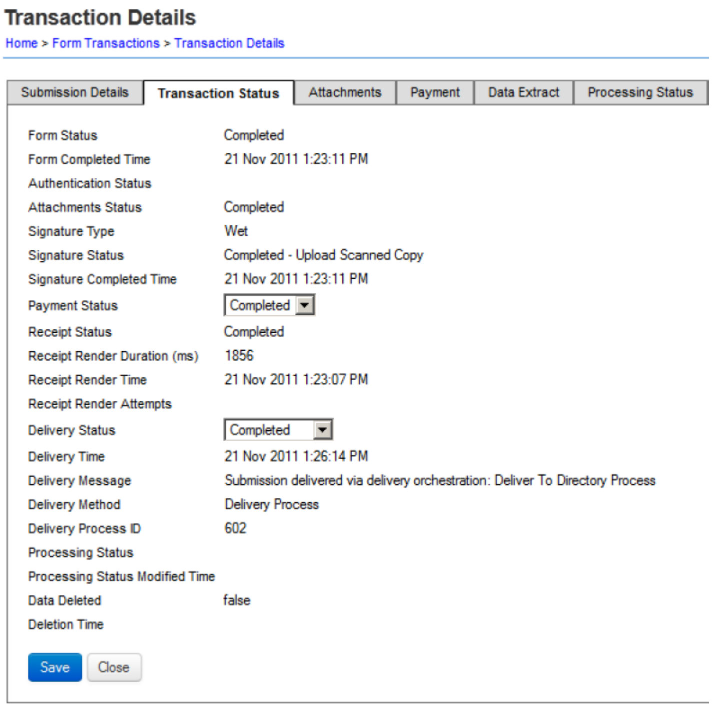
Digital signatures provided by a user need to be verified against a set of trusted certificates. At this point, Transaction Manager contains support for customer-specific digital signature verification services. If you would like to add support for digital signatures, please contact Avoka support to discuss development of a verification service that meets your needs.

Configuring eSignatures

Each organization that wants to use eSignatures needs to be set up with a certificate in Adobe LiveCycle. This certificate will be used to electronically sign receipts presented to the users.
After you have imported the certificate into Adobe LiveCycle, make a note of the certificate alias and edit the organization in Transaction Manager. Enter the certificate alias into the "Credential Alias" field and save.
To enable eSignatures for a form, edit it, select "eSignature" from the "Signature Type" dropdown and save. Note that this change will also disable anonymous fill-in for this form, as only registered users can perform eSignatures.
The Example Self Service Space allows new users to choose whether they wish to receive electronically signed receipts or not. Note that this setting cannot be changed after the user has been created.
When the Example Self Service Space encounters a form submission for which eSignatures are enabled, and if the user submitting the form has chosen to enable eSignatures, it will generate the electronically receipt and present it to the user. The user can choose to continue, or cancel the submission at this point.

Monitoring Signature Status for a Submission

For each submission, you can check whether signatures were requested and what the signature status is. To do so, go to "Operations > Form Transactions", view the submission details ( ) and switch to the "Transaction Status" tab.



The signature status of a submission

The "Signature Type" field indicates what type of signatures were requested. The "Signature Status" field indicates whether signatures have been completed, and the "Signature Completed Time" field indicates when signatures have been completed.

Tasks (Manager v5.0)

In addition to giving users the ability to fill in forms they are interested in, Transaction Manager also allows you to create tasks based on a specific form and assign a task to specific users and groups. The task can have prefill data associated with it, and completes when one of the assignees completes the task form.

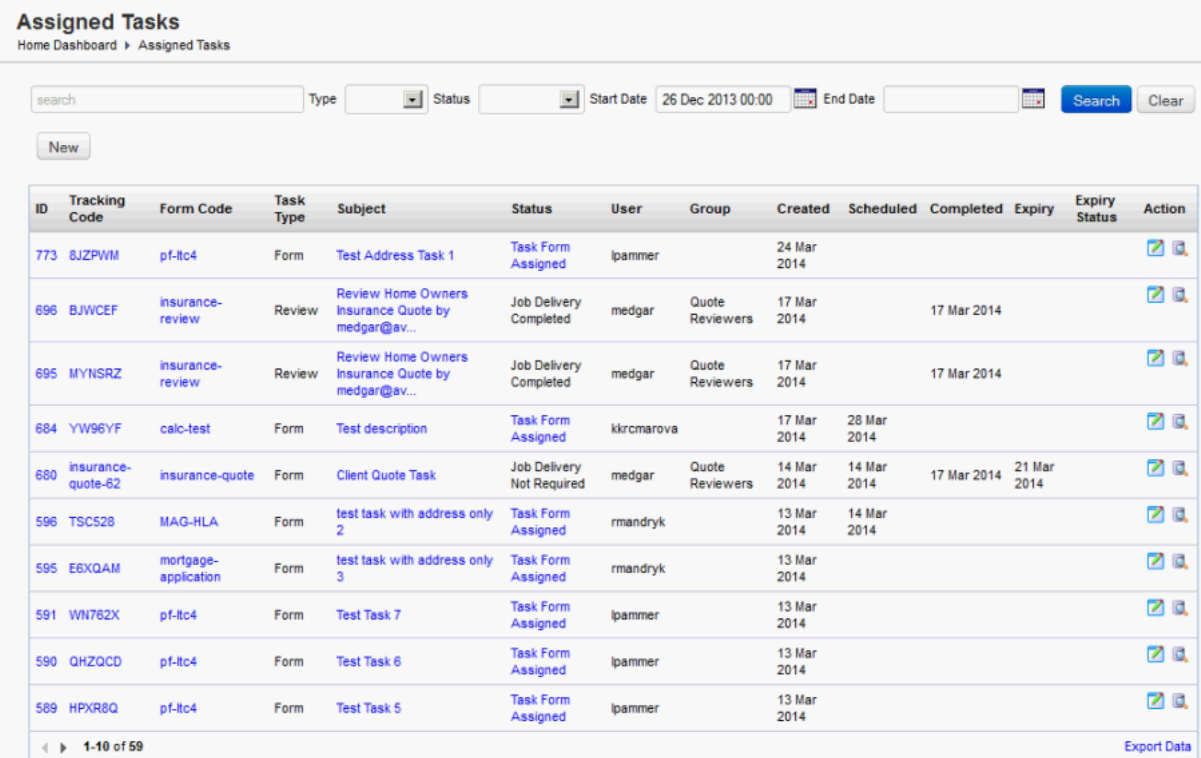
Search

Concepts

A *task* in Transaction Manager consists of a form that needs to be filled in by a user. The task may contain prefill data for customization (e.g. including reference numbers) and a *task subject* and *task message* that is used to identify the task to the user. When a task is created, TM creates a submission object that is marked as a task and contains all task data. As the user opens and completes the task, the submission is updated as normal.
There are currently two types of tasks: *Form tasks* are basic standalone tasks, whereas *review tasks* are associated a previous submission and are often prefilled with the submitted data for review purposes.
Tasks can be assigned to either a single user or a group of users. The *task status* indicates whether the task is assigned, completed or has expired (which is controlled via an optional *expiry date*). When a task expires, it can invoke a specified service to notify an external system about the task expiry. This service is configured as a service definition of type "Task Expiry Process".
Tasks can be created via the Transaction Manager management console or a web service interface. Please see the Transaction Manager Web Service API document for details about the web service interface.

Browsing the List of Tasks

To view the list of tasks that have been defined, go to "Operations > Assigned Tasks". You can filter the list by keyword (either task ID or a substring of the task subject or message), status and task creation date, and export the search results to Excel by clicking the "Export Data" link below the table.


Assigned Tasks
Home Dashboard ▸ Assigned Tasks

search | Type | Status | Start Date 26 Dec 2013 00:00 | End Date | Search | Clear

New

ID	Tracking Code	Form Code	Task Type	Subject	Status	User	Group	Created	Scheduled	Completed	Expiry	Expiry Status	Action
773	8JZPWM	pf-ltc4	Form	Test Address Task 1	Task Form Assigned	lpammer		24 Mar 2014					
696	BJWCEF	insurance-review	Review	Review Home Owners Insurance Quote by medgar@av...	Job Delivery Completed	medgar	Quote Reviewers	17 Mar 2014		17 Mar 2014			
695	MYNSRZ	insurance-review	Review	Review Home Owners Insurance Quote by medgar@av...	Job Delivery Completed	medgar	Quote Reviewers	17 Mar 2014		17 Mar 2014			
684	YW96YF	calc-test	Form	Test description	Task Form Assigned	kkrcmarova		17 Mar 2014	28 Mar 2014				
680	insurance-quote-62	insurance-quote	Form	Client Quote Task	Job Delivery Not Required	medgar	Quote Reviewers	14 Mar 2014	14 Mar 2014	17 Mar 2014	21 Mar 2014		
596	TSC528	MAG-HLA	Form	test task with address only 2	Task Form Assigned	rmandryk		13 Mar 2014	14 Mar 2014				
595	E6XQAM	mortgage-application	Form	test task with address only 3	Task Form Assigned	rmandryk		13 Mar 2014					
591	WN762X	pf-ltc4	Form	Test Task 7	Task Form Assigned	lpammer		13 Mar 2014					
590	QHZQCD	pf-ltc4	Form	Test Task 6	Task Form Assigned	lpammer		13 Mar 2014					
589	HPXR8Q	pf-ltc4	Form	Test Task 5	Task Form Assigned	lpammer		13 Mar 2014					

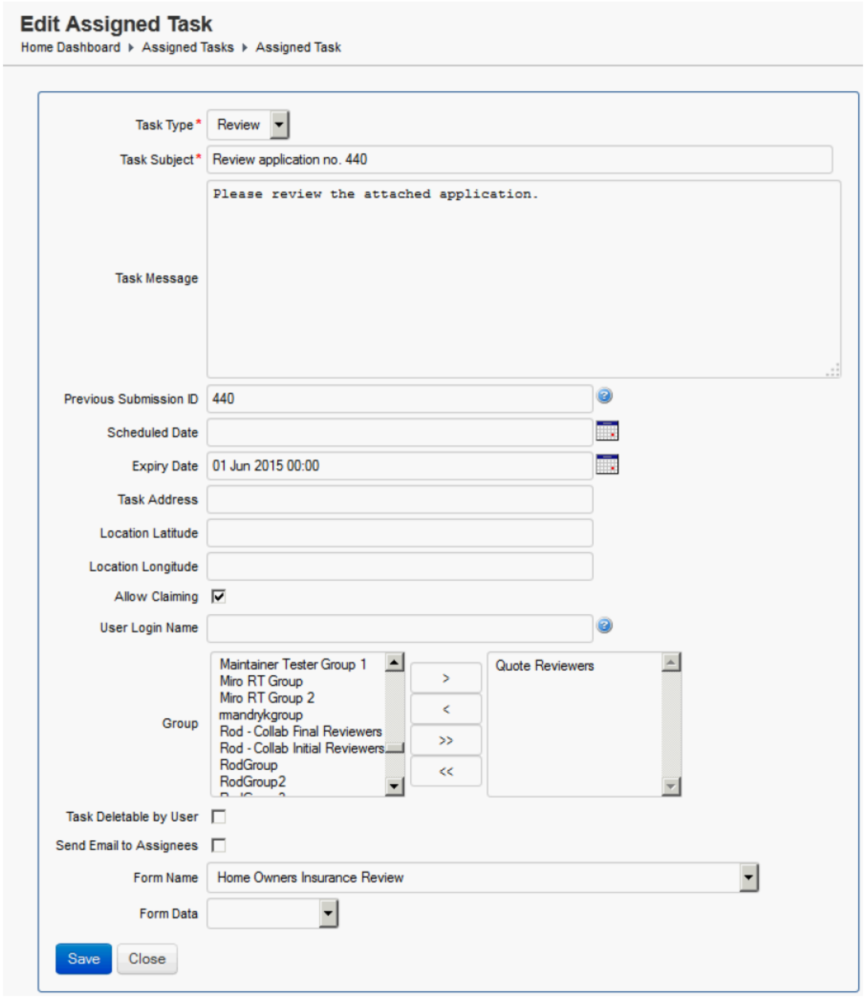
1-10 of 59 | Export Data

Viewing the list of tasks

This page allows you to create tasks as well as edit ( ) existing tasks. You can also view the full submission details via the "View Transaction Details" ( ) action. These operations will be described in the sections below.

Creating a Task

To create a new task via the Transaction Manager management console, go to "Operations > Assigned Tasks" and click the "New" button. You will be directed to the "Edit Assigned Task" page (an example is shown below). To find out how to add a task via the web service interface provided by Transaction Manager, please refer to the Transaction Manager Web Service API document.

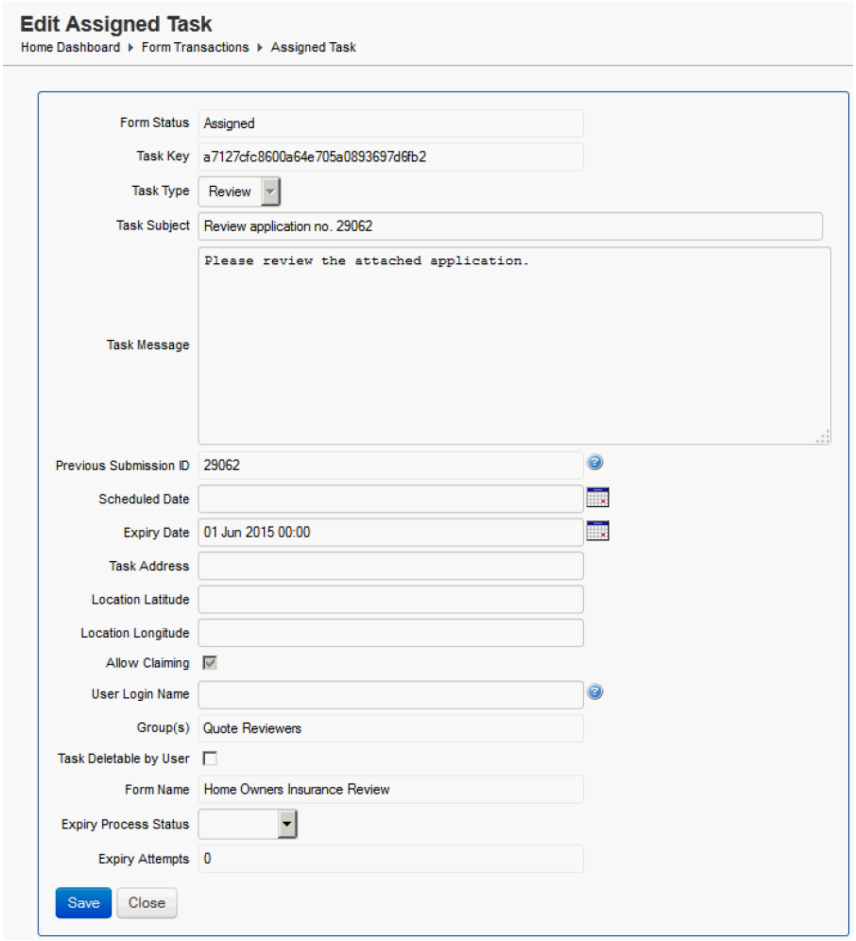


Creating a task

Choose the task type. Form tasks are basic tasks that can be created using either form XML or prefill XML data, whereas review tasks are based on a previous submission and do not accept prefill XML data (instead, they generally use the submission data).
Fill in a task subject and message that describes the task. This is shown to assignees on the user space, so it should contain any information that users will need to identify the task (e.g. reference numbers).
For review tasks, enter the submission ID that the task is based on.
If the assignee should be informed that the task should be done on a specific date, fill in the "Scheduled Date" field.
You can enter an expiry date if you want the task to expire at a specific date and time (for more on the mechanics of task expiry, see _sec_taskexpiry).
You can enter the physical address of a location or latitude and longitude information if relevant. This information will be displayed to the assignee (the default spaces include Google Map support). The latitude value needs to be in a range of -90 to 90 and the longitude value between -180 and 180.
You can assign the task to a specific user and/or a user group, in which case any member of the group will be able to view and complete the task. To assign the task to a specific user, enter the user's login name into the "User Login Name" field. To assign the task to a group, choose the group name from the "Group" dropdown. Note that users joining a group will be able to view and work on all tasks existing for the group.
Tick "Allow Claiming" to put the task into a special "claiming mode", where users have to claim the task before being able to work on it. In addition to being claimed, such tasks can be reassigned to other users by the claiming user or other space users with special privileges.
If you would like assignees to be able to delete the task, tick the "Task Deletable by User" checkbox. Note that assignees will not be able to delete completed tasks in order to preserve the audit trail.
You can choose to send email to the assignees to notify them of the new task, you can tick "Send Email to Assignees" checkbox. In this case, you can also fill in the "Email Subject" and "Email Message" in as appropriate (otherwise default values will be used). Note that this email will be sent out only once to the current assignees; if users later join the group the task is assigned to, they will not receive a notification email.
Choose the form that will need to be filled in to complete the task from the "Form Name" dropdown. The form can have payment and attachments associated with it as usual. For review tasks, you can leave the dropdown blank, in which case the form on which the original submission was based will be used.
If you would like to prefill the form that will be presented to the assignees, you have two options: If your task is of type "Form" and XML prefill is set up for your form, you can choose to use standard XML prefill (see chapter Form Version Configuration (Manager v5.0)#_sec_setupxmlprefillmappings) by choosing "Prefill XML" from the "Pre-population Data" dropdown and uploading the prefill XML file using the "Upload Prefill XML" field. Alternatively, you can also pass in a populated seed file (structurally matching the form seed XML data). To do so, choose "Form XML" from the "Pre-population Data" dropdown and upload the seed file using the "Upload Form XML" field.For tasks of type "Review", the XML data of the previous submission is automatically used if nothing else is specified.
Save the task. At this point a task submission will be created in form status "Assigned", and users will be able to fill in the form and complete the task using the user space associated with the form.

Editing a Task

To edit a task, go to "Operations > Assigned Tasks" and click the "Edit Task" icon ( ) next to the task. You will be directed to the "Edit Assigned Task" page.



Editing an existing task

This page allows you to edit most of the task data; however, some fields such as task type cannot be edited after task creation.

You can also manually change the "Expiry Process Status" field. Note that this should not be necessary during normal operations - the expiry process service is invoked by Transaction Manager, and the status is updated accordingly.

In addition, the task key generated by Transaction Manager is shown.

Task Expiry

If a task is not completed before the expiry date, a Transaction Manager background job (see Scheduled Jobs (Manager v5.0) #_sec_transactionprocessingjob) will change the task submission's form status to "Expired". At this point, the default user space will not allow assignees to complete the task. Expired task submissions cannot be progressed further.

Collaboration Jobs (Manager v5.0)

In addition to standalone transactions and tasks, Transaction Manager supports more complex workflows potentially involving multiple participants, steps and forms by providing collaboration jobs and the tools to manage them.

Search

Concepts

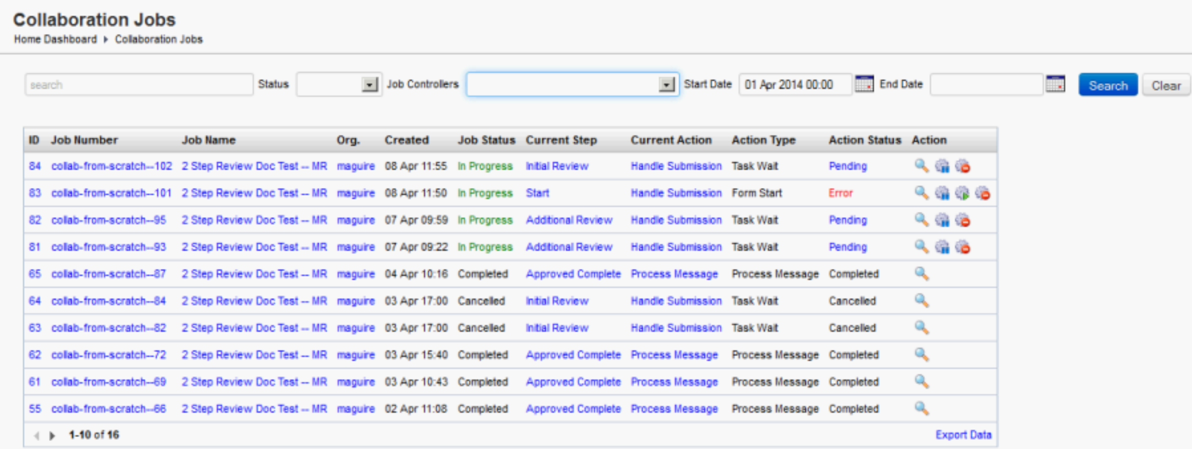
A *collaboration job* is a collection of submissions that are completed as part of a workflow. Collaboration jobs consist of a series of steps, which themselves can consist of a set of actions. Collaboration jobs are processed by a *job controller service*. When a submission is created and the form has been set up with a job controller service, the service is called and can create a collaboration job if appropriate.

Collaboration jobs may involve review steps and multiple users or groups of users. Additional submissions may be created as part of the job. Transaction Manager allows you to track and manage jobs and create custom services involved in job control.

For additional documentation on how to use jobs and create your own job controllers, see Avoka Transact Collaboration Jobs

Browsing the List of Collaboration Jobs

To view the list of collaboration jobs, go to "Operations > Collaboration Jobs".
You can filter the list of collaboration jobs by a substring of the job name, reference number, job key, status, job controller service and creation date.



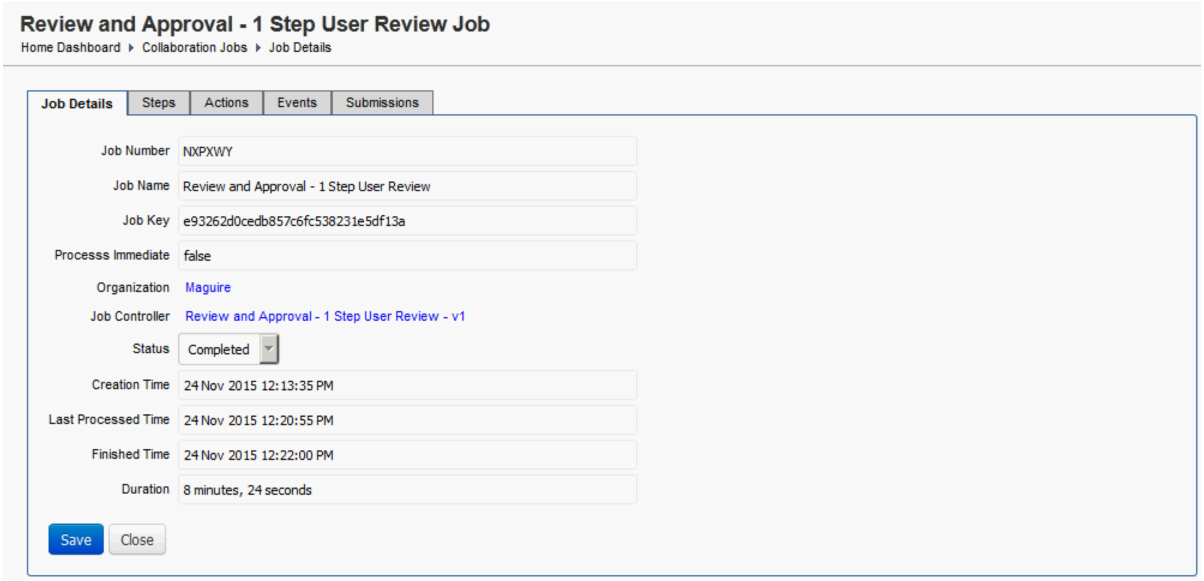
ID	Job Number	Job Name	Org.	Created	Job Status	Current Step	Current Action	Action Type	Action Status	Action
84	collab-from-scratch--102	2 Step Review Doc Test -- MR	maguire	08 Apr 11:55	In Progress	Initial Review	Handle Submission	Task Wait	Pending	
83	collab-from-scratch--101	2 Step Review Doc Test -- MR	maguire	08 Apr 11:50	In Progress	Start	Handle Submission	Form Start	Error	
82	collab-from-scratch--95	2 Step Review Doc Test -- MR	maguire	07 Apr 09:59	In Progress	Additional Review	Handle Submission	Task Wait	Pending	
81	collab-from-scratch--93	2 Step Review Doc Test -- MR	maguire	07 Apr 09:22	In Progress	Additional Review	Handle Submission	Task Wait	Pending	
65	collab-from-scratch--87	2 Step Review Doc Test -- MR	maguire	04 Apr 10:16	Completed	Approved Complete	Process Message	Process Message	Completed	
64	collab-from-scratch--84	2 Step Review Doc Test -- MR	maguire	03 Apr 17:00	Cancelled	Initial Review	Handle Submission	Task Wait	Cancelled	
63	collab-from-scratch--82	2 Step Review Doc Test -- MR	maguire	03 Apr 17:00	Cancelled	Initial Review	Handle Submission	Task Wait	Cancelled	
62	collab-from-scratch--72	2 Step Review Doc Test -- MR	maguire	03 Apr 15:40	Completed	Approved Complete	Process Message	Process Message	Completed	
61	collab-from-scratch--69	2 Step Review Doc Test -- MR	maguire	03 Apr 10:43	Completed	Approved Complete	Process Message	Process Message	Completed	
55	collab-from-scratch--66	2 Step Review Doc Test -- MR	maguire	02 Apr 11:08	Completed	Approved Complete	Process Message	Process Message	Completed	

Browsing the list of collaboration jobs

You can view job details ( ), pause ( ), process ( ) or permanently cancel a job that is in progress ( ). You can also delete ( ) cancelled jobs (especially useful in the when developing a job definition).
You cannot manually create collaboration jobs because they are always based on a submission whose form is associated with a job controller service.

Viewing Collaboration Job Details

To view the details of a collaboration job, go to "Operations > Collaboration Jobs" and click the "View" icon ( ) next to the job you are interested in. The screenshot below shows the job details for a completed job.

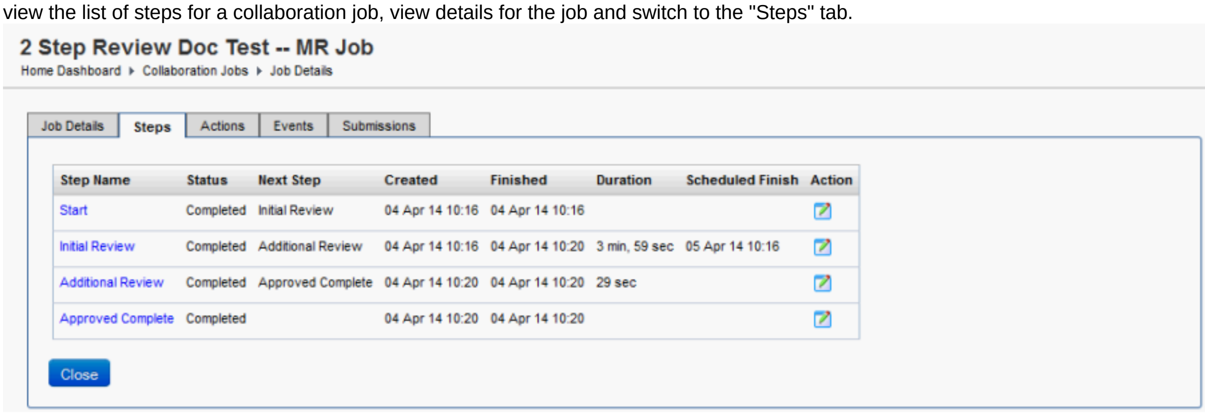


Collaboration job details

The "Job Details" tab contains basic information about the collaboration job: A name, a unique job key and job number, the job controller service managing the job, the status, and various dates from the job's life cycle.

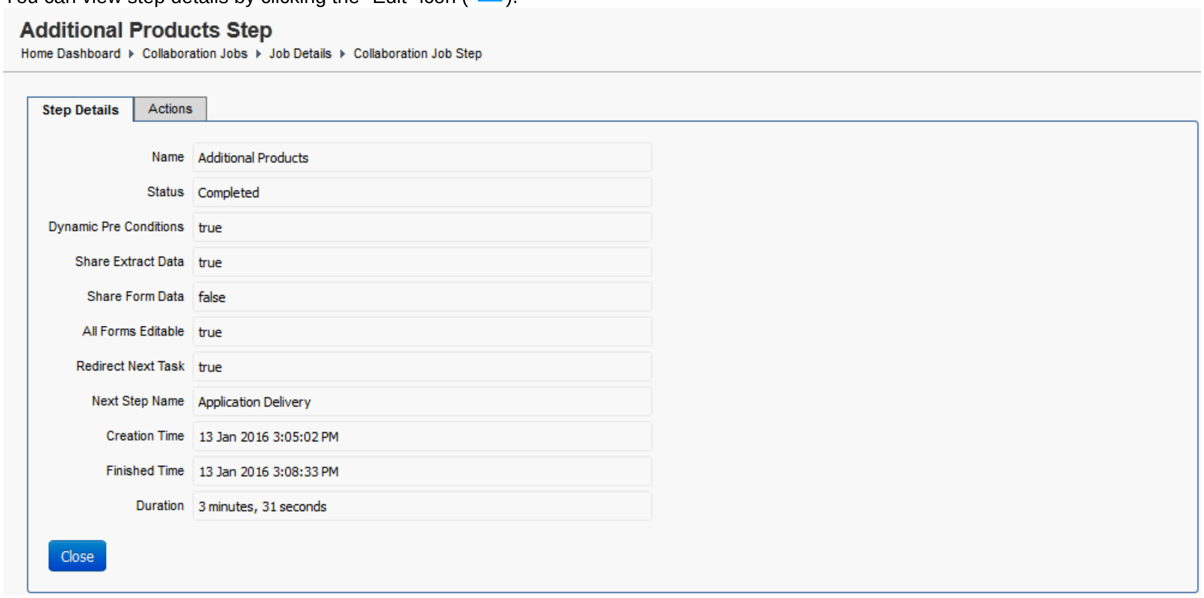
Viewing Collaboration Job Steps

Collaboration jobs consist of individual steps. The definition of which steps are possible and how they are linked resides in the job controller service. To view the list of steps for a collaboration job, view details for the job and switch to the "Steps" tab.



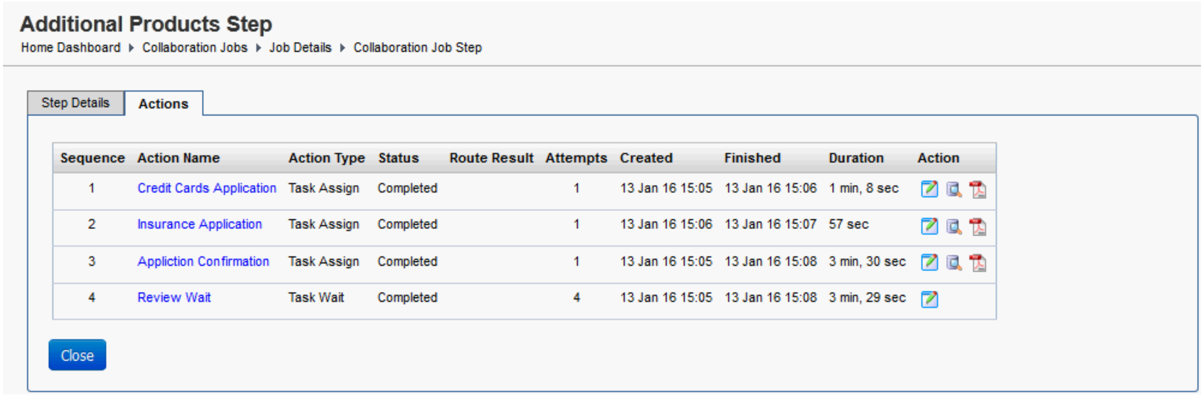
The list of steps for a collaboration job

You can view step details by clicking the "Edit" icon ( ).



Collaboration job step details

Each step has a name, status, creation and finished time and may have a scheduled completion time. Switch to the "Actions" tab to see all the actions for this step.



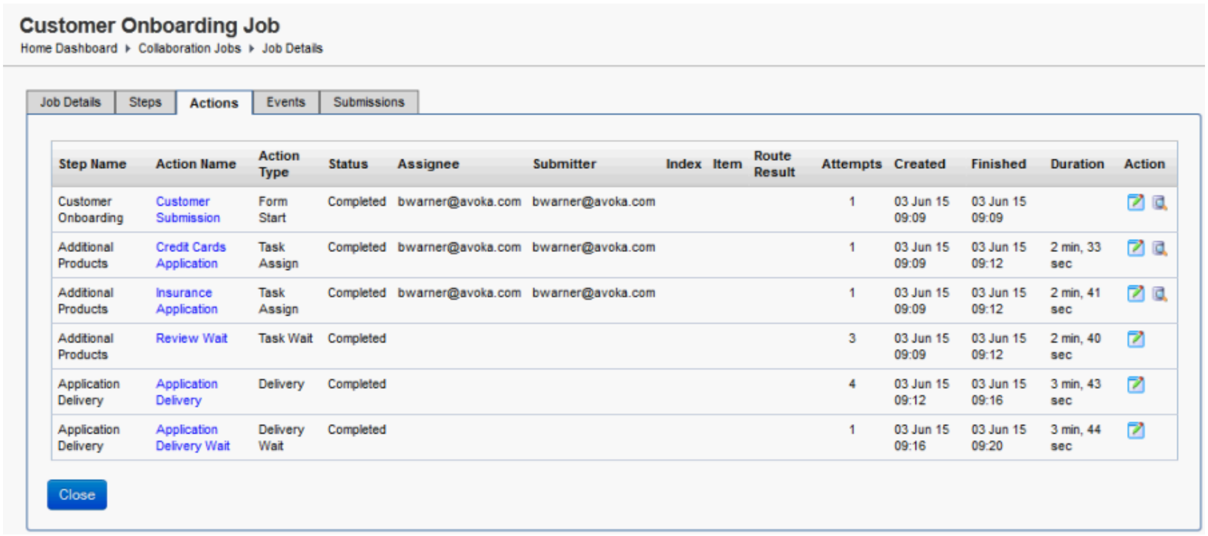
Sequence	Action Name	Action Type	Status	Route Result	Attempts	Created	Finished	Duration	Action
1	Credit Cards Application	Task Assign	Completed		1	13 Jan 16 15:05	13 Jan 16 15:06	1 min, 8 sec	
2	Insurance Application	Task Assign	Completed		1	13 Jan 16 15:06	13 Jan 16 15:07	57 sec	
3	Appliction Confirmation	Task Assign	Completed		1	13 Jan 16 15:05	13 Jan 16 15:08	3 min, 30 sec	
4	Review Wait	Task Wait	Completed		4	13 Jan 16 15:05	13 Jan 16 15:08	3 min, 29 sec	

The list of actions for a collaboration job step

You can edit ( ) job actions and view the details for the submission associated with the action ( ), if one exists. Please see the following section for more on job actions.

Viewing Collaboration Job Actions

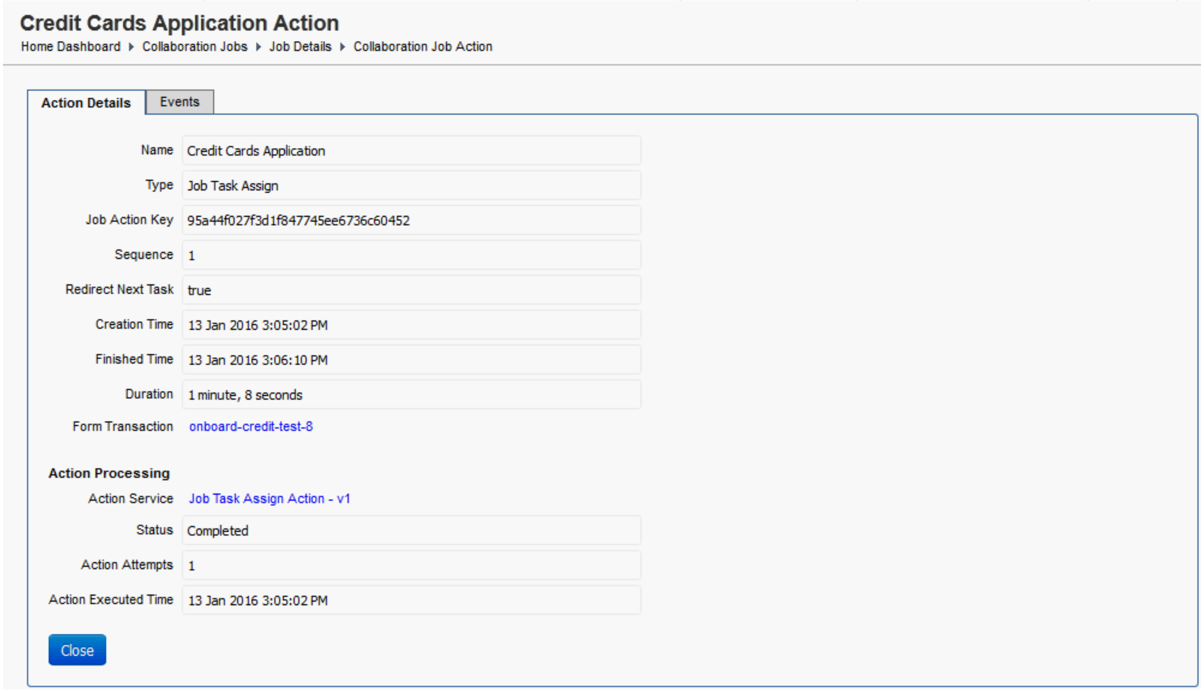
Each collaboration job step may break down into multiple actions. To view the list of actions for all steps in a collaboration job, view details for the job and switch to the "Actions" tab.



Step Name	Action Name	Action Type	Status	Assignee	Submitter	Index	Item	Route Result	Attempts	Created	Finished	Duration	Action
Customer Onboarding	Customer Submission	Form Start	Completed	bwarner@avoka.com	bwarner@avoka.com				1	03 Jun 15 09:09	03 Jun 15 09:09		
Additional Products	Credit Cards Application	Task Assign	Completed	bwarner@avoka.com	bwarner@avoka.com				1	03 Jun 15 09:09	03 Jun 15 09:12	2 min, 33 sec	
Additional Products	Insurance Application	Task Assign	Completed	bwarner@avoka.com	bwarner@avoka.com				1	03 Jun 15 09:09	03 Jun 15 09:12	2 min, 41 sec	
Additional Products	Review Wait	Task Wait	Completed						3	03 Jun 15 09:09	03 Jun 15 09:12	2 min, 40 sec	
Application Delivery	Application Delivery	Delivery	Completed						4	03 Jun 15 09:12	03 Jun 15 09:16	3 min, 43 sec	
Application Delivery	Application Delivery Wait	Delivery Wait	Completed						1	03 Jun 15 09:16	03 Jun 15 09:20	3 min, 44 sec	

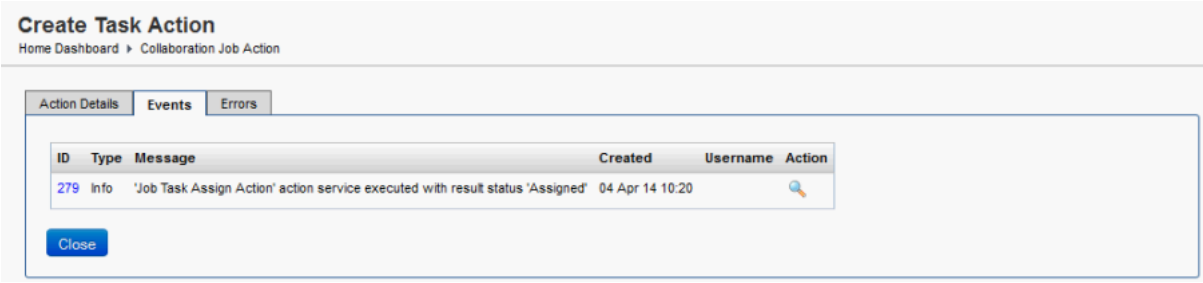
The list of actions for a collaboration job

You can edit each action ( ) and view the details of the submission associated with the action ( ), if one exists.
The screenshot below shows the details for a collaboration job action. Each action has a name, type, unique key, creation and finished time and may be associated with a submission ("Form Transaction" field).
The "Action Processing" section shows information about what service is responsible for handling the action, and what the processing result was.



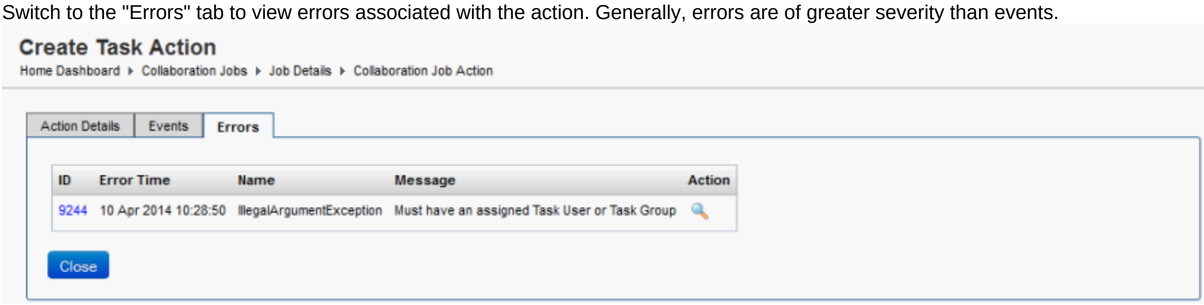
Details for a collaboration job action

Switch to the "Events" tab to see all events associated with this action.



Events associated with a collaboration job action

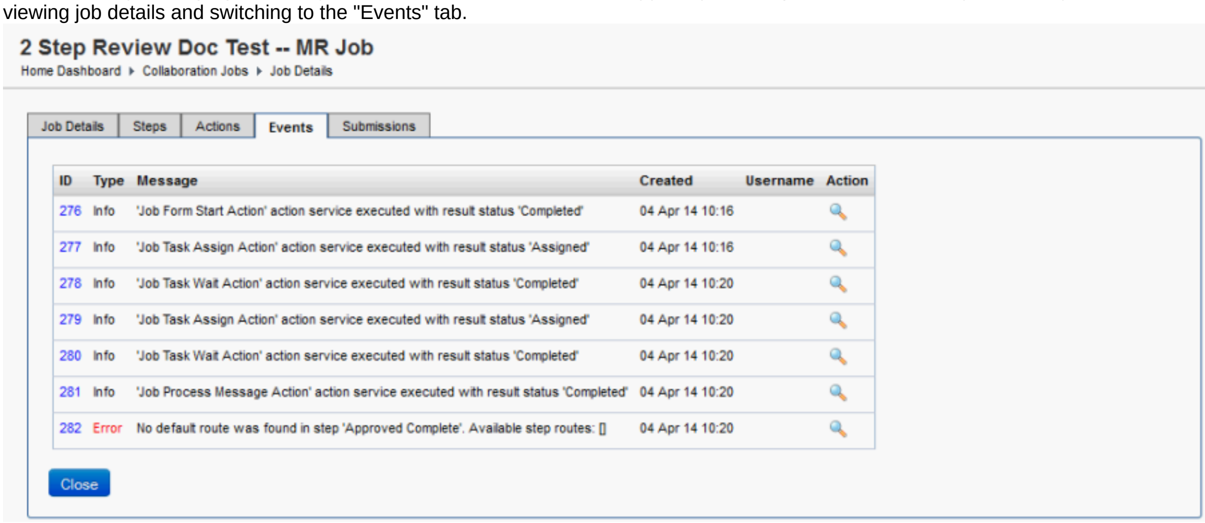
Switch to the "Errors" tab to view errors associated with the action. Generally, errors are of greater severity than events.



The list of errors for a collaboration job action

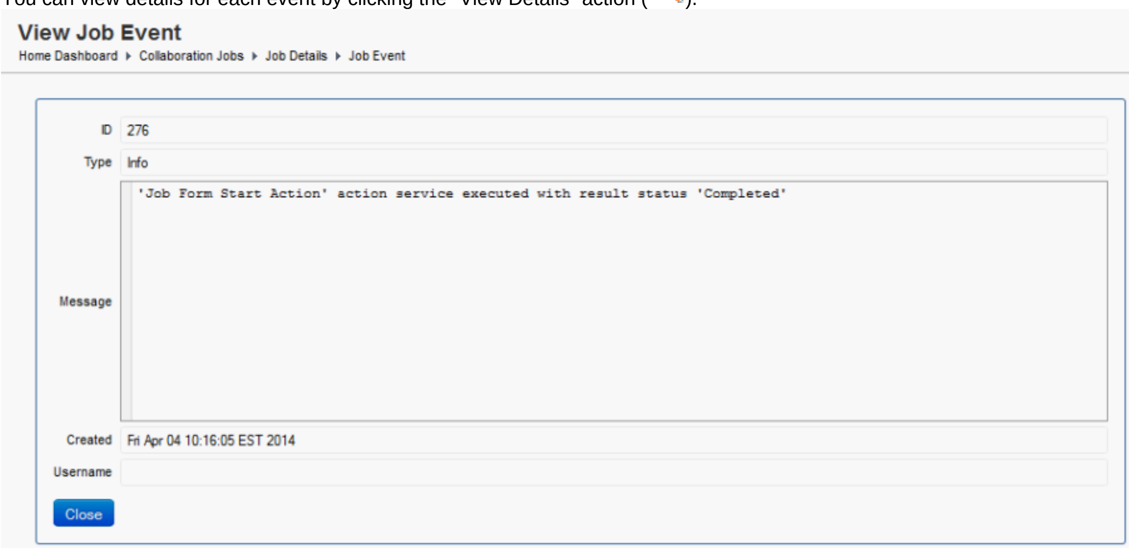
Viewing Collaboration Job Events

In addition to general-purpose TM system events, events can also be logged against a specific collaboration job. These job events can be viewed by viewing job details and switching to the "Events" tab.



The list of events associated with a collaboration job

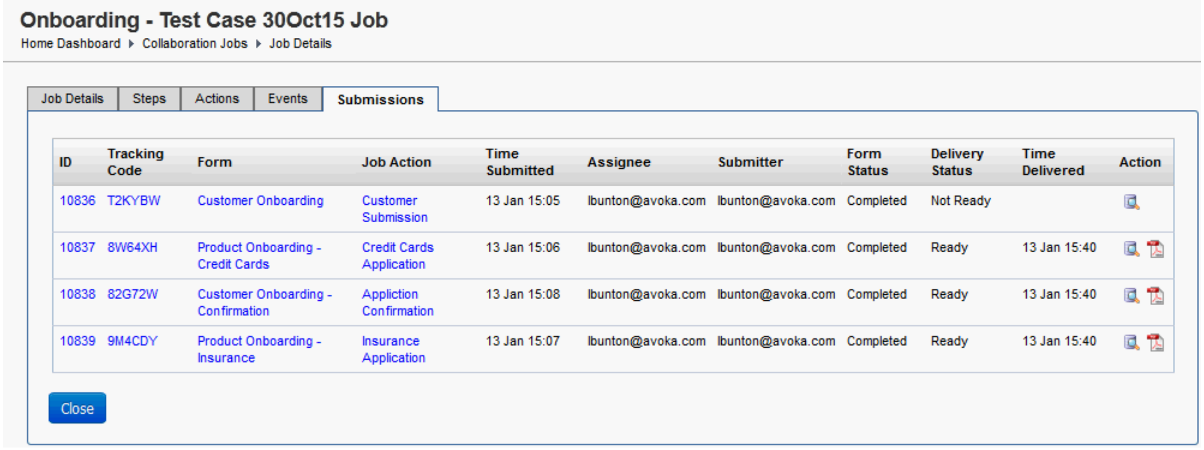
You can view details for each event by clicking the "View Details" action ( ).



Details for a collaboration job event

Viewing Submissions for a Collaboration Job

The set of submissions associated with a collaboration job can be accessed by viewing job details and switching to the "Submissions" tab.



The list of submissions for a collaboration job

For each submission, you can see the tracking code, which form was used, what job action is associated the submission and some other submission status values. You can view details for each submission ( ). See Operations (Manager v5.0) for more on submissions.

Offline Submission (Manager v5.0)

In addition to direct submission and saving forms online to Transaction Manager, PDF forms can also be saved to the user's computer (=offline save) and later be re-opened in Adobe Reader or Acrobat and submitted. Because the submission is done using Adobe Reader or Acrobat rather than a web browser running a PDF viewer plugin, the submission process has to be somewhat different.

Page Contents

Search

Concepts

Offline submission happens when the user submits a PDF form that had been saved locally, using Adobe Reader or Acrobat. In this case, Transaction Manager cannot send back a HTML page as usual, because Adobe Reader cannot interpret a HTML page. Instead, Transaction Manager detects that an offline submission was made and then populates and sends back a PDF (the *offline submission form*), which contains feedback for the user as well as a link back to Transaction Manager. When clicking on this link, a browser window will be opened, which will take the user to the appropriate page in Transaction Manager (e.g. the payment page).

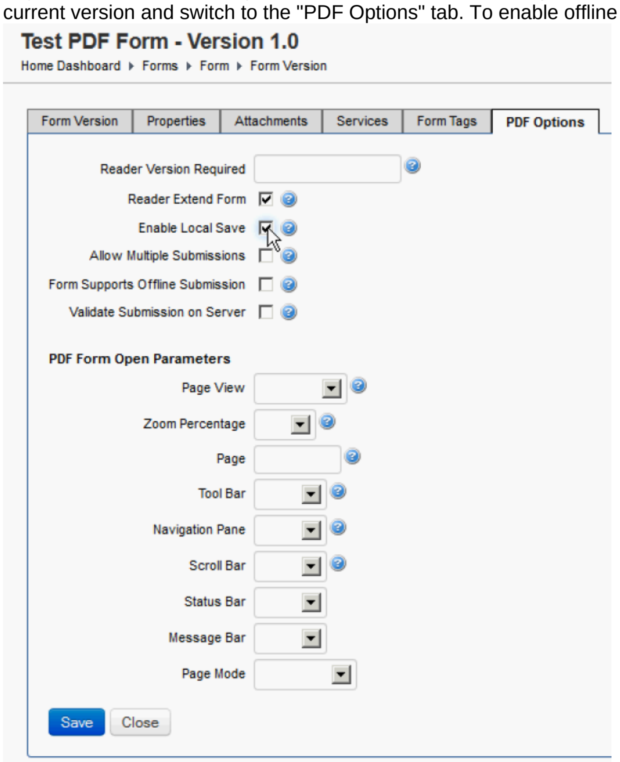
The offline submission form is dynamically populated according to the submission status (completed, attachments or payments outstanding, or failure due to errors) and can also be customized to some extent. While the offline submission form needs to support a certain data structure, you can elect to create your own offline submission form with a customized look and feel.

In addition, forms can be designed to support rendering the offline submission response themselves instead of using a separate form. This provides the user with a more streamlined experience.

Please note that for a form and the data filled in so far to be saved offline by your users, it needs to be reader-extended, and the form needs to be configured to support offline submission.

Enabling Offline Submission

Any PDF form can be configured to allow offline submission (HTML forms can only be saved online). To check a form's configuration, edit the form's current version and switch to the "PDF Options" tab. To enable offline submission for the form, tick the "Enable Local Save" checkbox and save.



Enabling offline submission for a form version

If offline submission is not enabled for a form and a user attempts to submit the form after having saved it locally, Transaction Manager will refuse to accept the submission. However, if a form does not support offline submission, it should be designed such that it will not allow the user to save their form locally in the first place to avoid issues at submission time.

Choosing Offline Submission Mode

In addition, you can decide whether you want the form itself to handle the offline submission response (note that it must be designed to do that) or instead use the offline submission form configured for the space.
To do so, edit the form version you want to configure, switch to the "PDF Options" tab and tick "Form Supports Offline Submission" to indicate that your form can handle the offline submission response, or leave it unticked to use the default offline submission form instead.

Offline Submission Scenarios

When the user submits a form offline, there are multiple scenarios to consider:

- The submission is complete, and the form is configured to skip the confirmation page
  The receipt is rendered back directly to Adobe Reader or Acrobat.
- The submission is complete, and the form is configured to show the confirmation page
  The user can be informed of their receipt number and will be redirected to the confirmation page.
- Attachments are needed
  The user needs to be made aware that attachments are outstanding. They will be redirected to the attachments page.
- Payment is needed
  The user needs to be made aware that payment needs to be made. They will be redirected to the payment page.
- Signatures are required
  The user needs to be made aware that the form needs to be signed.
- The form has expired
  The user attempted to submit a form after it has expired (applies only if an expiry date has been set up for the form). The submission is rejected and the user is informed that the form cannot be submitted.
- An error occurred during submission
  The user needs to be made aware that their form could not be submitted. They will be redirected to an error page.

The offline submission response will be populated with different text depending on what the scenario is. If you customize the text displayed to the user (as described in the next section), please make sure you clearly indicate to the user whether or not their submission is complete. The link back to Transaction Manager will be automatically generated and inserted into the response.

Authentication

If the form is configured to require users to be logged in, Transaction Manager ensures that the space user is authenticated before finalizing their offline submission.
If a user accesses a link to Transaction Manager in the offline submission form and they are not authenticated, they will be directed to the login page for the user space. Only after they have logged in will the submission go ahead. This applies even if the form has been completed and no other user work is outstanding - the user still needs to authenticate before Transaction Manager will process the form.

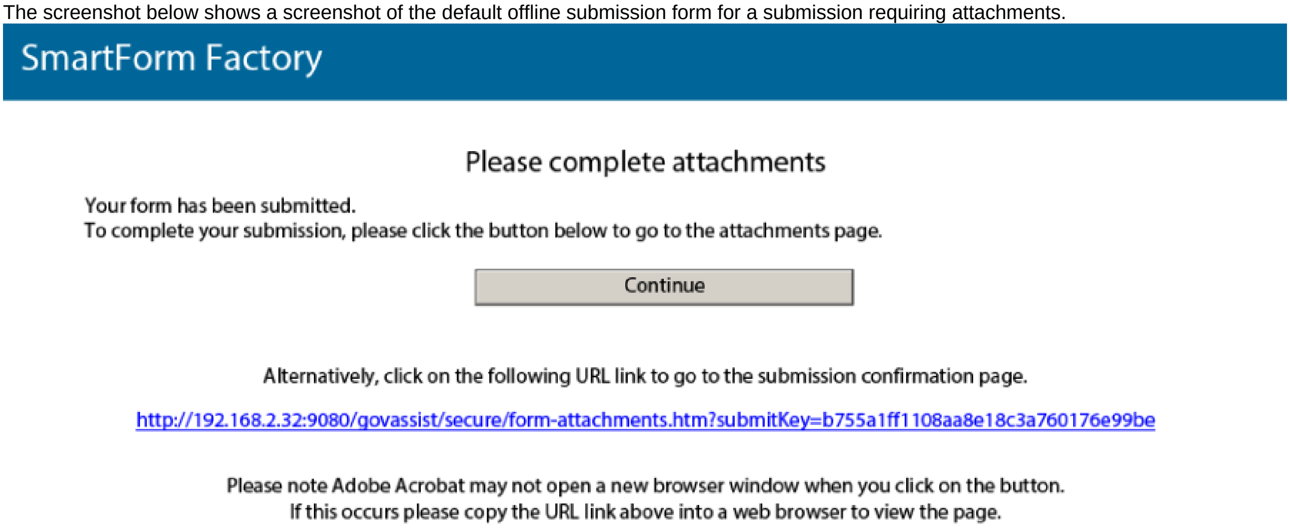
The Example Self Service Space also displays submissions for which authentication is required on the space user's To Do list. The user can then acknowledge submissions directly on the To Do list.

Offline Submission Form Customization

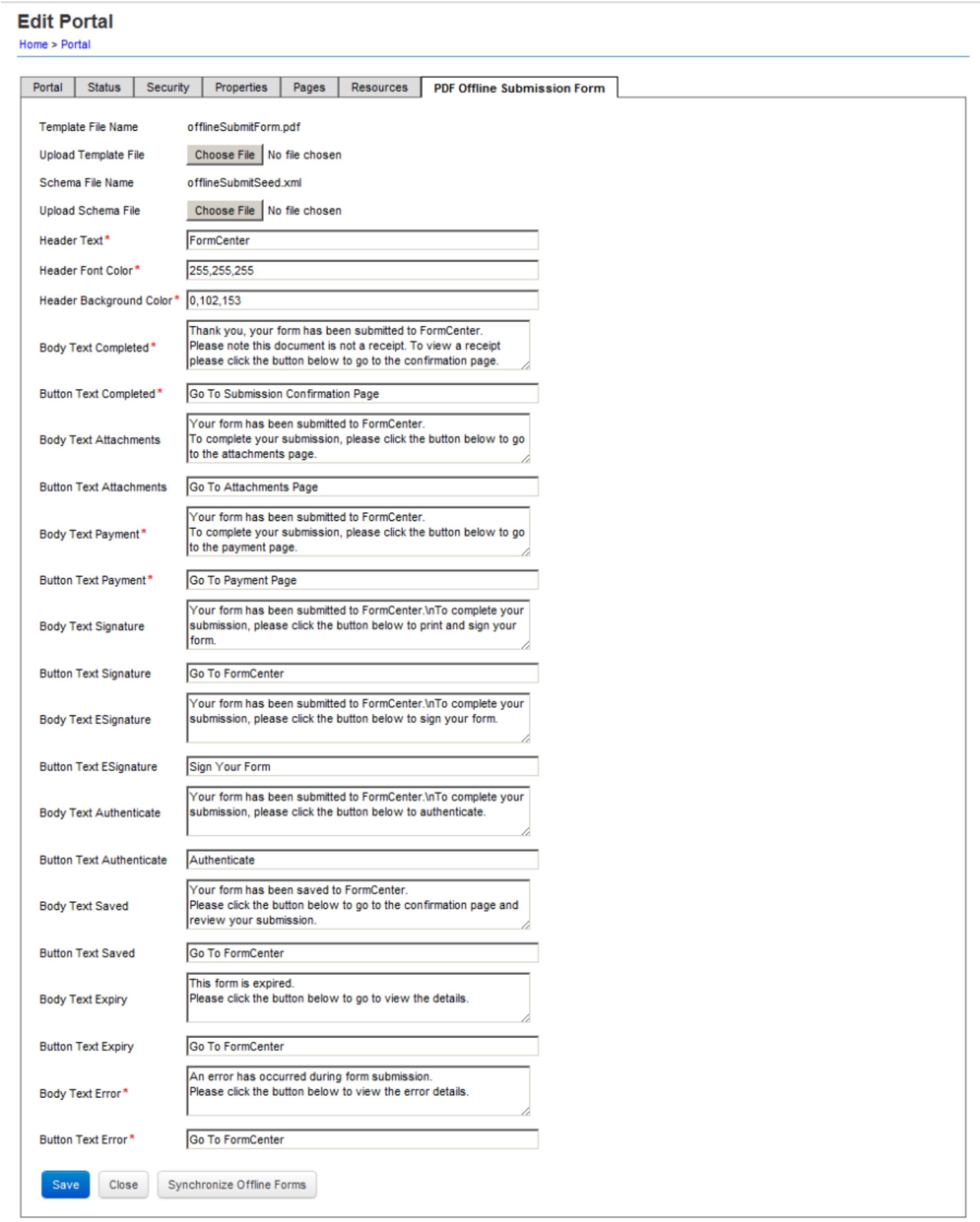
To cater for forms that are not designed to handle the offline submission response, Transaction Manager includes an offline submission form as part of the Example Self Service Space. This offline submission form is not branded and can be customized somewhat. Alternatively, you can also develop your own offline submission form using the same data configuration file and use it instead.

This section describes the various properties that can be configured for an offline submission form.

The screenshot below shows a screenshot of the default offline submission form for a submission requiring attachments.



The offline submission form
The following properties can be configured:
Header: In the screenshot, this is the top section titled "Transaction Manager". You can customize the title text as well as the font and background color.
Body text: In the screenshot, this is the text between the title and the button. You can customize the body text for each submission scenario.
Button: In the screenshot, this is the button labeled "Continue". You can specify a label for each of the submission scenarios.
Note that the section below the button is automatically populated and cannot be customized. If the submission is complete or has errors, the receipt number will be shown in the title.
To customize an offline submission form, go to "Forms > Form Spaces" and edit the user space (each space has its own offline submission form).
Switch to the "PDF Offline Submission Form" tab. The screenshot below shows the configuration of the default offline submission form.



Customizing an offline submission form

You can upload a new version of the offline submission form and its data configuration file by using the "Browse" buttons. Note that the data configuration file needs to match the one used by the default offline submission form.
Adjust the values as needed (the colors need to be specified as RGB values) and save.
You can click "Synchronize Offline Forms" to trigger an attempt to write out the form to the LiveCycle repositories.

Spaces (Manager v5.0)

Transaction Manager supports multiple user spaces, all of which expose their own set of forms. User spaces can be dynamically deployed, register themselves at startup and can be used to host forms after that.

Search

Concepts

A *space* is a Transaction Manager module contained in its own WAR file. One of these spaces is the Transaction Manager management console itself, another the BIRT report module; in addition, multiple user spaces may be installed.

A *user space* is a Transaction Manager module that is used to expose forms to your users and handle form requests, submissions and related activities. While Transaction Manager comes with an Example Self Service Space, user spaces are often customized and branded. Each form is associated with one or more user spaces, which can be used to render the form. If a form is associated with multiple spaces, the form (as well as saved and completed submissions of that form) will generally be visible on all these spaces.

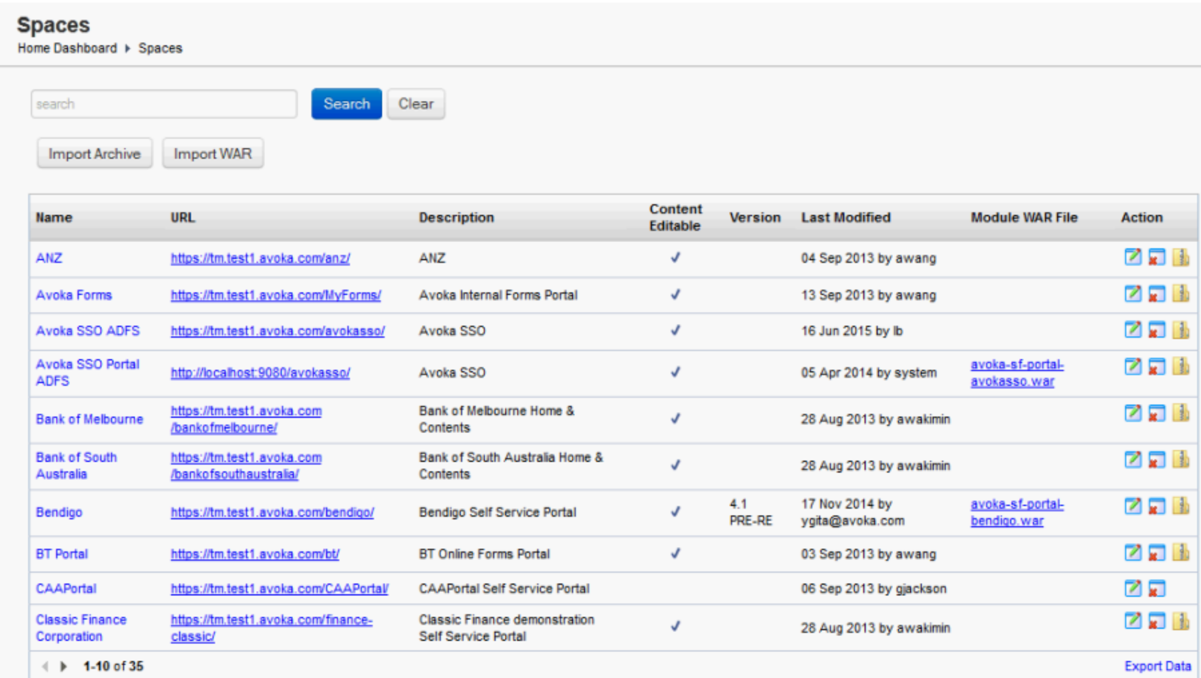
User spaces define a *security model*, which encapsulates strategies around handling user registration and login. Transaction Manager supports authentication against a local database and/or LDAP by default as well as space user self-registration, potentially combined with an approval step for new user accounts.

User spaces have a *status* indicating whether the user space is currently available. If the user space has been set to be unavailable (e.g. due to maintenance), requests made by space users are automatically redirected to a specific information page.
User spaces have a *space context path* (e.g. http://www.server.com/tmportal) and a set of *space pages*, which specify the HTML pages that provide specific functionality (e.g. landing page, payment page etc.). This allows Transaction Manager to call the user space without knowing about its implementation, and potentially allows you to customize space pages without having to rebuild the space EAR. User spaces also have customizable *space resources* (e.g. images) and *space properties*.

In addition, user spaces provide an *offline submission form*. This PDF form is sent back to users when they submit a form after saving it offline (see Offline Submission (Manager v5.0)). Transaction Manager has a default offline submission form you can use, but you can also create a branded version to use instead.

Browsing the List of Spaces

To view the list of spaces, go to "System > Modules" for system spaces such as the management console or to "Forms > Form Spaces" for form hosting spaces. You can filter the list using a substring of the space name, and export the search results to Excel by clicking the "Export Data" link below the table.



Name	URL	Description	Content Editable	Version	Last Modified	Module WAR File	Action
ANZ	https://tm.test1.avoka.com/anz/	ANZ	✓		04 Sep 2013 by awang		
Avoka Forms	https://tm.test1.avoka.com/MyForms/	Avoka Internal Forms Portal	✓		13 Sep 2013 by awang		
Avoka SSO ADFS	https://tm.test1.avoka.com/avokasso/	Avoka SSO	✓		16 Jun 2015 by lb		
Avoka SSO Portal ADFS	http://localhost:9080/avokasso/	Avoka SSO	✓		05 Apr 2014 by system	avoka-sf-portal-avokasso.war	
Bank of Melbourne	https://tm.test1.avoka.com/bankofmelbourne/	Bank of Melbourne Home & Contents	✓		28 Aug 2013 by awakimin		
Bank of South Australia	https://tm.test1.avoka.com/bankofsouthaustralia/	Bank of South Australia Home & Contents	✓		28 Aug 2013 by awakimin		
Bendigo	https://tm.test1.avoka.com/bendigo/	Bendigo Self Service Portal	✓	4.1 PRE-RE	17 Nov 2014 by ygita@avoka.com	avoka-sf-portal-bendigo.war	
BT Portal	https://tm.test1.avoka.com/bt/	BT Online Forms Portal	✓		03 Sep 2013 by awang		
CAAPortal	https://tm.test1.avoka.com/CAAPortal/	CAAPortal Self Service Portal			06 Sep 2013 by gjackson		
Classic Finance Corporation	https://tm.test1.avoka.com/finance-classic/	Classic Finance demonstration Self Service Portal	✓		28 Aug 2013 by awakimin		

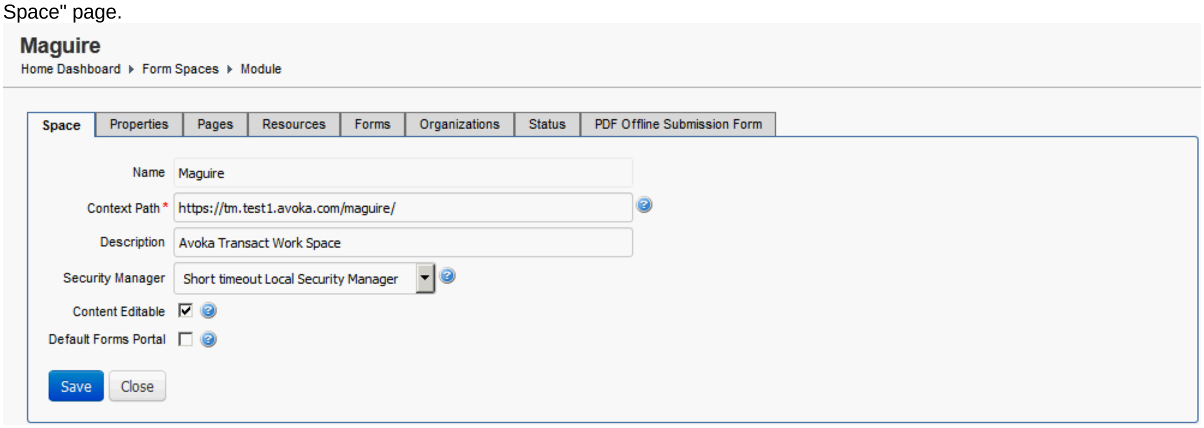
Browsing the list of spaces

This page allows you to add new spaces or edit ( ) and delete ( ) existing spaces as well as import and export space configurations.
The "WAR File Name" column shows the file name of the deployed space WAR file. Form space WAR files can be downloaded by clicking the link on the WAR file name.
You can also import an entire space WAR file by using the "Import WAR" facility. Please refer to _sec_hotdeployspace for more details.

Please note adding new spaces should not be necessary as user spaces generally create an entry in this table when they first start up. Also, the management console space cannot be deleted or renamed.
To assist you in auditing configuration changes, the last modified date and the user modifying the space are displayed in the "Last Modified" column.

Editing a Space

To edit a space, go to "System > Modules" or "Forms > Form Spaces" and click the "Edit Space" icon ( ) next to the space. You will see the "Edit Space" page.



Editing a space

The "Content Editable" checkbox determines whether administrators will be able to adjust space content at runtime. If the space is editable, you will be able to edit all fields and space content. If not, some fields will be read-only, and space content cannot be changed.

Be careful when editing the context path: When the Transaction Manager server starts up, an entry will be created for each space that has been deployed unless an entry with the same space name or context path already exists.
If needed, you can change the space context path to allow for Apache mappings etc. This should be the URL under which the space is accessible to users. Please note that the space context path is used during form submission - if the URL does not match the space context path (including port), the submission will not be accepted.

You can also edit the space description.

You can choose the security manager associated with the space from the "Security Manager" dropdown. If not set, the security manager currently marked as the default will be used. Please refer to Security (Manager v5.0)#_sec_securitymanagers for detailed information about security managers.

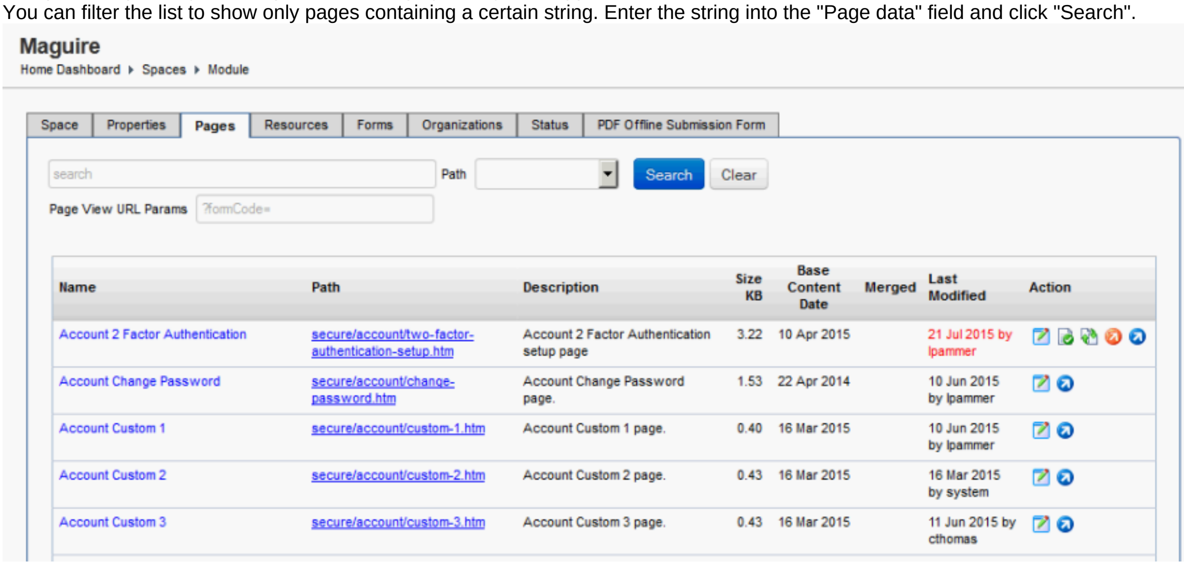
Tick the "Forms Default" flag if the space should be used as the default for rendering forms. This setting is mainly relevant when testing forms via the management console.

Save the space.

You can also configure the space's page paths, resources, properties, offline submission form, status and associated organizations as described in the sections below.

Configuring Pages for a Space

To configure a space's pages, edit the space, ensure the space is editable and switch to the "Space Pages" tab. You will see the list of pages defined for the space. Depending on the space, different pages may be mapped.
You can filter the list to show only pages containing a certain string. Enter the string into the "Page data" field and click "Search".



Viewing the list of space pages

You can see if the page was edited in the "Last Modified" column. If the page has been modified, the last modified date and user are shown. If the content is different to the base content (the page as defined in the space WAR file), the entry is highlighted in red, and additional actions (to view the differences between the current and base content and to restore the base content) are shown.

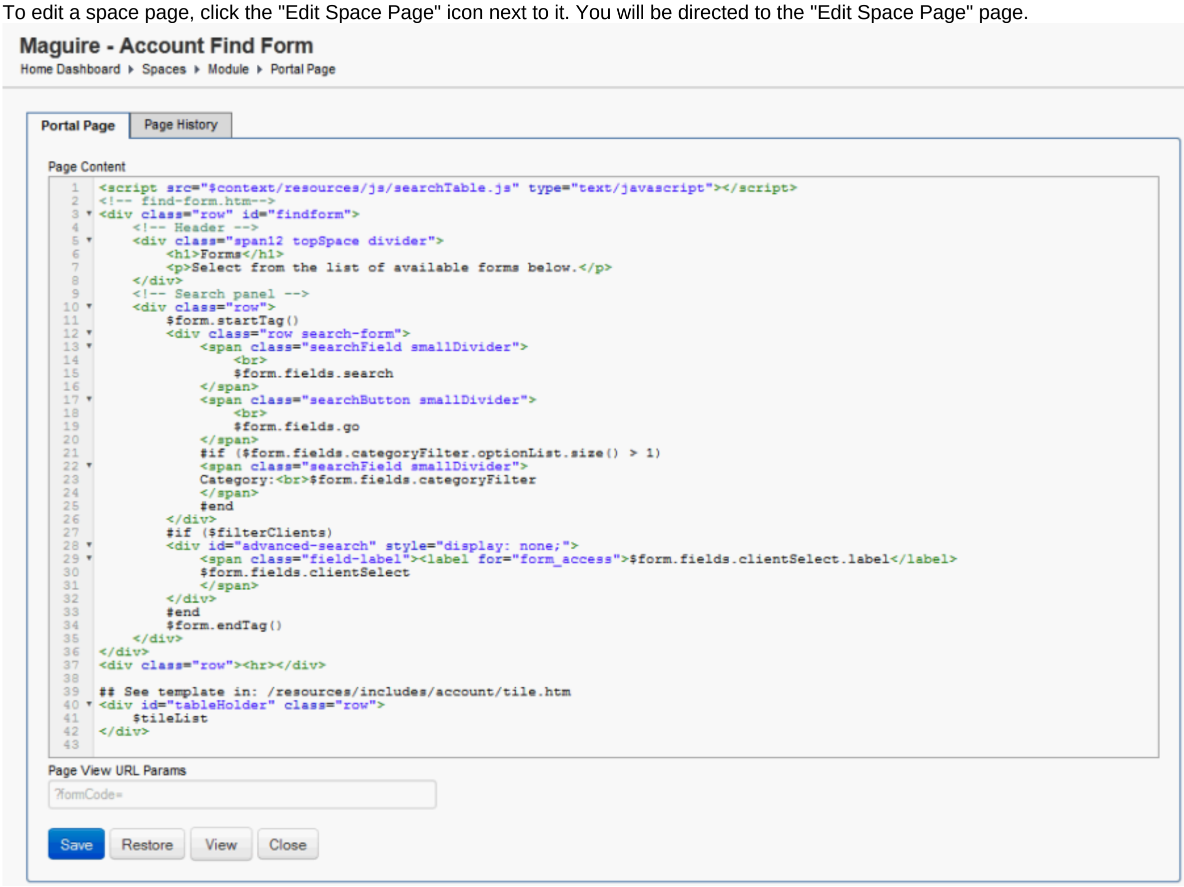
To promote an edited page to be the new baseline, use the "Mark Content Merged" ( ) icon.

Click on the URL or the "View Page" icon ( ) to view the page. If the page needs certain URL parameters to display correctly, for example a form code, enter the parameter(s) into the "Page View URL Params" field first.

You can edit individual pages using the "Edit Space Page" icon ( ). You can also restore the base content of all space pages by clicking "Restore Base Content".

Editing Space Pages

To edit a space page, click the "Edit Space Page" icon next to it. You will be directed to the "Edit Space Page" page.

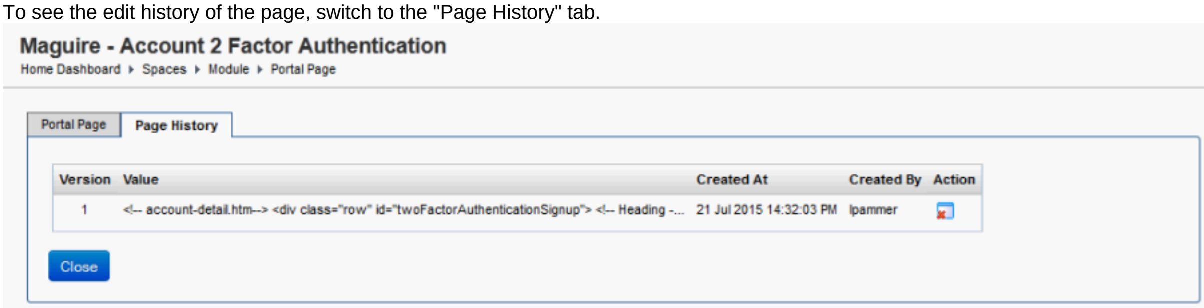


Modifying a space page

This page allows you to modify the HTML pages used by the space, for example to adjust styling at run time. Please be careful when doing this to avoid introducing issues with the page's styling or functionality.
Once you are done editing the page, save your changes.

You can then click "Preview" to inspect the new version. If the page needs certain URL parameters to display correctly, for example a form code, enter the parameter(s) into the "Page View URL Params" field first.
If you would like to restore the base content of the space page, click "Restore".

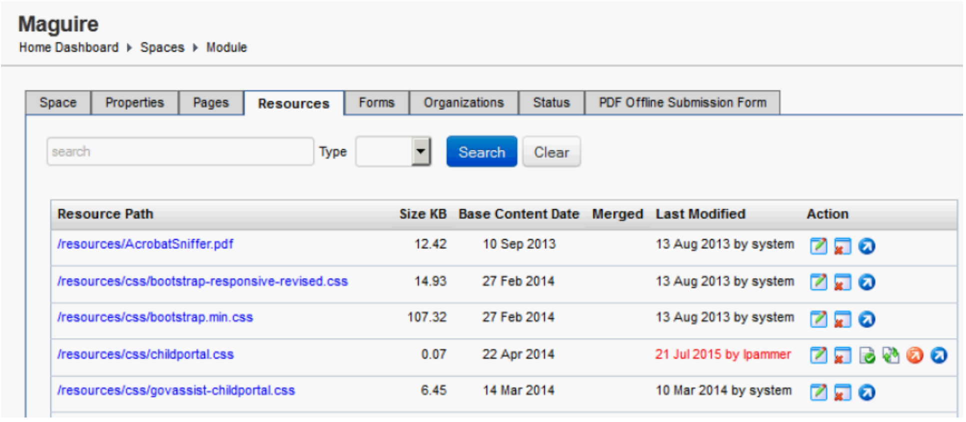
To see the edit history of the page, switch to the "Page History" tab.



Viewing the history for a portal page

Configuring Space Resources

Space resources are files other than HTML pages, such as CSS, images and localization-related files. You can view the list of space resources by editing the space, ensuring the space is editable and switching to the "Space Resources" tab.
You can filter the list to show only resources whose content includes a certain string. Enter the string into the "Page data" field and click "Search". You can also filter by resource type.



Viewing the list of space resources

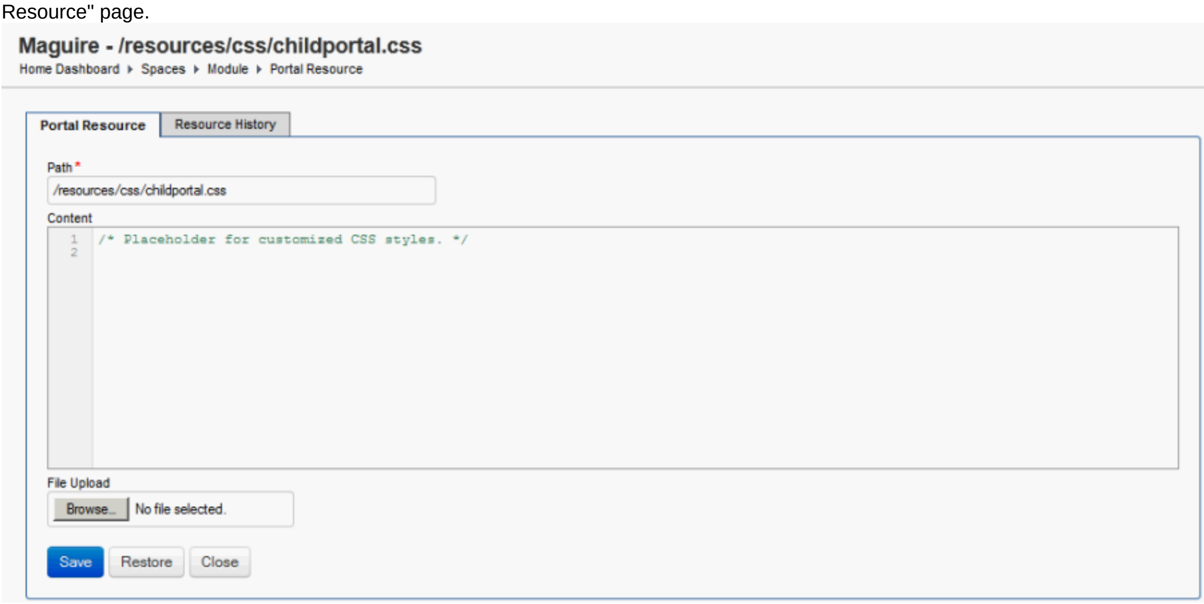
Click on the "View Content" icon ( ) next to the resource to load it in your browser.

You can create new resources by clicking "New" and edit each resource by clicking the "Edit Space Resource" icon ( ).
You can see if the resource was edited in the "Last Modified" column. If the resource has been modified, the last modified date and user are shown. If the content is different to the base content (the resource as defined in the space WAR file), the entry is highlighted in red, and additional actions (to view the differences between the current and base content and to restore the base content) are shown.

To promote an edited resource to be the new baseline, use the "Mark Content Merged" ( ) icon.
You can also revert all resources to their original version by clicking "Restore Base Content".

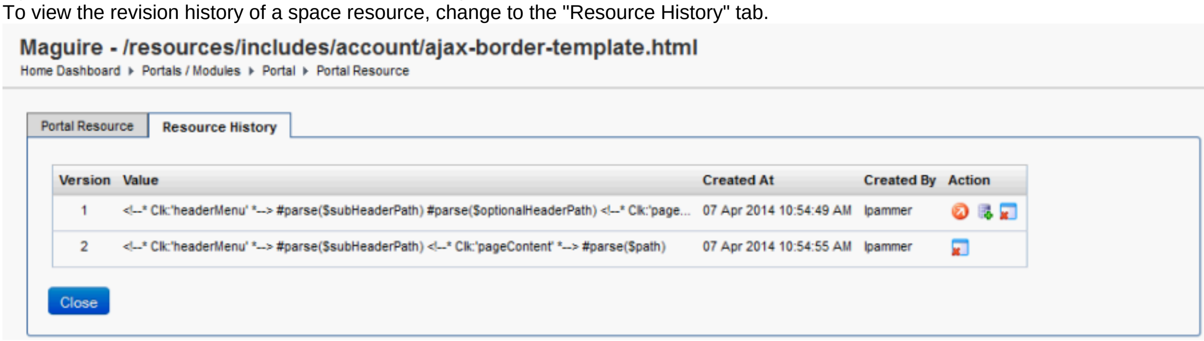
Editing Space Resources

To create a space resource, click "New"; or click the "Edit Space Resource" icon ( ) to edit a resource. You will be directed to the "Edit Space Resource" page.



Editing a space resource

The "Path" field contains path where the resource should be hosted (relative to the space context path).
You can edit textual resources directly, or upload the resource content using the file upload field.
Save the resource when you are done.
If you would like to restore the base content of the space resource, click "Restore".
To view the revision history of a space resource, change to the "Resource History" tab.

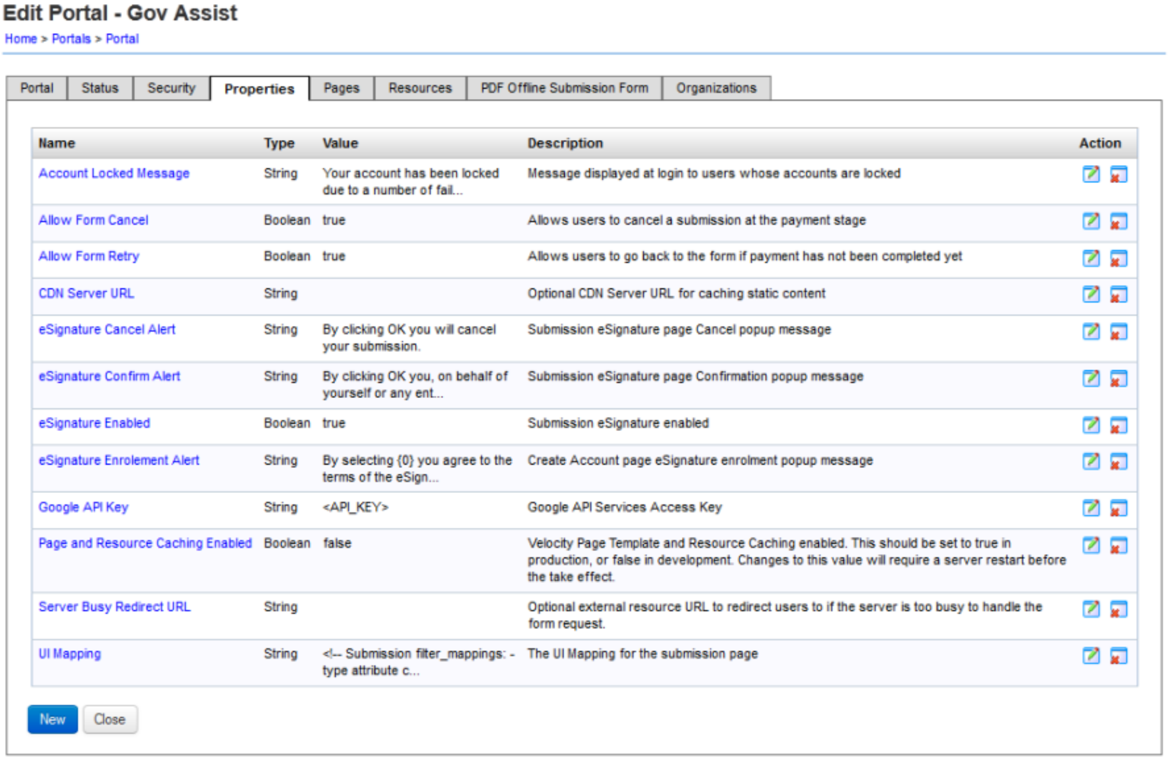


The history of a space resource

You can remove old revisions, restore the resource to the previous revision ( ) and compare the content of a revision to the current resource content ( ).

Configuring Space Properties

Space properties are arbitrary name-value pairs that you can use to customize your space. To view the list of space properties, edit the space and switch to the "Space Properties" tab.



Viewing the list of space properties

You can add, edit ( ) and delete ( ) space properties.

Editing Space Properties

To edit a space property, edit the space, switch to the "Space Properties" tab and click the "Edit" icon ( ) next to the space property. You will be directed to the "Edit Space Property" page.

Creating a space property

Modify the value and save your changes.

Standard Space Properties

Transaction Manager defines a set of standard space properties that can be set for every space. Most of these properties have a default value if the property is not present to avoid having to explicitly define these properties for all spaces.
The current list of standard user space properties is:

Property	Default	Description
Account Locked Message	Depends on space	Message displayed to a user whose account was locked due to too many failed login attempts. May not apply to all spaces depending on security configuration.
Allow Form Cancel	true	Controls whether users are able to cancel their submission on the payment page.
Allow Form Retry	true	Controls whether users are able to return to the form after submitting it, and potentially resubmit using different data.

Allow XML Redirect Target	false	Controls whether to check submission data for a custom HTTP redirect URL. If this setting is enabled and a custom redirect URL is found, the user is sent there after submission.	
Allow XML Request Param Form Prefill	false	Allows XML request parameters [xmlData | schemaSeed] to be used in form prefill.	schemaSeed] to be used in Form Prefill.
Anonymous Submission Access Timeout	15 min	The grace period in minutes during which anonymous users can access their submission.	
CDN Server URL	-	If set, instructs the space to load static resources such as CSS and images from a Content Distribution Network.	
Email Form Receipt Subject	…	Subject of the submission receipt email sent to space users	
Email Form Receipt Message	…	Body text of the submission receipt email sent to space users	
Email Lost Password Message	…	Lost password reset email message	
Email Lost Password Subject	…	Lost password reset email subject	
Email Saved Form Subject	…	Subject of the email sent to space users regarding a saved submission	
Email Saved Form Message	….	Body text of the email sent to space users regarding a saved submission	
eSignature Enabled	false	Controls whether the space will display information about eSignatures to the user.	
eSignature Confirm Alert	-	Text displayed to the user when they are accepting a digitally signed form receipt	
eSignature Cancel Alert	-	Text displayed to the user when they are rejecting a digitally signed form receipt	
eSignature Enrolement Alert	-	Text displayed to the user when creating a user space account on a space where the property "eSignature Enabled" is set to true	
Form Terms And Conditions	-	The form Terms and Conditions statement the user must accept before accessing the form.	
Google API Key	-	Google API Services Access Key	
Server Busy Redirect URL	-	If set, this URL will be used to redirect space users to if they attempt to render a form, but the system cannot handle the request due to high load. By default, the "Server Busy" space page will be used.	
Use Email for Login	false	Use the email address as the user login name for Local TM user accounts.	

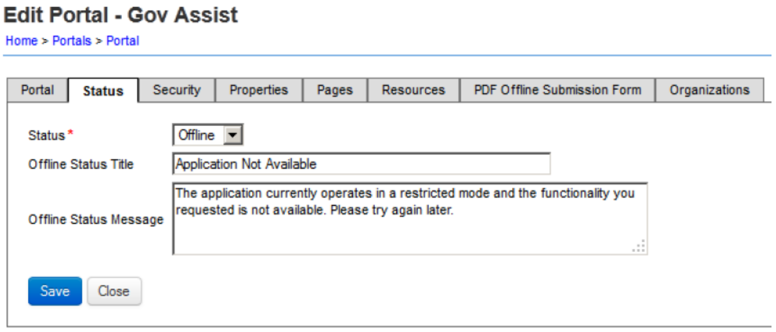
Note: If you are using a custom user space, some of these properties may not be implemented.

Configuring the Offline Submission Form for a Space

To configure a space's offline submission form, edit the space and switch to the "Offline Submission Form" tab (see Offline Submission (Manager v5.0)#_sec_offlinesubmissionformcustomization).

Editing the Space Status

Transaction Manager allows you to restrict access to each user space for maintenance purposes. This is achieved by maintaining a space status as well as some status messages. If a space is put into offline mode, users cannot access any content on that user space - instead, they will be redirected to a space-specific page that displays the status messages to the user. Once the space is put back into normal mode, all content can be accessed as before. Please note that the management console and Business report space do not support separate statuses.
To edit the space status, edit the space and switch to the "Status" tab. You will see the current status and the status messages (the screenshot below shows a space that is currently offline).

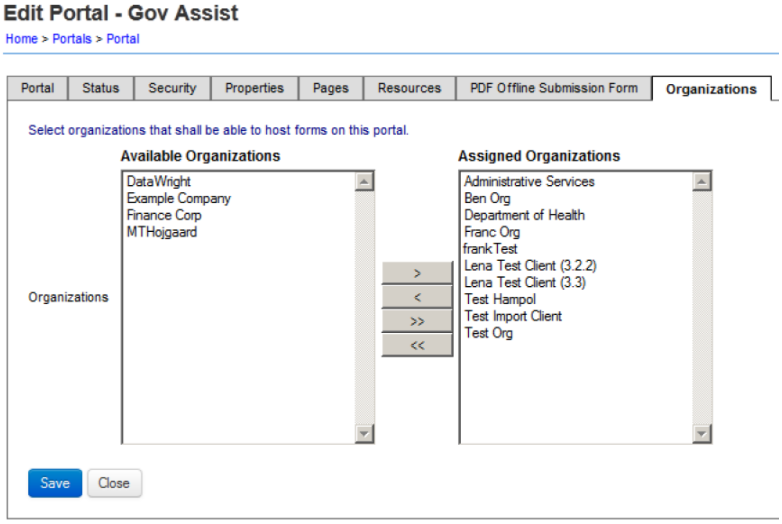


Editing the space status
Select the new status from the "Status" dropdown. Possible values are "Normal" and "Offline".
Modify the offline status title and message as necessary. These messages are relevant only when the space is in offline status. In this case, requests made for any content belonging to that space are redirected to the "Offline" space page (see _sec_configurepagesforspace for more on space pages). This page will use the offline status title as its page heading and the offline status message as the page body, so make sure to enter content that gives the space users the information they need (you can enter HTML content including links).
When you are done editing the status value and messages, save your changes.
In addition, requests for forms can be handled more intelligently by using a message specific to the form. To set up a customized message for a form,

edit the form and add a property of type "Offline Status Message". This message overrides the offline status message when space users attempt to access a page specific to the form (e.g. the landing page).

Associating Organizations with a Space

For an organization to be able to host forms on a specific user space, it needs to be associated with that space. This can be done for each organization (see Organization Configuration (Manager v5.0)#_sec_organizationspaceaccess), but you can also change the organizations associated with a space. To view the list of organizations associated with a space, edit the space and switch to the "Organizations" tab. You will see the list of available and assigned organizations.

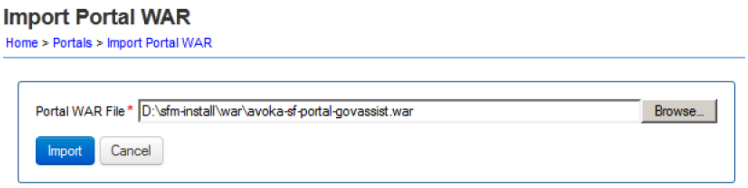


Assigning organizations to the space

You can use the ">" and "<" buttons to assign and unassign organizations. Click "Save" when you are done.

Hot Deploying a Space

"Hot deploying" a space means updating the underlying WAR file on the Transaction Manager application server without shutting down the server and restarting it. Transaction Manager now supports hot deployment of spaces by importing a WAR file via the management console.
To hot deploy a space, go to "Forms > Form Spaces" and click "Import WAR". The "Import Portal WAR" page will be displayed.



Hot deploying a space WAR file

Browse to the updated space WAR file on your hard disk and click "Import". If overwriting an existing WAR file, ensure the file name matches – otherwise conflicts may occur.
It is recommended to use this functionality on development servers only and carefully restrict access to the feature (using the "Import Space" permission) because accidental or malicious deployment of WAR files can be potentially very harmful. Also, when upgrading to a new Transaction Manager version, do not use this feature to update the WAR files, but instead follow the upgrade instructions provided.If you do use this feature, ensure that the space you have deployed starts up normally and is fully accessible afterwards.

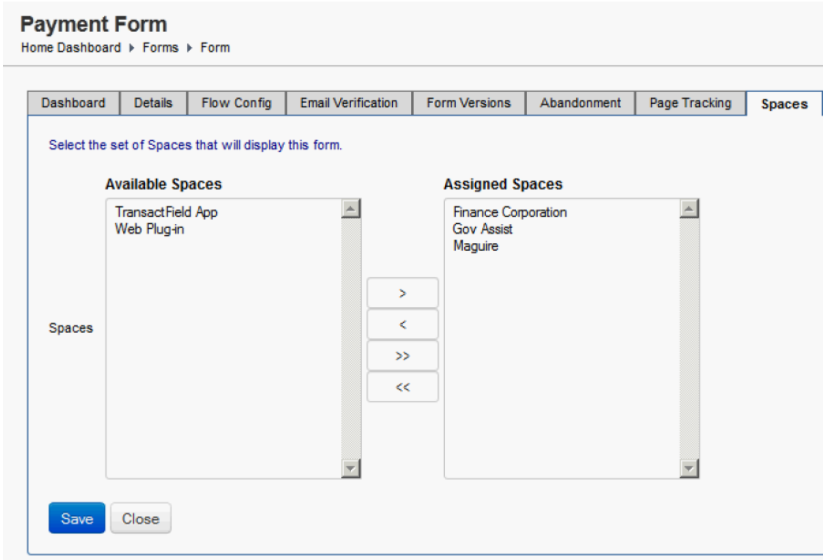
Deleting a Space

Before deleting a space, check if there are still forms associated with it. If so, these forms will not render after the space is deleted. You can fix this by associating them with a different space.

To delete a space entry, go to "System > Modules" or "Forms > Form Spaces" and click the "Remove Space" icon ( ) next to the space. You will be asked for confirmation before the space is deleted. Please note that if the space WAR file is still deployed on the application server, the space will re-create its entry when the server restarts.

Associating Forms with a Space

To associate a form with a particular space, go to "Forms > Forms" and edit the form. Change to the the "Space Access" tab, assign the desired set of spaces and save. From that point on, the form will be visible on the set of assigned spaces. By default, other user spaces will refuse to display the form.



Changing the space(s) a form is hosted on

Advanced Form Lookup (Manager v5.0)

Users can search for forms on the Example Self Service Space by using conventional search fields and filtering. If there is a large number of forms or if external resources should be referenced as well, however, more sophisticated form finding methods may be needed.

Search

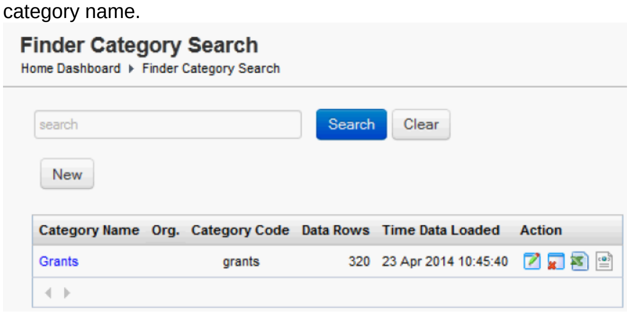
Concepts

Transaction Manager provides a way for users to search for forms or external resources by answering a series of questions about their needs. Each answer is used to filter the set of forms and resources.
A set of questions that are used to filter forms is called a *finder category*. It contains a form data file describing the forms and resources that shall be exposed as well as a user interface definition file. This separation is useful when a form data file is exposed multiple times via different user interface definitions.

The Transact Finder module is an example of how a finder category might be rendered, providing a user interface to progress through the set of questions and display matching forms. In addition, users can also search the entire set of forms and resources by keyword.

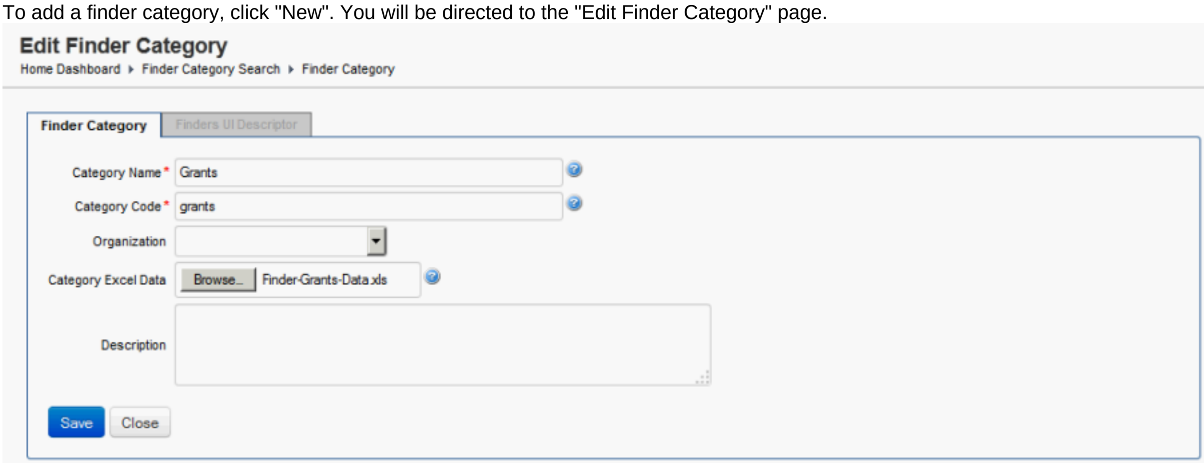
Setting Up Finder Categories

To view the list of finder categories in Transaction Manager, go to "Forms > Finder Categories". You can filter the list of categories by a substring of the category name.



Viewing the list of finder categories

You can use this page to create new finder categories or edit ( ) and delete ( ) existing categories. You can also view the form data file ( ) and the user interface definition file ( ) for each category.
To add a finder category, click "New". You will be directed to the "Edit Finder Category" page.



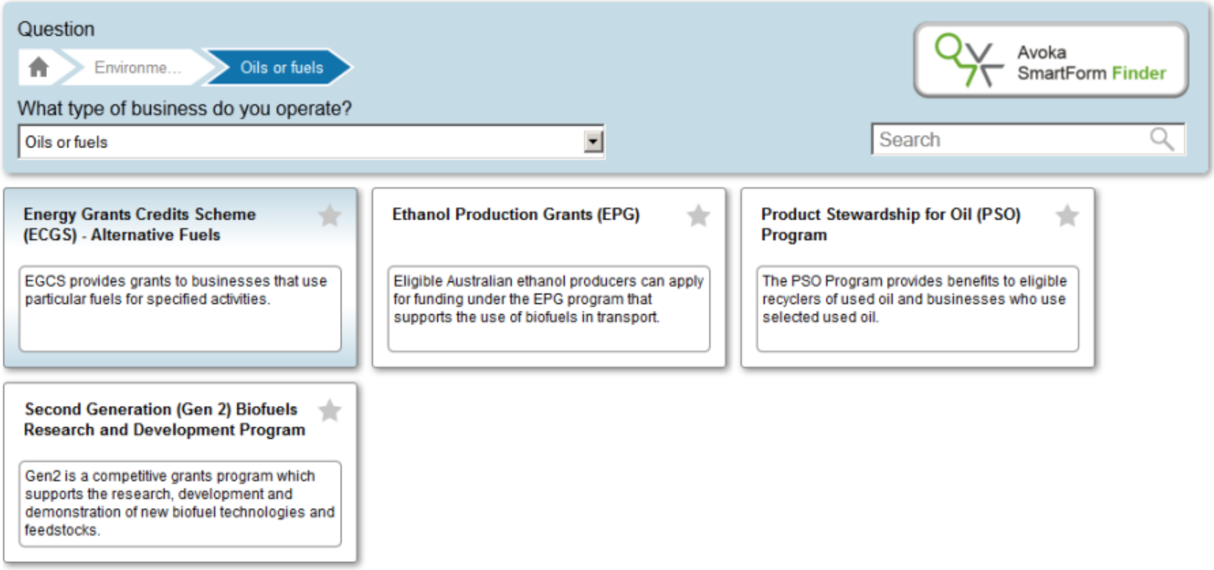
Creating a finder category

Enter a unique category name and a short unique category code.
Upload the form data file using the "Category Excel Data" field. The form data file is a Microsoft Excel file containing a line (or multiple lines) for a form or external resource. External resources define a URL, and forms are identified by the form code. All entries also have a description and an arbitrary set of attributes, which are used for filtering.
Save the finder category and switch to the "Finders UI Descriptor" tab.
Upload the user interface definition file using the "Upload XML File" field. The user interface definition file is an XML file defining the series of questions to be asked of the user, and what attribute in the form data file the questions correspond to.If you have already uploaded the user interface definition, you can also edit the XML data in the "Finders XML Descriptor" field directly.
When you are satisfied with the user interface definition, save your changes.

Accessing the Finder

Please note: The Transact Finder module contains an example page that will display the finder category with code "grants", if one exists. Generally, agencies would provide their own finder page using a Transaction Manager finder category. For more information around this, please refer to the Finder Examples module ("avoka-sf-finder-examples.war"), which is created by the Transaction Manager installer.

To be able to access the Finder module, it must be deployed on your Transaction Manager instance. The relevant deployment module is named "avoka-sf-finder.war".
To access Finder, create a finder category with code "grants" and then access <TM server root>/finder e.g. http://transactionmanager.com/finder
The screenshot below shows the Finder after the user has answered two questions. Note also the global search field and the quick navigation bar at the top that allows the user to jump to one of the previous questions.



The Finder

Scalability and Server Nodes (Manager v5.0)

Search

Concepts

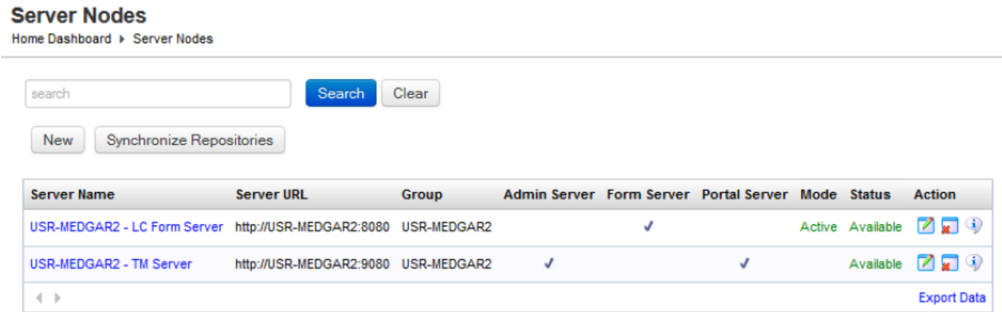
To allow a scalable architecture, Transaction Manager allows you to define *server nodes*. Each server node corresponds to a Java Virtual Machine involved in this Transaction Manager deployment. Server nodes may host Transaction Manager spaces and/or Adobe LiveCycle. There may be multiple nodes hosting the same space, and multiple LiveCycle nodes. A server node hosting Adobe LiveCycle is called a *form server node*.

For further details about the different deployment architectures that are supported, please refer to the TM Installation Guide, section "Server Nodes and Service Connections".
Note that all Transaction Manager nodes share the same database.

Configuring the Set of Server Nodes

To view the list of server nodes configured in Transaction Manager, go to "System > Server Nodes". You will see the currently configured servers (the screenshot below shows an example). The "Status" column is dynamically calculated based on recent health data. If the status in "Unavailable" for a server node, try reloading the page to see if the status persists, and if so, this could indicate an issue with the server node.

You should have a server node entry for each Java Virtual Machine hosting a Transaction Manager module or LiveCycle. Transaction Manager creates a default node at installation time representing a LiveCycle node, and a default node representing the Transaction Manager server. Ensure you correct the configuration (including IP address and credentials) for these nodes.

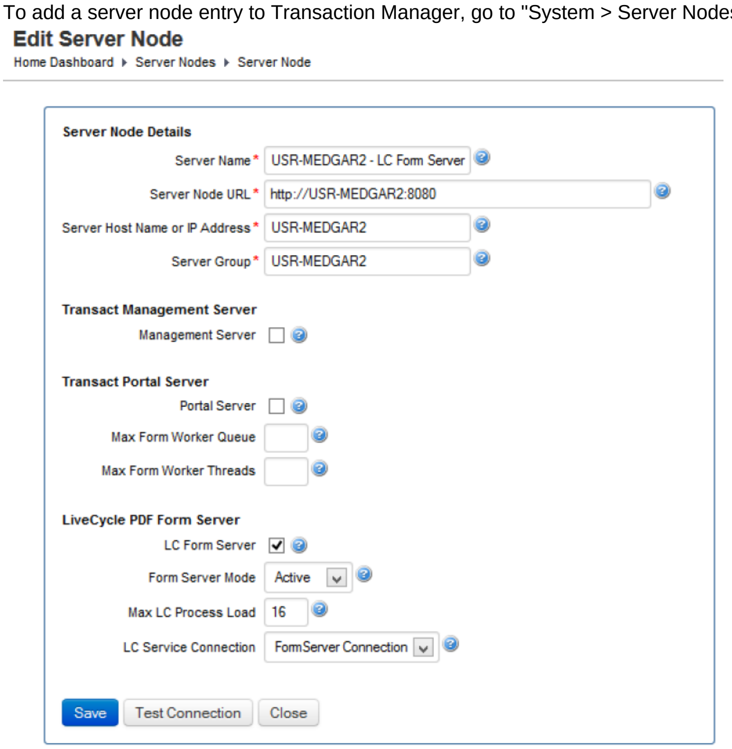


The list of server nodes

You can filter the list of servers by a substring of the server name. This page also allows you to add, edit ( ) and delete ( ) server nodes, as well as check the health status of a specific node ( ).
In addition, you can click "Synchronize Repositories" to synchronize the set of forms

Adding a Server Node

To add a server node entry to Transaction Manager, go to "System > Server Nodes" and click "New". You will see the "Edit Server Node" page.



Creating a new form server node

Enter a name for the server node into the "Server Name" field. This name is purely descriptive.
Enter the URL to the server into the "Server Node URL" field. The URL should use the domain name if there is one.
Enter the server's IP address into the "Server IP Address" field.
Enter the name of the server group the server node belongs to into the "Server Group" field. Generally, server nodes that can communicate via the network should be in the same server group; if your deployment consists of multiple sets of servers that cannot communicate via the network, each set of servers needs to be a separate server group. The group names themselves can be chosen arbitrarily.
Depending on what modules are deployed on the JVM that the server represents, tick one or more of the "Management Server" (management console), "Portal Server" (User Space) and "LC Form Server" checkboxes.
For portal servers, you can set the maximum length of the form worker queue and number of form worker threads.
The "Form Server Mode" dropdown can be used to indicate what status the LiveCycle form server node is in. "Active" signifies that the server is running and ready to take on tasks. "Standby" signifies that the server is running but not currently taking on Transaction Manager tasks. "Inactive" signifies that the server may not be running, and Transaction Manager will not query the server.
The "Max LC Process Load" field is relevant only to form server nodes and determines how many requests will be given to the server to process at a given time. A reasonable value for the maximum request load would be 10.
The "Service Connection" dropdown is typically used only for form servers. It points to an existing service connection, which basically contains connection details including authentication credentials and is used when requests are sent to the server node. Please refer to Services (Manager v5.0) #_sec_manageserviceconnections for more information on service connections and how to set them up.
Save your changes to create the new server.
If the server status is "Active" or "Standby", health data will be collected about the server.

Deactivating a Server

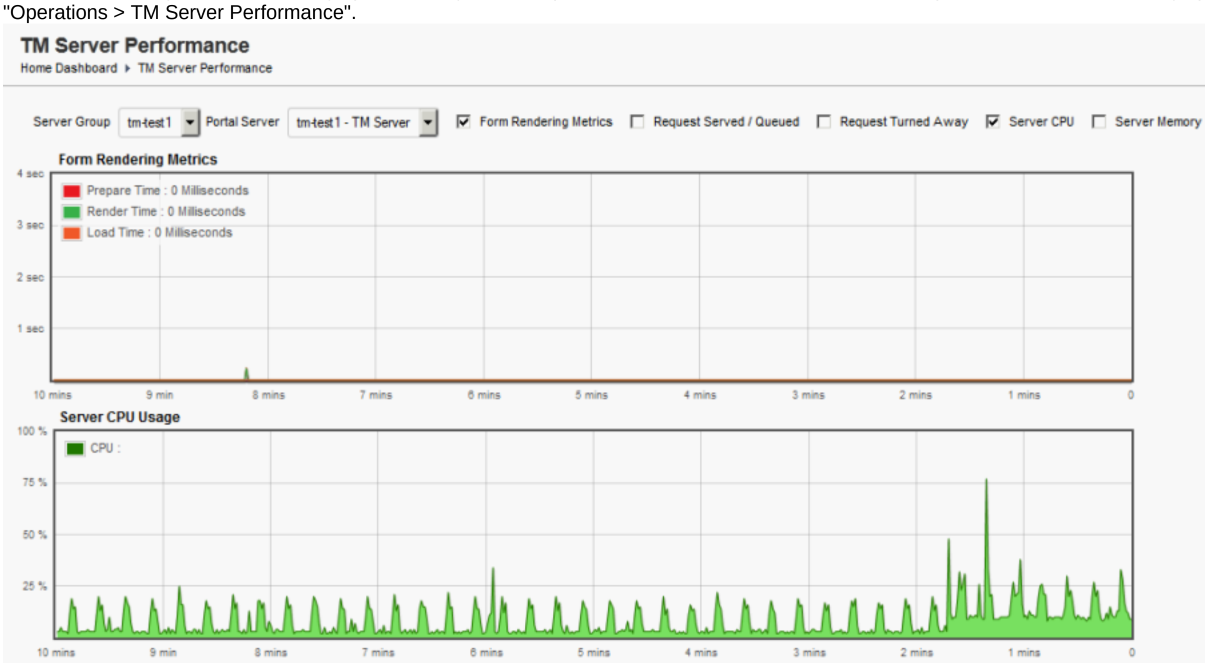
To temporarily deactivate a server, go to "System > Server Nodes", find the server and edit it. Set the status to "Inactive" and save.
The server will then no longer be selected when rendering forms or receipts. To ensure uninterrupted operations, do not deactivate all servers providing portal or LiveCycle services at the same time.

Removing a Server

To permanently remove a server, go to "System > Server Nodes", find the server and click the "Delete" icon ( ). Please note that some associated data, such as health records, will be permanently removed along with the server.

Monitoring Space Server Performance

Transaction Manager offers a set of graphs showing real-time performance measurements for the set of space servers. To view the graphs, go to "Operations > TM Server Performance".

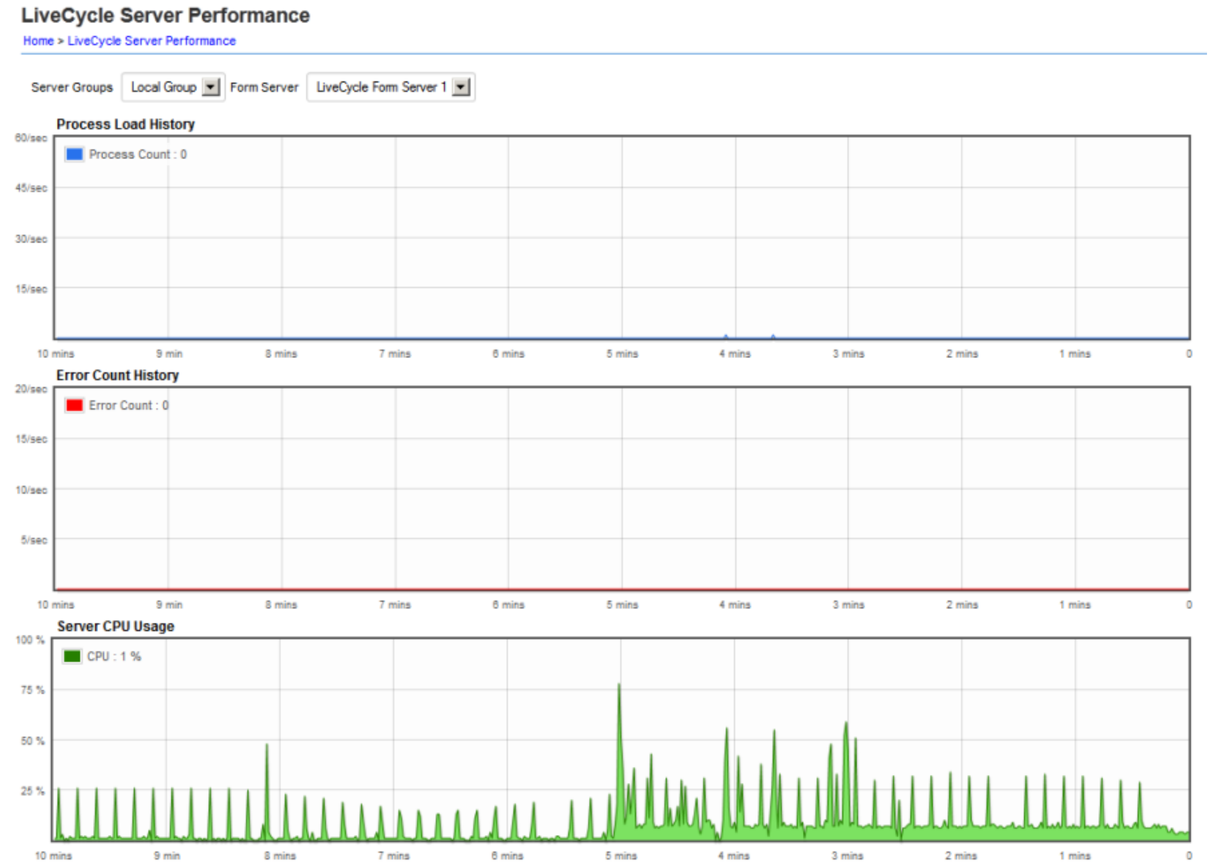


The space server performance graphs

You can select the server group and the space server and choose from a set of five graphs by ticking the checkboxes at the top.
The page will automatically update to show real-time metrics for the graphs you selected.

Monitoring Form Server Performance

Transaction Manager offers a set of graphs showing real-time performance measurements for the set of active form servers. To view the graphs, go to "Operations > LiveCycle Performance".
You will see three graphs that are constantly updating.



The form server performance graphs

You can filter the data by server group and form server.
The "Process Load History" chart shows the number of concurrent processes on the form server.
The "Server CPU Usage" chart shows CPU load for the form server over time.
The "Error Count History" chart shows the number of form server invocations that resulted in an error being thrown.

Multi-Environment Support (Manager v5.0)

Search

When forms are developed and hosted, there will often be several environments set up: Development and test server(s) as well as the production environment. As the form matures, it migrates from one environment to the other.
Transaction Manager was built with these processes in mind and allows you to easily migrate forms, form versions, organizations, metadata tag definitions, services, security managers and role configurations from one environment to the other. Each of these can be exported separately

Migration Order

If you want to migrate a set of entities (e.g. an organization and all its forms, or all roles, spaces and service definitions), you need to know in which order to import your export archives.
The following list shows the import order you should follow in general:

1. Services (service definitions and service connections)
2. Spaces
3. Security managers
4. Roles
5. Metadata tags
6. Organizations
7. Application packages
8. Forms

So if you want to migrate a space as well as an organization and its forms, you can export them in any order, but you need to import the space first, then the organization and finally the form.

Please note that services in particular contain large amounts of information that may be server-specific (such as endpoint URLs). When migrating services, you need to ensure the services are configured correctly for the target system by reviewing the services that have been imported. This is also true for other entities to a lesser extent (e.g. spaces need to have their context path adjusted).

Migrating an Organization

If needed, you can migrate an entire organization with its configuration including property types, metadata tags and document types across to another environment. After reimporting, the organization will be set up as it was on the original system (though the entity IDs may have changed). Please note that not all entities are included in the export/import process; for example, organization administrators are not exported along with the organization for security reasons. In addition, forms and associated data are not included in an organization export (they can be migrated separately).
Use this mechanism to migrate an organization to a system that doesn't have the organization yet or where organization-related properties are not up to date. It is also a good way to back up organization data without introducing dependencies on database IDs.

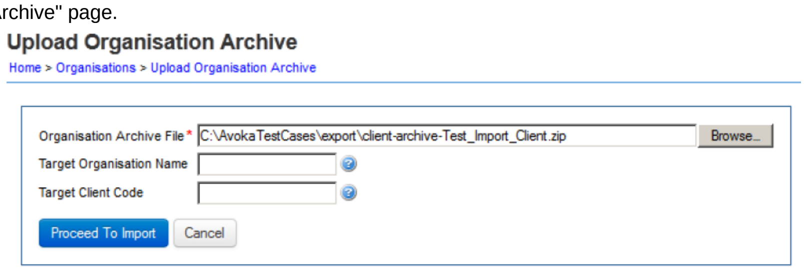
Exporting an Organization

To export an organization, go to "Forms > Organizations" and click the "Edit" icon (

) next to the organization. Click the "Export Organization" button on the "Organization" tab to download a ZIP file containing data describing the organization.

Importing an Organization

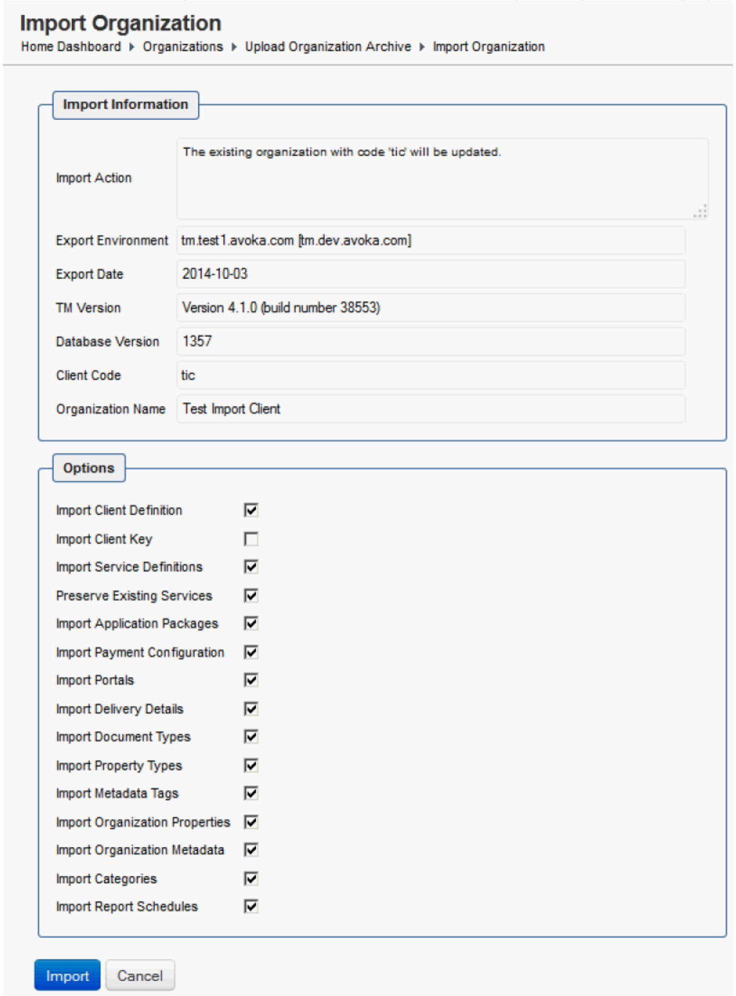
To import the organization on the target environment, go to "Forms > Organizations" and click the "Import" button. You will see the "Upload Organization Archive" page.



Importing an organization
Select the previously exported file using the "Browse" button.

If you want to create a new organization to import the data to, enter the desired organization name and client code into the "Target Organization Name" and "Target Client Code" fields. This can be useful when you want to rename or duplicate an organization during import.

Click "Proceed to Import". The file will be read and the "Import Organization" page will be displayed.



Confirming an organization to be imported

This page displays some information about the source system as well as the action that will be taken when you continue. Check the "Import Action" field specifically for any indication of import conflicts that may need to be resolved first.
You can choose what data shall be imported by changing the settings in the "Options" section. By default, the client key (a UUID uniquely identifying the organization) is not imported but will instead be regenerated if a new organization is created.
Click "Import" to import the organization data, and wait for the data to be integrated into the system. Please review the import message and record that are displayed to ensure all data was imported successfully. If any issues are found, a detail message will be added to the import record, so you can manually resolve the issue after importing.

Migrating an Application Package

While you can migrate forms and service definitions by themselves (which is discussed later in this chapter), it can be easier to create an application package containing a number of forms and organization specific services, and migrate the entire application package in one go. You still need to ensure the organization exists on the target system first (e.g. by importing it as described above), but apart from that, application packages can save you doing a lot of individual imports.

To learn more about how to create application packages, please see Organization Configuration (Manager v5.0)#_sec_setupapplicationpackages.

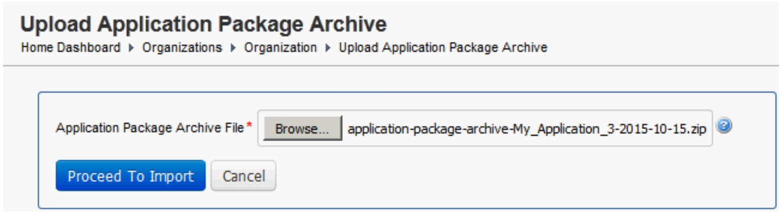
Exporting an Application Package

To export an application package, edit the organization and switch to the "Applications" tab. You have two options on what will be included in the export archive:

- export all services and forms with all form versions ( )
  This is the most comprehensive option. Use this when you need to migrate the entire set of form versions. However, if the forms contained in the application package have many versions, the export archive can become very large.
- export all services and the current version of each form ( )
  This is a good option if you simply need the current version of each form.

Importing an Application Package

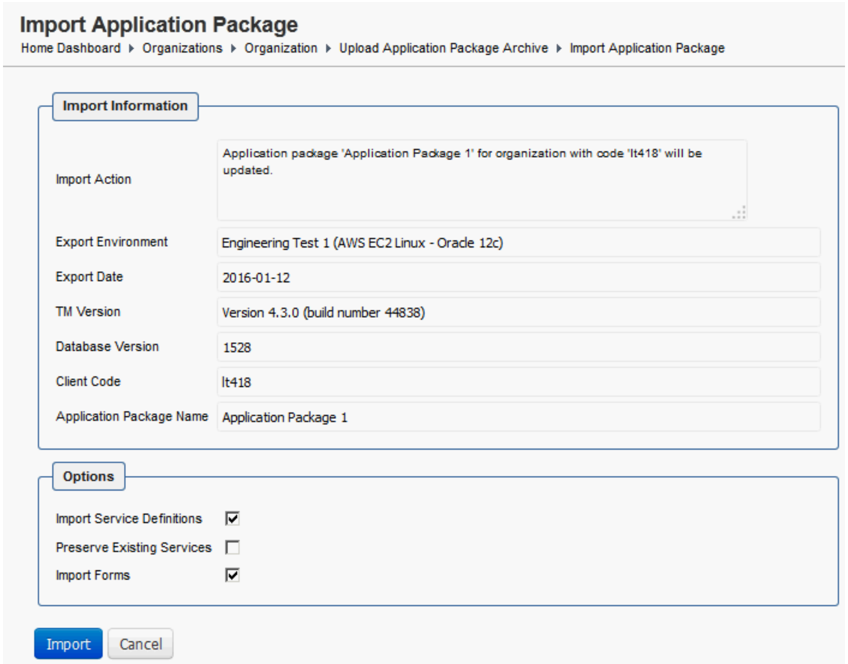
To import the application package on the target environment, go to "Forms > Organizations", edit the organization and switch to the "Applications" tab. Click the "Import" button. You will see the "Upload Application Package Archive" page as shown below.



Uploading an application package archive

Select the previously exported file using the "Browse" button.

Click "Proceed to Import". The file will be read and the "Import Application Package" page will be displayed.



Importing an application package

This page displays some information about the source system as well as the action that will be taken when you continue. Check the "Import Action" field specifically for any indication of import conflicts that may need to be resolved first.

You can choose what data shall be imported by changing the settings in the "Options" section. Mainly you can decide whether to import services and/or forms contained in the application package.
Additionally you can choose to import new services but leave any existing services alone by ticking "Preserve Existing Services" in addition to "Import Service Definitions".

Click "Import" to import the application package, and wait for the data to be integrated into the system. Please review the import message and record that are displayed to ensure all data was imported successfully. If any issues are found, a detail message will be added to the import record, so you can manually resolve the issue after importing.

Migrating a Form

If you are progressing a form to production, you will want to export the form and its configuration to the production server, but not necessarily transfer across all the organization data if the organization is already set up and up to date on the target system.
Transaction Manager allows you to export a single form or even a single form version and reimport it on a different Transaction Manager system. The process is very similar to exporting and reimporting an organization.

Exporting a Form and All Versions

You can export a form and all its form versions at once. This is best done if the form does not yet exist on the target system (and the entire version history is needed). If you need to migrate only a single form version, please refer to the following section.

Go to "Forms > Forms", and click the "Export Archive" icon ( ) next to the form to download a ZIP file containing data describing the form and all its versions.

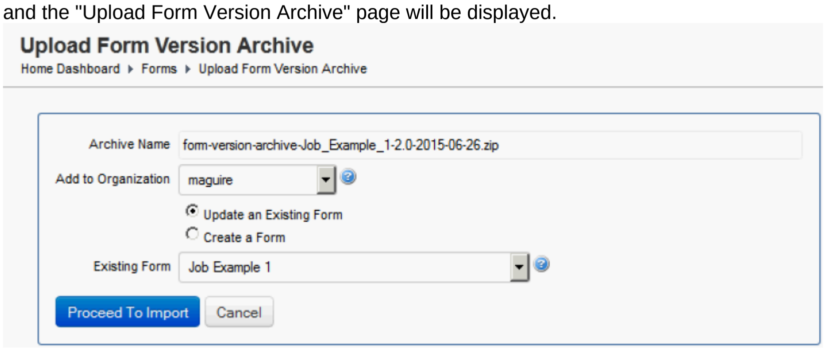
Exporting a Single Form Version

You can export a single form version if you would like to create or update just this version on the target system.

To export a form version, edit the form and switch to the "Form Versions" tab. Click the "Export Archive" icon ( ) next to the form version to download a ZIP file containing data describing the form and the chosen version.

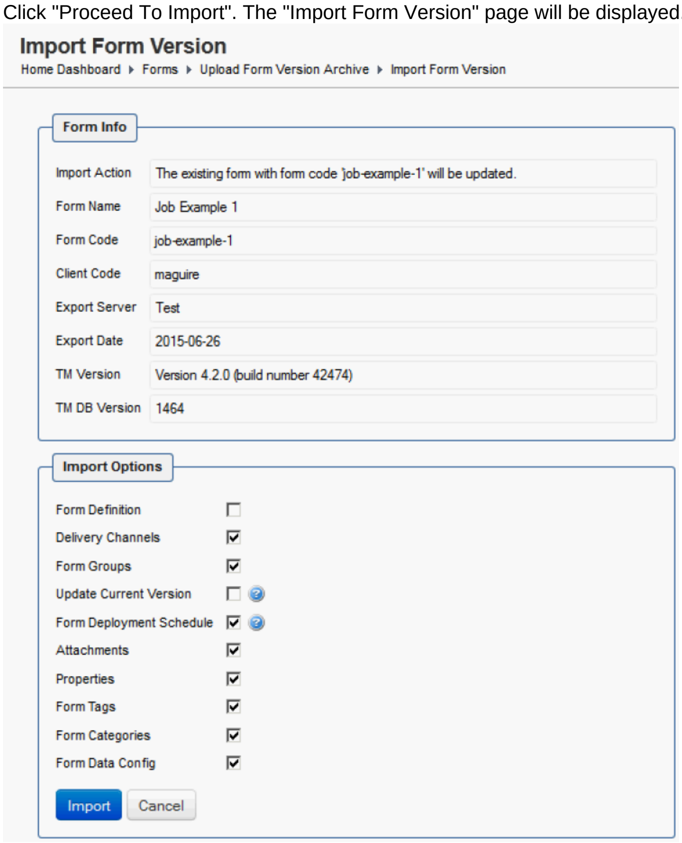
Importing a Form (One or Multiple Versions)

To import the form version archive on the target environment, go to "Forms > Forms" and click the "Import Form Version" button. The file will be read and the "Upload Form Version Archive" page will be displayed.



Uploading a form version archive

Select the previously exported file using the "Browse" button and click "Upload".
TM will scan the archive and load the target organization and form (which may not exist on the system you are importing to).
If you want to import the form to a specific organization, choose its client code from the "Add to Organization" field.
Now select whether you want to update an existing form or create a form.If you want to update a form, select it from the "Existing Form" dropdown. Note that if you are not updating the same form, be very careful as form versions are resolved by version number during import, and existing form versions will be overwritten if their version number matches the version number of a version in the import archive.If you want to create a form, enter a unique form code into the "New Form Code" field.
Click "Proceed To Import". The "Import Form Version" page will be displayed.



Importing a form

This page displays some information about the source system as well as the action that will be taken when you continue. Check the "Import Action" field specifically for any indication of import conflicts that may need to be resolved first.
You can choose what data shall be imported by changing the settings in the "Options" section.
Click "Import" to import the form data, and wait for the data to be integrated into the system. Please review the import message and record that are displayed to ensure all data was imported successfully. If any issues are found, a detail message will be added to the import record, so you can manually resolve the issue after importing.

Migrating Services

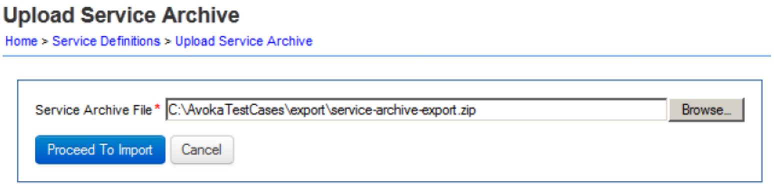
Transaction Manager allows you to export and import the set of service definitions and service connections. This allows you to configure and test your services on one server and then export them to your other environments.
Note: Only service connections that are referenced by service connections will be exported.

Exporting Services

To export the set of service definitions and the service connections they use, go to "System > Service Definitions" and click the "Export" button.
You can also export individual services, or filter the list of services by an application package and then click "Export" to export all services in the package.
A ZIP file containing information about the chosen service definitions, their parameters and service connections will be created and offered to you for download.

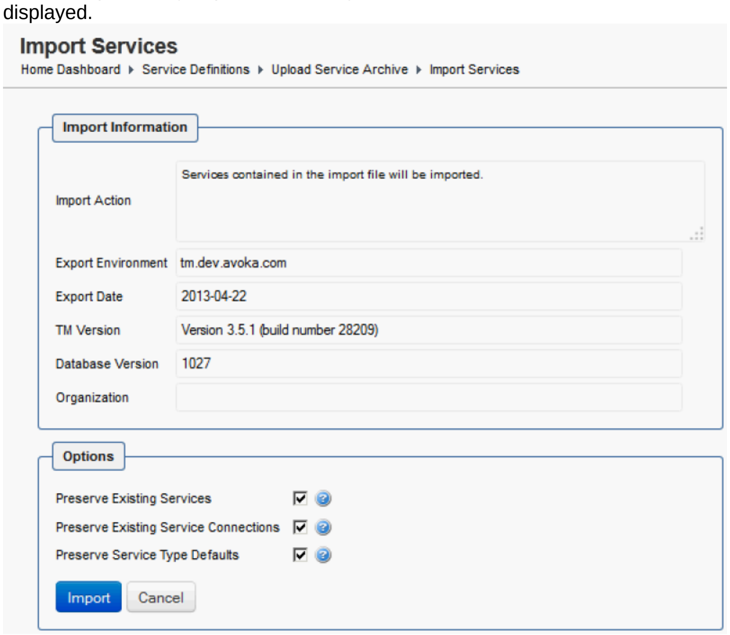
Importing Services

To import the set of services on the target environment, go to "System > Service Definitions" and click the "Import" button. You will see the "Upload Service Archive" page as shown below.



Importing the set of services

Select the previously exported file using the "Browse" button and click "Proceed To Import". The file will be read and the "Import Services" page will be displayed.



Import services confirmation page

This page displays some information about the source system as well as the action that will be taken when you continue. Check the "Import Action" field specifically for any indication of import conflicts that may need to be resolved first.

You can select several options to preserve data on the target system: Tick "Preserve Existing Services" to leave all service definitions on the target system intact. Tick "Preserve Existing Service Connections" to leave all service connections on the target system intact. Finally, tick "Preserve Service Type Defaults" to ignore the service type default flag for imported services; otherwise, imported services may be marked as the default for their type and existing services demoted.

Click "Import" to import the service data, and wait for the data to be integrated into the system. Please review the import message and record that are displayed to ensure all data was imported successfully. If any issues are found, a detail message will be added to the import record, so you can manually resolve the issue after importing.

Migrating Groups

You can use groups to define a set of users with certain privileges (mainly form access). TM allows you to export the full set of groups or a single group for migration to other servers.

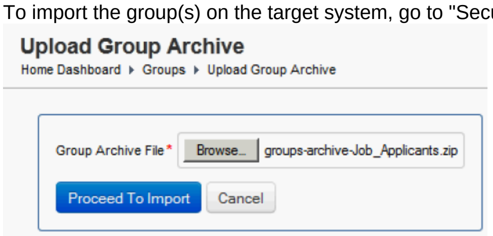
Note that the set of users associated with the group and the set of forms using the group is not part of the export. Only basic group settings (name, description) and user access control settings will be exported.

Exporting Groups

To export the full set of groups, go to "Security > Groups" and click the "Export" button. To export a single role, click the "Export Archive" icon ().In either case, a ZIP file containing the group data will be created and offered to you for download.

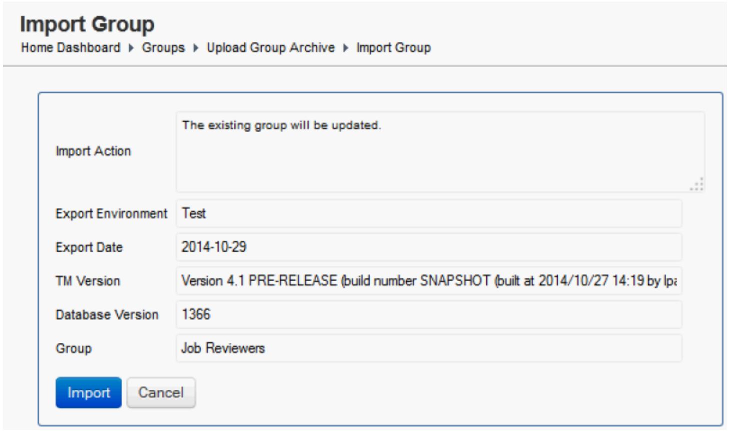
Importing Groups

To import the group(s) on the target system, go to "Security > Groups" and click "Import". You will see the "Upload Group Archive" page.



Importing a group

Select the previously exported file using the "Browse" button and click "Proceed To Import". The file will be read and the "Import Group" page will be displayed.



Import group confirmation page

This page displays some information about the source system as well as the action that will be taken when you continue. Check the "Import Action" field specifically for any indication of import conflicts that may need to be resolved first.

Click "Import" to import the group, and wait for the data to be integrated into the system. Please review the import message and record that are displayed to ensure all data was imported successfully. If any issues are found, a detail message will be added to the import record, so you can manually resolve the issue after importing.

Migrating Roles

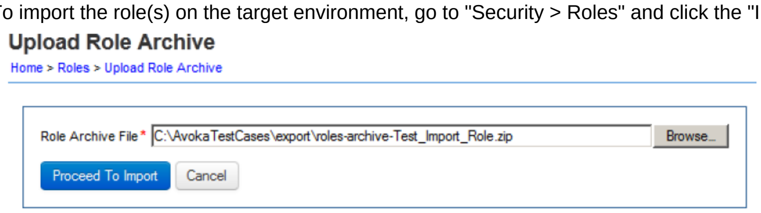
To save yourself the work of setting up the same set of roles on several servers, Transaction Manager allows you to export and import the set of roles, or individual roles. Please note that while the set of permissions is included in the export, role membership and other user-related data is not.

Exporting Roles

To export the full set of roles, go to "Security > Roles" and click the "Export Roles" button. To export a single role, click the "Export Archive" icon ( ).In either case, a ZIP file containing the role and permission data will be created and offered to you for download.
Note: To avoid lockout issues on the target system, the "Administrator" role will not be including when exporting the set of roles. The role can still be exported individually, however.

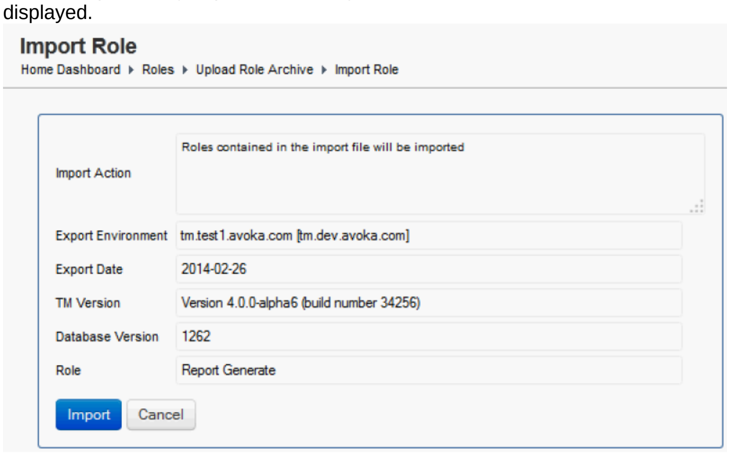
Importing Roles

To import the role(s) on the target environment, go to "Security > Roles" and click the "Import" button. You will see the "Upload Role Archive" page.



Importing the set of roles

Select the previously exported file using the "Browse" button and click "Proceed To Import". The file will be read and the "Import Role" page will be displayed.



Import roles confirmation page

This page displays some information about the source system as well as the action that will be taken when you continue. Check the "Import Action" field specifically for any indication of import conflicts that may need to be resolved first.

Click "Import" to import the role and permission data, and wait for the data to be integrated into the system. Please review the import message and record that are displayed to ensure all data was imported successfully. If any issues are found, a detail message will be added to the import record, so you can manually resolve the issue after importing.

Please note that importing roles and permission will not remove existing roles that are not contained in the import archive. Roles contained in the import archive will be reconfigured with exactly the permissions contained in the import archive. This means that roles may become more restrictive or more permissive depending on the contents of the import archive.

Migrating Security Managers

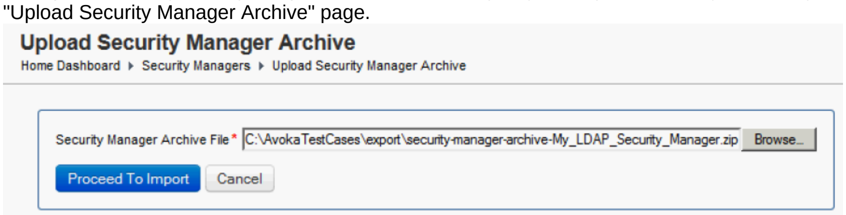
Transaction Manager allows you to export the set of security managers (or a single security manager) and import them to a different TM instance. This saves you configuration work and ensures the security managers are configured in the same way on multiple systems.

Exporting Security Managers

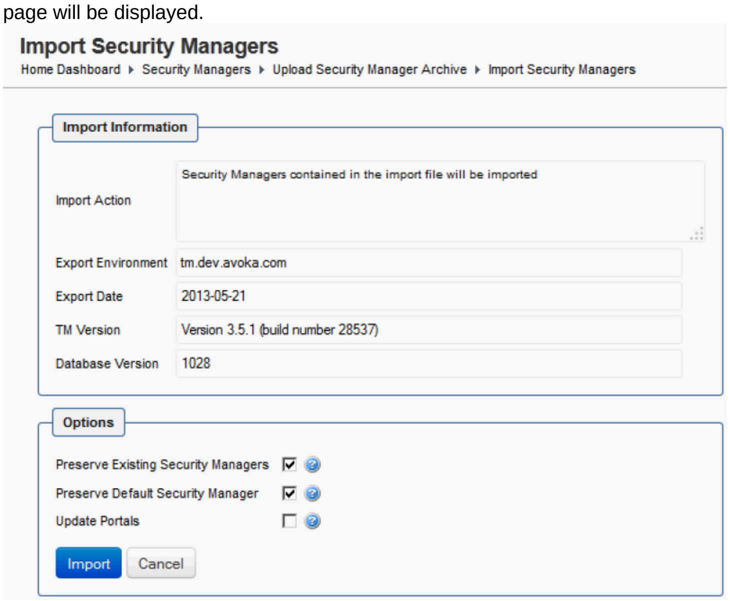
Go to "Security > Security Managers". To export the full set of security managers, click the "Export" button. To export a single security manager, click the "Export Archive" icon ( ) next to the security manager.In either case, a ZIP file containing the configuration data will be created and offered for download.

Importing Security Managers

To import a security manager export archive on the target system, go to "Security > Security Managers" and click the "Import" button. You will see the "Upload Security Manager Archive" page.



Uploading a security manager archive
Select the previously exported file using the "Browse" button and click "Proceed To Import". The file will be read and the "Import Security Managers" page will be displayed.



Importing a security manager
If you tick "Preserve Existing Security Managers", all security managers existing on the target system will not be modified, even if security managers with matching names exist in the export archive. New security managers will still be imported.

If you tick "Preserve Default Security Manager", the system default security manager will not be modified during import. None of the imported security managers will be promoted to the system default security manager.

If you tick "Update Portals", the import will also attempt to resolve all spaces that were associated with a security manager on the source system and ensure they are set to use the same security manager on the target system (note: space matching is done by name). Otherwise, spaces on the target system will use the same security manager as before.

Click "Import" to import the security managers in the archive with the settings you have selected.After the import process has completed, please review the import record that are displayed to ensure all data was imported successfully. If any issues are found, a detail message will be added to the import record, so you can manually resolve the issue after importing.

Migrating Metadata Tag Definitions

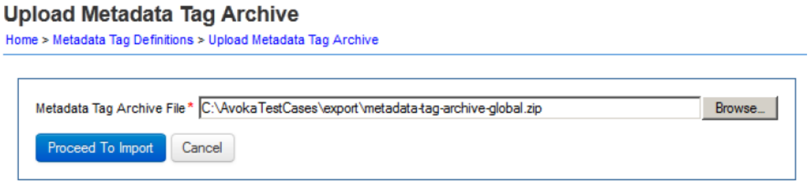
To save yourself the work of setting up the same set of metadata tag definitions on several servers, Transaction Manager also allows you to export and import the set of global metadata tag definitions and associated metadata tag list values.Please note that metadata tags created for a specific organization will not be exported (instead, these metadata tags are included when the organization itself is exported). In addition, concrete metadata values will not be part of the export.

Exporting Metadata Tags

Go to "Forms > Metadata Tag Definitions" and click the "Export Global Tags" button to download a ZIP file containing data describing the set of global metadata tags.

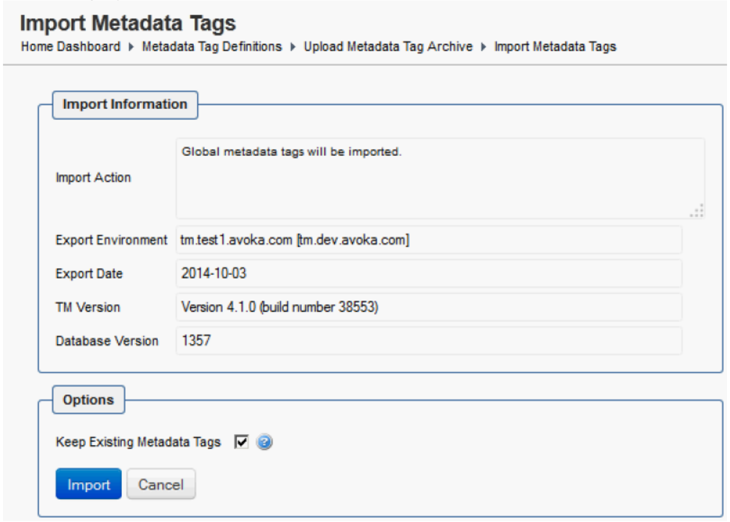
Importing Metadata Tags

To import the metadata tag definitions on the target environment, go to "Forms > Metadata Tag Definitions" and click the "Import" button. You will see the "Upload Metadata Tag Archive" page.



Importing the set of metadata tags

Select the previously exported file using the "Browse" button and click "Proceed To Import". The file will be read and the "Import Metadata Tags" page will be displayed.



Import metadata confirmation page

This page displays some information about the source system as well as the action that will be taken when you continue. Check the "Import Action" field specifically for any indication of import conflicts that may need to be resolved first.
By default, all existing global metadata tags are kept. You can untick the "Keep Existing Metadata Tags" checkbox to cause any existing global metadata tags that are not contained in the import archive to be deleted (effectively restoring the set of tags in the import file).

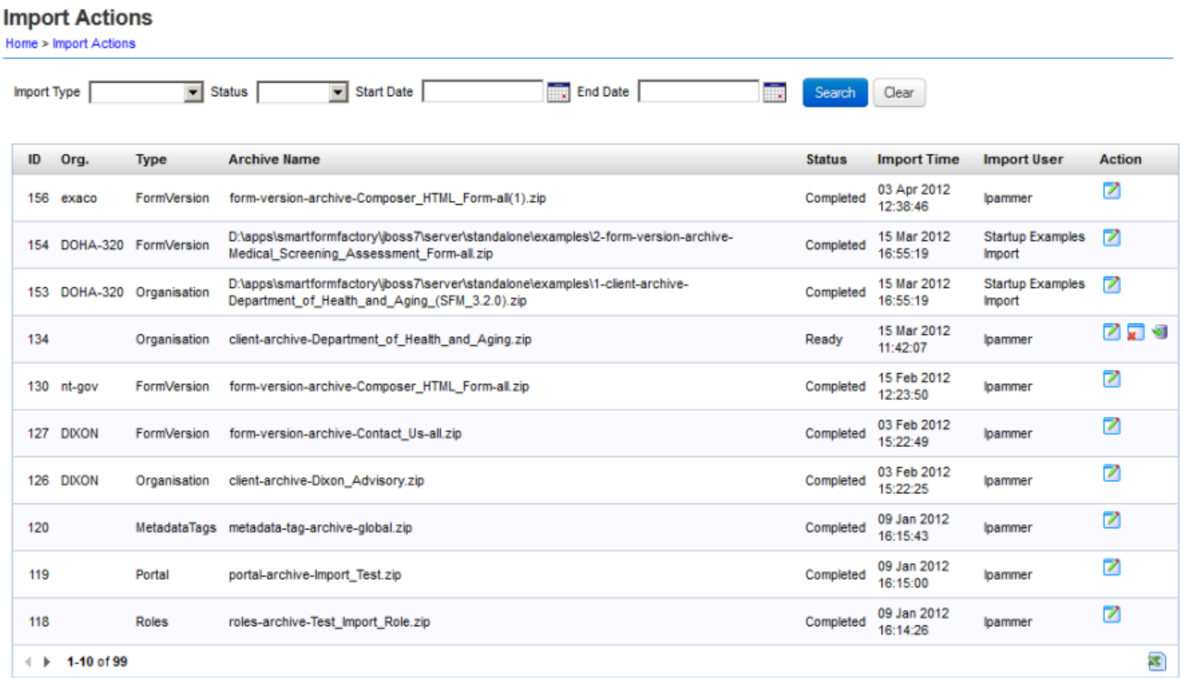
Click "Import" to import the metadata tags, and wait for the data to be integrated into the system. Please review the import message and record that are displayed to ensure all data was imported successfully. If any issues are found, a detail message will be added to the import record, so you can manually resolve the issue after importing.

The Import Log

To facilitate tracking and auditing, Transaction Manager keeps a log of all import operations that have been performed, and records the outcome of each import as well as any warning messages.

Viewing the Import Log

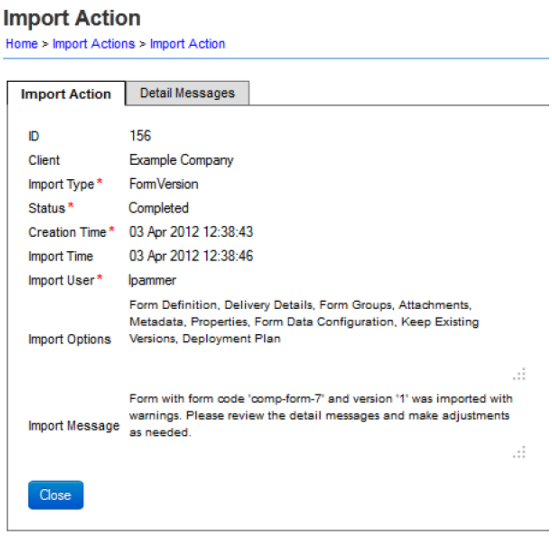
To browser the import log, go to "System > Import Log". You will see the list of import operations.
You can filter the list by import type (e.g. "Organization"), import status and the import completion date, and export the search results to Excel by clicking the "Export Data" link below the table.



ID	Org.	Type	Archive Name	Status	Import Time	Import User	Action
156	exaco	FormVersion	form-version-archive-Composer_HTML_Form-all(1).zip	Completed	03 Apr 2012 12:38:46	lpammer	
154	DOHA-320	FormVersion	D:\apps\smartformfactory\jboss7\server\standalone\examples\2-form-version-archive-Medical_Screening_Assessment_Form-all.zip	Completed	15 Mar 2012 16:55:19	Startup Examples Import	
153	DOHA-320	Organisation	D:\apps\smartformfactory\jboss7\server\standalone\examples\1-client-archive-Department_of_Health_and_Aging_(SFM_3.2.0).zip	Completed	15 Mar 2012 16:55:19	Startup Examples Import	
134		Organisation	client-archive-Department_of_Health_and_Aging.zip	Ready	15 Mar 2012 11:42:07	lpammer	
130	nt-gov	FormVersion	form-version-archive-Composer_HTML_Form-all.zip	Completed	15 Feb 2012 12:23:50	lpammer	
127	DIXON	FormVersion	form-version-archive-Contact_Us-all.zip	Completed	03 Feb 2012 15:22:49	lpammer	
126	DIXON	Organisation	client-archive-Dixon_Advisory.zip	Completed	03 Feb 2012 15:22:25	lpammer	
120		MetadataTags	metadata-tag-archive-global.zip	Completed	09 Jan 2012 16:15:43	lpammer	
119		Portal	portal-archive-Import_Test.zip	Completed	09 Jan 2012 16:15:00	lpammer	
118		Roles	roles-archive-Test_Import_Role.zip	Completed	09 Jan 2012 16:14:26	lpammer	

Browsing the import log

You can click the "Edit" ( ) icon to view an individual import entry.



Viewing details for an import entry

The "Import Action" tab shows general information about the import and whether it has been completed. To view a list of warning messages generated during the import, switch to the "Detail Messages" tab.

Viewing detail messages for an import entry

Operations (Manager v5.0)

This chapter explains how to keep track of what is happening on Transaction Manager: what submissions have been made, what their status is, whether there were any problems, what changes have been made to the Transaction Manager configuration and by whom and how you can post and view important notifications. This knowledge will help you resolve issues raised by organizations or users.

Search

Tracking Submissions

An integral part to understanding what is happening on Transaction Manager is knowing what forms users are submitting. This information can be found on the submission log, with additional data available via the request log. Many questions asked by users can be answered by investigating these logs.

The Submission Life Cycle

A submission object is created in TM as soon as the user opens a form (or even earlier for task submissions that are assigned to users to complete). As the user performs actions (e.g. online save, submission, payments), the submission is updated by TM. Even after the user has completed their work, there may be additional steps in the submission process, such as virus scanning and delivery.
Therefore, the submission life cycle can consist of many stages. This section lays out all the tasks that could potentially be part of the submission process.
User-related actions:

- Render form
  Transaction Manager status: Entries are created in the request log and submission log. The submission form status is "Opened".
  Next actions: save, save offline, submit
- Save form to Transaction Manager
  Transaction Manager status: The submission form status is "Saved".
  Next actions: save to Transaction Manager, save offline, submit
- Save form offline
  Transaction Manager status: The submission record in TM is not updated.
  Next actions: submit
- Mark form as abandoned, effectively canceling the submission
  Transaction Manager status: The submission form status is "Abandoned".
  Next actions: abandonment delivery if configured
- Submit form
  Transaction Manager status: The submission form status is "Submitted" if further user tasks need to be done, otherwise "Completed".
  Next actions: authenticate, verify email address, add attachments, make payment, sign
- Authenticate / acknowledge submission
  Note: This action is needed only when a form that requires authentication is submitted while the user is not logged on (this is commonly the case after a form has been saved offline).
  Transaction Manager status: The submission form status is "Submitted", the authentication status is "Completed" once the user has acknowledged the submission.
  Next actions: add attachments, make payment, sign
- Reject submission
  Transaction Manager status: The submission form status is "Submitted", the authentication status is "Cancelled" once the user has rejected the submission.
  Next actions: None - as the user has rejected the submission, it cannot progress further.
- Verify email address
  Note: This action happens in parallel with other actions such as adding attachments and payment. At submission time, an email is sent to the user asking them to click on a link to verify their email address if submission email verification is enabled. Once they do, this step will complete. Refer to Form Configuration (Manager v5.0)#_sec_submissionemailverification for more on submission email verification.
  Transaction Manager status: The submission form status is "Submitted", the email verification status is "Completed" once the user has verified their email address.
  Next actions: add attachments, make payment, sign
- Complete attachments
  Transaction Manager status: The submission form status is "Submitted" or "Completed", the attachment status is "Completed".
  Next actions: make payment, sign
- Make payment
  Transaction Manager status: The submission form status is "Submitted" or "Completed", the payment status is "Completed".
  Next actions: sign
- Accept electronically signed submission
  Transaction Manager status: The submission form status is "Submitted" or "Completed", the signature status is "Completed" and the signature type is "ESignature".
  If all the previous steps have been completed, the submission is now complete from the user's perspective.
- Reject electronically signed submission
  Transaction Manager status: The submission form status is "Submitted", the signature status is "Cancelled" and the signature type is "ESignature".
  Next actions: None - as the user has rejected the submission, it cannot progress further. If this is configured, the submitted data may be delivered as part of abandonment delivery.
- Sign the form
  Transaction Manager Status: The submission form status is "Submitted" or "Completed", the signature status is "Completed" and the signature type is "Wet".
  If all the previous steps have been completed, the submission is now complete from the user's perspective.

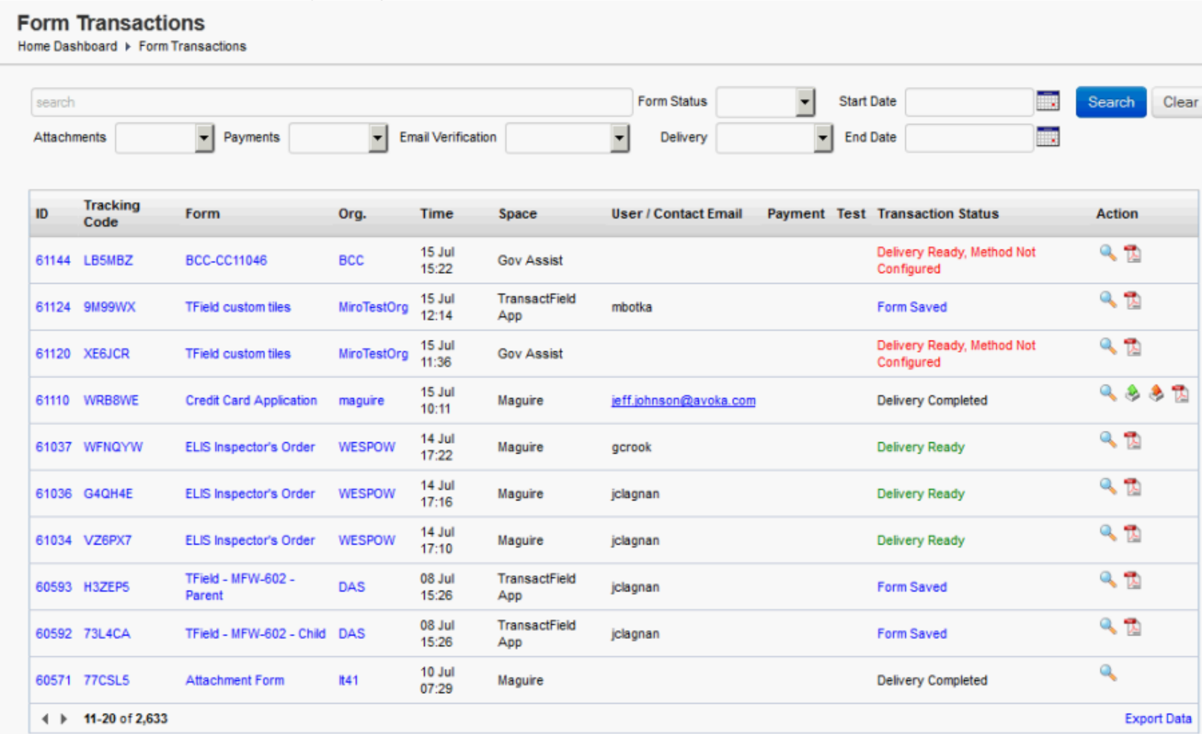
Please note that some of these actions may not occur or happen in a different order (e.g. attachments and payment can generally completed in either order).
Tasks performed by Transaction Manager:

- Receipt Render
  This is done for a form that has been submitted, either when the user requests a receipt or otherwise as part of a routine job. If a form requires eSignatures, the receipt that is generated will contain the electronic signature of the organization hosting the form.
- Virus scan
  This is done for submissions with attachments, if a virus scanning service is configured. If applicable, submissions are not considered complete until virus scanning has been done.
- Delivery
  This is the final step and happens only once the user has completed the submission and all other Transaction Manager tasks have also been completed successfully.

Please note that during certain stages, the submission can go into an error state. An example for an error to be corrected by the user is a payment error, which is often caused by invalid credit card details. An example for an error occurring during Transaction Manager processing is a delivery error, which might be caused by invalid delivery email addresses.
Transaction Manager will indicate the status, including errors, in the submission log.

Browsing and Filtering Submissions

To view the list of all submissions, go to "Operations > Form Transactions".


Form Transactions
Home Dashboard ▸ Form Transactions

search | Form Status | Start Date | Search | Clear
Attachments | Payments | Email Verification | Delivery | End Date

ID	Tracking Code	Form	Org.	Time	Space	User / Contact Email	Payment	Test	Transaction Status	Action
61144	LB5MBZ	BCC-CC11046	BCC	15 Jul 15:22	Gov Assist				Delivery Ready, Method Not Configured	
61124	9M99WX	TField custom tiles	MiroTestOrg	15 Jul 12:14	TransactField App	mbotka			Form Saved	
61120	XE6JCR	TField custom tiles	MiroTestOrg	15 Jul 11:36	Gov Assist				Delivery Ready, Method Not Configured	
61110	WRB8WE	Credit Card Application	maguire	15 Jul 10:11	Maguire	jeff.johnson@avoka.com			Delivery Completed	
61037	WFNQYW	ELIS Inspector's Order	WESPOW	14 Jul 17:22	Maguire	gcrook			Delivery Ready	
61036	G4QH4E	ELIS Inspector's Order	WESPOW	14 Jul 17:16	Maguire	jclagnan			Delivery Ready	
61034	VZ6PX7	ELIS Inspector's Order	WESPOW	14 Jul 17:10	Maguire	jclagnan			Delivery Ready	
60593	H3ZEP5	TField - MFW-602 - Parent	DAS	08 Jul 15:26	TransactField App	jclagnan			Form Saved	
60592	73L4CA	TField - MFW-602 - Child	DAS	08 Jul 15:26	TransactField App	jclagnan			Form Saved	
60571	77CSL5	Attachment Form	lt41	10 Jul 07:29	Maguire				Delivery Completed	

11-20 of 2,633 | Export Data

The submission log
You can filter this list by organization, the submission ID or key, a substring of the form name, receipt / tracking number, the login name or email address of the user who made the submission, submission date, as well as form, attachment, payment, email verification and delivery status.
The key column is "Transaction Status" which shows a message indicating the current status (see the previous section for details about the submission life cycle). Depending on the status, various actions will be available, which are described in greater detail below.

Submission-Related Actions

For each submission, Transaction Manager allows you to view additional information or take action as appropriate. These actions are displayed in the "Action" column (see screenshot above). Depending on the submission's current status, different actions will be available.
The available actions are:

- View Details ( )
  Availability: Always
  Opens the "View Form Transaction" page (see _sec_submissiondetails) for the selected submission
- Open Saved Form ( )
  Availability: Saved forms
  Opens the saved form, which can be completed by the administrator on behalf of the user (e.g. as part of a support request). As this is a sensitive operation, this action is only available to administrators whose roles include the permission "Submission Save Edit".
- View Receipt ( )
  Availability: Submitted forms, unless submission data is unavailable
  Opens the form receipt.
- Form Attachments ( )
  Availability: Submitted forms for which attachments are outstanding
  Opens the "Form Attachments" page on the user space. You can send the URL to anonymous users who have lost access to their submission (e.g. by closing the browser window).
- Credit Card Payment Page ( )
  Availability: Submitted forms for which credit card payment is outstanding
  Opens the payment page on the user space where credit card payment can be made. This can be used when a space user contacts support to assist them with their credit card payment. The support administrator can then access the credit card payment page and enter the user's details.
- Retry Delivery ( )
  Availability: Completed forms where delivery has not been completed and submission data is available

Attempts to deliver the submission to the current delivery details set up in the form. This can be used to retry delivery, e.g. after changing delivery details. For submissions with email delivery, this action can also be used after delivery has been completed.

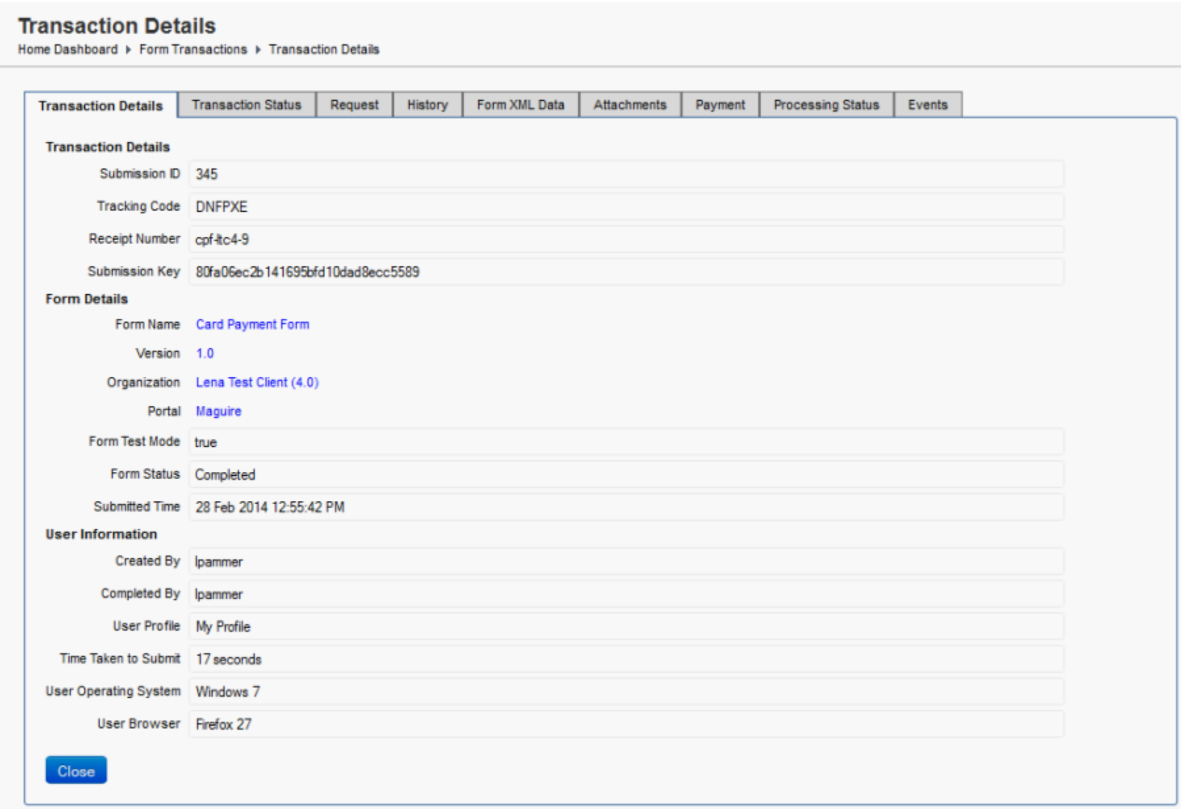
- Retry Delivery from Scratch ( )
  Availability: Completed forms where delivery has not been completed and submission data is available
  Attempts to deliver the submission to the current delivery details set up in the form after deleting all existing delivery checkpoints for the submission. If no delivery checkpoints exist, this is equivalent to the retry delivery option above.
- View Email Delivery Confirmation ( )
  Availability: Submissions with secure email delivery waiting for acknowledgment, where submission data is available
  Opens the "Email Delivery" page that allows authorized administrators to acknowledge delivery of a submission. This only applies if the delivery method is "Email Secure" (see Delivery Configuration (Manager v5.0)#_sec_secureemaildelivery).
  Note that this works only if the submission data has not been deleted.

These actions can be very valuable for finding out what happened to a submission and make the submission log the best starting point when investigating submission-related issues.

Submission Details

Each submission is described in more detail on its own page, including the various statuses, the submitter's browser and PDF viewer version, attachments, payment, previous submission stages, data extracted from the submission, errors and events associated with the submission, the current processing status at the organization, and optionally auditing information.

To view this page, go to "Operations > Form Transactions", find the submission and click the "View Details" ( ) action. You will be directed to the "View Form Transaction" page for that submission.



Viewing submission details

Basic Submission Details

The "Transaction Details" tab shown above lists basic submission data. This section describes some fields of interest.
Submit Key: An alternate key uniquely identifying the submission. This key, as opposed to the submission ID, is not generated by the database and will therefore persist through database migrations.
Tracking Code: The tracking code is a short random alphanumeric code that is generated when the form is opened. By default, it is displayed to the user as a reference number for the submission. See Form Configuration (Manager v5.0)#_sec_trackingcode for more on tracking codes.

Receipt Number: Another unique identifier for the submission, the receipt number is generated at save/submission time and can be displayed on the receipt. The receipt number can be generated according to your specifications; please refer to Form Configuration (Manager v5.0)#_sec_trackingcode fo r more information. You can also configure your form to use the receipt number in place of the tracking code.

Form Test Mode: Indicates whether the submission was made when the form was in test mode or not. This can be used in reporting as well as in determining which set of delivery details will be used (test or production).

Task Key: The task key of the task associated with this submission, if applicable (see Tasks (Manager v5.0) for more on tasks).

Form XML Hash: A hash value calculated from the submitted XML data. This is used during web service delivery to verify that submission data has been transferred correctly. The hash is kept even if the XML data for a submission has been deleted.

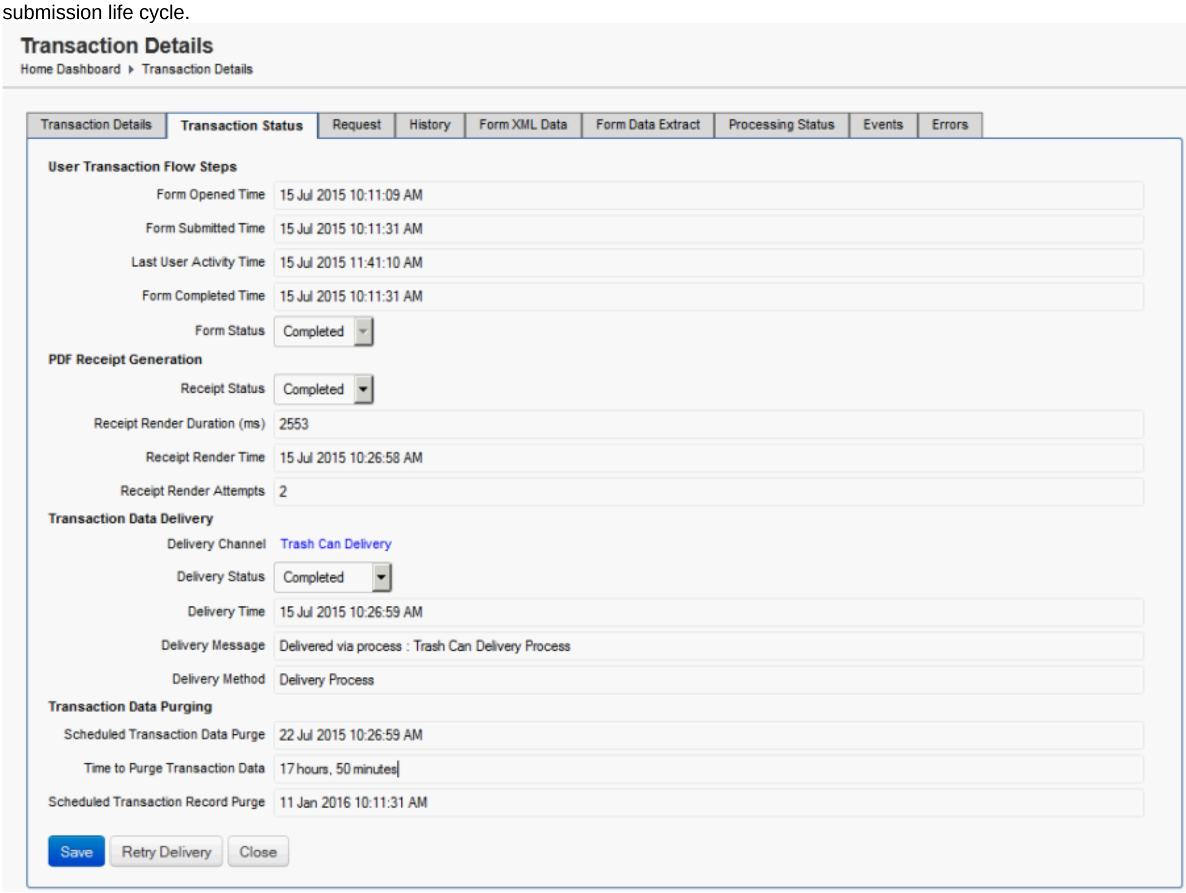
User Information section: This section displays important information about the user(s) involved in the submission. The information displayed depends on submission status and history.

Contact Email Address: The user's email address. This is automatically extracted from the submitted XML data if the configuration mapping "Contact Email Address" is set for the form data configuration (see Form Version Configuration (Manager v5.0)#_sec_configurationmappings).

Offline Submission (PDF forms only): This field indicates whether a submission was made via the browser (online) or via a form that was saved to the user's computer (offline).

Submission Status Values

The "Transaction Status" tab contains all status variables relevant to the submission. See _sec_submissionlifecycle for an outline of the basic submission life cycle.



Viewing the submission status

In addition, this tab lists some additional information about delivery, whether or not the submission data has been deleted and the current processing status (see _sec_submissionprocessingstatus).
If the submission has not yet been completed (form status is "Assigned", "Saved" or "Submitted"), administrators can set the status to "Abandoned". This can be used to manually expire a submission and prevent the user from accessing it again.
Transaction Manager allows you to manually mark an outstanding payment to be completed. This should be done only if payment for a submission has been made outside of Transaction Manager. To mark payment as completed, select "Completed" from the "Payment Status" dropdown and click "Save". An entry will be added to the event and audit logs.

Similarly, you can mark submission attachments as "Completed" if required. If the form status is "Submitted" and attachments are outstanding, you can change the value of the "Attachments Status" dropdown to "Completed" and save.
You can also change the status of submission email verification (if applicable). Submission email verification means that an email is sent to the user's email address (as entered into the form) and the user must click on a link in this email to complete the submission (see also _sec_submissionemailverification).You can manually set submission email verification to completed or reset it to required status to restart the email verification process.

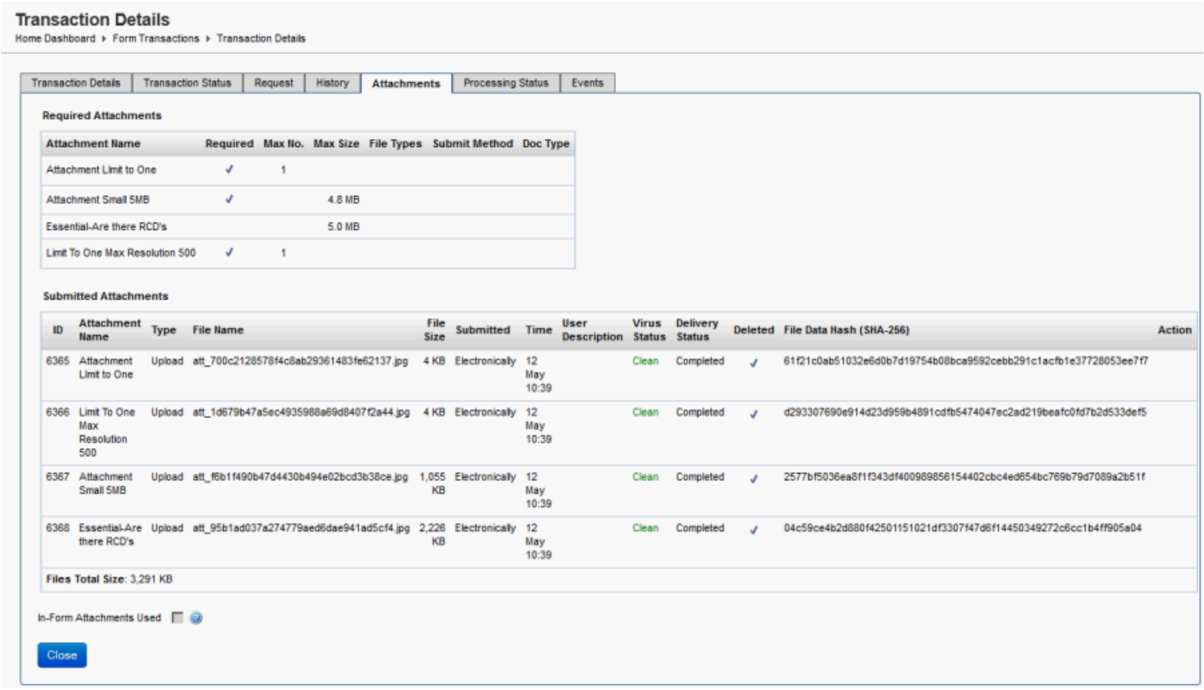
If the receipt needs to be regenerated (e.g. due to errors in the receipting process), you can manually change the receipt status from "Completed" to "Ready" and save. If the submission data is still available, this will cause TM to regenerate the receipt. Once this is done, you can then choose to redeliver the submission if necessary (see below).

Transaction Manager also allows you to change the delivery status of a submission. This should be done only for three purposes: Marking a submission as undeliverable, marking delivery as completed, and marking delivery as ready so a submission is delivered a second time. Note also that marking a submission as undeliverable affects only submission delivery, and should not be used in other scenarios.
Note that you cannot progress submissions which are not yet ready for delivery, and that you cannot modify the delivery status of submissions for which no submission data is available (e.g. because it has already been purged).

To change a submission's delivery status, choose the new status from the "Delivery Status" dropdown and save. The status will be updated, and the delivery message will indicate that the status was manually modified. In addition, an entry will be added to the audit log (entity name "Submission").
For more options on submission delivery monitoring and bulk status changes, please refer to Delivery Configuration (Manager v5.0)#_monitordelivery.

Attachments

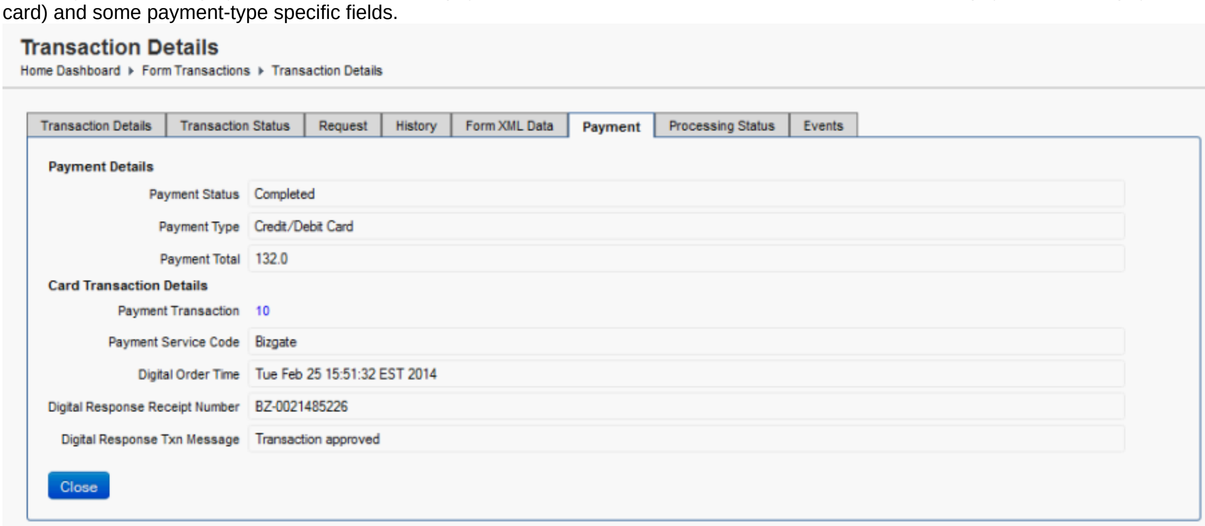
The "Attachments" tab shows the set of attachments associated with the form and allows you to view individual attachments. Please refer to Managing Form Attachments (Manager v5.0)#_sec_viewattachmentsforsubmission for an in-depth description.



Viewing the attachments for a submission

Payment

The "Payment" tab displays information about the payment associated with the submission. This includes the payment status, payment type (e.g. credit card) and some payment-type specific fields.

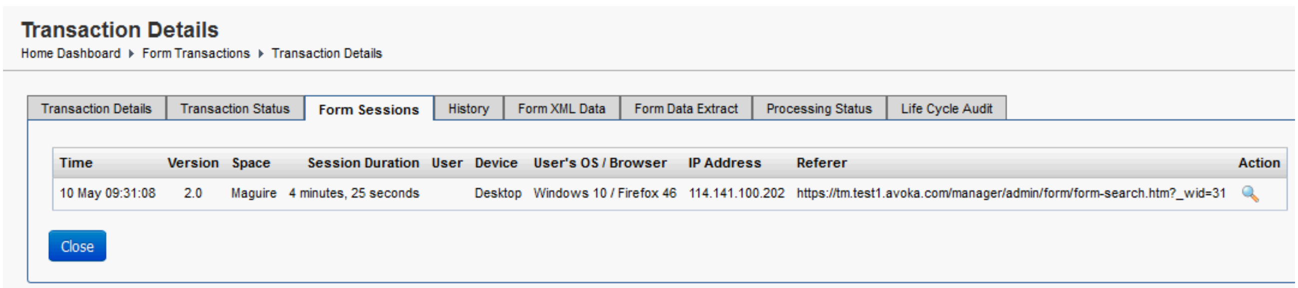


Submission payment status

Form Sessions

The "Form Sessions" tab shows the set of form requests made as part of this submission. One request is made when the submission is started, and additional requests may occur as the submission is saved and reopened.

Each request can be inspected by clicking the "View" link ( ). See section _sec_viewrequestlog on the information available for each request.

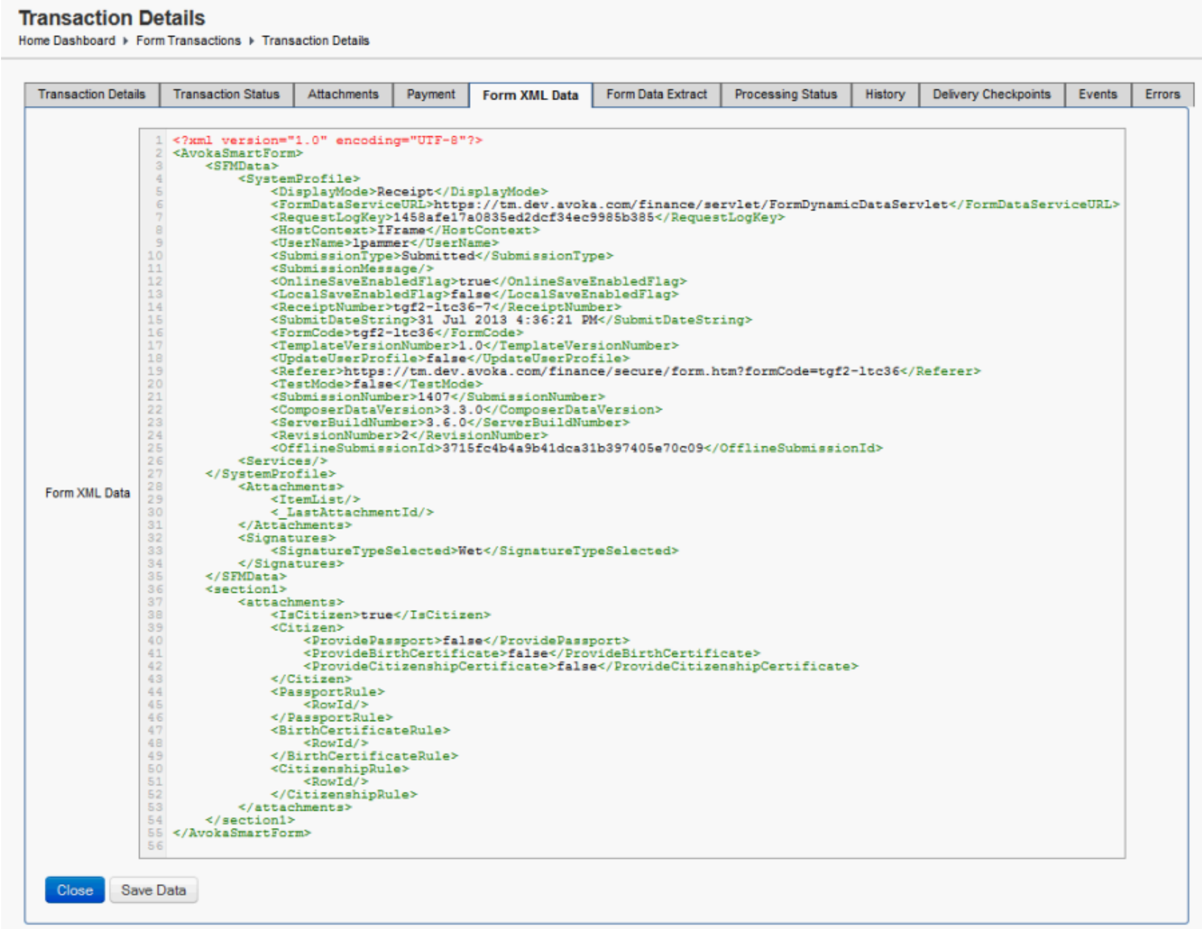


The form sessions for a submission

Form XML Data

The "Form XML Data" tab displays the XML data submitted by the user. Under very specific circumstances (the form has been completed by the user, form data has not yet been purged, and an error is preventing receipt generation or delivery), the XML data can be edited by administrators with the

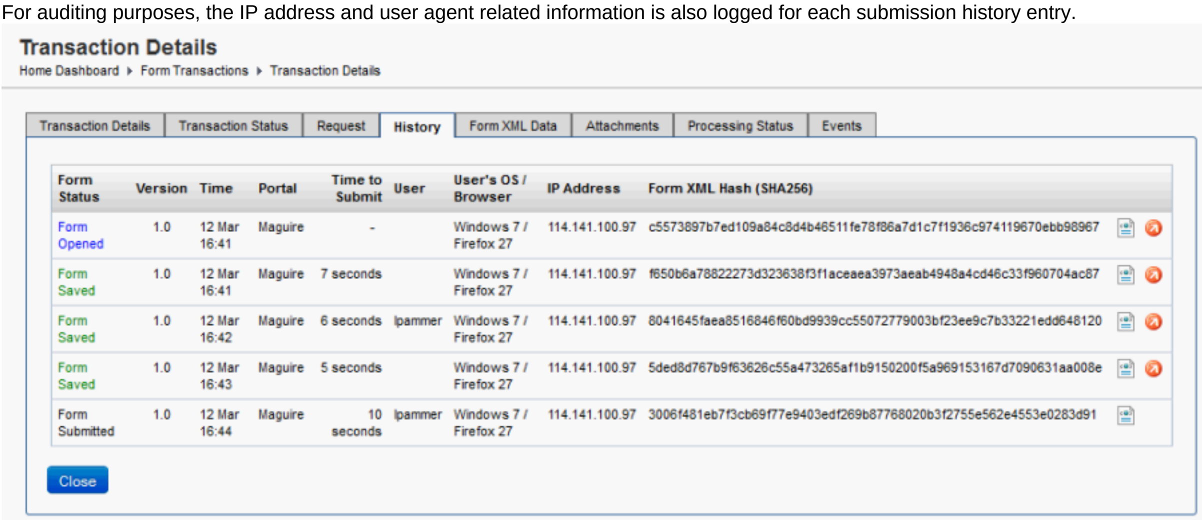
appropriate permissions. This can be useful if the user has entered characters that cause issues with back end processing, which the administrator can then remove to allow delivery to proceed.
Because viewing and editing the form data is a very sensitive operation, permissions are used to control access to the tab ("Submission Xml Data View" for read-only access, "Submission Xml Data Edit" to be able to modify the XML).In addition, whenever an administrator views or changes the data, TM logs the operation to the audit log. When an administrator changes the data and saves the changes, the resulting XML is verified and stored as a new submission history entry (see following section on the submission history), preserving the data previously submitted by the user.



Viewing the submitted XML

Submission History

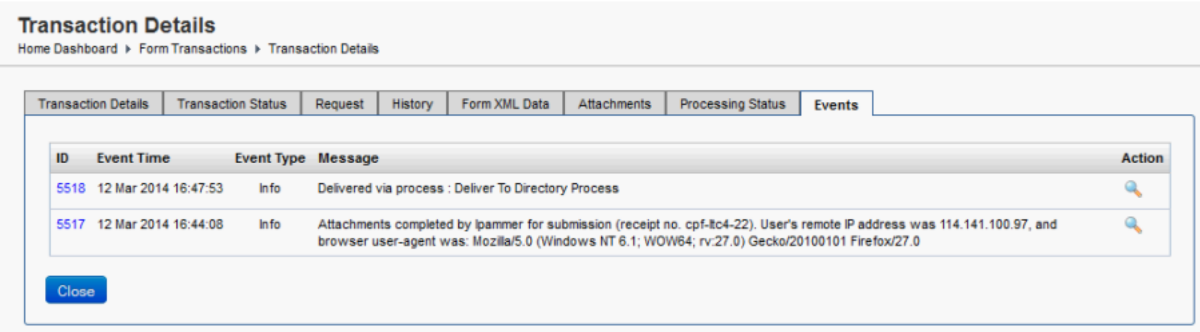
The "History" tab shows the list of all versions of a submission. A version is created when a submission assigned (task submission only), when the user opens a form, every time a submission is saved to Transaction Manager and, finally, when it is submitted. For each version, the submission XML is stored and can be accessed by clicking on the "View XML" icon ( ). For each version but the current one, you can also click the "View XML Diff" icon ( ) to see how that version is different to the current XML.
For auditing purposes, the IP address and user agent related information is also logged for each submission history entry.



The version history of a submission

Submission Events and Errors

In addition, you can see any errors or events that were logged against the submission by navigating to the "Events" and "Errors" tabs. In the example shown below, there are events showing when payments, attachments and signatures were completed. The submission event log also records delivery events.

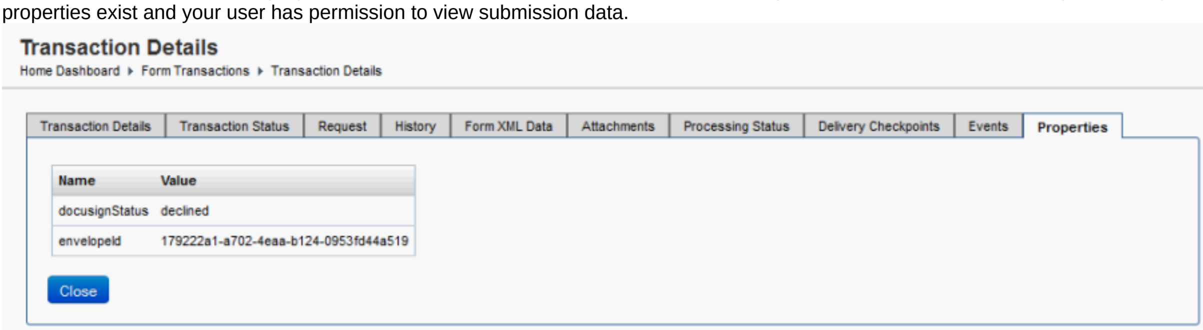


Viewing submission-related events

These tabs can be very helpful when debugging submission errors. However, we also recommend searching the error log around the time of the defect.

Submission Properties

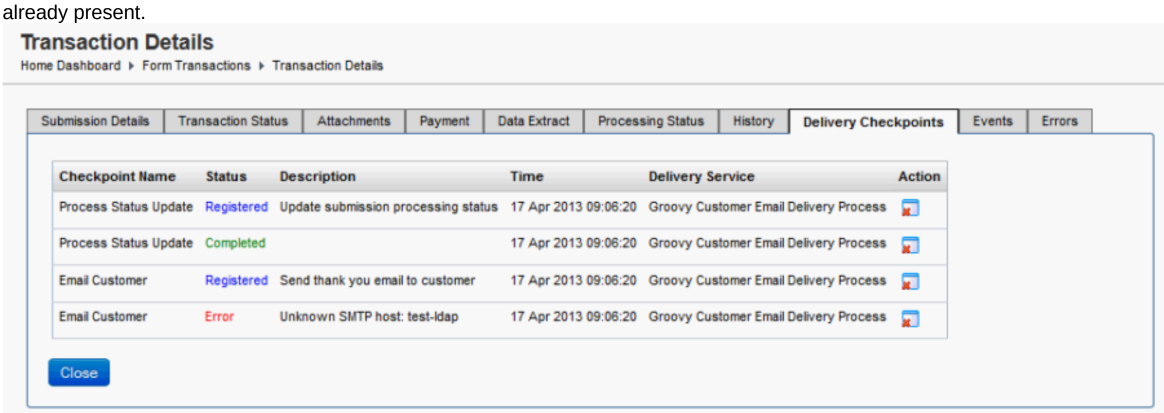
Submissions can have a set of arbitrary properties (key-value pairs). For example, these properties may be generated and referenced by Groovy services to store additional status values for the submission.
To view the list of submission properties, view submission details and switch to the "Properties" tab. Note that the tab is present only if submission properties exist and your user has permission to view submission data.



The set of properties for a submission

Delivery Checkpoints

Complex delivery services may include multiple steps that need to be completed. If a delivery failure occurs, it may be important to know which step failed (both for debugging purposes and in order to be able to retry delivery in such a way that previously completed steps are repeated only as necessary).
Transaction Manager allows delivery services to record arbitrary checkpoints as they proceed through delivery for a submission, and query existing checkpoints to ensure the correct behavior is executed in case of a retry after a previous delivery failure.Because delivery services can execute custom logic (e.g. a Groovy script), TM does not have any knowledge of delivery checkpoint semantics but merely provides an easily accessible API to record and query existing checkpoints. In addition, administrators can manually delete delivery checkpoints if necessary.
To view the list of delivery checkpoints for a submission, view submission details and switch to the "Delivery Checkpoints" tab. Note that most delivery services will not record checkpoint information; for a custom Groovy service, you have to determine which checkpoints make sense for a customized delivery service and then enhance the Groovy script to log checkpoints and potentially execute smart retry behavior by querying which checkpoints are already present.

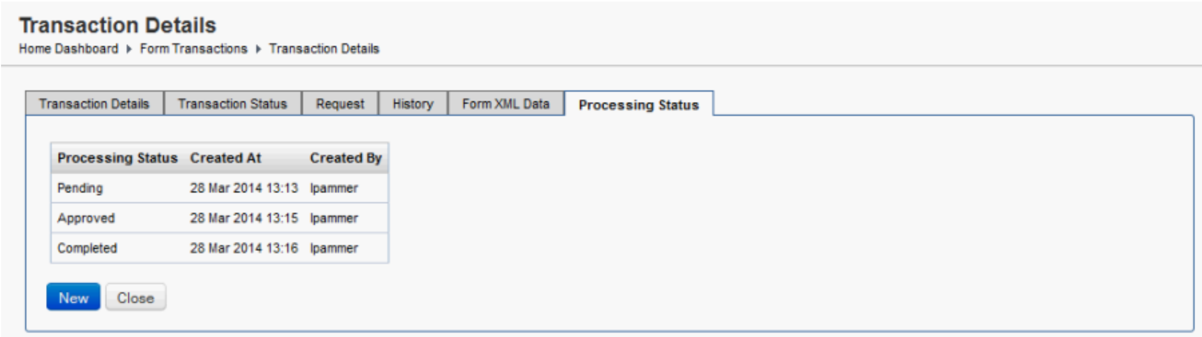


The list of submission delivery checkpoints for a submission

Please refer to the Transact Integration Guide, section "Delivery Checkpoints" for an example Groovy script.

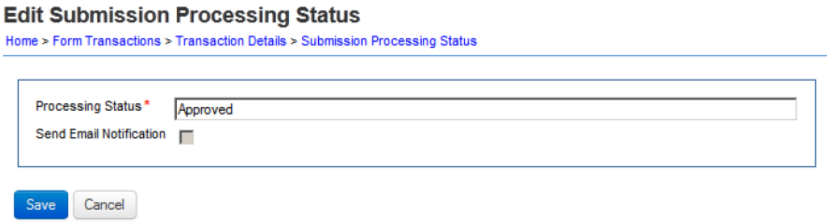
Processing Status

In addition to the Transaction Manager status values described earlier, there is also a status variable called "Processing Status". This value is used to indicate whether the organization has started processing the submission and what the current status is. It can be updated by the organization using a web service call. Refer to the separately provided TIA Administration Guide for more information.
In the Example Self Service Space, users can elect to be notified when the processing status for one of their submission changes.
The "Processing Status" tab shows the history of all changes made to the processing status, along with the time that the change was made and the user who made the change.



Processing status entries for a submission

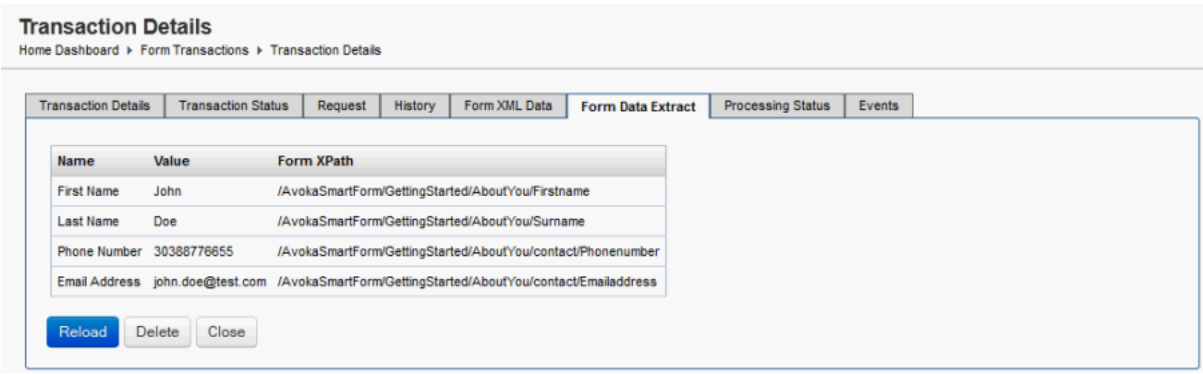
You can also manually update the processing status by clicking "New". On the "Edit Submission Processing Status" page, enter the new processing status. If the submission is associated with a Transaction Manager space user, you can tick "Send Email Notification" to make sure the user is notified of the status change.



Updating the processing status

Data Extract Values

Transaction Manager can be instructed to extract items of interest from the submitted XML and store them separately (see Form Version Configuration (Manager v5.0)#_setupdataextractmappings on how to set up data extract mappings).
To view extracted data, edit the submission and switch to the "Data Extract" tab. As shown below, all items extracted from the submission will be shown, with the concrete value for the current submission listed in the "Value" column.



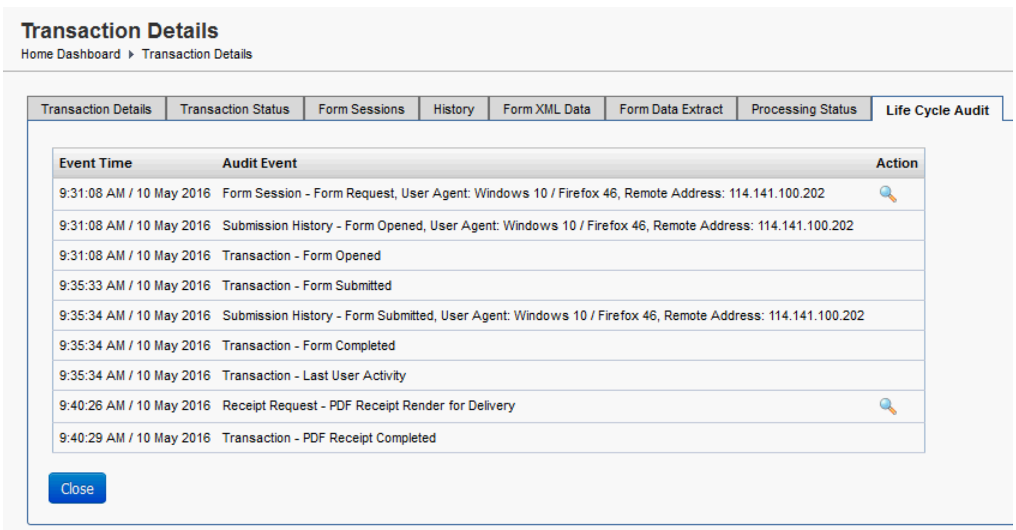
Viewing data extracted from the submission XML

You can delete all values if necessary, or reload the values from the submission XML using the configured data extract mappings by clicking "Reload".

Life Cycle Audit

Submissions can process through a number of different steps. While TM offers ways to track all possible steps and events, it can be hard to get the full picture of what happened in which order. To help with this, TM provides a full life cycle audit for each submission.

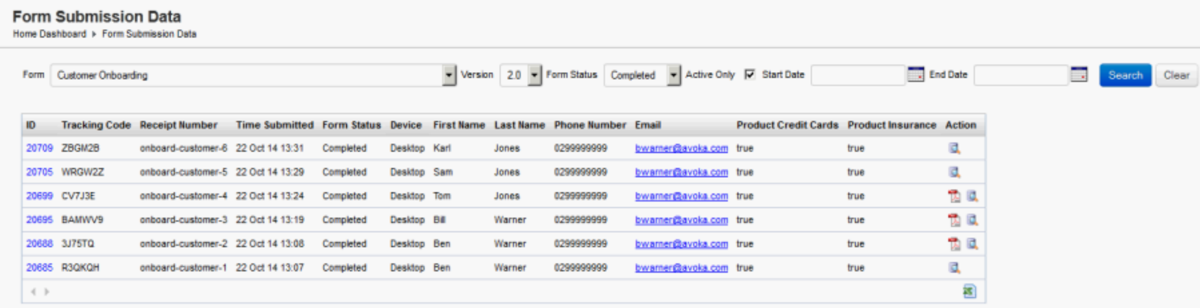
To access the life cycle audit, view the submission and switch to the "Life Cycle Audit" tab.



Viewing a submission's life cycle

Form Submission Data

The form submission data page lists the submission data extract values for a set of submission in tabular form. It can be used to quickly review the extracted data for multiple submissions.
To view form submission data, go to "Operations > Form Submission Data". Select a form and form version and possibly a date range, and click "View".



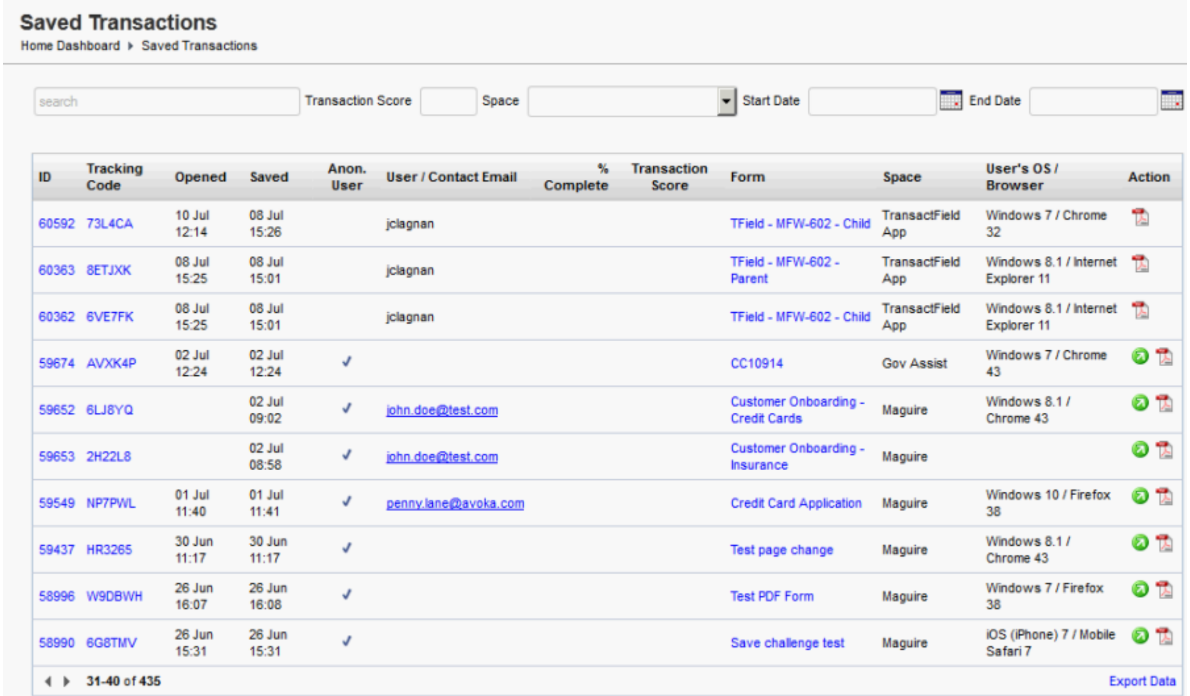
ID	Tracking Code	Receipt Number	Time Submitted	Form Status	Device	First Name	Last Name	Phone Number	Email	Product Credit Cards	Product Insurance	Action
20709	ZBGM2B	onboard-customer-6	22 Oct 14 13:31	Completed	Desktop	Karl	Jones	0299999999	bwarner@avoka.com	true	true	
20705	WRGW2Z	onboard-customer-5	22 Oct 14 13:29	Completed	Desktop	Sam	Jones	0299999999	bwarner@avoka.com	true	true	
20699	CV7J3E	onboard-customer-4	22 Oct 14 13:24	Completed	Desktop	Tom	Jones	0299999999	bwarner@avoka.com	true	true	
20695	BAMWV9	onboard-customer-3	22 Oct 14 13:19	Completed	Desktop	Bill	Warner	0299999999	bwarner@avoka.com	true	true	
20688	3J75TQ	onboard-customer-2	22 Oct 14 13:08	Completed	Desktop	Ben	Warner	0299999999	bwarner@avoka.com	true	true	
20685	R3QKQH	onboard-customer-1	22 Oct 14 13:07	Completed	Desktop	Ben	Warner	0299999999	bwarner@avoka.com	true	true	

Viewing submission data extract values for a form

Each entry corresponds to a submission for which data extract mappings were defined at submission time, and data was extracted. You can navigate to the submission details ( ), or view the receipt ( ).
You can also export the entire set of entries as an Excel file by clicking the "Export Data" link below the table.

Saved Submissions

Transaction Manager provides a specialized page showing saved submissions only. To access it, go to "Operations > Saved Transactions".

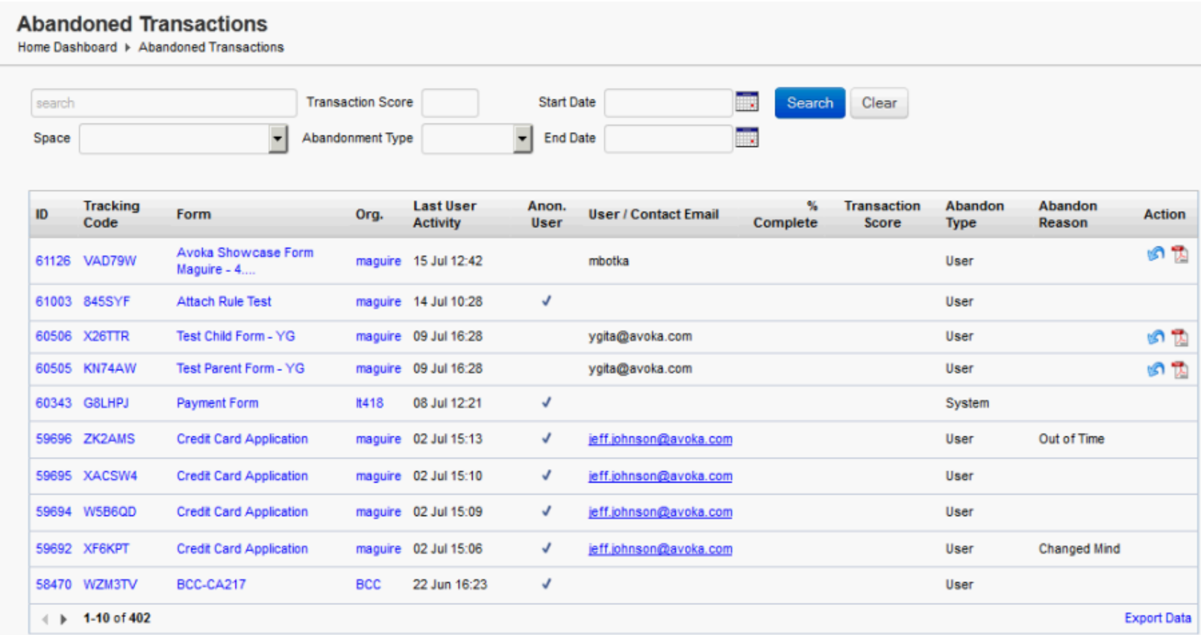


ID	Tracking Code	Opened	Saved	Anon. User	User / Contact Email	% Complete	Transaction Score	Form	Space	User's OS / Browser	Action
60592	73L4CA	10 Jul 12:14	08 Jul 15:26		jclagnan			TField - MFW-602 - Child	TransactField App	Windows 7 / Chrome 32	
60363	8ETJXK	08 Jul 15:25	08 Jul 15:01		jclagnan			TField - MFW-602 - Parent	TransactField App	Windows 8.1 / Internet Explorer 11	
60362	6VE7FK	08 Jul 15:25	08 Jul 15:01		jclagnan			TField - MFW-602 - Child	TransactField App	Windows 8.1 / Internet Explorer 11	
59674	AVXK4P	02 Jul 12:24	02 Jul 12:24	✓				CC10914	Gov Assist	Windows 7 / Chrome 43	
59652	6LJ8YQ		02 Jul 09:02	✓	john.doe@test.com			Customer Onboarding - Credit Cards	Maguire	Windows 8.1 / Chrome 43	
59653	2H22L8		02 Jul 08:58	✓	john.doe@test.com			Customer Onboarding - Insurance	Maguire		
59549	NP7PWL	01 Jul 11:40	01 Jul 11:41	✓	penny.lane@avoka.com			Credit Card Application	Maguire	Windows 10 / Firefox 38	
59437	HR3265	30 Jun 11:17	30 Jun 11:17	✓				Test page change	Maguire	Windows 8.1 / Chrome 43	
58996	W9DBWH	26 Jun 16:07	26 Jun 16:08	✓				Test PDF Form	Maguire	Windows 7 / Firefox 38	
58990	6G8TMV	26 Jun 15:31	26 Jun 15:31	✓				Save challenge test	Maguire	iOS (iPhone) 7 / Mobile Safari 7	

The list of saved submissions
You can filter the list by organization, form code, submission OID, tracking and receipt number, minimum transaction score, space and saved time.
For each saved submission, you can view submission details by clicking the link in the "ID" or "Tracking Code" columns. If the submission supports it, you can open it by clicking the "Open Saved Form" action ( ). This can be useful if users ask for help in completing their submissions.As long as submission data is available, you can also view the receipt for the currently filled in data by clicking the "View Receipt" action ( ).

Abandoned Submissions

Transaction Manager provides a list of abandoned submissions, which includes submissions abandoned by users (explicitly by clicking a button to abandon the submission, or implicitly by closing the browser or never returning to a saved form) and submissions marked as abandoned by administrators. You can access the list of abandoned submissions via "Operations > Abandoned Transactions".



Abandoned Transactions

Home Dashboard ▸ Abandoned Transactions

ID	Tracking Code	Form	Org.	Last User Activity	Anon. User	User / Contact Email	% Complete	Transaction Score	Abandon Type	Abandon Reason	Action
61126	VAD79W	Avoka Showcase Form Maguire - 4....	maguire	15 Jul 12:42		mbotka			User		
61003	845SYF	Attach Rule Test	maguire	14 Jul 10:28	✓				User		
60506	X26TTR	Test Child Form - YG	maguire	09 Jul 16:28		ygita@avoka.com			User		
60505	KN74AW	Test Parent Form - YG	maguire	09 Jul 16:28		ygita@avoka.com			User		
60343	G8LHPJ	Payment Form	lt418	08 Jul 12:21	✓				System		
59696	ZK2AMS	Credit Card Application	maguire	02 Jul 15:13	✓	jeff.johnson@avoka.com			User	Out of Time	
59695	XACSW4	Credit Card Application	maguire	02 Jul 15:10	✓	jeff.johnson@avoka.com			User		
59694	W5B6QD	Credit Card Application	maguire	02 Jul 15:09	✓	jeff.johnson@avoka.com			User		
59692	XF6KPT	Credit Card Application	maguire	02 Jul 15:06	✓	jeff.johnson@avoka.com			User	Changed Mind	
58470	WZM3TV	BCC-CA217	BCC	22 Jun 16:23	✓				User		

1-10 of 402 Export Data

The list of abandoned submissions

You can filter the list by organization, form code, submission OID, tracking and receipt number, abandonment type ("User" denoting explicit user abandonment, "System" implicit user abandonment and "Administrator" submissions abandoned by an administrator), minimum transaction score, space and last user activity time.

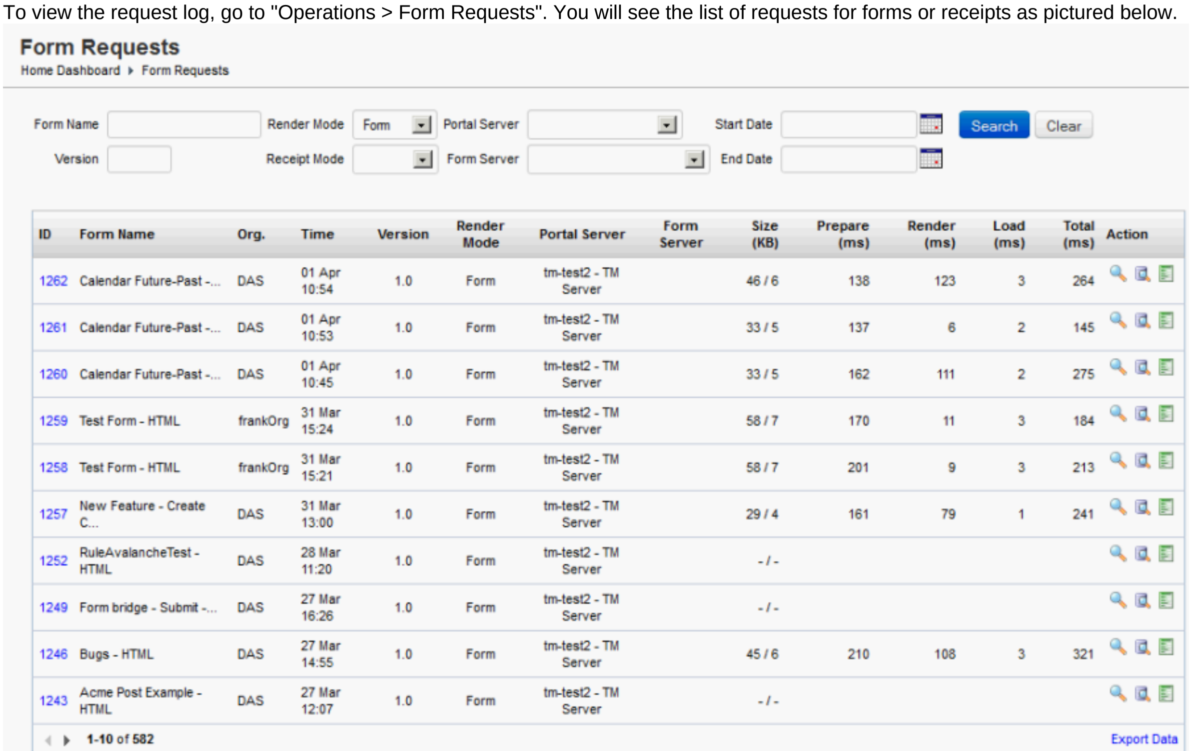
If submission data is present, you can revert the submission into "Saved" status (e.g. if the original submitter would like to complete the submission now) by using the "Make Abandoned Transaction Saved" action ( ) or view the receipt by using the "View Receipt" action ( ).

Request Log

The request log contains information about user requests for forms and receipts; i.e. whenever a user opens a form, a request log entry is created and associated with the submission log entry. Like the submission log, the request log contains information that can be very helpful in reporting and debugging.

Browsing the Request Log

To view the request log, go to "Operations > Form Requests". You will see the list of requests for forms or receipts as pictured below.



Form Requests

Home Dashboard ▸ Form Requests

ID	Form Name	Org.	Time	Version	Render Mode	Portal Server	Form Server	Size (KB)	Prepare (ms)	Render (ms)	Load (ms)	Total (ms)	Action
1262	Calendar Future-Past -...	DAS	01 Apr 10:54	1.0	Form	tm-test2 - TM Server		46 / 6	138	123	3	264	
1261	Calendar Future-Past -...	DAS	01 Apr 10:53	1.0	Form	tm-test2 - TM Server		33 / 5	137	6	2	145	
1260	Calendar Future-Past -...	DAS	01 Apr 10:45	1.0	Form	tm-test2 - TM Server		33 / 5	162	111	2	275	
1259	Test Form - HTML	frankOrg	31 Mar 15:24	1.0	Form	tm-test2 - TM Server		58 / 7	170	11	3	184	
1258	Test Form - HTML	frankOrg	31 Mar 15:21	1.0	Form	tm-test2 - TM Server		58 / 7	201	9	3	213	
1257	New Feature - Create C...	DAS	31 Mar 13:00	1.0	Form	tm-test2 - TM Server		29 / 4	161	79	1	241	
1252	RuleAvalancheTest - HTML	DAS	28 Mar 11:20	1.0	Form	tm-test2 - TM Server		- / -					
1249	Form bridge - Submit -...	DAS	27 Mar 16:26	1.0	Form	tm-test2 - TM Server		- / -					
1246	Bugs - HTML	DAS	27 Mar 14:55	1.0	Form	tm-test2 - TM Server		45 / 6	210	108	3	321	
1243	Acme Post Example - HTML	DAS	27 Mar 12:07	1.0	Form	tm-test2 - TM Server		- / -					

1-10 of 582 Export Data

Viewing the request log

You can filter the request log by organization, the form name, the form version, the "Render Mode" ("Form" to display only requests made for forms, "Receipt" to display only requests made for receipts), the "Receipt Mode" for receipt requests ("Web" indicating that the receipt was requested by a user via a web browser, "Email" indicating that the receipt was generated when a space user asked for an email copy, "Delivery" indicating that the receipt was generated during submission delivery), the server node serving the request and the request date and time.

You can also export the search results to Excel by clicking the "Export Data" link below the table.

The request list shows the basic data (organization, form, form version, request time, render mode) as well as some performance-related measurements (note that the latter are logged only if the form was rendered successfully).

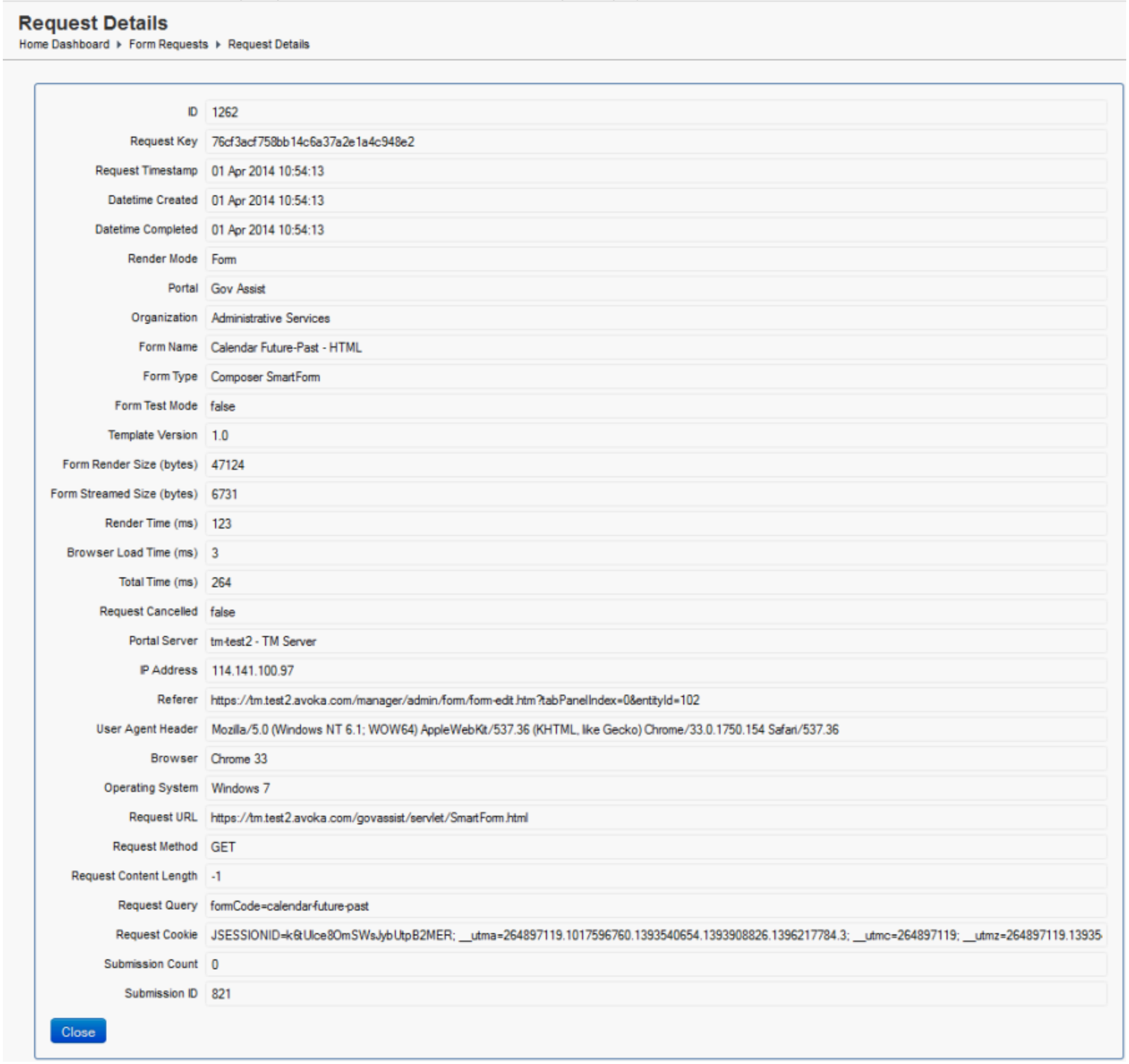
The "Size (KB)" column shows the size of the form produced by the render process.

The "Prepare (ms)" column contains the time it took to prepare the render request (including creating the request log entry). The "Render (ms)" column contains the time it took to render the form on the server. The "Load (ms)" contains the time that was needed to stream the rendered form to the user's browser. The "Total (ms)" column contains the total time from when the request was made to when the form was displayed in the browser.

You can view detailed information on each request (see the following section) by clicking the "View" icon ( ). You can view the submission associated with the request by clicking the "View Submission Details" icon ( ), and the form that was requested by clicking the "View Form Details ( ) icon.

Viewing a Request

If you want to view detailed information about a form request (e.g. IP address or referer), go to "Operations > Form Requests", locate the request and click the "View Details" icon ( ). You will see the "View Form Request" page.

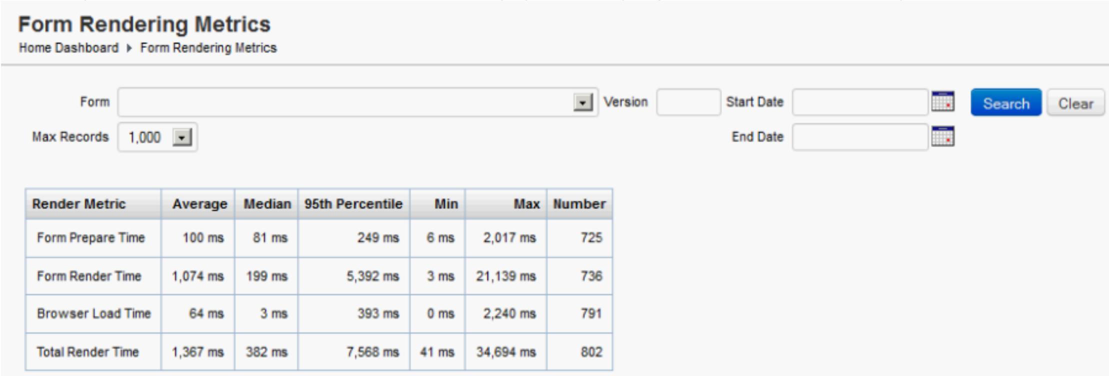


Viewing detailed request information

The "View Form Request" page shows a number of fields containing request log relevant data. Some of the important fields are described in this section.
Request Key: An alternate key uniquely identifying the request. This key, as opposed to the request ID, is not generated by the database and will therefore persist through database migrations.
Form Type: The form type of the form being requested. See Form Configuration (Manager v5.0)#_sec_formtypes on the form types available.
Form Test Mode: Indicates whether the request was made when the form was in test mode or not. This can be used for reporting or debugging purposes.
Task: The task on which the request was based (if applicable).
Remote User / External User ID: The identifier of the user requesting the form (if the request was not anonymous). The fields displayed are "Remote User" and "Profile Name" if the user space is using Transaction Manager user accounts, and "External User ID" and "External Profile ID" if the user space is using a custom login mechanism (see Spaces (Manager v5.0)#_sec_standardspaceproperties for the space property controlling this behavior).
IP Address, Referer, Browser, User Agent Header: These fields contain data related to the user and their environment, which can be helpful for debugging.
Referer: This field contains the referer HTML header. If the user accessed a form via the space landing page, this field will contain the referer that was set when the user requested the landing page.
Request Query: This field shows the exact request made by the user's browser (basically the URL request parameters).
Request Cookie: This field shows the contents of the cookie header.
Submission Count: The number of submissions resulting from a form request. Transaction Manager by default will accept only a single submission and reject subsequent ones (see Deployment Properties (Manager v5.0)#_sec_duplicateformsubmissions).
Submission Expiry Date: The date after which a submission for this request will be rejected.

Form Rendering Metrics

You can get an overview on how fast forms are rendering by accessing "Operations > Form Rendering Metrics".



Render Metric	Average	Median	95th Percentile	Min	Max	Number
Form Prepare Time	100 ms	81 ms	249 ms	6 ms	2,017 ms	725
Form Render Time	1,074 ms	199 ms	5,392 ms	3 ms	21,139 ms	736
Browser Load Time	64 ms	3 ms	393 ms	0 ms	2,240 ms	791
Total Render Time	1,367 ms	382 ms	7,568 ms	41 ms	34,694 ms	802

Viewing form rendering metrics

The table shows various metrics (including mean, median and range) of the total render time and its components for all form requests matching the search criteria.

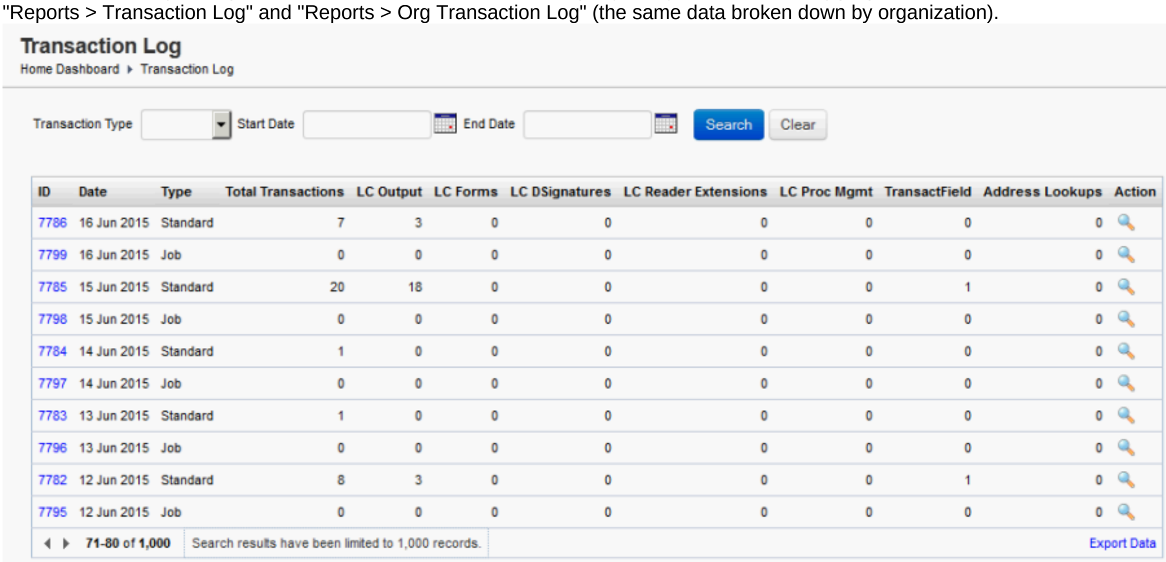
The total render time refers to the total time from when the request was made to when the form was displayed in the browser and includes three subtasks shown as separate rows: The "Form Prepare Time" refers to the time it took to prepare the render request (including creating the request log entry). The "Form Render Time" refers to the time it took to render the form on the server. The "Browser Load Time" contains the time that was needed to stream the rendered form to the user's browser.

To change the set of form requests on which the metrics are based, enter different values into the search fields and click "Search". You can filter by form and form version as well as render date. The "Max Records" field allows you to a limit on the number of requests to calculate the metrics from.

Note: If you choose a large number of records to use for calculation (the "Max Records" field), the metrics calculation may take longer.

Transaction Log

Transaction Manager keeps a record of the transactions occurring in your system for licensing purposes. You can review this information by accessing "Reports > Transaction Log" and "Reports > Org Transaction Log" (the same data broken down by organization).



ID	Date	Type	Total Transactions	LC Output	LC Forms	LC DSignatures	LC Reader Extensions	LC Proc Mgmt	TransactField	Address Lookups	Action
7786	16 Jun 2015	Standard	7	3	0	0	0	0	0	0	
7799	16 Jun 2015	Job	0	0	0	0	0	0	0	0	
7785	15 Jun 2015	Standard	20	18	0	0	0	0	1	0	
7798	15 Jun 2015	Job	0	0	0	0	0	0	0	0	
7784	14 Jun 2015	Standard	1	0	0	0	0	0	0	0	
7797	14 Jun 2015	Job	0	0	0	0	0	0	0	0	
7783	13 Jun 2015	Standard	1	0	0	0	0	0	0	0	
7796	13 Jun 2015	Job	0	0	0	0	0	0	0	0	
7782	12 Jun 2015	Standard	8	3	0	0	0	0	1	0	
7795	12 Jun 2015	Job	0	0	0	0	0	0	0	0	

71-80 of 1,000 Search results have been limited to 1,000 records. Export Data

The transaction log

Transaction data is calculated and aggregated per day. Transactions mostly correspond to form submissions. Additional information such as usage of Adobe LiveCycle modules or the TransactField App is also recorded. For further details on transaction licensing in Transaction Manager, please contact your account manager.

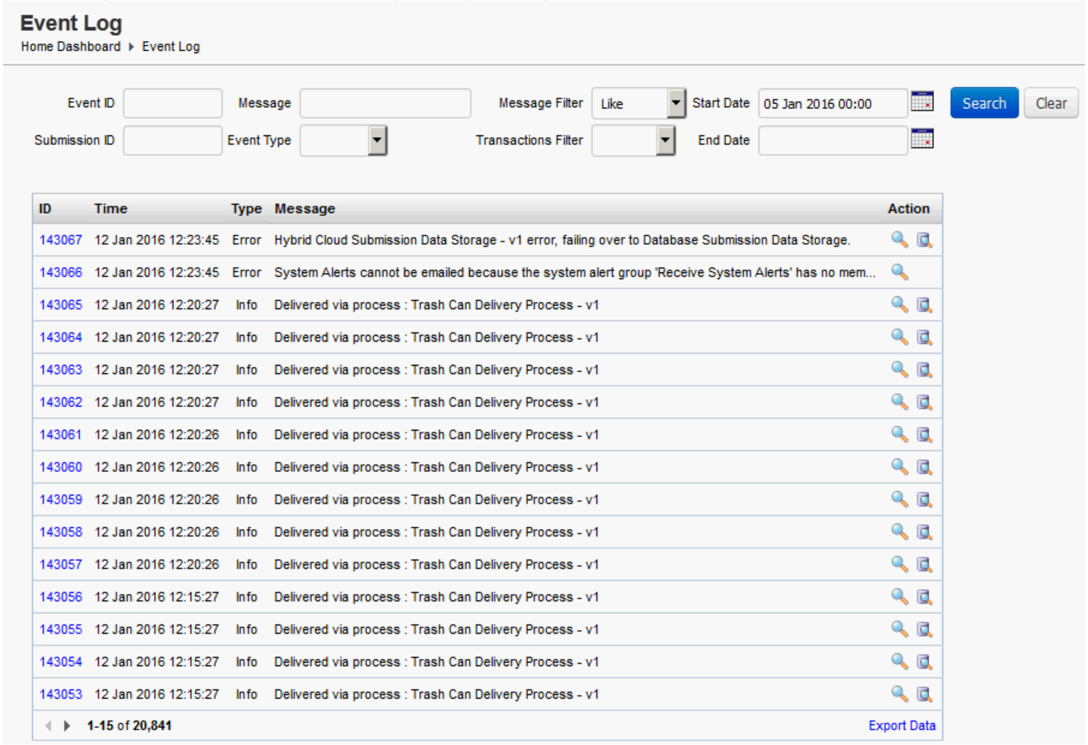
Transaction Manager also provides four BIRT reports containing transaction information. Please see Reports (Manager v5.0)
#_sec_transactionsbydayreport onwards.

Event Log

The event log contains various messages logged by Transaction Manager, mostly of informative character. Examples for event log entries are messages recording when Transaction Manager is itself started or shut down or when a submission is delivered to the organization.

To view the event log, go to "System > Event Log". You can filter the list of events by ID, type ("Info", "Warning", "Error" or "Security"), using a substring of the message, the submission which the event relates to (where applicable) and the date and time the event was logged. The "Message Filter" dropdown lets you control how the message substring is used in the search. You can include or exclude submission related events by using the "Transactions Filter" dropdown.

You can export the search results to Excel by clicking the "Export Data" link below the table.



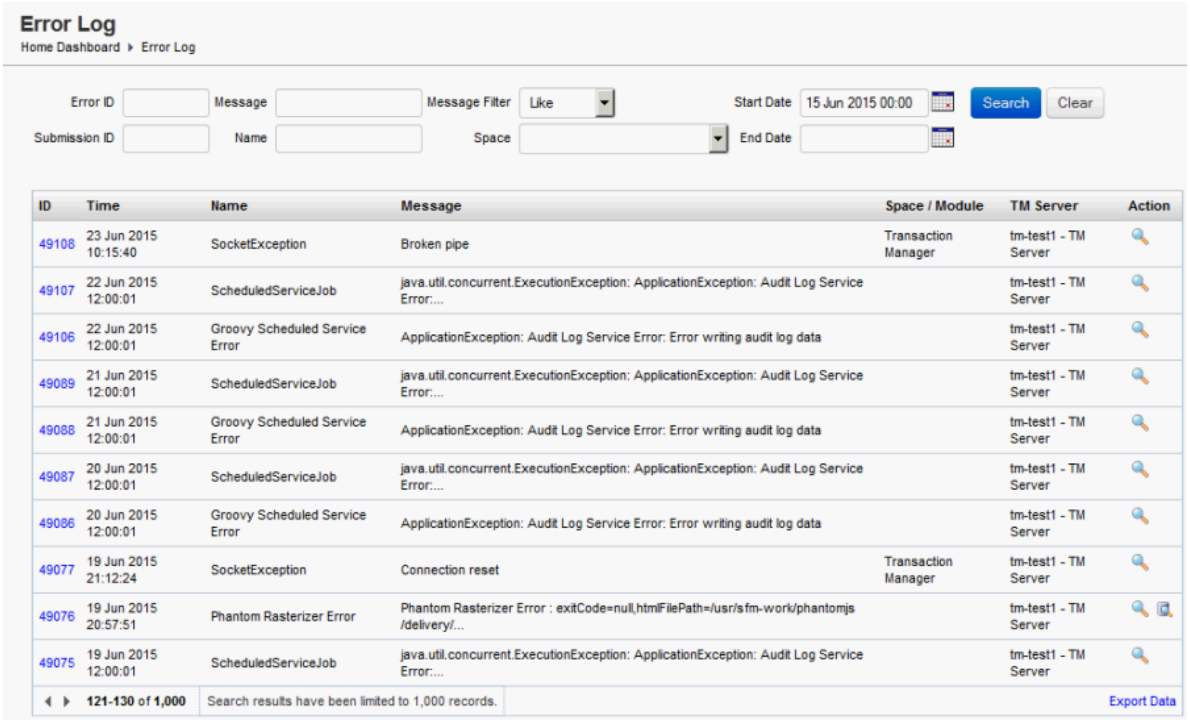
The event log

From this page, you can also view the details for each event using the "View Details" icon ( ). This takes you to the "View Event Log" page. Even though this page does not contain any additional fields beyond what is shown in the event log table, it can be useful to see the full text content for long messages.

Details for an event log entry

Error Log

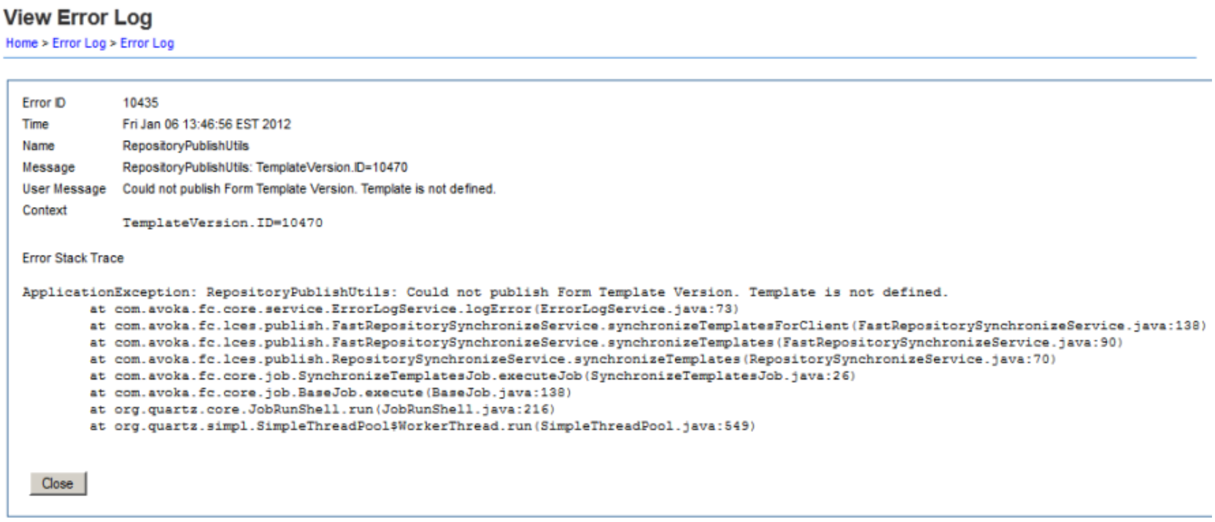
The error log is similar to the event log. However, messages logged to the error log are generally of a higher severity and may contain additional information such as a stack trace and the request URL. Examples for error log entries are problems resulting from lost database connections, but also faulty form data configurations or delivery errors due to misconfiguration.
To view the error log, go to "System > Error Log". You can filter the list by organization, using the error ID, the error name, a substring of the message, the submission ID (where applicable) and the date and time the error was logged. The "Message Filter" dropdown lets you control how the message substring is used in the search.
You can export the search results to Excel by clicking the "Export Data" link below the table.



The error log

You can also view detailed information for each entry by clicking the "View Details" icon ( ). This takes you to the "View Error Log" page for the entry. In addition to the basic data displayed in the error log, you can also see the full stack trace, name, context, user message, solution and the submission ID where applicable. If a user session existed for the error, the user's IP address, user name and user agent string are displayed.

Some errors associated with failed submissions may also contain the XML data that was being submitted. If this is the case, an additional icon ( ) will be displayed. Click on this icon to view the XML data.Note: This icon is displayed only if you have the "Submission XML Data View" permission.



Details for an error log entry

The information you find here should go a long way towards diagnosing problems, and the error log should be monitored regularly to ensure smooth operations.

System Information

Transaction Manager provides information about runtime load and parameters. Go to "System > System Info" to access the system information.

System Information

Home Dashboard ▸ System Information

Runtime Info

Current Process ID	2608
Free Memory	1,205 MB
Total Memory	2,041 MB
Max Memory	2,041 MB
Number CPU / Cores	2
CPU Speed	2.35 GHz
Server Uptime	1,035 hours

Build Properties

Name	Value
database.product	sqlserver
form.types	Composer SmartForm,Dynamic PDF Form,Static PDF Form,Other Document
formcenter.version	4.0 BETA 2
groovy.client.data.isolation	true
groovy.enabled	true
installation.dir	d:/avoka/transact/manager
livecycle.version	10.0.2
revision.number	35145

System Properties

Name	Value
RUN_UNDER_ECLIPSE	false
awt.toolkit	sun.awt.windows.WToolkit
catalina.home	d:\avoka\transact\manager\server\standalone\tmp
cayenne.server.contexts_sync_strategy	false
com.sun.jndi.ldap.connect.pool.protocol	plain ssl
com.sun.jndi.ldap.connect.pool.timeout	300000
file.encoding	Cp1252
file.encoding.pkg	sun.io
file.separator	\
java.awt.graphicsenv	sun.awt.Win32GraphicsEnvironment

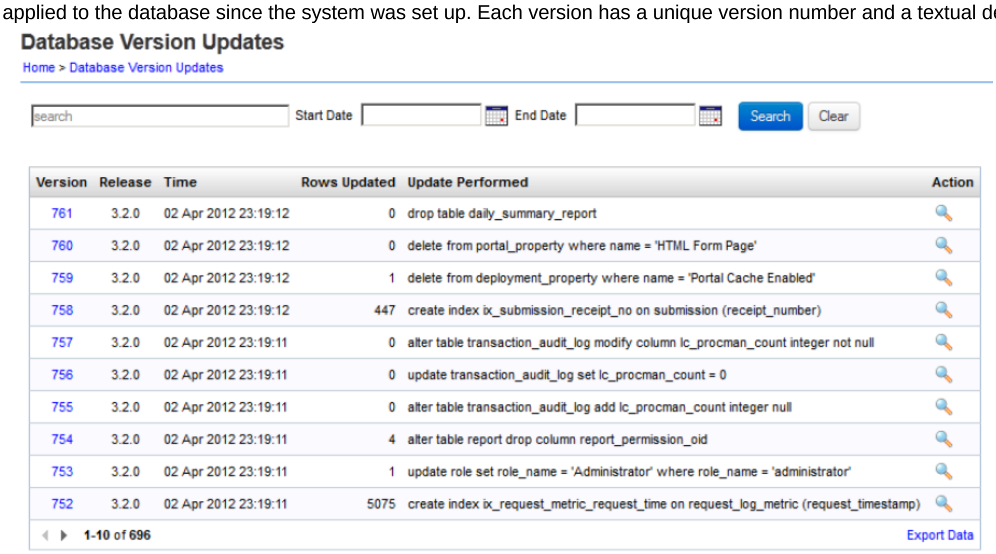
1-10 of 98

Viewing system information

The page consists of three sections: some runtime statistics at the top, properties that were used to build your version of Transaction Manager in the center and the Java system properties in the table below.

Database Version History

When Transaction Manager is started, it checks the database to see what version the database schema is at and runs update statements as necessary. All these automatic updates are logged to the database version history, which can be viewed via the management console.
To view the database version history, go to "System > Database Version History". You will see the "Database Version Updates" page listing the updates applied to the database since the system was set up. Each version has a unique version number and a textual descriptor identifying the release.



Database Version Updates

Home > Database Version Updates

search | Start Date | End Date | Search | Clear

Version	Release	Time	Rows Updated	Update Performed	Action
761	3.2.0	02 Apr 2012 23:19:12	0	drop table daily_summary_report	
760	3.2.0	02 Apr 2012 23:19:12	0	delete from portal_property where name = 'HTML Form Page'	
759	3.2.0	02 Apr 2012 23:19:12	1	delete from deployment_property where name = 'Portal Cache Enabled'	
758	3.2.0	02 Apr 2012 23:19:12	447	create index ix_submission_receipt_no on submission (receipt_number)	
757	3.2.0	02 Apr 2012 23:19:11	0	alter table transaction_audit_log modify column lc_procman_count integer not null	
756	3.2.0	02 Apr 2012 23:19:11	0	update transaction_audit_log set lc_procman_count = 0	
755	3.2.0	02 Apr 2012 23:19:11	0	alter table transaction_audit_log add lc_procman_count integer null	
754	3.2.0	02 Apr 2012 23:19:11	4	alter table report drop column report_permission_oid	
753	3.2.0	02 Apr 2012 23:19:11	1	update role set role_name = 'Administrator' where role_name = 'administrator'	
752	3.2.0	02 Apr 2012 23:19:11	5075	create index ix_request_metric_request_time on request_log_metric (request_timestamp)	

1-10 of 696 Export Data

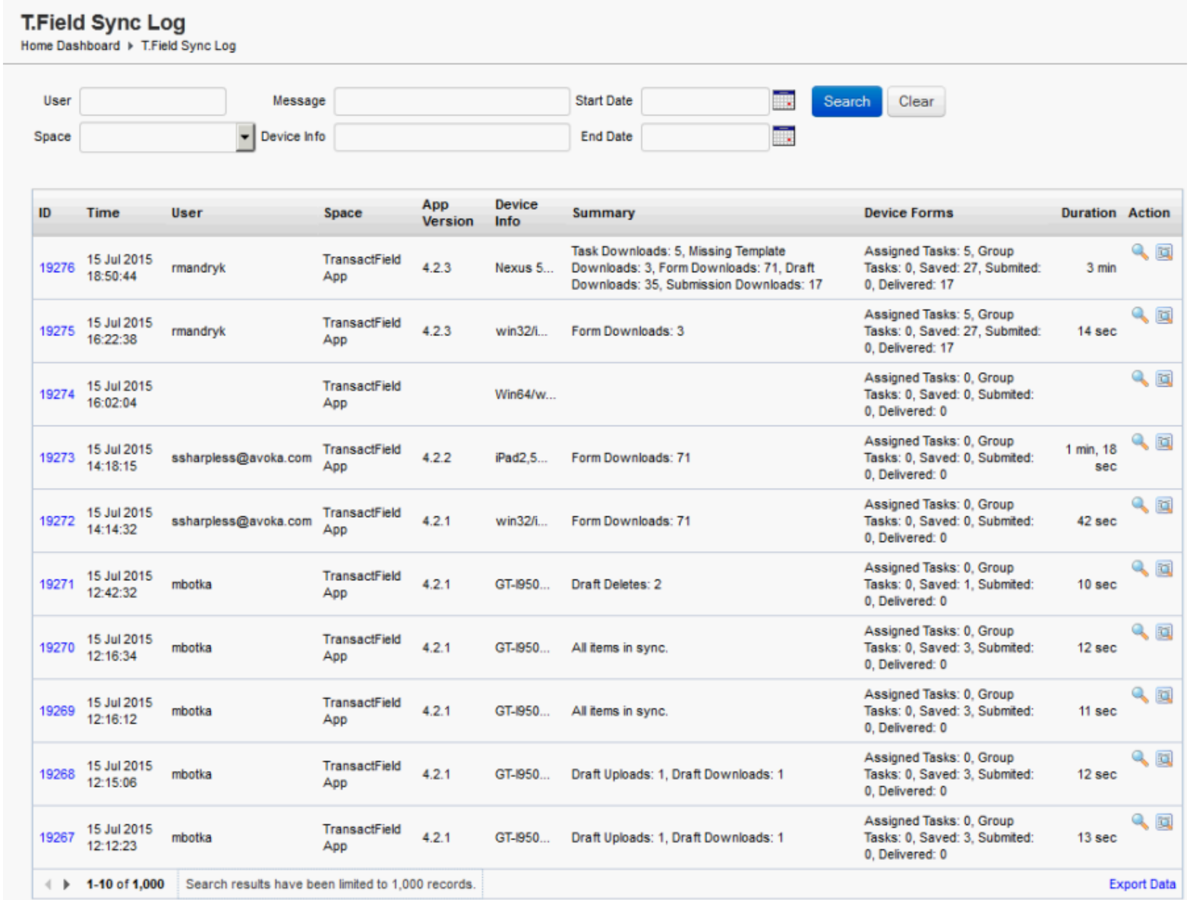
Browsing the list of database updates
You can filter the list by version number, a substring of the release identifier and the date the update was applied, and export the search results to Excel

by clicking the "Export Data" link below the table.

You can also click the "View" icon ( ) to see the full statement that was executed and how many rows were modified.
This log can be useful in determining which version your database is up to; other than that, it is not relevant for day-to-day operations.

TransactField App Synchronization Log

To aid in debugging TransactField App issues, Transaction Manager stores a log of operations invoked by the TransactField App. You can access the TransactField App synchronization log via "System > T.Field Sync Log".



ID	Time	User	Space	App Version	Device Info	Summary	Device Forms	Duration	Action
19276	15 Jul 2015 18:50:44	rmandryk	TransactField App	4.2.3	Nexus 5...	Task Downloads: 5, Missing Template Downloads: 3, Form Downloads: 71, Draft Downloads: 35, Submission Downloads: 17	Assigned Tasks: 5, Group Tasks: 0, Saved: 27, Submited: 0, Delivered: 17	3 min	
19275	15 Jul 2015 16:22:38	rmandryk	TransactField App	4.2.3	win32/i...	Form Downloads: 3	Assigned Tasks: 5, Group Tasks: 0, Saved: 27, Submited: 0, Delivered: 17	14 sec	
19274	15 Jul 2015 16:02:04		TransactField App		Win64/w...		Assigned Tasks: 0, Group Tasks: 0, Saved: 0, Submited: 0, Delivered: 0		
19273	15 Jul 2015 14:18:15	ssharpless@avoka.com	TransactField App	4.2.2	iPad2,5...	Form Downloads: 71	Assigned Tasks: 0, Group Tasks: 0, Saved: 0, Submited: 0, Delivered: 0	1 min, 18 sec	
19272	15 Jul 2015 14:14:32	ssharpless@avoka.com	TransactField App	4.2.1	win32/i...	Form Downloads: 71	Assigned Tasks: 0, Group Tasks: 0, Saved: 0, Submited: 0, Delivered: 0	42 sec	
19271	15 Jul 2015 12:42:32	mbotka	TransactField App	4.2.1	GT-I950...	Draft Deletes: 2	Assigned Tasks: 0, Group Tasks: 0, Saved: 1, Submited: 0, Delivered: 0	10 sec	
19270	15 Jul 2015 12:16:34	mbotka	TransactField App	4.2.1	GT-I950...	All items in sync.	Assigned Tasks: 0, Group Tasks: 0, Saved: 3, Submited: 0, Delivered: 0	12 sec	
19269	15 Jul 2015 12:16:12	mbotka	TransactField App	4.2.1	GT-I950...	All items in sync.	Assigned Tasks: 0, Group Tasks: 0, Saved: 3, Submited: 0, Delivered: 0	11 sec	
19268	15 Jul 2015 12:15:06	mbotka	TransactField App	4.2.1	GT-I950...	Draft Uploads: 1, Draft Downloads: 1	Assigned Tasks: 0, Group Tasks: 0, Saved: 3, Submited: 0, Delivered: 0	12 sec	
19267	15 Jul 2015 12:12:23	mbotka	TransactField App	4.2.1	GT-I950...	Draft Uploads: 1, Draft Downloads: 1	Assigned Tasks: 0, Group Tasks: 0, Saved: 3, Submited: 0, Delivered: 0	13 sec	

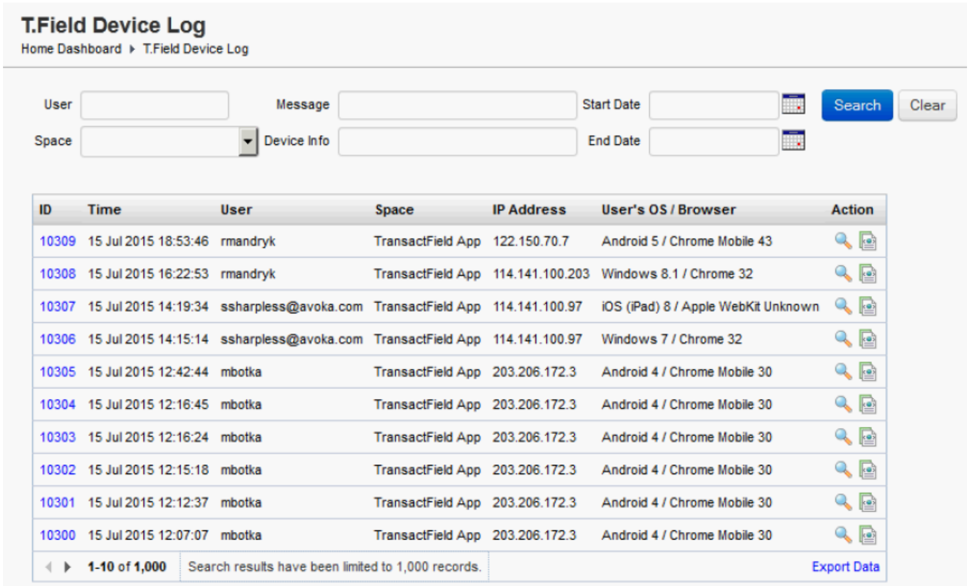
1-10 of 1,000 Search results have been limited to 1,000 records. Export Data

The TransactField App synchronization log

You will see the list of operations invoked by devices running the TransactField App. You can filter the list (among other things) by space, user and date. In addition, TransactField App users can elect to upload their device log to the TM server; these log files can be viewed via the TransactField App device log, which is described in the following section.

TransactField App Device Log

TransactField App users can upload the log file maintained by the TransactField App to the TM server. This can be very helpful in debugging issues occurring on the device running the TransactField App.
To view the list of uploaded TransactField App device logs, go to "System > T.Field Device Log".



ID	Time	User	Space	IP Address	User's OS / Browser	Action
10309	15 Jul 2015 18:53:46	rmandryk	TransactField App	122.150.70.7	Android 5 / Chrome Mobile 43	
10308	15 Jul 2015 16:22:53	rmandryk	TransactField App	114.141.100.203	Windows 8.1 / Chrome 32	
10307	15 Jul 2015 14:19:34	ssharpless@avoka.com	TransactField App	114.141.100.97	iOS (iPad) 8 / Apple WebKit Unknown	
10306	15 Jul 2015 14:15:14	ssharpless@avoka.com	TransactField App	114.141.100.97	Windows 7 / Chrome 32	
10305	15 Jul 2015 12:42:44	mbotka	TransactField App	203.206.172.3	Android 4 / Chrome Mobile 30	
10304	15 Jul 2015 12:16:45	mbotka	TransactField App	203.206.172.3	Android 4 / Chrome Mobile 30	
10303	15 Jul 2015 12:16:24	mbotka	TransactField App	203.206.172.3	Android 4 / Chrome Mobile 30	
10302	15 Jul 2015 12:15:18	mbotka	TransactField App	203.206.172.3	Android 4 / Chrome Mobile 30	
10301	15 Jul 2015 12:12:37	mbotka	TransactField App	203.206.172.3	Android 4 / Chrome Mobile 30	
10300	15 Jul 2015 12:07:07	mbotka	TransactField App	203.206.172.3	Android 4 / Chrome Mobile 30	

1-10 of 1,000 Search results have been limited to 1,000 records. Export Data

The TransactField App device log

You can view ( ) individual log entries to see the full TransactField App device log. You can also view the corresponding synchronization log entry ( ).

Preventative Measures

To avoid problems altogether or at least detect them as soon as possible, you can take preventative measures. This section outlines some things you can do to keep Transaction Manager running smoothly.

Monitor the System Logs

A good way to detect issues is to keep an eye on the logs provided by Transaction Manager. Inspect the error log and the event log regularly to be alerted to irregularities and problems of all sorts. Make sure you look into issues that you do not fully understand, using additional logs such as the submission and request log to track users' activities, the audit log to track changes made by administrators or the log files maintained by the application server.
If you have multiple administrators set up in Transaction Manager, the audit log should also be monitored to make sure no unintended or harmful changes have been made.

Check Scheduled Job Status

All jobs are started on a specific schedule and should not require maintenance. However, you should make sure all jobs are running normally from time to time by checking the "Scheduled Jobs" page (for active jobs, the status should be "Normal", the "Next Run" time should be in the future, and the "Last Run" time should be in the past, with the interval between them equal to "Repeat Interval").

Run Status Reports

If you have status reports set up for Transaction Manager, you should view them from time to time to check for aberrations or unexpected trends (e.g. a decline in submission levels may indicate some sort of problem). Better yet, set up a report schedule to get the reports emailed to you automatically (see Reports (Manager v5.0) for more).

Test Changes

Whenever any administrator makes changes, whether modifying organization properties or uploading a new form, these changes should be tested carefully, especially if they will affect forms that are in production.
This helps detect any errors or misconfiguration early on and lowers the risk of your users encountering unwanted behavior.

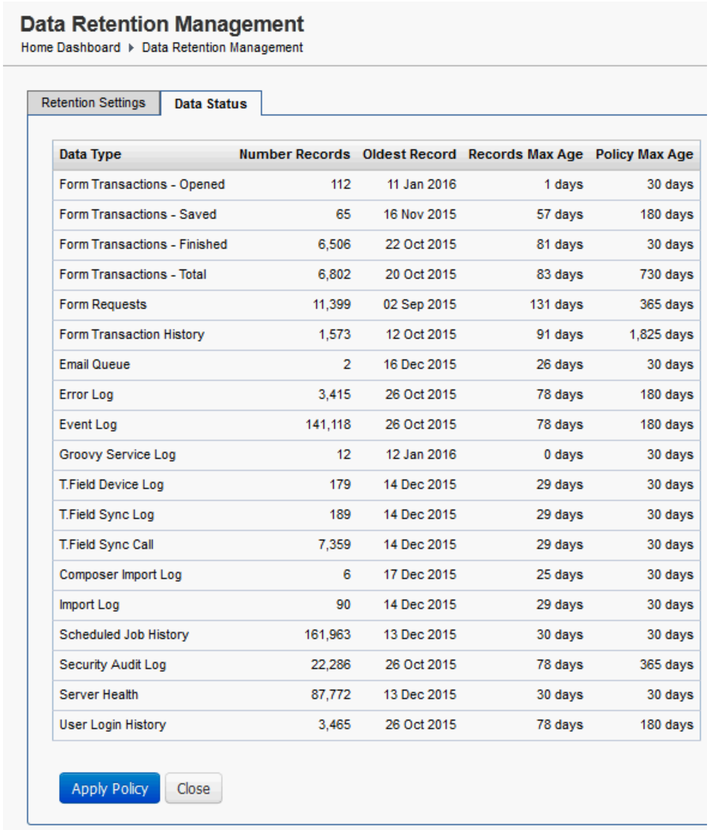
Data Retention

To maintain the system over time, data storage and retention needs to be considered. Storing a large amount of data in the Transaction Manager database can cause various operations to slow down, as well as the database itself to grow larger. Transaction Manager provides data purging capabilities as well as flexible data storage to allow you to keep your system functioning well.
Generally, you need to weigh the need for keeping the amount of data manageable against the need to preserve data, especially submission-related data that your space users have access to (such as receipts).
Transaction Manager already archives submission and request log metadata to a set of lightweight reporting tables, which can be used for reporting and auditing purposes after the original records have been purged. In addition, you can customize data retention and storage as described in the sections below.
Note that you can set the data retention and storage settings globally and override important settings for individual organizations as needed.

Please refer to Data Retention Management (Manager v5.0) for an article explaining the available settings.

Data Retention Configuration (Global)

To configure data retention and purging, go to "System > Data Retention Management".



Data Type	Number Records	Oldest Record	Records Max Age	Policy Max Age
Form Transactions - Opened	112	11 Jan 2016	1 days	30 days
Form Transactions - Saved	65	16 Nov 2015	57 days	180 days
Form Transactions - Finished	6,506	22 Oct 2015	81 days	30 days
Form Transactions - Total	6,802	20 Oct 2015	83 days	730 days
Form Requests	11,399	02 Sep 2015	131 days	365 days
Form Transaction History	1,573	12 Oct 2015	91 days	1,825 days
Email Queue	2	16 Dec 2015	26 days	30 days
Error Log	3,415	26 Oct 2015	78 days	180 days
Event Log	141,118	26 Oct 2015	78 days	180 days
Groovy Service Log	12	12 Jan 2016	0 days	30 days
T.Field Device Log	179	14 Dec 2015	29 days	30 days
T.Field Sync Log	189	14 Dec 2015	29 days	30 days
T.Field Sync Call	7,359	14 Dec 2015	29 days	30 days
Composer Import Log	6	17 Dec 2015	25 days	30 days
Import Log	90	14 Dec 2015	29 days	30 days
Scheduled Job History	161,963	13 Dec 2015	30 days	30 days
Security Audit Log	22,286	26 Oct 2015	78 days	365 days
Server Health	87,772	13 Dec 2015	30 days	30 days
User Login History	3,465	26 Oct 2015	78 days	180 days

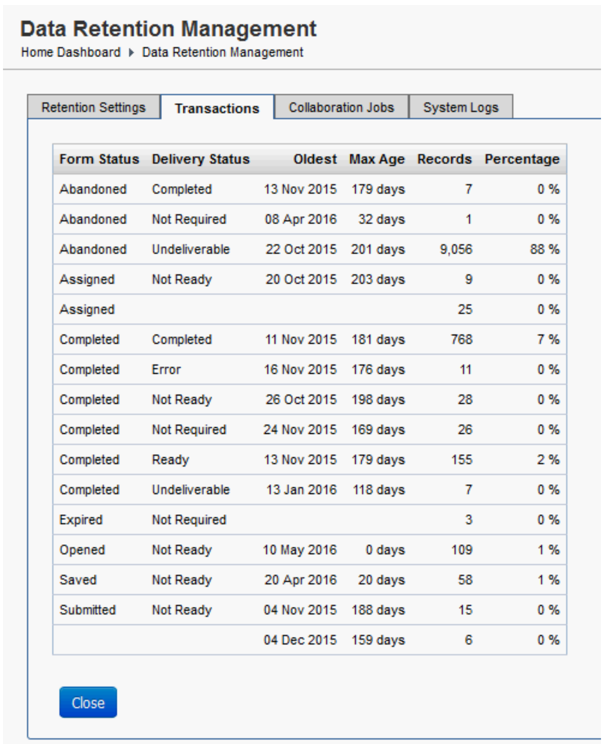
Managing data retention

You will see the currently configured retention settings. To modify the retention settings for a specific entity, choose a new value and click "Save". After saving, you can click "Apply Policy" to apply the retention policies and purge records exceeding the maximum age. The records will be purged, and Transaction Manager will display the number of records that were purged.

Some data retention settings can be overridden on the organization and form level. If you would like to restrict the available options on the organization levels such that they cannot be set to longer retention times than the system settings, tick the "Enforce Global Threshold" checkbox.

If you would like an overview of the data in the system, switch to one of the other tabs.

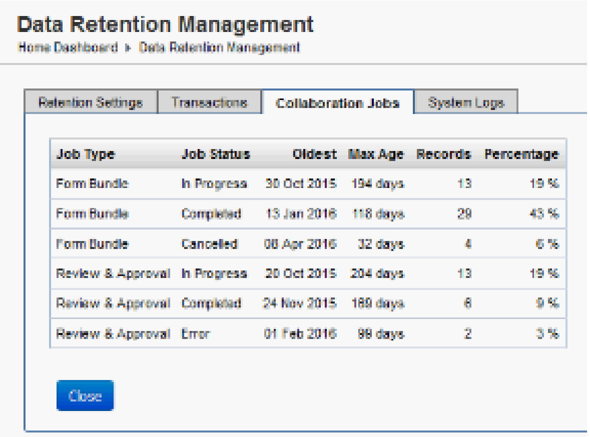
Information about submission data (excluding submissions associated with collaboration jobs) is provided on the "Transactions" tab.



Form Status	Delivery Status	Oldest	Max Age	Records	Percentage
Abandoned	Completed	13 Nov 2015	179 days	7	0 %
Abandoned	Not Required	08 Apr 2016	32 days	1	0 %
Abandoned	Undeliverable	22 Oct 2015	201 days	9,056	88 %
Assigned	Not Ready	20 Oct 2015	203 days	9	0 %
Assigned				25	0 %
Completed	Completed	11 Nov 2015	181 days	768	7 %
Completed	Error	16 Nov 2015	176 days	11	0 %
Completed	Not Ready	26 Oct 2015	198 days	28	0 %
Completed	Not Required	24 Nov 2015	169 days	26	0 %
Completed	Ready	13 Nov 2015	179 days	155	2 %
Completed	Undeliverable	13 Jan 2016	118 days	7	0 %
Expired	Not Required			3	0 %
Opened	Not Ready	10 May 2016	0 days	109	1 %
Saved	Not Ready	20 Apr 2016	20 days	58	1 %
Submitted	Not Ready	04 Nov 2015	188 days	15	0 %
		04 Dec 2015	159 days	6	0 %

Viewing submission data retention status

Information about collaboration jobs is provided on the "Collaboration Jobs" tab.



Data Retention Management

Home Dashboard ▸ Data Retention Management

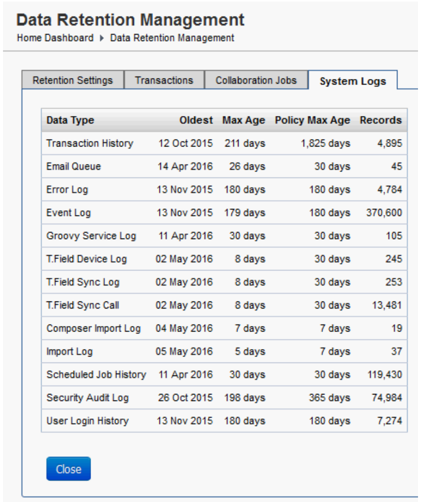
Retention Settings | Transactions | Collaboration Jobs | System Logs

Job Type	Job Status	Oldest	Max Age	Records	Percentage
Form Bundle	In Progress	30 Oct 2015	194 days	13	19 %
Form Bundle	Completed	13 Jan 2016	118 days	29	43 %
Form Bundle	Cancelled	08 Apr 2016	32 days	4	6 %
Review & Approval	In Progress	20 Oct 2015	204 days	13	19 %
Review & Approval	Completed	24 Nov 2015	169 days	6	9 %
Review & Approval	Error	01 Feb 2016	98 days	2	3 %

Close

Viewing collaboration job data retention status

Information about other entities is provided on the "System Logs" tab.



Data Retention Management

Home Dashboard ▸ Data Retention Management

Retention Settings | Transactions | Collaboration Jobs | System Logs

Data Type	Oldest	Max Age	Policy Max Age	Records
Transaction History	12 Oct 2015	211 days	1,825 days	4,895
Email Queue	14 Apr 2016	26 days	30 days	45
Error Log	13 Nov 2015	180 days	180 days	4,784
Event Log	13 Nov 2015	179 days	180 days	370,600
Groovy Service Log	11 Apr 2016	30 days	30 days	105
T.Field Device Log	02 May 2016	8 days	30 days	245
T.Field Sync Log	02 May 2016	8 days	30 days	253
T.Field Sync Call	02 May 2016	8 days	30 days	13,481
Composer Import Log	04 May 2016	7 days	7 days	19
Import Log	05 May 2016	5 days	7 days	37
Scheduled Job History	11 Apr 2016	30 days	30 days	119,430
Security Audit Log	26 Oct 2015	198 days	365 days	74,984
User Login History	13 Nov 2015	180 days	180 days	7,274

Close

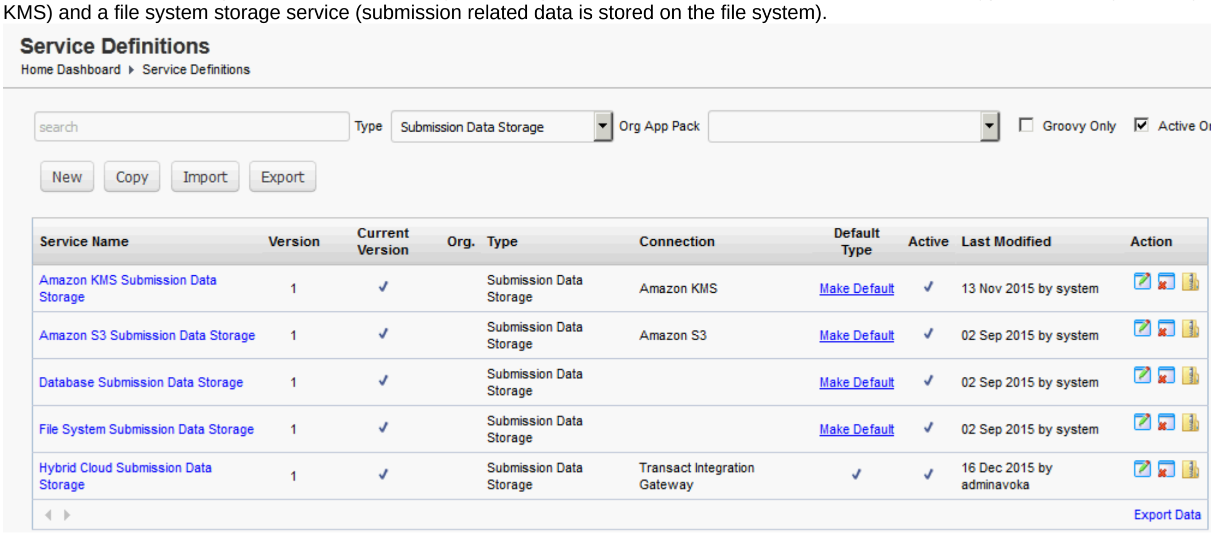
Viewing system log data retention status

For each entity that can be purged automatically, you can review the number of these entities currently stored in the system and the age of the oldest record in days and as a date.

Data Storage Configuration (Global)

In addition to data retention, you can also configure submission data storage. This means that you can choose how and where submission-related data (submission data, attachments and submission history data for saved forms) will be stored and retrieved by Transaction Manager.
By default, submission-related data is stored in the Transaction Manager database. However, data storage can be configured and changed at runtime. The functionality is encapsulated in a set of service definitions of type "Submission Data Storage". The storage service that is marked as the default for this service type will be used when storing submission data.

To view the list of storage services, go to "System > Service Definitions" and filter by service type "Submission Data Storage" (see below). By default, Transaction Manager includes a database storage service (submission-related data is held in the database), an Amazon S3 storage service (submission related data is held in S3), an Amazon KMS storage service (submission data is held in the TM database, encrypted with a key held in your Amazon KMS) and a file system storage service (submission related data is stored on the file system).



Service Definitions

Home Dashboard ▸ Service Definitions

search | Type: Submission Data Storage | Org App Pack | Groovy Only | Active Or

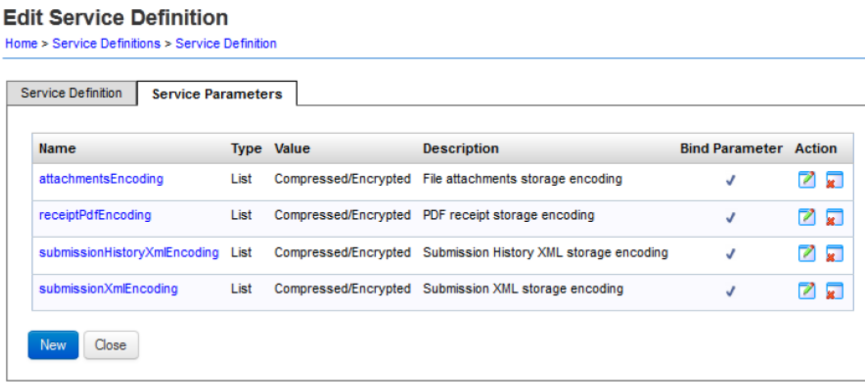
New | Copy | Import | Export

Service Name	Version	Current Version	Org.	Type	Connection	Default Type	Active	Last Modified	Action
Amazon KMS Submission Data Storage	1	✓		Submission Data Storage	Amazon KMS	Make Default	✓	13 Nov 2015 by system	
Amazon S3 Submission Data Storage	1	✓		Submission Data Storage	Amazon S3	Make Default	✓	02 Sep 2015 by system	
Database Submission Data Storage	1	✓		Submission Data Storage		Make Default	✓	02 Sep 2015 by system	
File System Submission Data Storage	1	✓		Submission Data Storage		Make Default	✓	02 Sep 2015 by system	
Hybrid Cloud Submission Data Storage	1	✓		Submission Data Storage	Transact Integration Gateway	✓	✓	16 Dec 2015 by adminavoka	

Export Data

The list of storage services

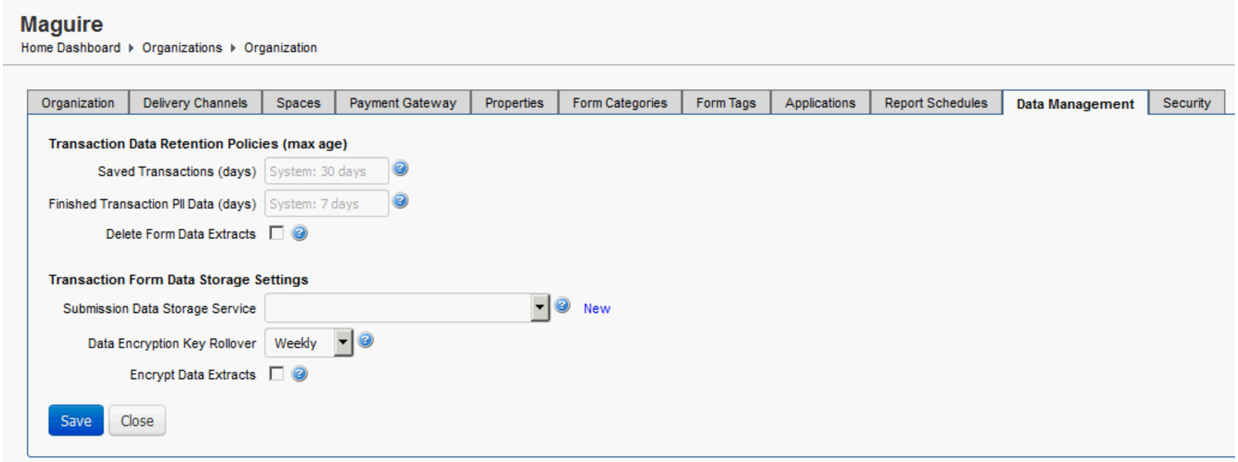
To view detailed settings for a storage service, click the "Edit" icon ( ) next to it. Switch to the "Service Parameter" tab.



Storage service parameters
For each storage service, you can define how the various submission data items (submission data, receipt data, attachments and submission history data from saved forms) will be stored.
You can opt to store the data compressed or uncompressed. In addition, you can choose to have the data encrypted. By default, the data will be both compressed and encrypted to increase security and save on storage space.

Data Retention and Storage Configuration (per Organization)

You can override important form-related settings for each organization individually. To do so, go to "Forms > Organizations" and edit the organization. Switch to the "Data Management" tab to view or change the current configuration (see below for an example).



Adjusting the data retention and storage settings for an organization
The data retention policies you can configure are the age after which saved submissions are automatically abandoned and the maximum age of data for finished (delivered or undeliverable) submissiondata. If you would like the global settings to apply for this organization, leave the field blank.

If you would like submission data extract values to be deleted as soon as the submission XML is deleted, tick "Delete Form Data Extracts". If not set, submission data extract values will be retained until the entire submission record is deleted. This setting is available only in relaxed data retention policy mode (see the knowledge base article Data Retention Management (Manager v5.0) for details on data retention).

The "Transaction Form Data Storage Settings" section allows you to customize how submission data is stored for this organization.

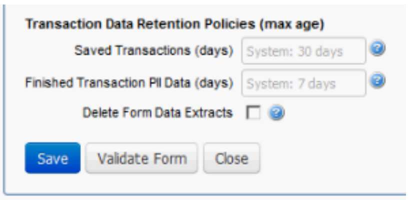
You can choose a specific submission data storage service (see Services (Manager v5.0)#_sec_submissiondatastorageservice) for this organization by selecting one from the "Submission Data Storage Service" dropdown. If left blank, the system default service will be used.

You can set a rollover interval for the security key for the organization by adjusting the value of the "Data Encryption Key Rollover" dropdown. The security key is used by TM to encrypt submission-related data. All encrypted data is associated with its security key, so data encrypted before a security key change will still be accessible.If you choose a value other than "Never", a new security key will be automatically generated by a background job when required.

Finally, you can choose to encrypt the all submission data extract values for this organization by ticking the "Encrypt Data Extracts" checkbox. If ticked, all values will be encrypted across the board; otherwise each data extract mapping can be configured to encrypt the value or not (see Form Version Configuration (Manager v5.0)#_setupdataextractmappings).
Note that changes to this setting are not applied retroactively to existing submission data extract values.

Data Retention and Storage Configuration (per Form)

You can override important form-related settings for each form individually. To do so, go to "Forms > Forms" and edit the form. Switch to the "Details" tab to view or change the current configuration in the "Transaction Form Data Retention (max age)" section.



Adjusting the data retention settings for a form

The data retention policies you can configure are time after which saved submissions will be automatcally abandoned and the maximum age of data for finished (completed and delivered or abandoned) submission data. If you would like the global/organization settings to apply for this form, leave the field blank.

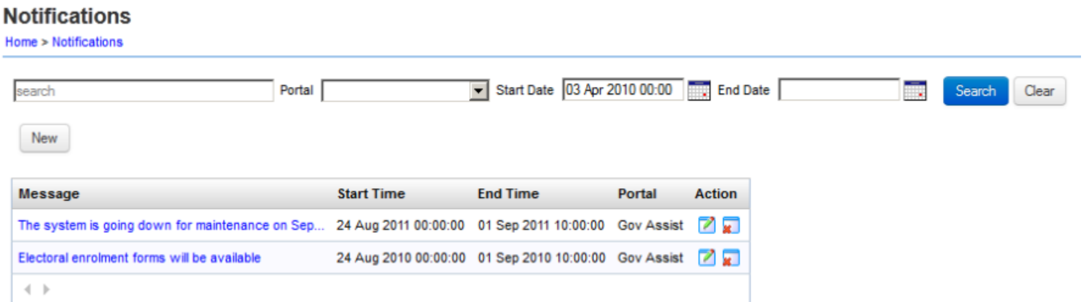
If you would like submission data extract values to be deleted as soon as the submission XML is deleted, tick "Delete Form Data Extracts". If not set, submission data extract values will be retained until the entire submission record is deleted. This setting is available only in relaxed data retention policy mode (see the knowledge base article Data Retention Management (Manager v5.0) for details on data retention).

Notification Messages

To keep your users informed about current announcements and maintenance notices, Transaction Manager offers functionality to post notification messages to the management console or a user space.

Browsing the List of Notifications

To manage notifications, go to "System > Notification Messages". You will see the list of notification messages that have been set up in Transaction Manager.



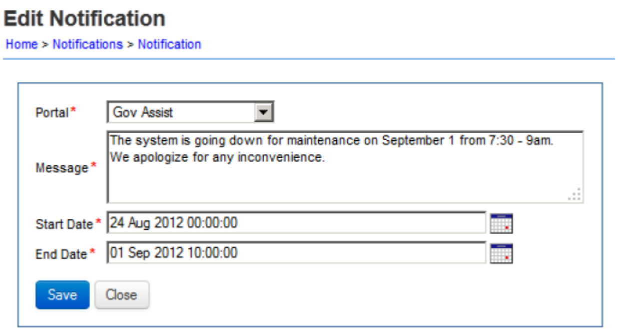
Viewing the list of notification messages

You can filter the list using a substring of the message, by space and date.

Using this page, you can create new messages and edit ( ) or delete ( ) existing messages.

Adding a Notification Message

To add a notification message for a space, go to "System > Notification Messages" and click "New". You will be directed to the "Edit Notification" page.



Creating a notification message

Choose the space on which the message should be displayed.
Enter a detailed message to be displayed to space users.
Enter a start and end date and time between which the notification should be displayed on the space.
Save the notification. After it has been saved, the notification will be displayed on the space during the specified time period.

Deleting a Notification Message

To delete a notification, go to "System > Notification Messages" and click the "Delete" icon ( ) next to the notification. You will be asked to confirm this action before the notification message is removed.

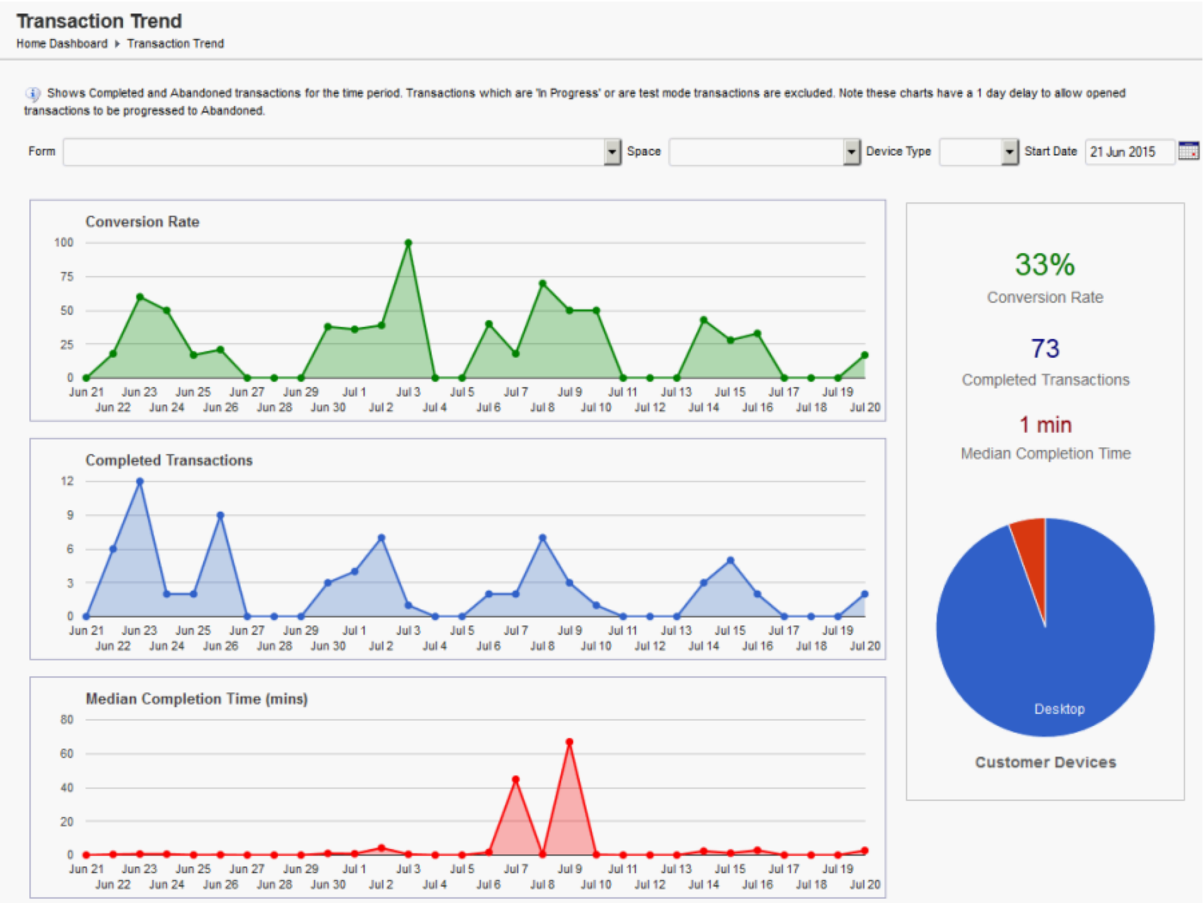
Charts

For a quick overview of submissions and form requests, you can use the charts built into Transaction Manager.
Transaction Manager provides a transaction trend showing submission statistics.

In addition, there are several individual charts built into Transaction Manager: The Form A/B Testing chart, the Submission Trend chart, the Form Submissions chart, the Referer Submissions chart, the Browser Submissions chart, the OS Submissions chart and the Reader Versions chart.
Note: Most charts are based on an additional set of database tables holding a subset of submission and request data. These tables are regularly updated, and are more lightweight than the submission/request tables themselves. The data contained in the tables can be purged automatically and independently from the submission and request tables (see _sec_dataretention).

The Transaction Trend Chart

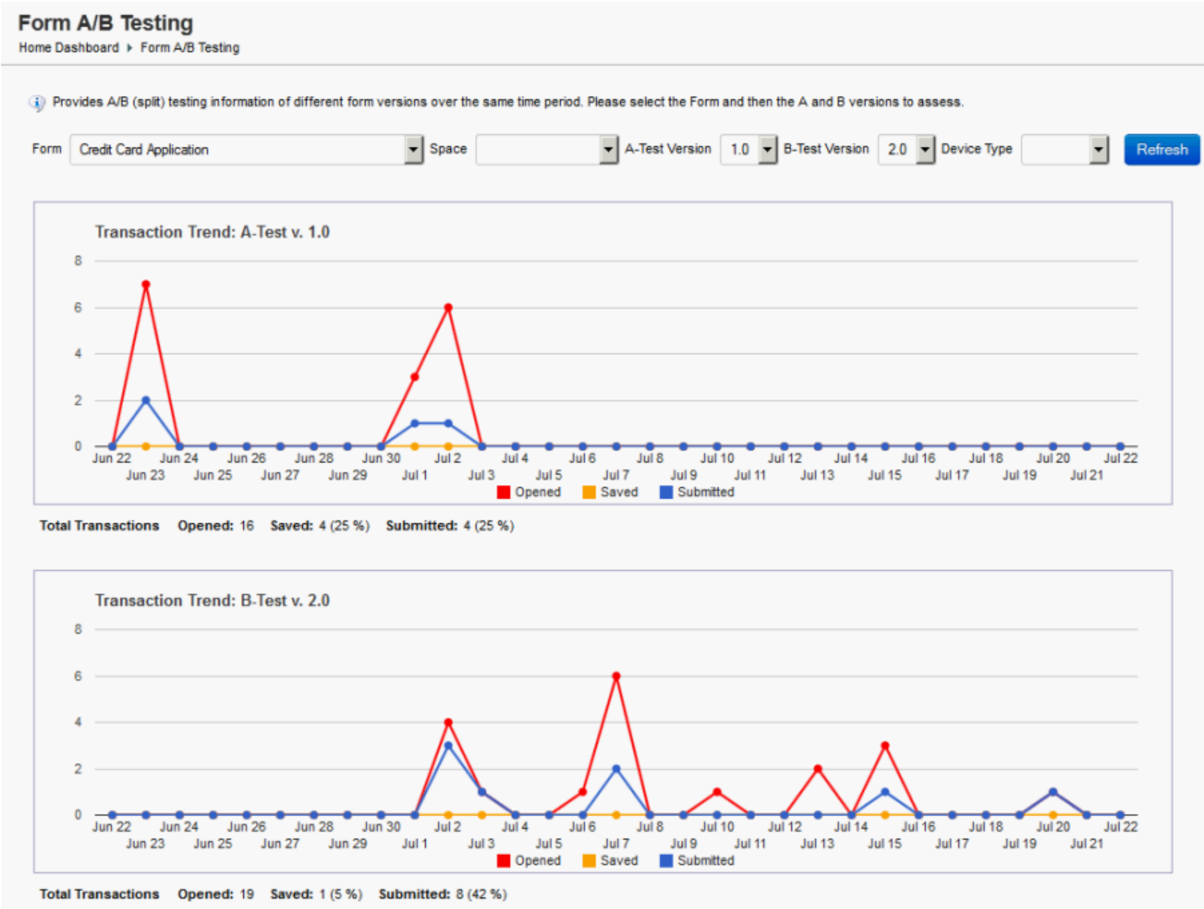
The Transaction Trend chart consists of several charts showing transaction trends in regards to conversion rate (what percentage of submissions were completed by users), number of transactions and median time taken by users to complete a submission. A pie chart shows the distribution of device types (desktop, tablet, phone).
To access the Transaction Trend chart, go to "Analytics > Transaction Trend". You can filter the charts by organization, form, space, device type and date.



The Transaction Trend chart

The Form A/B Testing Chart

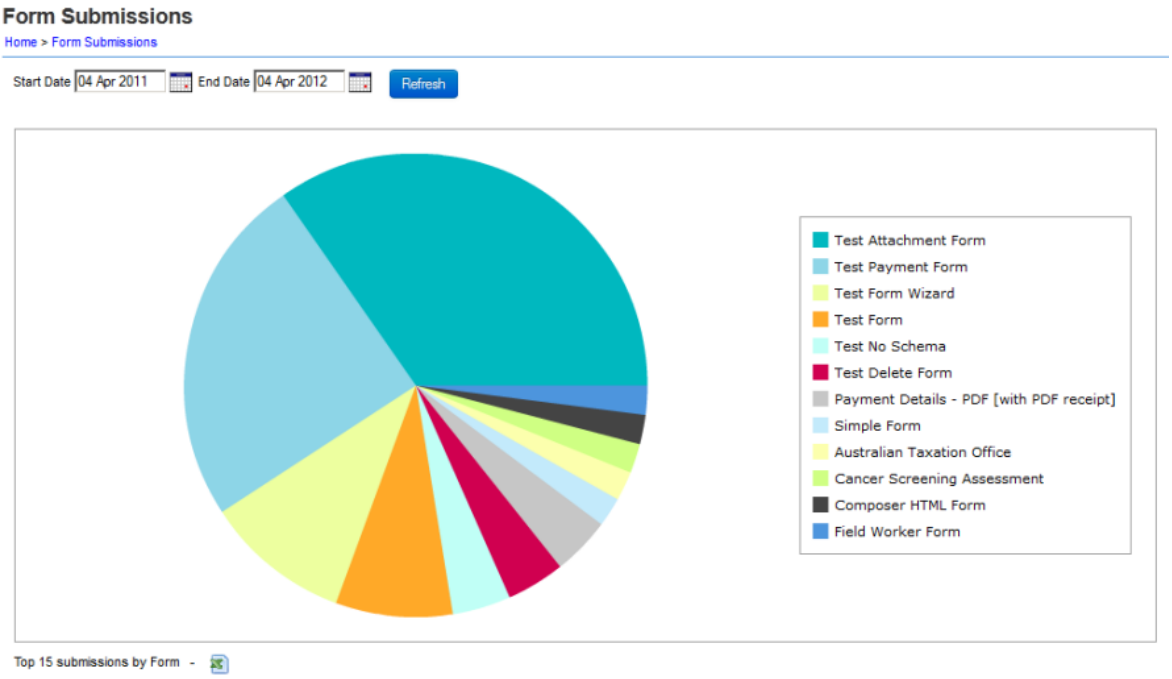
The Form A/B Testing chart can be used to compare submission statistics for two form versions side by side.
To access the Form A/B Testing chart, go to "Analytics > Form A/B Testing". Select the form and the two versions you want to compare to show data for both versions for the past month. You can also filter the data by space and device type (desktop, tablet or phone).



The Form A/B Testing chart

The Form Submissions Chart

The Form Submissions chart shows a pie chart of the number of all submissions per form during a certain date range. The chart can be filtered by organization.
To access the Form Submission chart, go to "Analytics > Form Submissions". If you want to modify the date range used, enter new dates into the "Start Date" and "End Date" fields and click "Refresh".
You can export the search results to Excel by clicking the "Export Data" icon below the chart.

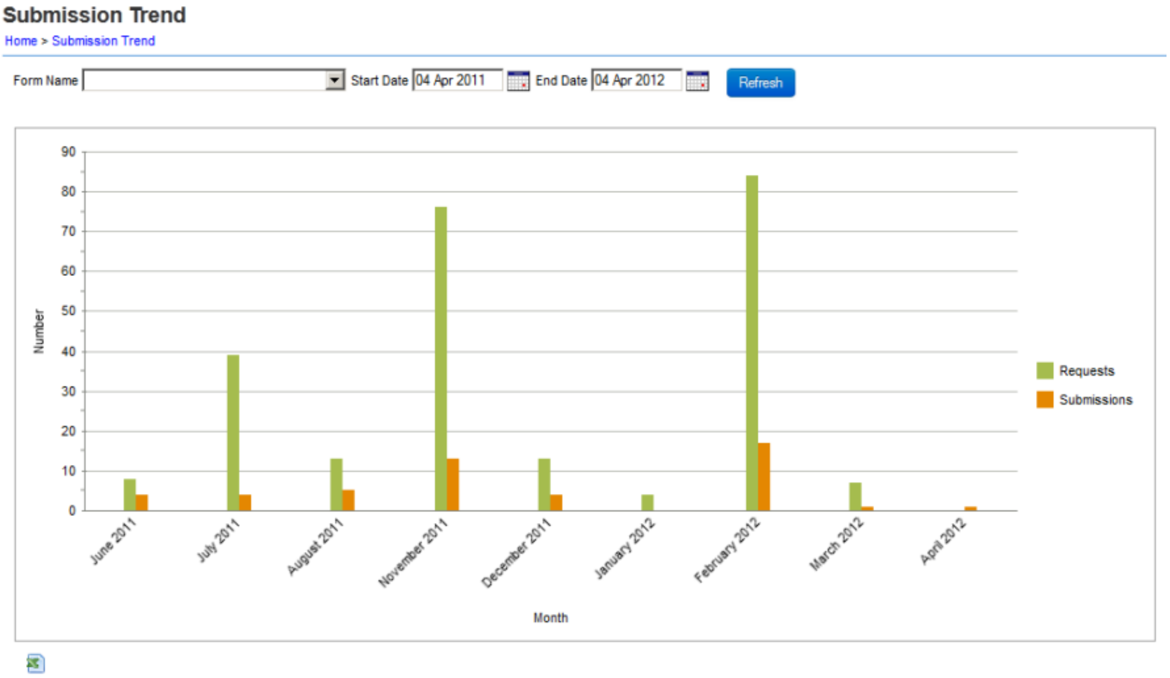


The Form Submissions chart

The Submission Trend Chart

The Submission Trend Chart shows the number of form requests and submissions for all forms during a date range. The data is grouped by month, and can be filtered by organization or form.
To access the Submission Trend chart, go to "Analytics > Submission Trend". If you want to modify the form or date range used, select a form from the

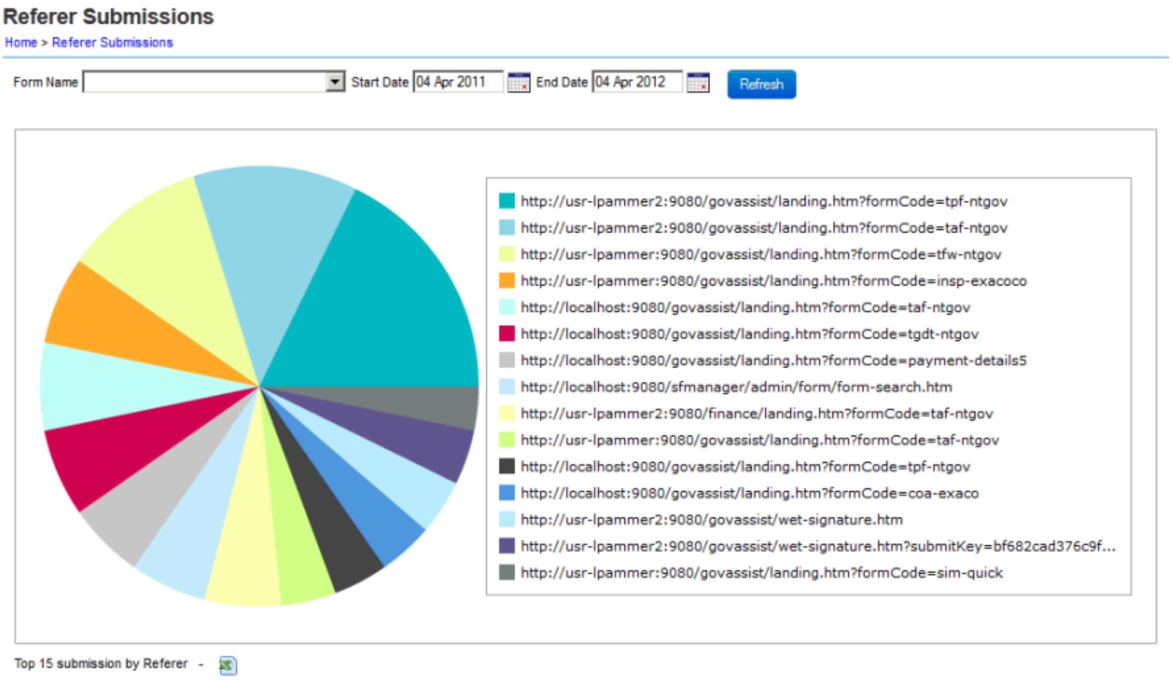
"Form Name" field, enter new dates into the "Start Date" and "End Date" fields and click "Refresh".
You can export the search results to Excel by clicking the "Export Data" icon below the chart.



The Submission Trend chart

The Referer Submissions Chart

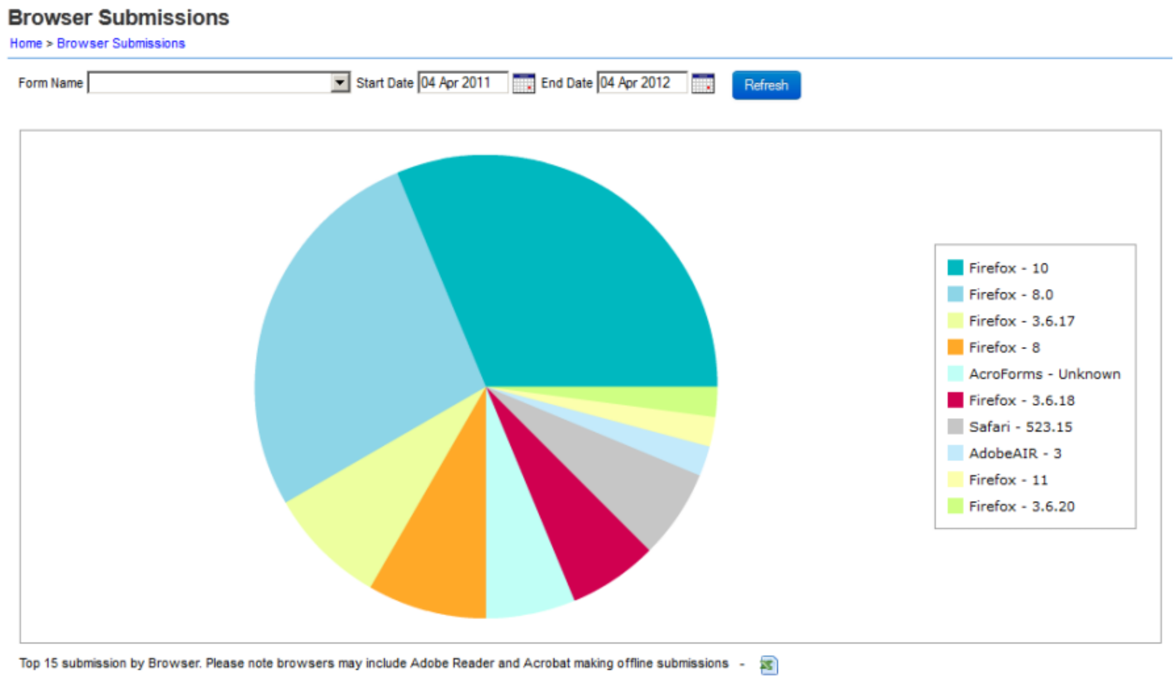
The Referer Submissions chart shows the referer URLs (URLs from which the space users navigated to the form or landing page) as a pie chart. It also lists the referring URLs along with a percentage.
To access the Referer Submissions chart, go to "Analytics > Referer Submissions". To filter by form, choose a value from the "Organization" dropdown and select a value from the "Form Name" dropdown. If you want to modify the date range used, enter new values into the "Start Date" and "End Date" fields and click "Refresh".



The Referer Submissions chart

The Browser Submissions Chart

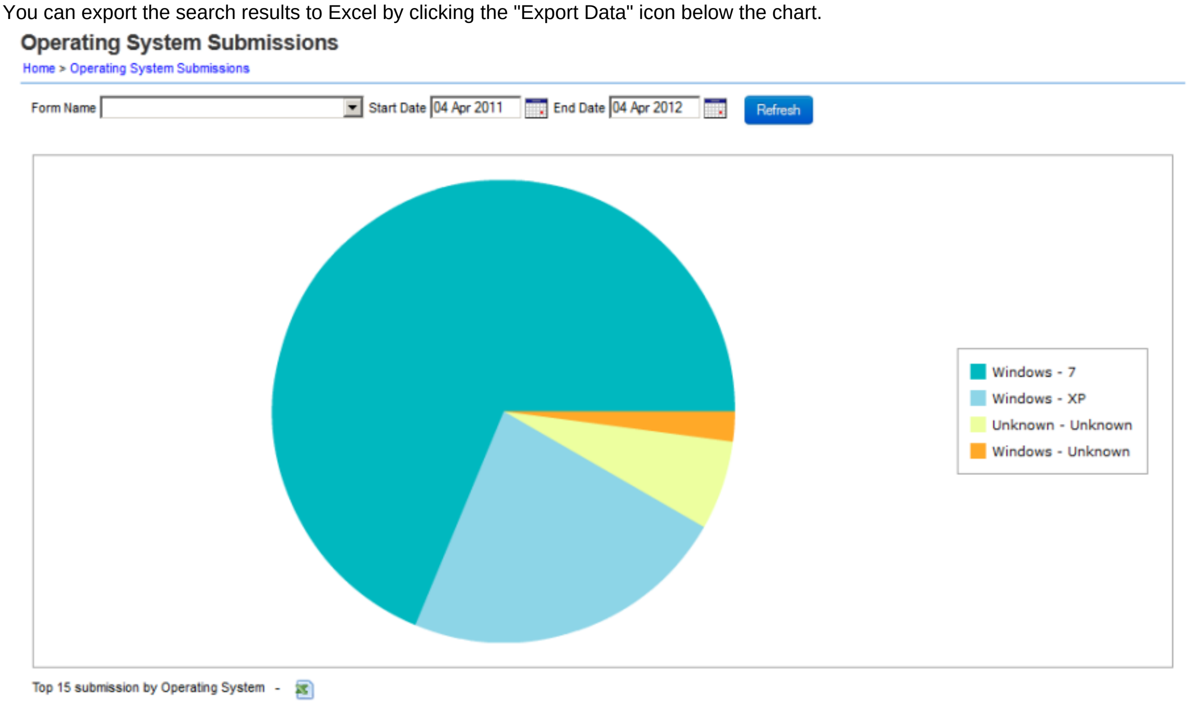
The Browser Submissions chart shows a breakdown of the most common browser versions used to submit forms. The data can be filtered by organization, form and date.
To access the Browser Submissions chart, go to "Analytics > Browser Submissions". To filter by form, choose a value from the "Organization" dropdown and select a value from the "Form Name" dropdown. If you want to modify the date range used, enter new values into the "Start Date" and "End Date" fields and click "Refresh".
You can export the search results to Excel by clicking the "Export Data" icon below the chart.



Browser Submissions chart

The Operating System Submissions Chart

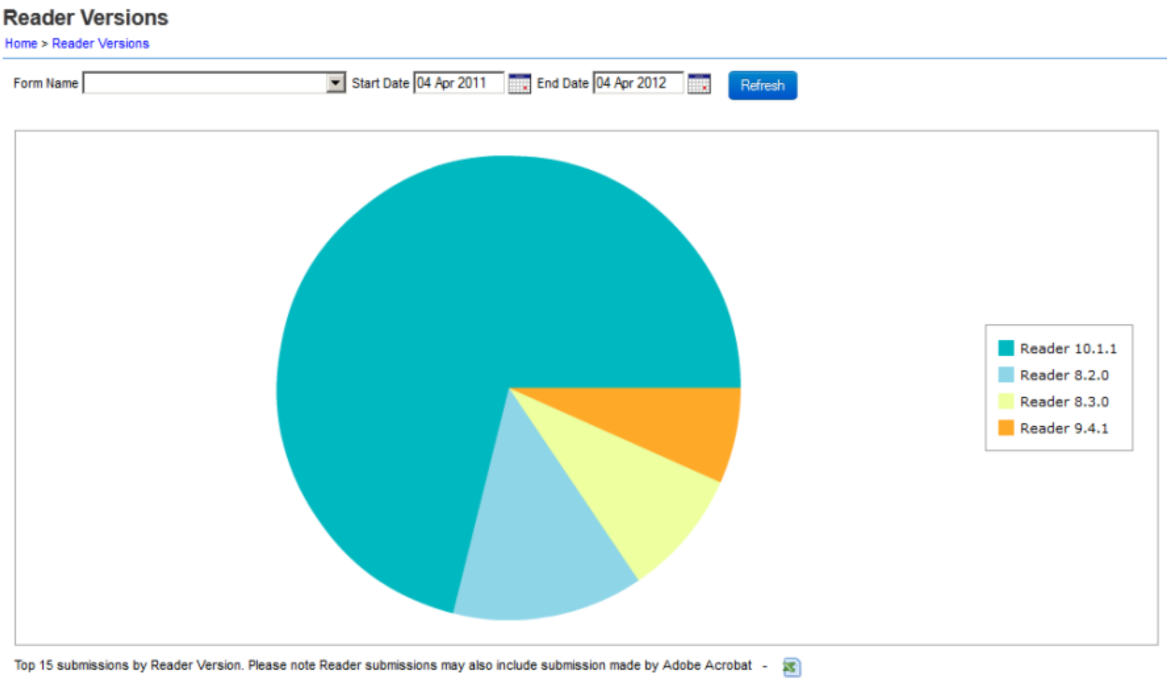
The OS Submissions chart shows a breakdown of the most common operating systems used to submit forms. The data can be filtered by organization, form and date.
To access the Operating System Submissions chart, go to "Analytics > OS Submissions". To filter by form, choose a value from the "Organization" dropdown and select a value from the "Form Name" dropdown. If you want to modify the date range used, enter new values into the "Start Date" and "End Date" fields and click "Refresh".
You can export the search results to Excel by clicking the "Export Data" icon below the chart.



The Operating System Submissions chart

The Reader Versions Chart

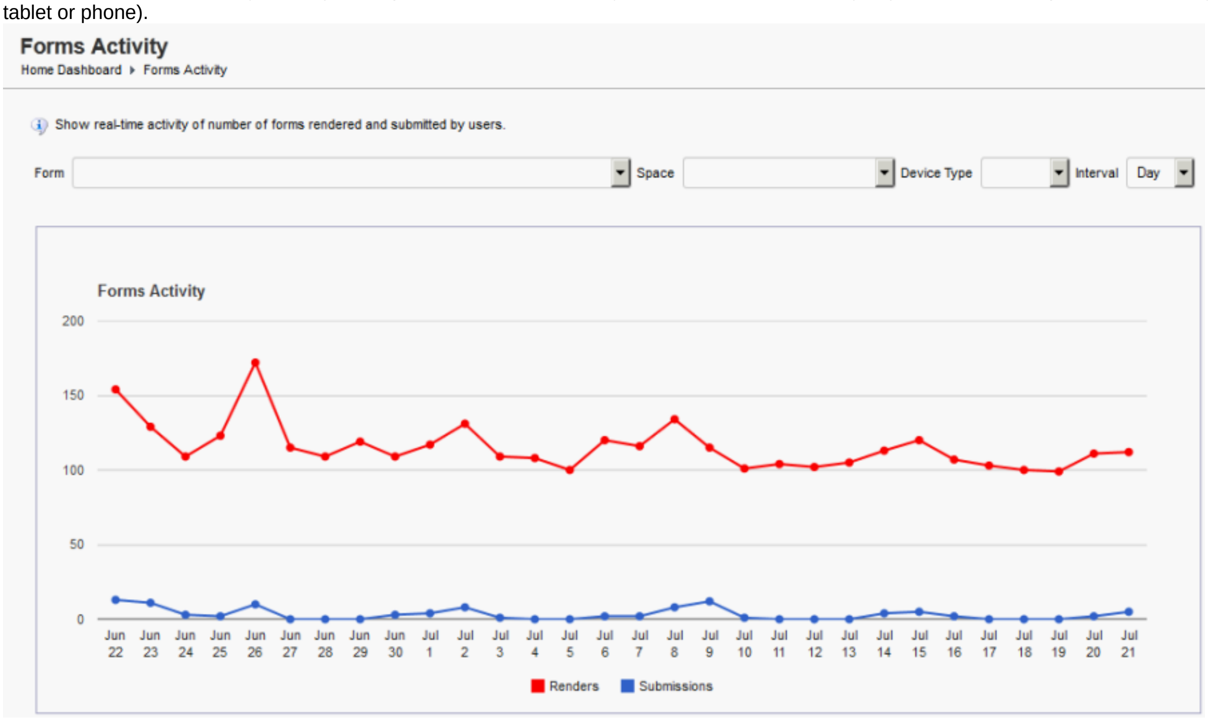
The Reader Versions chart shows a breakdown of the most common Adobe Reader versions used to submit forms. The data can be filtered by organization, form and date.
To access the Reader Versions chart, go to "Analytics > Reader Versions". To filter by organization, select a value from the "Organization" dropdown. If you want to modify the date range used, enter new values into the "Start Date" and "End Date" fields and click "Refresh".



The Reader Versions chart

The Forms Activity Chart

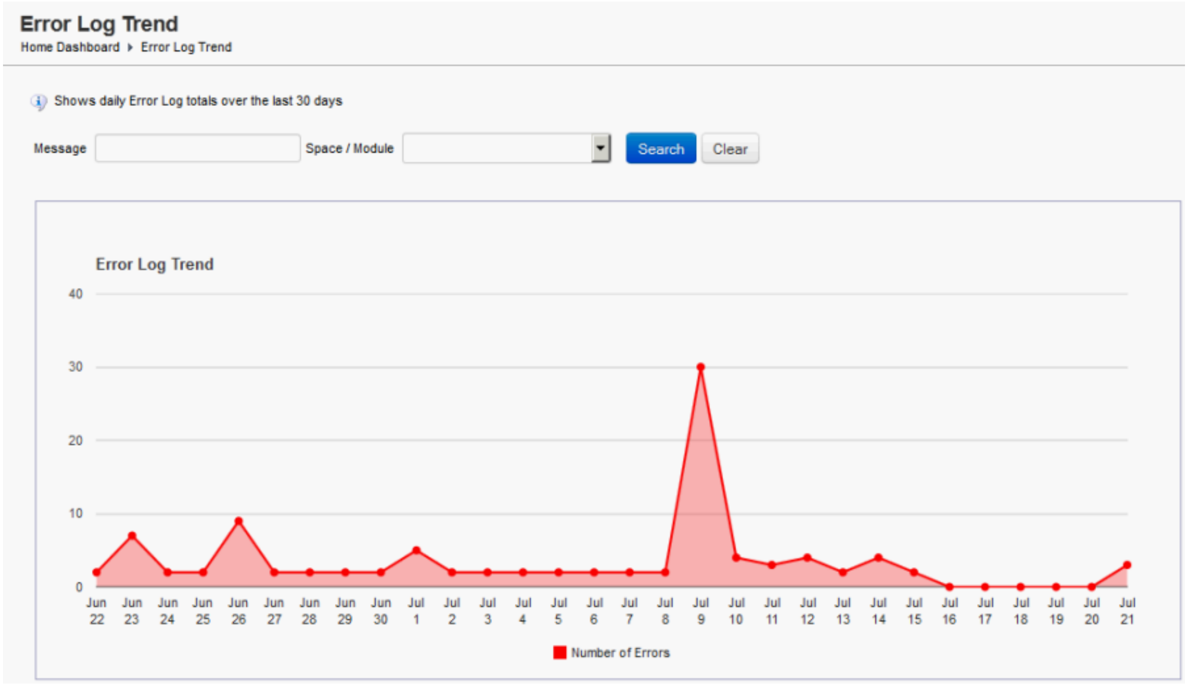
The Forms Activity chart is based on live submission data to show recent activity in TM. It shows the number of render requests and submissions for the last month (or day depending on the interval selected).
To access the Forms Activity chart, go to "Operations > Forms Activity". You can filter the data by organization, form, space and device type (desktop, tablet or phone).



The Forms Activity chart

The Error Log Trend Chart

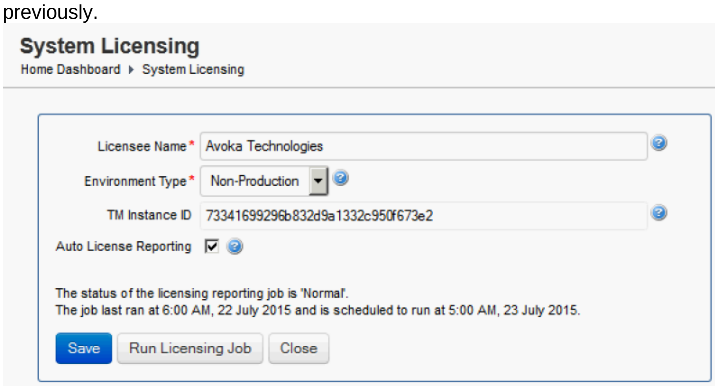
The Error Log Trend chart shows the number of error log entries logged in TM for the last month. It provides a quick way of checking on the health of the TM instance.
To access the Error Log Trend chart, go to "System > Error Log Trend". You can filter the data by space and a substring of the error message.



The Error Log Trend chart

System Licensing

The TM licensing model includes automated reports sent to an Avoka server, containing daily summaries on the number of submissions, broken down further by organization and some other metrics (e.g. how many transactions involving T.Field were completed).
The licensing data that is collected and reported on can be viewed in the transaction log (see _sec_transactionlog).
The licensing reports that are sent to a secure Avoka server need to be configured with some information about the TM system such as environment name and type. To view the current settings, go to "System > System Licensing". These settings are also displayed if they have not been filled in previously.



System licensing settings

To configure licensing details for your server, enter the licensee name (your company name + the server identification, e.g. "My Company UAT") and choose the server type (production or non production). Save the changes.
By default licensing reporting data is automatically transmitted securely to the Avoka reporting server. If this mechanism is not suitable for your TM server, please contact Avoka to obtain a key for manual license reporting for your server (you can untick "Auto License Reporting" and click the link to request a license key via email).Once you receive your manual license reporting key, go to "System > System Licensing" again, untick "Auto License Reporting" and enter your key into the "Manual License Reporting Key" field. Also enter the directory on the TM server where licensing reports shall be stored into the "License Report Directory" field. TM must be able to write to this directory.Save your changes to switch to manual licensing reporting. TM will store the licensing reports to the configured folder, and you will need to provide these report files to Avoka when requested.

Outgoing Email (Manager v5.0)

Transaction Manager uses email for a variety of notifications to both administrators and space users. This chapter talks about configuring SMTP server information and how emails can be tracked and sent using the management console.

Page Contents

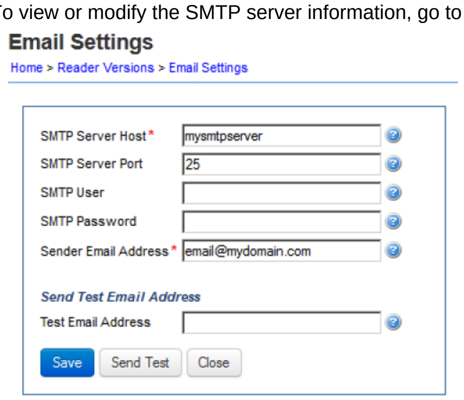
Search

Concepts

Transaction Manager requires an SMTP server to send its emails. Please note that Transaction Manager does not currently process incoming emails. Emails are either sent directly as part of some processes (e.g. delivery, form promotion, scheduled reports) or alternatively can be scheduled on an *email queue*. The email queue provides a mechanism to send arbitrary emails, keeping track of emails and their current status and sending them with a configurable number of retries.

Configuring SMTP Server Information

For Transaction Manager to be able to send email, it needs to be configured with connection parameters to an SMTP server.
To view or modify the SMTP server information, go to "System > Email Settings" (see below for what you should be seeing).

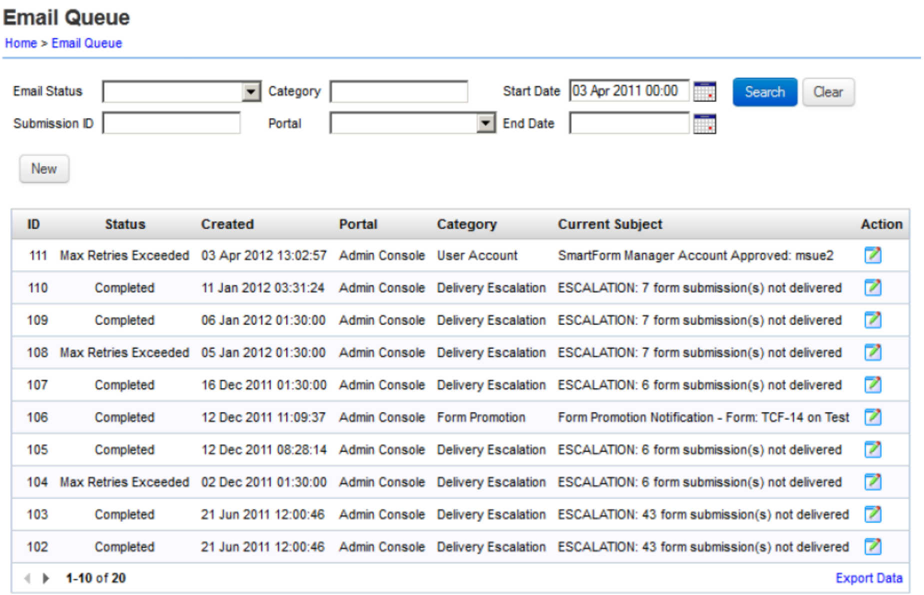


/

The SMTP configuration for Transaction Manager
Edit the fields to set up the connection to the SMTP server, then save.
You can send a test email by entering a valid email address into the "Test Email Address" field and clicking "Send Test".

The Email Queue

The email queue enables administrators or Transaction Manager services to create emails, which will be automatically sent, retried if necessary and logged. This helps track whether emails actually get sent and allows you to resolve errors such as invalid recipients.
To view the email queue, go to "System > Email Queue". You will see the list of emails in the queue (including those that have been sent already).
You can filter the list of emails by status ("Ready", "Completed", "Error", "Max Retries Exceeded" or "Purged"), a substring of the category, the submission ID associated with the email, the space and the creation date.You can also export the search results to Excel by clicking the "Export Data" link below the table.


Email Queue
Home > Email Queue

Email Status | Category | Start Date 03 Apr 2011 00:00 | Search | Clear
Submission ID | Portal | End Date

New

ID	Status	Created	Portal	Category	Current Subject	Action
111	Max Retries Exceeded	03 Apr 2012 13:02:57	Admin Console	User Account	SmartForm Manager Account Approved: msue2	
110	Completed	11 Jan 2012 03:31:24	Admin Console	Delivery Escalation	ESCALATION: 7 form submission(s) not delivered	
109	Completed	06 Jan 2012 01:30:00	Admin Console	Delivery Escalation	ESCALATION: 7 form submission(s) not delivered	
108	Max Retries Exceeded	05 Jan 2012 01:30:00	Admin Console	Delivery Escalation	ESCALATION: 7 form submission(s) not delivered	
107	Completed	16 Dec 2011 01:30:00	Admin Console	Delivery Escalation	ESCALATION: 6 form submission(s) not delivered	
106	Completed	12 Dec 2011 11:09:37	Admin Console	Form Promotion	Form Promotion Notification - Form: TCF-14 on Test	
105	Completed	12 Dec 2011 08:28:14	Admin Console	Delivery Escalation	ESCALATION: 6 form submission(s) not delivered	
104	Max Retries Exceeded	02 Dec 2011 01:30:00	Admin Console	Delivery Escalation	ESCALATION: 6 form submission(s) not delivered	
103	Completed	21 Jun 2011 12:00:46	Admin Console	Delivery Escalation	ESCALATION: 43 form submission(s) not delivered	
102	Completed	21 Jun 2011 12:00:46	Admin Console	Delivery Escalation	ESCALATION: 43 form submission(s) not delivered	

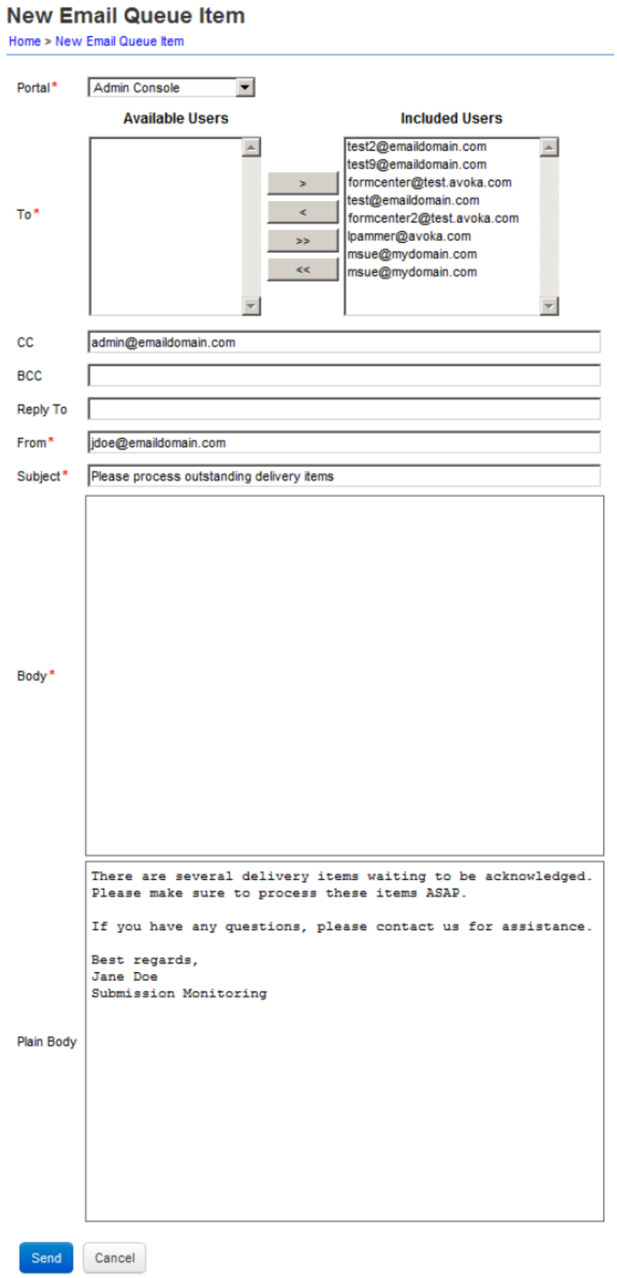
1-10 of 20 Export Data

Viewing the email queue

This page allows you to create new emails and edit ( ) existing emails. These operations are described in the sections below.

Adding an Email to the Queue

To add a new email, click "New". This will take you to the "New Email Queue Item" page.



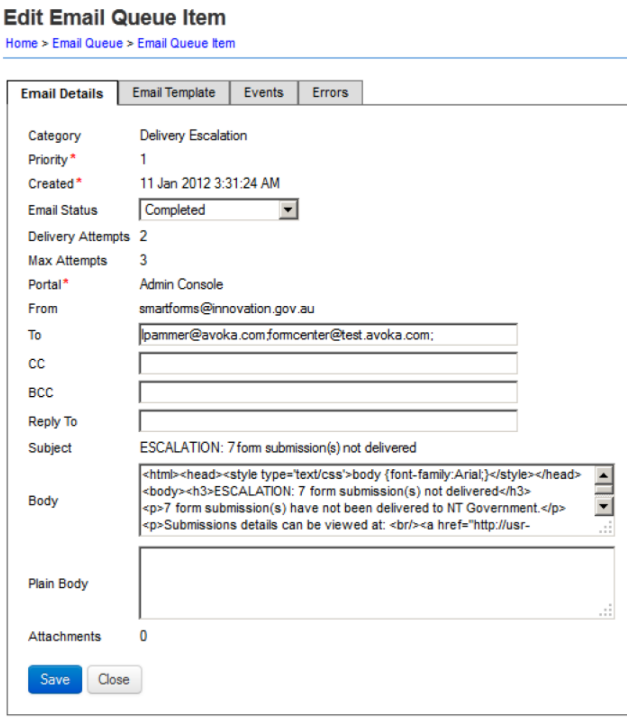
Adding an email to the email queue

Select a space to whose users you wish to send email. This will populate the "Available Users" list with all users associated with the space.
Select one or more of the email addresses from "Available Users" and use ">" to add them as recipients. To add all users, use ">>". The "<" and "<<" buttons can be used to remove a selection of or all users from the recipients list.
You can also enter additional email addresses into the "CC" and "BCC" fields.
Edit the "From" field if you would like to use a custom sender address. By default, this field is populated with the value of the "Email Default Sender" deployment property (which can be edited using "System > Email Settings"; see also section "Configuring SMTP Server Information").
The "Reply To" field can be used to set a reply address that is different to the "From" address. It will be used when a user replies to an email sent out via the email queue.
Enter the email subject into the "Subject" field.
Enter the email message into the "Body" (for HTML email content) and/or the "Plain Body" (for plain text content) fields.
When you are done, save the email. It will be automatically sent by the Email Queue job. If there is an error, there will be a maximum of two automatic retries. Refer to the following section for getting information on the email progress and how to retry sending the email.

Editing an Email

Once an email has been added to the queue, it can be edited using the "Edit" icon (

) next to the email. The screenshot below shows the "Edit Email Queue Item" page.



Editing a queued email

The "Email Details" tab allows you to edit some of the fields such as the recipients and email body. This allows you to resolve email delivery issues.In addition, if an email was resent unsuccessfully so often that it has exceeded the maximum number of retries, you can reset its status by changing the "Email Status" field to "Ready".

Once you are done editing the email, save your changes.

The "Email Template" tab contains various template values from which the actual email fields are calculated (if applicable). The actual values are calculated when the email is being sent.

The "Events" and "Errors" tabs show events and errors that occurred when attempting to send the email. This can be useful for debugging email failures.

Reports (Manager v5.0)

Transaction Manager includes extensive reporting capabilities. You can set up your own reports that generate submission statistics, trends and help organizations monitor user activities. Transaction Manager uses the BIRT framework to run reports.
These reports can be viewed on screen or scheduled to be automatically emailed to one or more recipients.

Search

Concepts

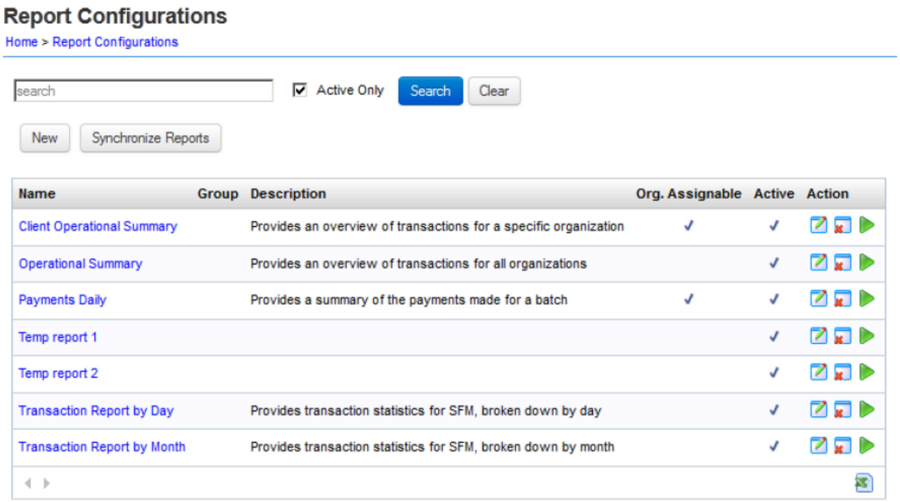
A *report* in Transaction Manager consists of a BIRT report file as well as *report parameters* that are passed into the report at runtime. There are two types of reports: *System reports* are global reports that are not organization specific, whereas *organization assignable reports* are reports that display data for a particular organization.
Each report allows you to view the report on screen (i.e. as HTML) as well as to define multiple *report schedules*. Report schedules are used to automatically generate and email the current report for a particular date range. Reports can be sent out in PDF, CSV or XLS (Microsoft Excel) format.

Configuring Reports

This section describes how to set up and run reports in Transaction Manager. For a new report, you need a BIRT report file to base the report on. You can create and upload your own reports without having to rebuild Transaction Manager, albeit with some restrictions as described below.

Browsing the List of Reports

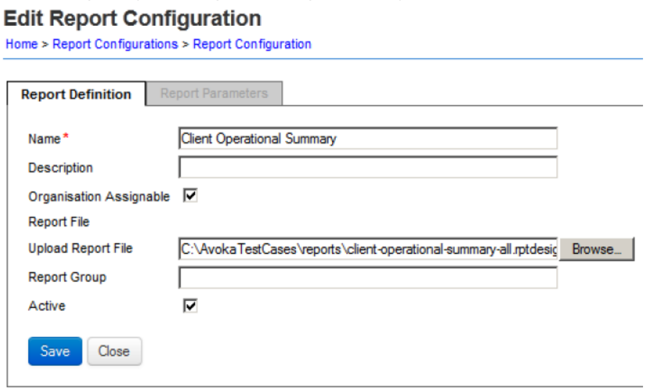
To view the list of reports set up in Transaction Manager, go to "Reports > Report Configurations".



Browsing the list of reports
You can filter this list a substring of the report name. In addition, you can choose to show inactive reports by unticking the "Active Only" checkbox.You can also export the search results to Excel by clicking the "Export Data" link below the table.
This page allows you to create new reports and edit ( ) or delete ( ) existing ones, as well as rendering reports on-screen ( ).
You can also click "Synchronize Reports" to recreate the report directory where Transaction Manager stores all reports so BIRT can access them. Doing this will delete the contents of the report directory and create a folder for each report containing the report design as uploaded by the administrator.

Adding a Report

To add a report, go to "Reports > Report Configurations" and click the "New" button. You will be directed to the "Edit Report Configuration" page.



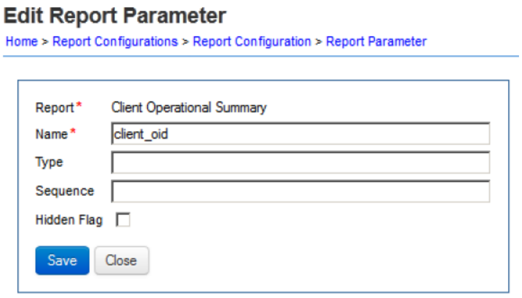
Creating a report

Give the report a unique name and (optionally) a description.
The "Organization Assignable" checkbox determines whether your report should be schedulable for specific organizations or whether it is a system report.
Upload a BIRT report design file (*.rptdesign) using the "Browse" button next to "Upload Report File". The file will be stored in the database as well as the reports directory on the file system so it is accessible to BIRT (the reports directory is configured via the deployment property "Config Directory"; see also Deployment Properties (Manager v5.0)#_sec_directories)
You can enter the category to which the report belongs (e.g. "Global Submission Statistics") into the "Report Group" field. Specifying a report group is currently purely descriptive, i.e. Transaction Manager is not using this information, but merely stores it.
The "Active" checkbox can be used to mark a report as active or obsolete. To run or schedule the report, make sure the checkbox is ticked.
Save the report and verify that it is shown in the list of reports. Before you can use it, you may need to add one or more report parameters.

Setting Up Report Parameters

Report parameters are passed into the BIRT report when it is being rendered on-screen (there are also invisible parameters that are passed into a report when it is generated and emailed by the report scheduling component – these are discussed in _sec_reportscheduleexecution).
At present, there is only one parameter that Transaction Manager supports here, which is "client_oid". If the report you are setting up is organization assignable, add a report parameter named "client_oid" to the Transaction Manager report. At render time, the organization's ID is automatically retrieved and passed into the report (see _sec_renderorganizationreport on how to render an organization assignable report).
Please note: For this to work, your BIRT report needs to define a parameter named "client_oid" and use that parameter in its queries to retrieve organization-specific data.

To add the parameter, go to "Reports > Report Configurations" and click the "Edit" icon ( ) next to the report. On the "Edit Report Configuration" page, switch to the "Report Parameters" tab and click the "New" button. You will be directed to the "Edit Report Parameter" page.



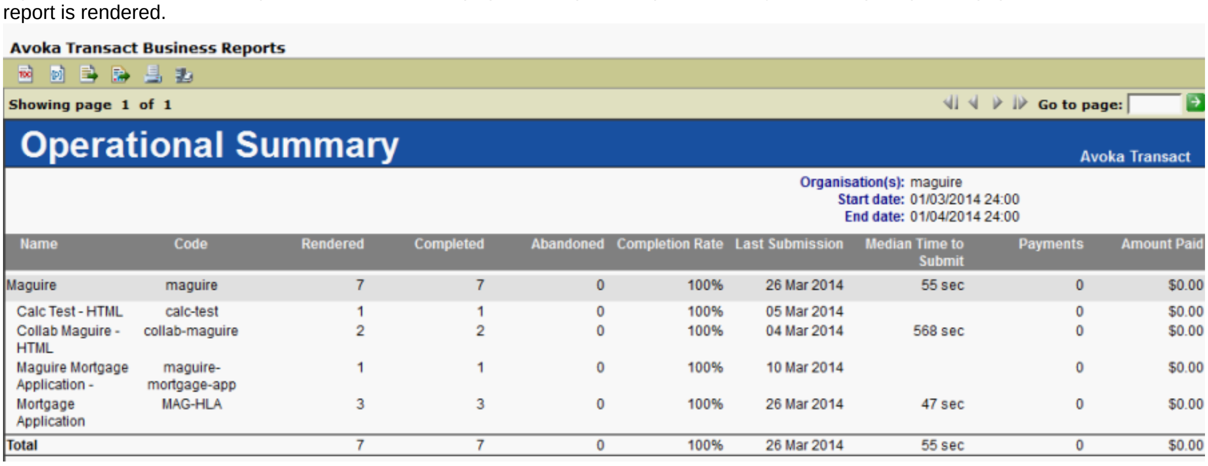
Adding a report parameter
Enter the parameter name (in this case "client_oid", as explained above) into the "Name" field. The other fields can remain blank.
Save the report parameter. Now the report is fully configured and can be rendered on-screen.

Configuring Users to View Reports

The BIRT reporting engine is set up as a space in Transaction Manager. For a user to be able to view reports, they must be associated with the "Business Reports" space (see Security (Manager v5.0)#_sec_changespacesforuseraccount on how to do this). In addition, when viewing reports on-screen, the user needs to log on with their credentials.

Rendering a System Report

To render a system report on-screen, go to "Reports > Report Configurations" and click the "View" icon ( ) next to the report. This will render the report in the current window (see below for an example). If the report has parameters, you will be prompted to populate them with values before the report is rendered.



Avoka Transact Business Reports

Showing page 1 of 1 — Go to page:

Operational Summary — Avoka Transact

Organisation(s): maguire
Start date: 01/03/2014 24:00
End date: 01/04/2014 24:00

Name	Code	Rendered	Completed	Abandoned	Completion Rate	Last Submission	Median Time to Submit	Payments	Amount Paid
Maguire	maguire	7	7	0	100%	26 Mar 2014	55 sec	0	$0.00
Calc Test - HTML	calc-test	1	1	0	100%	05 Mar 2014		0	$0.00
Collab Maguire - HTML	collab-maguire	2	2	0	100%	04 Mar 2014	568 sec	0	$0.00
Maguire Mortgage Application -	maguire-mortgage-app	1	1	0	100%	10 Mar 2014		0	$0.00
Mortgage Application	MAG-HLA	3	3	0	100%	26 Mar 2014	47 sec	0	$0.00
Total		7	7	0	100%	26 Mar 2014	55 sec	0	$0.00

Rendering a report
The BIRT report viewer packaged with Transaction Manager renders the report as HTML. The viewer allows you to re-run the report, export the data and print the report.

Rendering an Organization Assignable Report

Organization assignable reports typically have a hidden parameter "client_oid" as described in _sec_setupreportparameters. To ensure that the correct organization ID is passed into the report when it is rendered, go to "Reports > Report Configurations" and select the desired organization from the "Organization" dropdown, Then click the "View" icon ( ) next to the report.

Removing a Report

You can delete reports and report parameters if needed. Be aware that deleting a report will also remove all schedules associated with it.

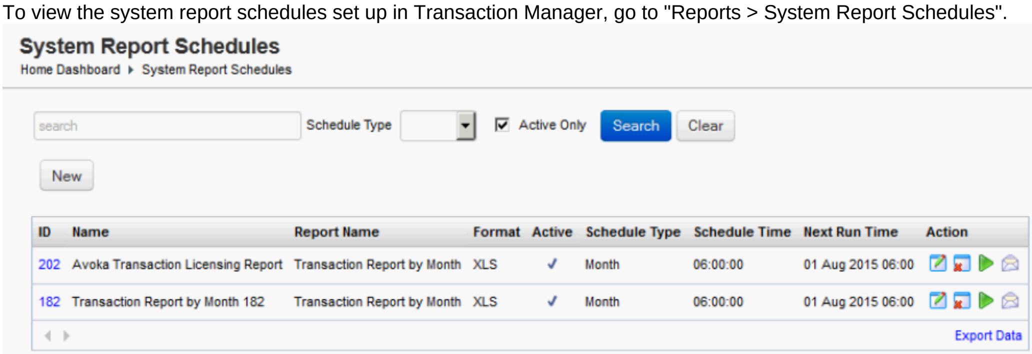
To delete a report, go to "Reports > Report Configurations" and click the "Delete" icon ( ) next to the report. After you confirm your decision, the report and all its schedules will be irrevocably deleted.
If you do not want to physically delete the report, you can also make it inactive. To do this, go to "Reports > Report Configurations" and click the "Edit" icon ( ) next to the report. On the "Edit Report" page, untick the "Active" checkbox and save. This marks the report as obsolete; you can later reactivate it by simply re-ticking the checkbox.

Scheduling Reports

This section describes how you can set up schedules to automatically run reports and send them out via email.

Browsing System Report Schedules

To view the system report schedules set up in Transaction Manager, go to "Reports > System Report Schedules".

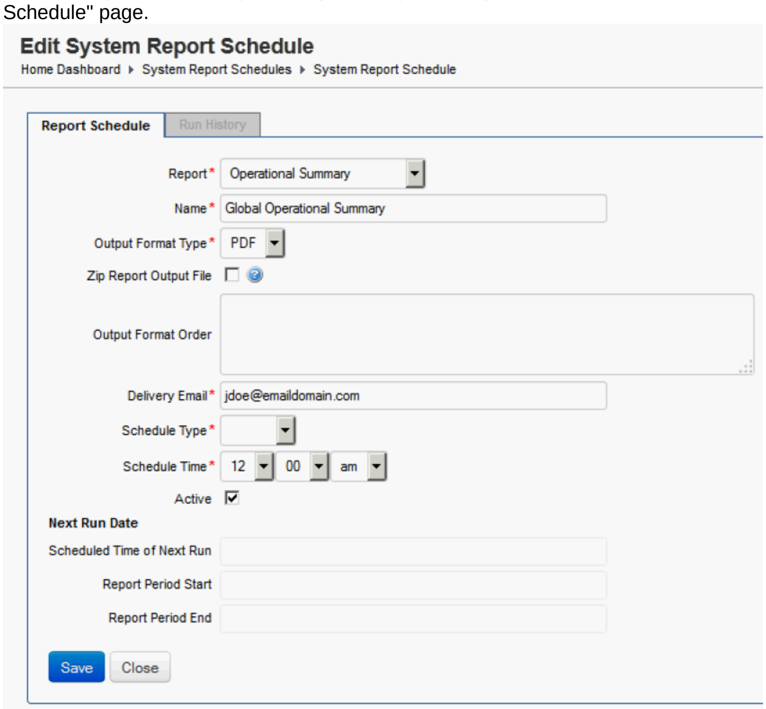


Viewing the list of system report schedules
You can filter this list using a substring of the report name and by schedule type (day, week or month). If you tick the "Active Only" checkbox, only report schedules that have the "Active" flag set will be displayed.You can also export the search results to Excel by clicking the "Export Data" link below the table.

This page allows you to create new system report schedules and trigger ( ), edit ( ) and delete ( ) existing report schedules as well as render the report on-screen ( ).

Adding a System Report Schedule

To add a report schedule, go to "Reports > System Report Schedules" and click the "New" button. You will be directed to the "Edit System Report Schedule" page.



Creating a system report schedule

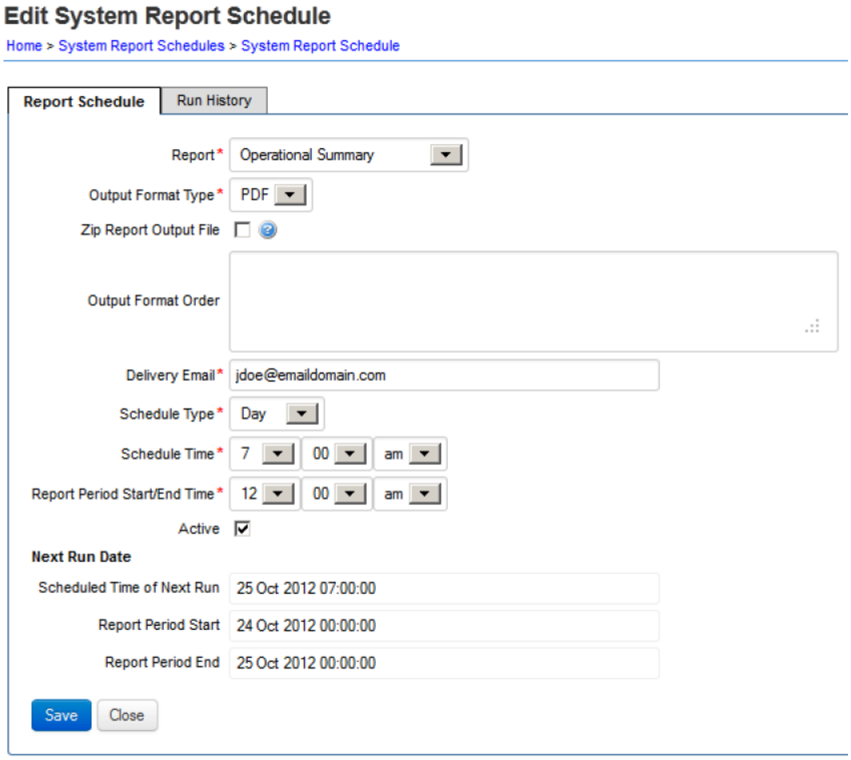
Configuring a Report Schedule

Select the report you want to schedule from the "Report" dropdown. If you are creating a system report schedule, the dropdown will contain only system reports; if you are scheduling an organization assignable report, you will be offered only organization assignable reports.
Enter a globally unique report name.
The "Output Format Type" can be PDF, CSV or XLS. If you want your reports to be easily readable by humans, choose PDF; if you want to process

them automatically, CSV is the better choice. Choose XLS to obtain the report in Microsoft Excel format.
The checkbox "Zip Report Output File" allows you to automatically compress the report file and attach the resulting ZIP file to the report email rather than the uncompressed original. This setting helps when report files are getting very large, especially with CSV and XLS reports.
The "Output Format Order" is relevant only if you have chosen CSV as the output format. It allows you to specify the order in which the columns extracted from the report are written to the CSV file. For some more details on this, see _sec_reportscheduleexecution.
Enter the email addresses you want the report sent to into the "Delivery Email" field, using commas to separate them.
The "Active" checkbox at the bottom of the form indicates that the report schedule is ready for use; untick this only if you want to deactivate the schedule to stop reports from being emailed out automatically.
All other fields contain scheduling information. You have a choice of three basic schedule intervals: daily, weekly and monthly. See below on how to configure these schedule types. Please also refer to _sec_reportscheduleexecution to understand how the configured report periods are passed into the report.

Setting Up a Daily Schedule

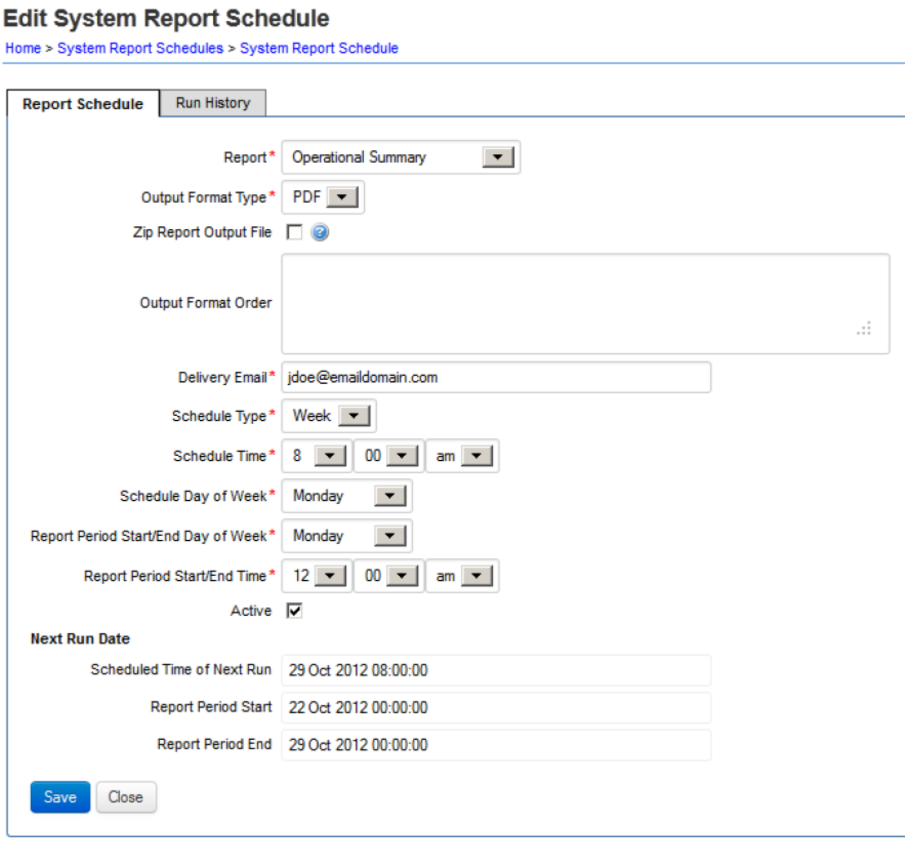
Daily schedules, as their name implies, are run once a day. If the report is configured to generate results for a certain date range, Transaction Manager will run it for a 24-hour period. This means that every day, the recipient(s) will get a report containing one day's worth of data.
The screenshot below shows an example configuration for a daily report schedule.



A daily report schedule
Set the "Schedule Type" to "Day". Set the "Schedule Time" to the time that you want the report to run (format HH:MM:SS).
Set "Report Period Start/End Time" to a time of day (format HH:MM:SS). If you use "12:00:00", each report will contain data from 12pm of one day to 12pm on the following day.

Setting Up a Weekly Schedule

Weekly schedules are run once a week. If the report is configured to generate results for a certain date range, Transaction Manager will run it for a 7-day period. This means that every week, the recipient(s) will get a report containing one week's worth of data.
The screenshot below shows an example configuration for a weekly report schedule.

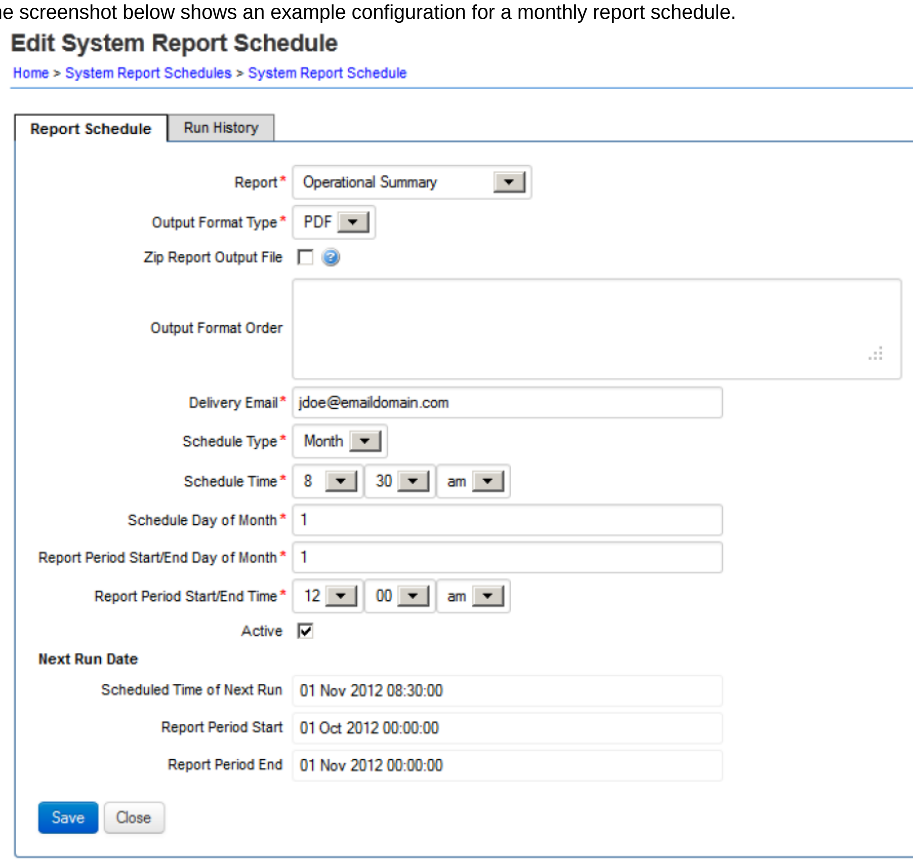


A weekly report schedule

Set the "Schedule Type" to "Week". Set the "Schedule Time" to the time that you want the report to run (format HH:MM:SS).
Select the day of week you want the report to run on from the "Schedule Day of Week" dropdown.
Set "Report Period Start/End Day of Week" to the day of week you want the report to use for its reporting period. For example, if you use "Sunday", Transaction Manager will instruct the report to retrieve data from a Sunday to the Sunday the week after.
Set "Report Period Start/End Time" to a time of day (format HH:MM:SS). If you use "12:00:00", each report will contain data from 12pm of one day to 12pm seven days later.

Setting Up a Monthly Schedule

Monthly schedules are run once a month. If the report is configured to generate results for a certain date range, Transaction Manager will run it for a 1-month period (the number of days varies, as it must).
The screenshot below shows an example configuration for a monthly report schedule.

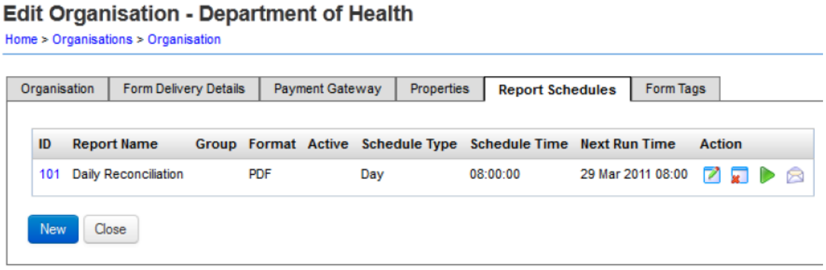


A monthly report schedule

Set the "Schedule Type" to "Month". Set the "Schedule Time" to the time that you want the report to run (format HH:MM:SS).
Set "Schedule Day of Month" to the day of month you want the report to be run on. You can use values 1-31, with each value standing for the *n*th day of a calendar month. To avoid problems when using values > 28, you can also use zero and negative values: For example, 0 will cause the report to run on the last day of the month, however many days that month contains. Likewise, -1 will cause the report to run on the second-last day, and so on.
Set "Report Period Start/End Day of Month" to the day of month you want to use as the boundary between reports. As with "Schedule Day of Month", use a value from 1 to 31 or zero and negative values.
Set "Report Period Start/End Time" to a time of day (format HH:MM:SS). If you use "12:00:00", each report will contain data from 12pm of one day to 12pm on the same day but one month later.

Browsing an Organization's Report Schedules

To view the report schedules set up for an organization, go to "Forms > Organizations", edit the organization and switch to the "Report Schedules" tab.

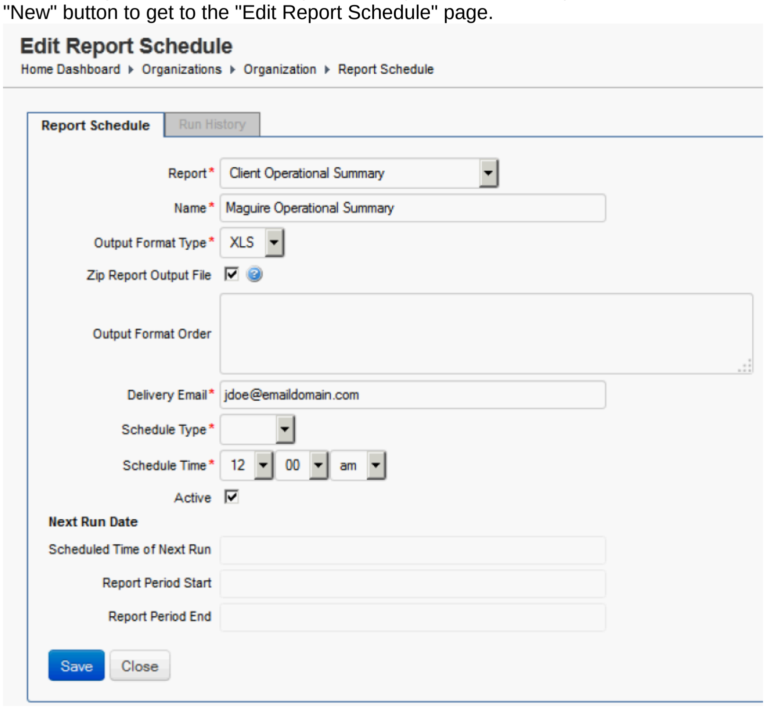


The list of report schedules for an organization

This page allows you to create new report schedules and trigger ( ), edit ( ) and delete ( ) existing report schedules as well as render the report onscreen ( ).

Adding an Organization Report Schedule

To add a report schedule to an organization, go to "Forms > Organizations", edit the organization and switch to the "Report Schedules" tab. Click the "New" button to get to the "Edit Report Schedule" page.



Creating a report schedule for an organization
Now configure the report schedule as described in _configurereportschedule onward.

Report Schedule Execution

When a BIRT report is executed according to a Transaction Manager schedule, the report has to have some way of knowing what date range it should be generating data for. Similarly, an organization assignable report needs to know what organization it is being run against.
The way this is accomplished is by using parameters and naming conventions. Transaction Manager will examine the BIRT report and its input parameters. It will look for and populate report parameters with the following names:

- "client_oid"This parameter will be populated with the organization ID (a number).
- "start_date"This parameter will be populated with the start date and time of the reporting period.
- "end_date"This parameter will be populated with the end date and time of the reporting period.

The end date is always the closest possible date and time before the time that the report is run. For example, if a weekly report is configured to return data from Sunday 12pm to the next Sunday 12pm, you will get the most recent results possible by running the report Sunday afternoon.
The other thing to be aware of is the way report output is generated if the CSV output format type is used. In this case, the BIRT report will be processed by Transaction Manager, the table named "data_table" (if it exists) will be parsed and its contents returned in the CSV file. All other tables will be ignored. This allows you to either re-use the table displayed when viewing the report on-screen, or designate another table for CSV export.
For CSV reports, you can also instruct Transaction Manager to only return a sorted subset of the columns defined in the report's dataset by entering the column names you wish to have extracted into the "Output Format Order".

Triggering a Report Schedule

If you need to execute a report schedule that is not yet due to run, you can trigger it manually. This will generate the report email as per the settings for the report schedule and send it out to the configured recipients.
To trigger a scheduled report, find the report (system reports: "Reports > System Report Schedules", organization assignable reports: "Forms > Organizations") and click the "Trigger the report schedule" icon ( ) next to the report. A new window will open to process your request. It can be safely closed once it has finished loading (the window will remain blank).

Removing a Report Schedule

If you want to remove a report schedule, you have two choices: You can either irrevocably delete the report schedule, or you can deactivate it with the option of reactivating it later.
To delete a report schedule, go to "Reports > System Report Schedules" (system reports) or "Forms > Organizations", edit the organization and switch to the "Report Schedules" tab (organization assignable reports). Click the "Delete" icon ( ) next to the report schedule. You will be asked to confirm this action before the report schedule is deleted.

To deactivate a report schedule, click the "Edit" icon ( ) next to it. On the "Edit Scheduled Report" page, untick the "Active" checkbox and save. As long as the schedule is inactive, it will not be executed, and no emails will be sent out (though you can still manually trigger the schedule).

Reports Included in Transaction Manager

Transaction Manager comes with a set of sample reports that allow you to monitor form submissions. These reports will be loaded into a newly installed system by default.
To use these reports, set up report schedules as described in the previous sections.

The "Operational Summary" Report

Operational Transaction Summary — Avoka Transact

Organisation(s): All
Start date: 17/10/2014 24:00
End date: 17/11/2014 24:00

Form Name	ID	Total	Completed	System Abandoned	Admin Abandoned	User Abandoned	Completion Rate	Last Completed	Median Time to Submit	Payments	Amount Paid
A Secret Hidden Organisation	20	10	0	10	0	0	0%			0	$0.00
Insurance Application	224	7	0	7	0	0	0%			0	$0.00
USA Bank - Consumer Loan Appli	174	3	0	3	0	0	0%			0	$0.00
FinanceCorp	3	1	1	0	0	0	100%	20 Oct 14	<1 min	0	$0.00
Home Owners Insurance Quote	3	1	1	0	0	0	100%	20 Oct 14	<1 min	0	$0.00
Lena Test Client (4.0)	10	2	1	0	0	1	50%	27 Oct 14	5 min	0	$0.00
Test Attachment Form	53	2	1	0	0	1	50%	27 Oct 14	5 min	0	$0.00
Maguire	7	4	3	1	0	0	75%	24 Oct 14	<1 min	3	$119.85
Credit Card Application	225	1	0	1	0	0	0%			0	$0.00
SmartForm PDF Test	228	3	3	0	0	0	100%	24 Oct 14	<1 min	3	$119.85
Training	24	37	10	24	3	0	27%	12 Nov 14	2 min	0	$0.00
Avoka TransactWeb Workshop For	240	16	5	11	0	0	31%	11 Nov 14	1 min	0	$0.00
CT Application	230	1	1	0	0	0	100%	03 Nov 14	<1 min	0	$0.00
CT Changing receipt type on Pu	233	1	0	1	0	0	0%			0	$0.00
CT Form Sharing	229	1	1	0	0	0	100%	03 Nov 14	4 min	0	$0.00
CT Form Sharing	241	4	1	3	0	0	25%	12 Nov 14	3 min	0	$0.00
CT Form Sharing - LC Output Re	235	9	2	4	3	0	22%	03 Nov 14	4 min	0	$0.00
Populating DDL with an initial	238	5	0	5	0	0	0%			0	$0.00
Total		54	15	35	3	1	28%	12 Nov 14	<1 min	3	$119.85

The "Operational Summary" report
This report gives a fairly detailed overview of all organizations that have been set up on the system and submissions for their forms over a certain period. The metrics include the number of completed and abandoned forms, completion rate and the date and time of the most recent completed submission. Note that submissions that are in progress (saved, submitted with user tasks outstanding) are not included in the report.
This system report accepts three parameters: Start and end date, and the organization to report on (or "All Clients" to show everything).

The "Client Operational Summary" Report

Operational Transaction Summary											Avoka Transact
										Organisation: maguire **Start date:** 17/09/2014 24:00 **End date:** 17/10/2014 24:00	
Form Name	ID	Total	Completed	System Abandoned	Admin Abandoned	User Abandoned	Completion Rate	Last Completed	Median Time to Submit	Payments	Amount Paid
Account Opening 2	33	3	0	3	0	0	0%			0	$0.00
Account Opening 3	34	5	0	5	0	0	0%			0	$0.00
Custom Block Test	35	2	0	2	0	0	0%			0	$0.00
Customer Onboarding	15	492	30	462	0	0	6%	16 Oct 14	<1 min	0	$0.00
Customer Onboarding - Triage	9	7	7	0	0	0	100%	22 Sep 14	<1 min	0	$0.00
Customer Onboarding - Triage -	13	1	1	0	0	0	100%	22 Sep 14	<1 min	0	$0.00
Ezidebit Form	14	1	0	1	0	0	0%			0	$0.00
Product Onboarding - Credit Ca	11	1	0	1	0	0	0%			0	$0.00
Product Onboarding - Credit Ca	16	14	12	2	0	0	86%	26 Sep 14	<1 min	0	$0.00
Product Onboarding - Insurance	10	1	0	1	0	0	0%			0	$0.00
Product Onboarding - Insurance	17	13	8	5	0	0	62%	25 Sep 14	<1 min	0	$0.00
Product Onboarding - Retiremen	18	8	8	0	0	0	100%	25 Sep 14	<1 min	0	$0.00
QUS - Paramedical Examination	54	2	2	0	0	0	100%	16 Oct 14	1 min	0	$0.00
Total		550	68	482	0	0	12%	16 Oct 14	<1 min	0	$0.00

The "Client Operational Summary" report
This report is similar to the operational summary report, but it is designed to be used as an organization assignable report. Users cannot choose the organization at render time, which makes it safe for use by organization administrators.

The "Payment Reconciliation" Report

Daily Payment Report					Avoka Transact
				Organisation: Northern Territory Government **Batch:** 20130125	
Submission ID	Receipt Number	Payment ID	Payment Time	Payment Receipt	Amount
15	tpf-ntgov-3	3	25 Jan 2013 12:16	140312301	$120.00
Total					$120.00

The "Payment Reconciliation" report
This report provides a list of credit card payments made for an organization in a 24-hour period ("batch"). It can be used by the organization to reconcile payments against the reports they receive from the payment gateway provider.
Please refer to Online Payment (Manager v5.0) for more information about credit card payment in Transaction Manager and the concept of a batch.
This organization assignable report can be scheduled and emailed out as PDF, XLS or CSV (for automated reconciliation and processing).

The "Transactions By Day" Report

Transaction Report												Avoka Transact
										Start date: 24/07/2013 24:00 **End date:** 31/07/2013 24:00		
Date	Transactions	Renders	Saves	Email Verifications	LC Forms	LC Reader Extensions	LC Output	LC Digital Signatures	LC Proc Mgmt	FieldWorker	Address Lookups	Static PDF
Jul 24, 2013	11	33	9	0	1	1	10	0	0	6	0	1
Jul 25, 2013	9	90	1	0	0	0	9	0	0	1	0	0
Jul 26, 2013	8	20	3	0	3	3	8	0	0	4	0	0
Jul 27, 2013	0	0	0	0	0	0	0	0	0	0	0	0
Jul 28, 2013	0	0	0	0	0	0	0	0	0	0	0	0
Jul 29, 2013	23	63	3	0	3	3	23	1	1	6	0	0
Jul 30, 2013	20	63	8	0	2	2	20	0	0	14	3	0
Total	71	269	24	0	9	9	70	1	1	31	3	1

The "Transactions By Day" report
This system report is used to collect transaction metrics for licensing purposes. It will show transaction data for each day in a specified date range.
You can view this report to see how many billable transactions have been processed in your system. Alternatively, refer to Operations (Manager v5.0) #_sec_transactionlog to see how you can review transaction log data via the Transaction Manager management console.

The "Transactions By Month" Report

Transaction Report													Avoka Transact
											Start date: 08/2012 **End date:** 07/2013		
Date	Number of Days	Transactions	Renders	Saves	Email Verifications	LC Forms	LC Reader Extensions	LC Output	LC Digital Signatures	LC Proc Mgmt	FieldWorker	Address Lookups	Static PDF
04/2013	21	190	336	39	1	65	63	189	1	1	69		1
05/2013	31	219	498	56	5	42	42	219	0	0	109		0
06/2013	30	15	87	5	0	1	1	14	0	0	4	5	0
07/2013	31	292	982	110	0	96	96	290	1	1	58	6	2
Total	113	716	1903	210	6	204	202	712	2	2	240	11	3

The "Transactions By Month" report
This system report is used to collect transaction metrics for licensing purposes. It will show transaction data for the last twelve months, grouped by month (note that the report in the screenshot was taken from a server with less than a year of available licensing data).
You can view this report to see how many billable transactions have been processed in your system. Alternatively, refer to Operations (Manager v5.0) #_sec_transactionlog to see how you can review transaction log data via the Transaction Manager management console.

The "Organization Transactions By Day" Report

Transaction Report Avoka Transact

Organisation: fincorp
Start date: 24/07/2013 24:00
End date: 31/07/2013 24:00

Date	Transactions	Renders	Saves	Email Verifications	LC Forms	LC Reader Extensions	LC Output	LC Digital Signatures	LC Proc Mgmt	FieldWorker	Address Lookups	Static PDF
Jul 24, 2013	9	33	9	0	0	0	8	0	0	6	0	1
Jul 25, 2013	4	90	1	0	0	0	4	0	0	1	0	0
Jul 26, 2013	8	20	3	0	3	3	8	0	0	4	0	0
Jul 27, 2013	0	0	0	0	0	0	0	0	0	0	0	0
Jul 28, 2013	0	0	0	0	0	0	0	0	0	0	0	0
Jul 29, 2013	14	63	3	0	0	0	14	0	0	6	0	0
Jul 30, 2013	15	63	5	0	0	0	15	0	0	14	0	0
Total	50	269	21	0	3	3	49	0	0	31	0	1

The "Organization Transactions By Day" report

This organization report is used to collect transaction metrics for licensing purposes. It will show transaction data for each data in a specified date range.
You can view this report to see how many billable transactions have been processed for a specific organization. Alternatively, refer to Operations (Manager v5.0)#_sec_transactionlog to see how you can review transaction log data via the Transaction Manager management console.

The "Organization Transactions By Month" Report

Transaction Report Avoka Transact

Organisation: fincorp
Start date: 08/2012
End date: 07/2013

Date	Number of Days	Transactions	Renders	Saves	Email Verifications	LC Forms	LC Reader Extensions	LC Output	LC Digital Signatures	LC Proc Mgmt	FieldWorker	Address Lookups	Static PDF
04/2013	21	89	336	28	1	1	1	88	0	0	43		1
05/2013	31	126	498	22	5	2	2	126	0	0	66		0
06/2013	30	7	87	2	0	0	0	6	0	0	3	0	0
07/2013	31	170	982	104	0	5	5	168	0	0	55	0	2
Total	113	392	1903	156	6	8	8	388	0	0	167	0	3

The "Organization Transactions By Month" report

This organization report is used to collect transaction metrics for licensing purposes. It will show transaction data for the last twelve months, grouped by month (note that the report in the screenshot was taken for an organization with just three months of available licensing data).
You can view this report to see how many billable transactions have been processed for a specific organization. Alternatively, refer to Operations (Manager v5.0)#_sec_transactionlog to see how you can review transaction log data via the Transaction Manager management console.

Configuring BIRT Logging

BIRT can be configured to log information to the file system. This information can be useful for debugging purposes, but the log file needs to be monitored and purged if necessary.
Transaction Manager provides configuration options to tune the BIRT log level (or turn logging off completely) and to configure where the log files are written.
To configure the log level, edit the space named "Business Reports" and modify the space property "BIRT Log Level". Acceptable values are "OFF", "SEVERE", "WARNING", "INFO", "CONFIG", "FINE", "FINER" and "FINEST".
To configure the log directory, modify the space property "BIRT Log Directory".

Services (Manager v5.0)

This chapter explains what Transaction Manager services are, what functions the standard services perform, and how to manage services.

Search

Concepts

A *service* in Transaction Manager is a software module providing specific functionality on demand. Services have a *service type* describing the functionality (e.g. virus scanning) they implement. There can be multiple services per service type, one of which will be set up as the default service for that type. Services can be specific to an organization, which can make sense for services that can be associated with specific organizations or forms and should not be visible to other organizations.

Services are uniquely identified by their *name* and *version number*. Services with the same name are considered to be different versions of the same service.

Services may require a *service connection*, which is basically a set of connection parameters to a certain endpoint (e.g. the LiveCycle server). Services can also have a set of arbitrary *service parameters* controlling their behavior.

Because of the big part custom services often play, TM supports setting up unit tests for custom services. These can be used to validate service functionality after making changes.

Service Types Defined in Transaction Manager

Transaction Manager uses several services for various tasks such as rendering forms or virus scanning files. This section lists and describes these service types.

The Data Retention Management Service

This service is periodically called by the Data Retention job. It is responsible for purging a variety of entities within Transaction Manager (such as submission data and error logs). While this service comes with a set of default settings, configuring a suitable data retention policy is very important. Please refer to Operations (Manager v5.0)#_sec_dataretention for more information about data retention.

The Delivery Process Service

This service is called when a submission with a delivery method of "LC Process" is being delivered. It is responsible for passing the submission data across to the LiveCycle orchestration that is being invoked.

The Dynamic Data Service

This service can be called by forms to obtain data dynamically while they are being filled in (e.g. data lookup or validation using external services). By default, a Groovy Dynamic Data service is provided, which will perform the lookup using a Groovy script configured as a service parameter. This means that the lookup behavior can be changed by modifying this service parameter, without requiring a restart or rebuild of Transaction Manager.

The Email Queue Service

This service is responsible for processing emails stored in the email queue, including retries (see Outgoing Email (Manager v5.0)#_sec_emailqueue).

The Find Delivery Process ID Service

This service can update the delivery process ID assigned by Adobe LiveCycle for submissions delivered to an Adobe LiveCycle process. This information can be used for debugging purposes.
Please note that when the service connects to Adobe LiveCycle to retrieve the process ID, it uses the service connection defined for the Delivery Process service that delivered the submission originally. Therefore, it is not necessary to define a service connection for the Find Delivery Process ID service itself.

The Form Meta Data Service

This service is called by the user space to render the metadata related to the form as HTML.

The Form PDF Pre-Render Service

This service is called when a new Composer SmartForm with an XDP form template is created. It uses an Adobe LiveCycle orchestration to pre-render a PDF from the XDP template. This PDF is used for form rendering, which speeds up rendering and reduces the load on LiveCycle's processes.

The Form Prefill Service

This service is called at form render time. It is responsible for retrieving prefill XML data for the form. The exact service used can be defined at a form version level, and a global default service can be defined.

The Form Saved Processor Service

This service is called when a form is saved to Transaction Manager, and the form version in question has been associated with a Form Saved Processor service.
The default service provided by TM allows you to specify custom logic in a Groovy script.

The Form Security Filter Service

This service is called when a new submission is started, or a saved submission is resumed. It can be used to implement custom security checks in addition to the checks performed by TM. TM includes a Groovy service template as a starting point.

Once you have a service, you can associate it with specific form versions as described in Form Version Configuration (Manager v5.0)#_sec_configureservicesforversion.

The Form Server Load Balancer Service

This service is called by various operations involving LiveCycle (e.g. rendering forms and receipts), and is used to ensure capacity to perform the operation is available. It also performs load balancing if more than one LiveCycle server is available.

The Form Version Selector Service

This service is used when a form is rendered and is responsible for choosing the appropriate form version to render. The default service named "Current Version Selector" will simply return the current form version unless a specific version was requested.
If you would like to customize the way versions are selected (e.g. for A/B testing), you can create your own form version selector service and implement custom logic in a Groovy script.
The form version selector service can be configured on the form level (see Form Configuration (Manager v5.0)#_sec_formdetails). If not specified for the form, the form version selector service that is marked as the system default will be used.

The Groovy Service

This service can be used to provide Groovy code that can be reused by other Groovy services in the system. TM itself does not directly call services of this type.

The Job Action Service

This service provides a "Job Action" service related to collaboration jobs.

The Job Action Wait Service

This service provides a "Job Action Wait" service related to collaboration jobs.

The Job Controller Service

The Job Controller service is responsible for creating new collaboration jobs for a submission and then processing the collaboration job. It needs to be associated with the form version.

The Job Delivery Service

This service provides a "Job Delivery" service related to collaboration jobs.

The Job Delivery Wait Service

This service provides a "Job Delivery Wait" service related to collaboration jobs.

The Job Form Start Service

This service provides a "Job Form Start" service related to collaboration jobs.

The Job Process Message Service

This service provides a "Job Process Message" service related to collaboration jobs.

The Job Receipt Wait Service

This service provides a "Job Receipt Wait" service related to collaboration jobs.

The Job Task Assign Service

This service provides a "Job Task Assign" service related to collaboration jobs.

The Job Task Wait Service

This service provides a "Job Task Wait" service related to collaboration jobs.

The Payment Gateway Service

This service is used to process payments. It is used to connect to a payment gateway and make a transaction. Please refer to Online Payment (Manager v5.0) for more on payments in Transaction Manager.

The Payment Reminder Service

This service is periodically called by the Payment Reminder job. It sends out reminder emails about submissions with outstanding payments older than a certain age (configured as the "reminderAgeHours" service parameter). Emails will be sent to administrators belonging to the "Receive Outstanding Payment Alerts" group.

The Receipt Number Service

This service is used by Transaction Manager to calculate receipt numbers for new submissions. It can be configured on the form level (see Form Configuration (Manager v5.0)#_sec_trackingcode), with the default Receipt Number service serving as a fallback.

The Render Composer HTML Form Service

This service is responsible for rendering Composer SmartForms.

The Render eSignature Receipt Service

This service is used to render an electronically signed receipt to the user.

The Render Form Service

This service is called when a form is opened for fill-in. The service will use an orchestration in LiveCycle ES to render the form.

The Render Offline Submission Form Service

This service is called when a form is submitted after being saved offline. The service will use an orchestration in LiveCycle ES to render the form.

The Render Receipt Service

This service is called when a form receipt is opened. The service will use an orchestration in LiveCycle ES to render the receipt. Typically, receipts are rendered statically, with all interactive form features (such as buttons and text fields) disabled.

The Repository Publish Service

This service is used by Transaction Manager to publish a single form to the LiveCycle repositories.

The Repository Synchronize Service

This service is used to bring the LiveCycle repositories up to date after the set of form versions has changed.

The Scheduled Service

This service can be configured to execute custom logic, which is defined in a service parameter containing a Groovy script. The Scheduled Service job can be configured to call this service on a custom schedule.

The Submission Completed Processor Service

This service will be called only if it is configured as the "Submission Completed Processor" service for a form version (see Form Version Configuration (Manager v5.0)#_sec_configureservicesforversion). It is called when a submission is completed by the user and can perform further checks or actions at this point (e.g. assign a new review task based on the submission data).
Transaction Manager provides a service that allows you to specify the desired behavior as a Groovy script, and another that allows the specification of simple actions as XML.

The Submission Data Extraction Service

This service is called on form submission. It passes the submitted form to LiveCycle and retrieves the submission XML data.

The Submission Data Storage Service

This service is responsible for storing and retrieving submission, receipt and attachment data.

The Submission Data Validator Service

This service is responsible for validating submission data when a submission is submitted (not saved or abandoned) by the user. This can be used to prevent users from making fraudulent submissions.
If the service returns an error, the submission is accepted but the errors are recorded and the submission is delivered using a specialized delivery channel for submissions with data validation errors.
You can create your own submission data validator service by using the Groovy service template provided.

The Submission Delivery Controller Service

This service is responsible for a number of features around submission delivery (such as delivery itself, marking submissions as undeliverable, sending reminders etc).

The Submission Email Verification Service

This service is responsible for sending email verification requests to users who have submitted a form. It is also called by the Submission Email Verification Reminder job to send reminders for users who have yet to complete outstanding email verification requests.

The Submission Preprocessor Service

This service is called when a form is submitted, but before the submission is stored to the database. It has the opportunity to check and/or modify submission content and metadata, and can reject a submission entirely.
This service can be configured per form version, and a global default service can be used as a fallback.

The Submission Receipt Service

This service is called periodically by the Transaction Processor service. It is responsible for creating and storing receipts for new submission, and re-attempting to render receipts for submissions where previous attempts failed.

The Task Expiry Process Service

This service is called when a task expires (which happens when the expiry date is reached without the task having been completed). The service is responsible for invoking an external process to notify it of the task having expired.

The Tracking Number Service

This service is used by Transaction Manager to calculate tracking codes for new submissions. It can be configured on the form level (see Form Configuration (Manager v5.0)#_sec_trackingcode), with the default tracking number service serving as a fallback.

The Transaction History Creation Service

This service is called by the Transaction Processor service and extracts relevant metrics from form requests and submission records for reporting purposes.

The Transaction History Publish Service

This service can be called by a scheduled job to publish transaction history data to a customer system.

The Transaction Processor Service

The Transaction Processor service is called by the Transaction Processing job and performs submission processing in the following order: Abandonment, receipting, delivery, task expiry, transaction history creation and data retention.

Two Factor Authentication Service

This service is responsible for validating the second factor in two-factor authentication. Please refer to the Transact Integration Guide for more on two-factor authentication in TM.

The Virus Scan Service

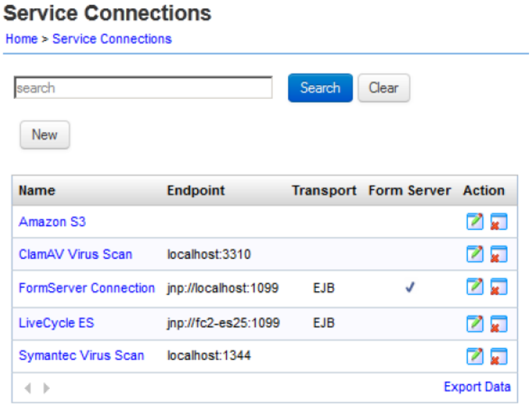
This service is used to scan attachments stored in Transaction Manager for viruses. If this service is not configured, no virus scanning will be done.

Managing Service Connections

Many services need to connect to a server to function properly. These connection details are set up using a service connection. Service connections are managed separately to the services themselves, which allows you to reuse and change service connections easily.
Please note that some service definitions (notably those related to rendering forms and receipts) do not use their own service connection. Instead individual requests are assigned an available form server node, which handles the request and defines its own service connection. Refer to Scalability and Server Nodes (Manager v5.0) for more information on server nodes.

Browsing the List of Service Connections

To see all service connections set up in Transaction Manager, go to "System > Service Connections" (see below).
The list can be filtered using a substring of the service connection name and the endpoint value, and export the search results to Excel by clicking the "Export Data" link below the table.

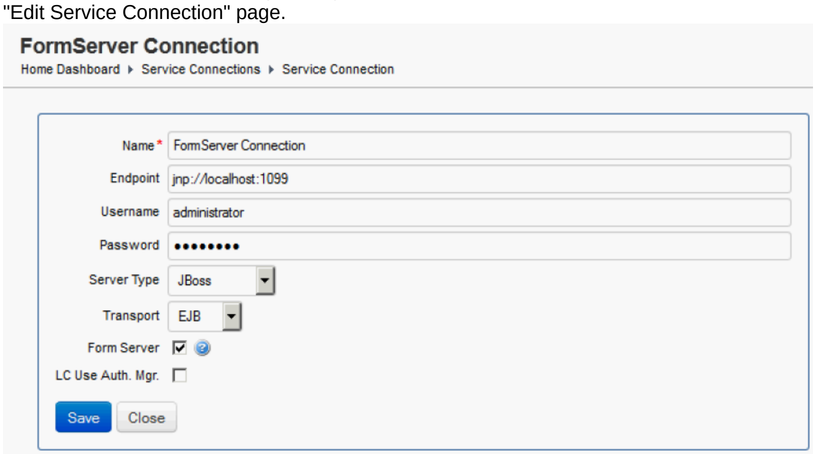


Browsing the list of service connections

This page allows you to create new service connections, edit existing connections ( ) and delete service connections ( ).

Editing a Service Connection

To edit a service connection, go to "System > Service Connections" and click the "Edit" icon ( ) next to the connection. You will be directed to the "Edit Service Connection" page.



Editing a service connection
Change the fields as required (depending on the service using the connection, not all fields will be used) and save the connection.
One important field is the "Form Server" checkbox. This should be ticked if the service connection is being set up for use with a form server node, which is used to host LiveCycle (see Scalability and Server Nodes (Manager v5.0)). If ticked, all communication with the remote server will take place using the "Form Server" Transaction Manager module (which must be deployed on the server). In addition, the "Endpoint" field must then contain the URL of the endpoint relative to the "FormServer" module (which is typically on the same server, so is referenced by using "localhost" or a local IP address). For service connections where the "Form Server" checkbox is not ticked, the "Endpoint" field must still contain the URL as accessible to Transaction Manager itself.

Deleting a Service Connection

While you can delete service connections, make sure to update all services using the service connection before you do so. Otherwise a service may be left without a service connection, which will most likely cause it to fail.

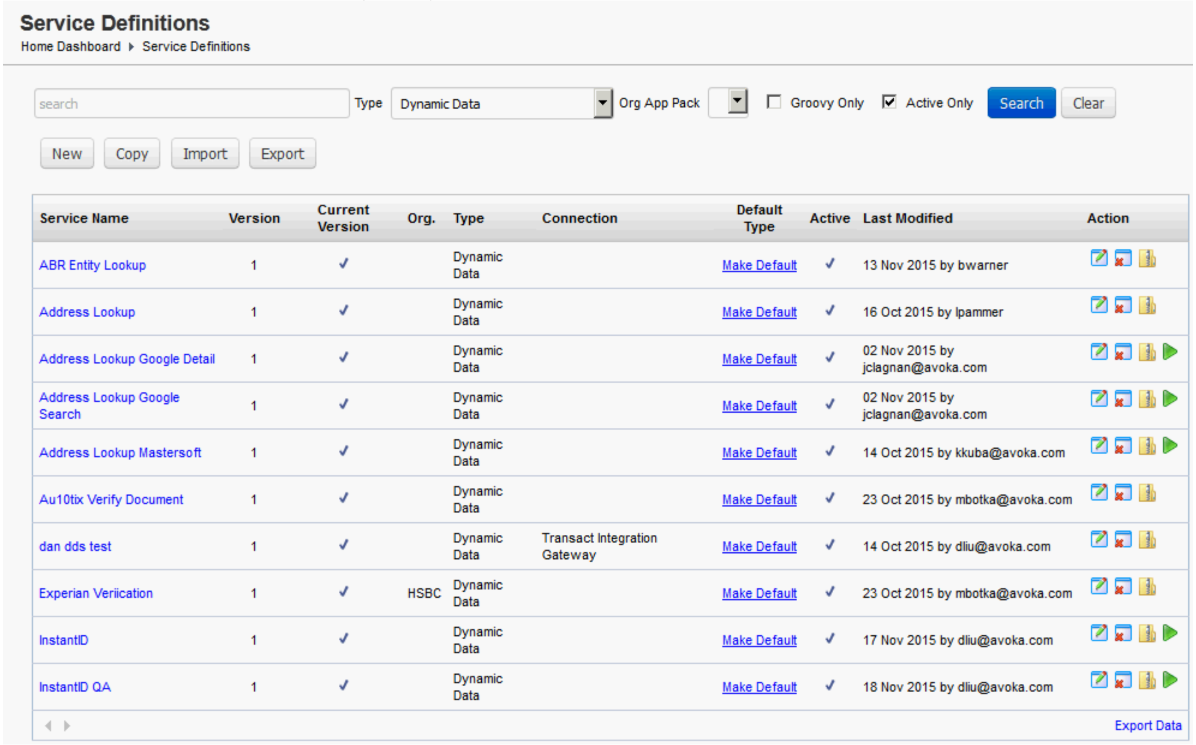
To delete a service connection, go to "System > Service Connections" and click the "Delete" icon ( ) next to the service connection. You will be asked to confirm this action before the connection is irrevocably deleted.

Managing Service Definitions

This section describes how to view and manage the service definitions set up in Transaction Manager.

Browsing the List of Service Definitions

To view the list of all service definitions, go to "System > Service Definitions".



Service Name	Version	Current Version	Org.	Type	Connection	Default Type	Active	Last Modified	Action
ABR Entity Lookup	1	✓		Dynamic Data		Make Default	✓	13 Nov 2015 by bwarner	
Address Lookup	1	✓		Dynamic Data		Make Default	✓	16 Oct 2015 by lpammer	
Address Lookup Google Detail	1	✓		Dynamic Data		Make Default	✓	02 Nov 2015 by jclagnan@avoka.com	
Address Lookup Google Search	1	✓		Dynamic Data		Make Default	✓	02 Nov 2015 by jclagnan@avoka.com	
Address Lookup Mastersoft	1	✓		Dynamic Data		Make Default	✓	14 Oct 2015 by kkuba@avoka.com	
Au10tix Verify Document	1	✓		Dynamic Data		Make Default	✓	23 Oct 2015 by mbotka@avoka.com	
dan dds test	1	✓		Dynamic Data	Transact Integration Gateway	Make Default	✓	14 Oct 2015 by dliu@avoka.com	
Experian Veriication	1	✓	HSBC	Dynamic Data		Make Default	✓	23 Oct 2015 by mbotka@avoka.com	
InstantID	1	✓		Dynamic Data		Make Default	✓	17 Nov 2015 by dliu@avoka.com	
InstantID QA	1	✓		Dynamic Data		Make Default	✓	18 Nov 2015 by dliu@avoka.com	

Browsing the list of service definitions

You can filter the list using the organization, a substring of the service name, the service type (e.g. "Render Form"), an application package (see Organization Configuration (Manager v5.0)#_sec_setupapplicationpackages), whether the service is Groovy based and whether the service is marked as active. When filtering by organization, you will be shown all services available to that organization, which includes global services. Organization administrators can see but not modify global service definitions.

To assist you in auditing configuration changes, the last modified date and the user modifying the service are displayed in the "Last Modified" column. This information is also available for individual service parameters.

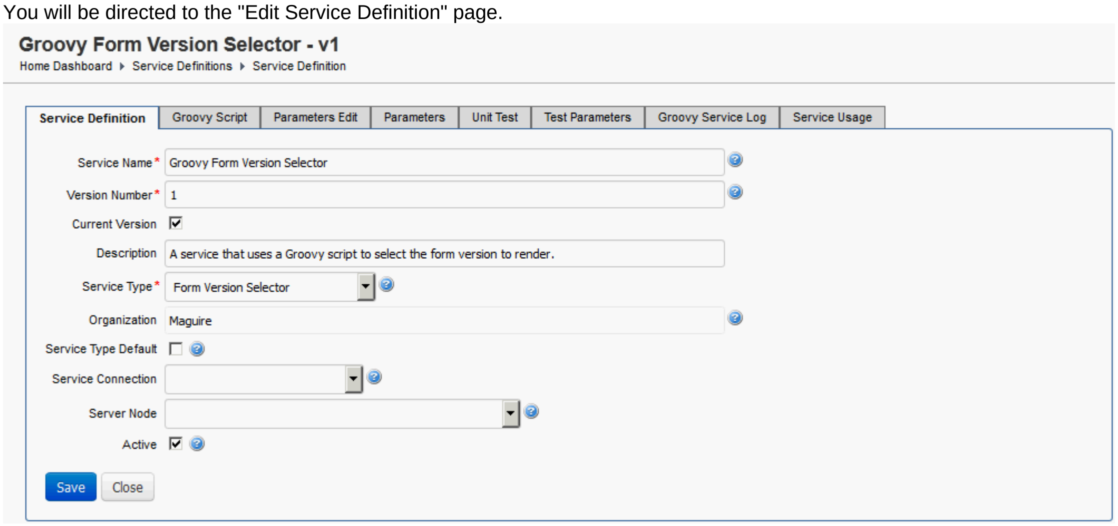
This page allows you to create new service definitions from a template, copy, edit ( ) and delete existing service definitions ( ) as well as setting a service definition as the default for its service type ("Make Default"). You can also export and import the set of service definitions.

If a service supports unit testing, you can run the unit tests for it ( ).

Editing a Service Definition

To edit an existing service definition, click the "Edit" icon ( ) next to it.

You will be directed to the "Edit Service Definition" page.



Editing a service definition

The service name combined with the version number needs to be globally unique, even if the service belongs to a specific organization.

If you are a global administrator, you can choose to assign the service definition to a specific organization. If you do so, the service definition is invisible to organization administrators of other organizations and be associated only with the same organization or its forms. System related service types such as "Email Queue" should be global, while other services such as "Dynamic Data" or "Delivery Process" might be candidates for organization-specific services. If you change the organization a service is assigned to, please be aware that all entities belonging to other organizations (e.g. form versions) that are currently associated with this service will be disassociated and may have to be assigned new services. In most cases, it is better to create a new service for a particular organization rather than change a global service to be organization-specific.

The service type determines what this service is used for. Tick the "Service Type Default" checkbox to make this service the system default one for its type. This setting may not be relevant for some service types; for example, services of type "Dynamic Data" are always resolved by name and the "Service Type Default" setting is not used.

The class name of the service's Java class (or possibly the Spring bean name) is displayed in the "Classname / Bean Name" field. The class must be one of the classes pre-defined in Transaction Manager (while new service classes can be added in theory, this is beyond the scope of this document).

If the service requires a connection to a server, select the appropriate service connection from the "Connection" dropdown.

If the service needs to connect to a specific server node defined in TM (e.g. delivery service only present on a dedicated server), select the server node from the "Server Node" dropdown.

If you wish to deactivate a service, uncheck the "Active" checkbox. This will prevent the service from being displayed in various contexts, including on the form version edit page. You can later reactivate the service if it needs to be reinstated.

Click the "Save" button to save your changes. From now on, the service will use the new configuration when called.

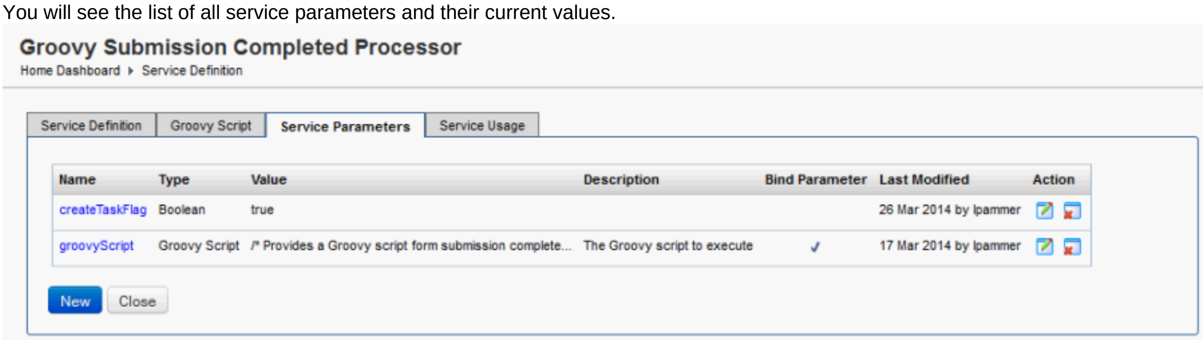
Editing the Groovy Script

The "Groovy Script" tab is displayed for services based on a Groovy script. It allows you to edit the Groovy script directly.

The Groovy script for a service

Managing Service Parameters

Services can define service parameters, which can be used to change how the service behaves. For example, the submission data storage service allows you to specify whether submission XML shall be encrypted and compressed before storage.
To view the list of service parameters, go to "System > Service Definitions" and edit the service definition, then switch to the "Service Parameters" tab. You will see the list of all service parameters and their current values.



Viewing the list of service parameters

Note that the time the parameter was last modified and the user who made the change are shown in the "Last Modified" column.

You can also edit the parameters to change their value. However, as services are vital to Transaction Manager, please make sure to make changes only if you know how the service and its parameters work.

If you want to edit a service parameter, click the "Edit" icon ( ) next to it. You will be directed to the "Edit Service Parameter" page (see below), where you can enter a new value for the parameter. Depending on the parameter type, you may be offered a specialized editor.

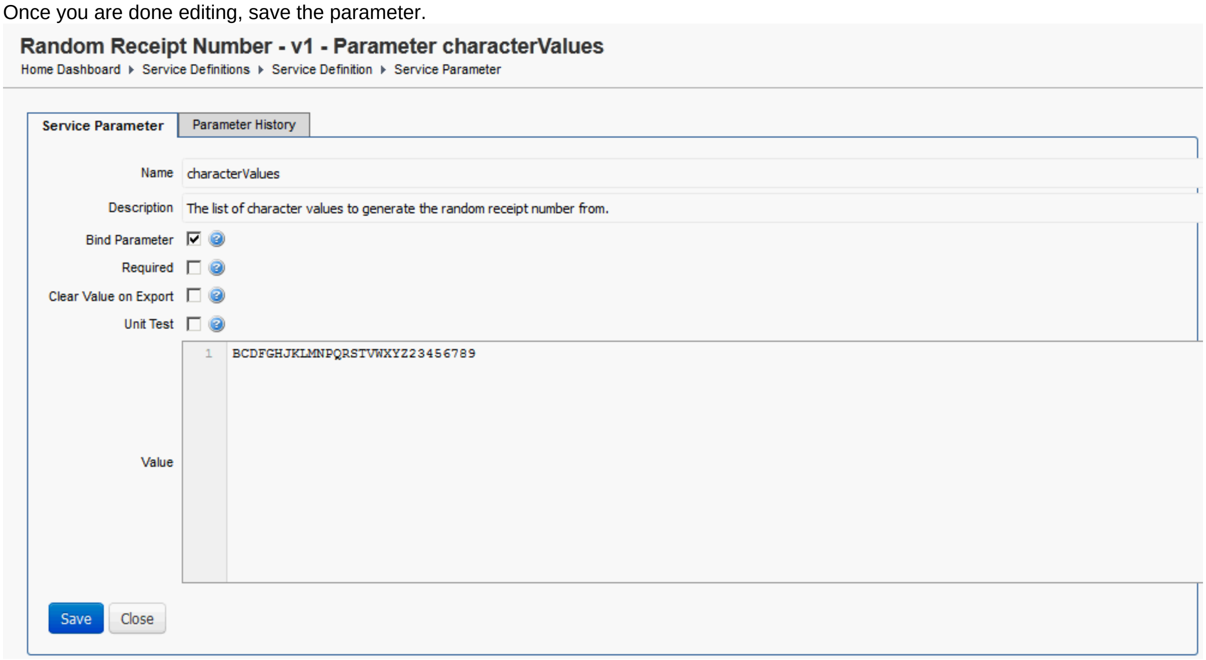
Please do not change the value of the "Bind Parameter" checkbox. If unticked, the parameter will not be automatically passed to the service upon instantiation; such parameters are used to augment customizable services (such as Spring Bean and Groovy services).

If a parameter is absolutely required for the service to work and cannot be left blank, tick the "Required" flag. Note that the service may not execute if required parameters are not set.

Tick the "Clear Value on Export" checkbox if you would like the parameter value to be excluded when exporting the service. This can be useful for values that need to be reconfigured on every environment, or sensitive values like passwords.

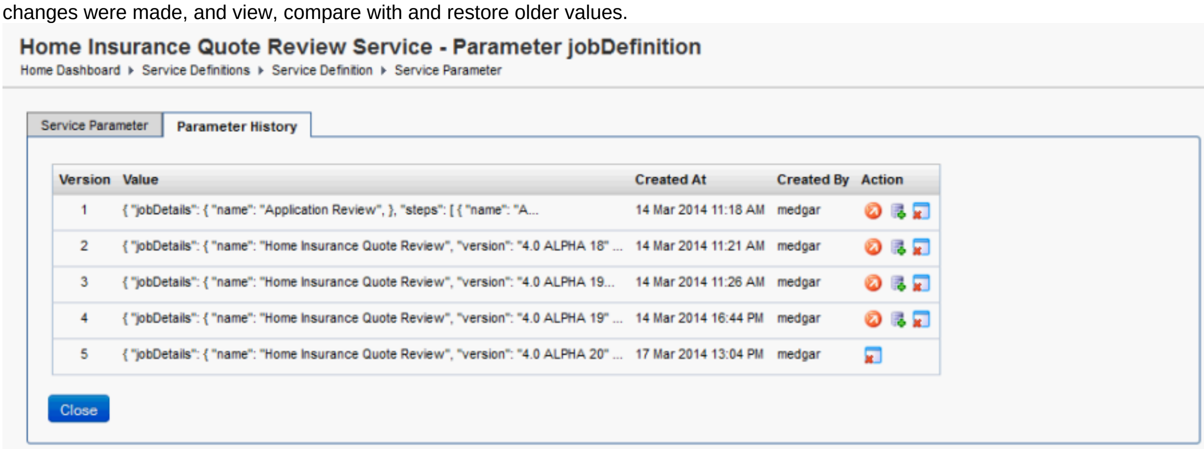
If the parameter is used only for service unit testing, tick the "Unit Test" checkbox.

Once you are done editing, save the parameter.



Editing a service parameter

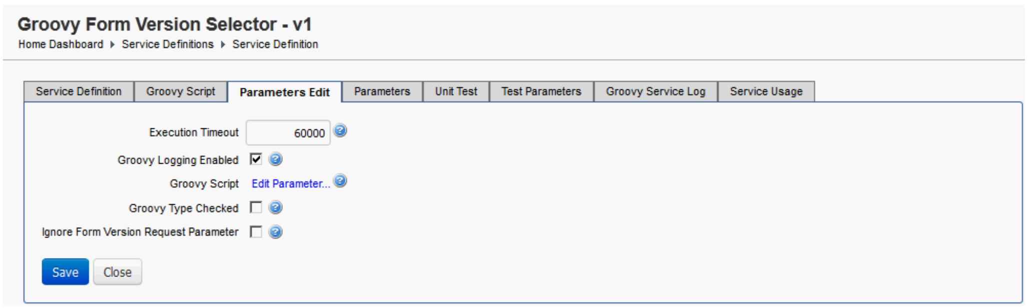
To view a history of the changes made to a service parameter, you can switch to the "Parameter History" tab. You can see exactly when and by whom changes were made, and view, compare with and restore older values.



Change history for a service parameter

Changing Service Parameter Values

If you just need to change parameter values, you can use the parameter editor page. Edit the service definition and switch to the "Parameters Edit" tab.

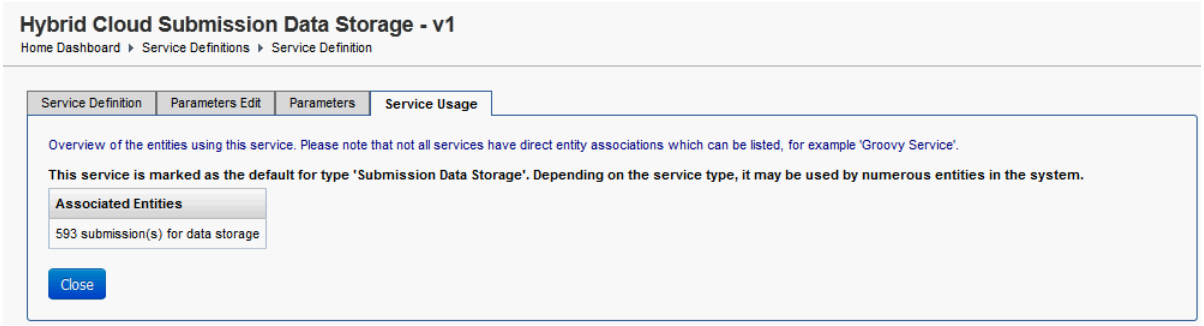


Changing the values of service parameters

Enter values for any parameters you want to change and click "Save".

Service Usage

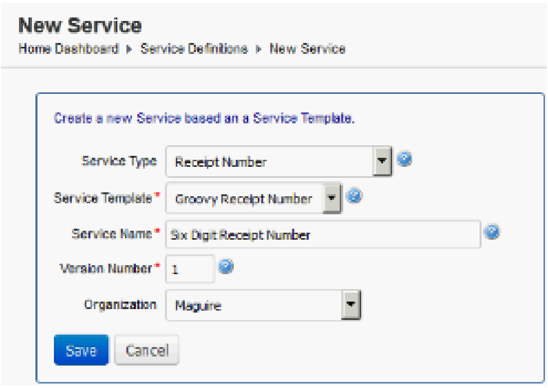
The "Service Usage" tab is shown for service types that can be explicitly associated with entities such as form versions, organizations and delivery channels. It helps you determine where the service is used, so you can avoid unwanted side effects when modifying / deleting the service.



Service usage for a service of type "Submission Data Storage"

Creating a Service Definition from a Template

To create a new service, go to "System > Service Definition" and click "New". You will be able to create a new service based on a template on the "New Service" page.



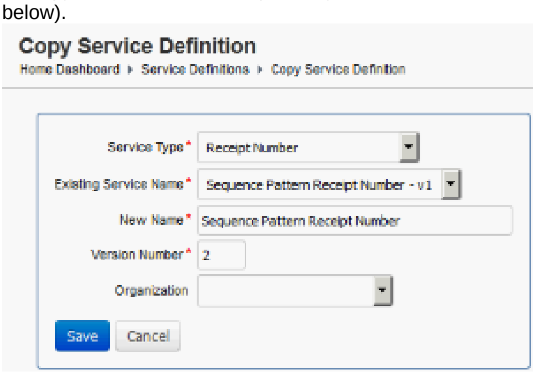
Creating a service definition
First, choose the service type from the "Service Type" dropdown. Each service type may have multiple templates; please choose the most suitable from the "Service Template" dropdown.

Give the new service a name (if this is a totally new service, give it a globally unique name; if a new version of an existing service, you can use the same name as the existing service with a new version number).

If you are a global administrator, you can associate the service definition with a specific organization. If you do so, the service can be associated only with forms and other entities belonging to that organization; if you do not select an organization, the service will be globally available.
Click "Save" to create the service. You will then be taken to the "Service Edit" page, where you can configure the service as needed (see _sec_editservicedefinition).

Copying a Service Definition

You can create a copy of an existing service definition that inherits its base configuration and all service parameters from the original service rather than a service template.
To copy a service definition, go to "System > Service Definitions" and click "Copy". You will be redirected to the "Copy Service Definition" page (see below).



Creating a copy of a service definition
Choose the service type for which you want to create a new service definition.

Now select the service you would like to create a copy of from the "Existing Service Name" dropdown.

If you are just creating a new version of the service, leave the name as is and check the version number (TM will suggest a suitable version number for you). Otherwise, enter a unique combination of service name and version number.

If you would like to restrict the new service to a specific organization, select the organization from the "Organization" dropdown and save. This option is not available to organization administrators; for them, the new service definition will always be associated with their organization.
Transaction Manager creates a new service, which is identical to the original apart from the name. If you would like to make any changes to the new service, edit it and change service parameters as described in the preceding sections.

Deleting a Service Definition

When deleting a service definition, make sure it is not used anywhere (e.g. form versions) to avoid errors. It is not recommended to delete a service that is the default for a service type.

To delete a service definition, go to "System > Service Definitions" and click the "Delete" icon ( ) next to it. You will be asked to confirm this action before the service definition is irrevocably deleted.

Importing / Exporting Service Definitions

You can export service definitions, their parameters and associated service connections, to be imported to a different Transaction Manager instance or for backup purposes.
To learn about how to export and import services, please refer to Multi-Environment Support (Manager v5.0)#_sec_migrateservices.

Groovy Based Services

Service definitions that can be swapped out and configured at runtime add a lot of flexibility to your system; however, often the requirements for the functionality that a service provides vary widely and can change over time. Transaction Manager addresses this issue by providing services that are based on a Groovy script (Groovy is a programming language related to Java). The Groovy script encapsulates the basic functionality that the service provides (depending on service type; for example, custom delivery logic for a submission delivery service), and can be freely adjusted at runtime. Groovy services make it possible to integrate with a wide variety of third party systems, or even just to provide customizations or adjustments that fall outside the scope of service parameters provided by Transaction Manager. Do note that Groovy services could be used to run malicious code on Transaction Manager, and while Transaction Manager contains safeguards to mitigate this issue, it is important that you regulate access to Groovy based services carefully.
For more on specific Groovy services such as submission delivery, dynamic data retrieval and form prefill, please refer to the Transact Integration Guide.

For unit test support, please refer to the Transact Services Guide included in the TM management console ("System > Transact Services Guide").

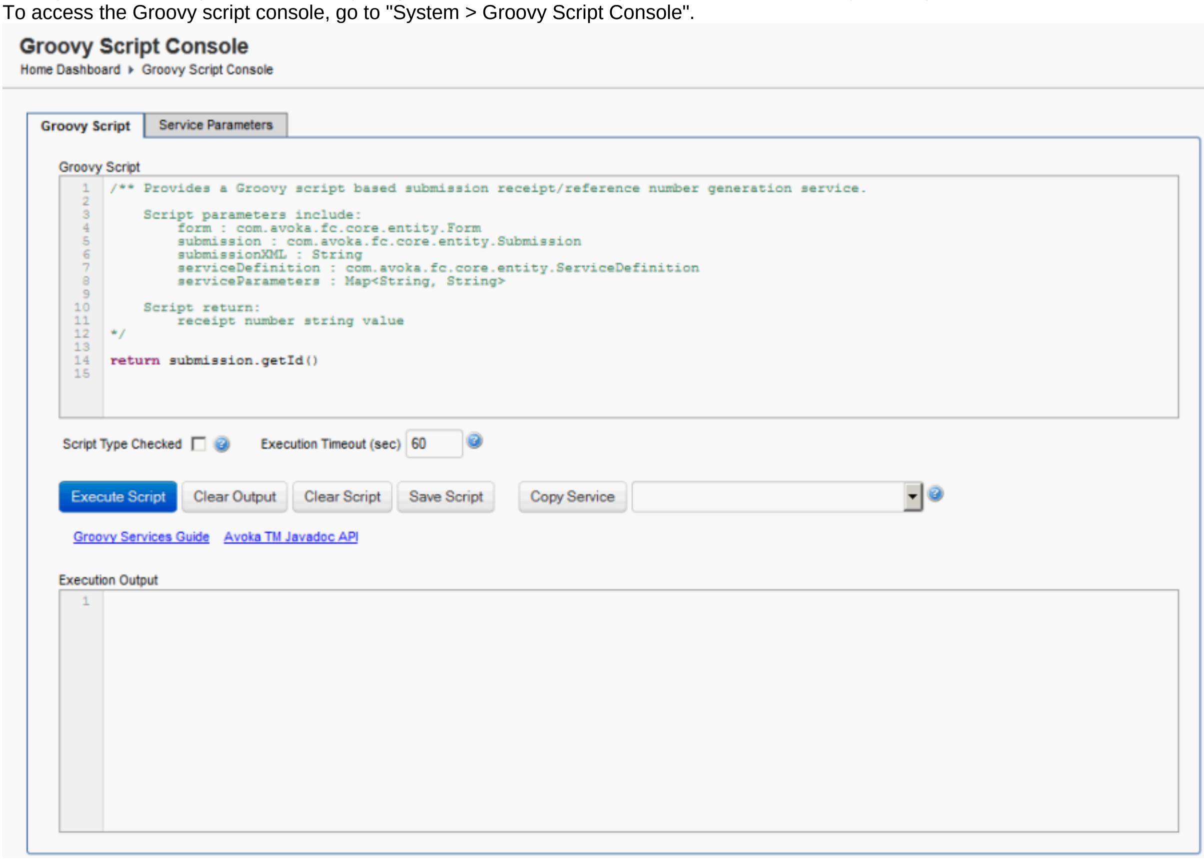
The Groovy Services Guide

To help you develop your own Groovy scripts, TM comes with its own service development guide that contains some Groovy basics, tips and tricks as well as documentation about the Groovy extension points provided in Transaction Manager.
The Groovy services guide can be accessed via "System > Transact Services Guide".

The guide also explains how to set up unit testing for a service.

The Groovy Script Console

Groovy scripts must be constructed and tested carefully to avoid runtime errors. To make script development easier, the Transaction Manager management console provides a Groovy script console, which allows you to enter and evaluate your scripts.
To access the Groovy script console, go to "System > Groovy Script Console".



The Groovy script console
Enter the script in the "Groovy Script" area and click "Execute Script" to evaluate the script and see the results. Tick "Script Type Checked" to enabled Groovy type checking (see http://docs.codehaus.org/display/GROOVY/Type+checking+extensions for more on type checking in Groovy).

Note that output printed to System.out (e.g. via "println") will be shown in the "Script output" window. However, output printed to System.err will not be shown. To see the value your script returns, you'll need to print it to System.out as well.

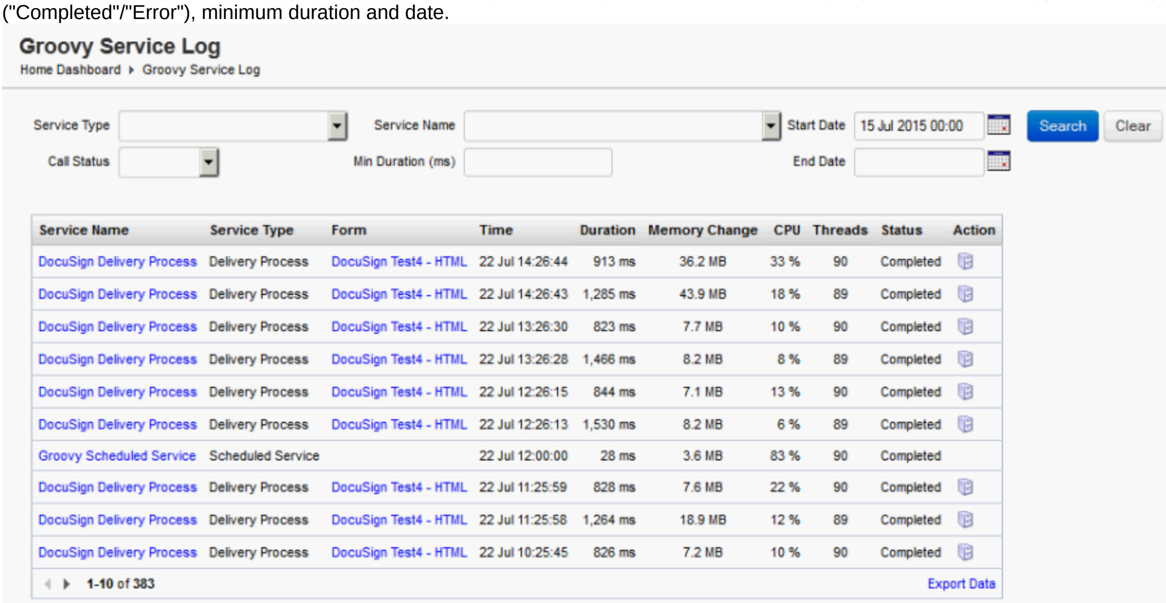
You can also initialize the script from an existing service definition by selecting the service from the "Service" dropdown and clicking "Copy Service". If the service has other service parameters, these will also be loaded for testing purposes.

You can save the current script against your user account preferences by clicking "Save Script" or clear the stored script by clicking "Clear Script".

Finally, you can create service parameters that will be used when testing your script on the "Service Parameters" tab.

The Groovy Service Log

The Groovy service log contains records logging the execution of Groovy services when they complete successfully or in specific error cases.
Groovy service logging can be globally disabled by setting the deployment property "Groovy Logging Enabled" to "false". In addition, Groovy service logging can be enabled or disabled for each service by setting the parameter "groovyLoggingEnabled" to "true" or "false".
To view the Groovy service log, go to "System > Groovy Service Log". You can filter the list by organization, service type, service (by name), status ("Completed"/"Error"), minimum duration and date.



Service Name	Service Type	Form	Time	Duration	Memory Change	CPU	Threads	Status	Action
DocuSign Delivery Process	Delivery Process	DocuSign Test4 - HTML	22 Jul 14:26:44	913 ms	36.2 MB	33 %	90	Completed	
DocuSign Delivery Process	Delivery Process	DocuSign Test4 - HTML	22 Jul 14:26:43	1,285 ms	43.9 MB	18 %	89	Completed	
DocuSign Delivery Process	Delivery Process	DocuSign Test4 - HTML	22 Jul 13:26:30	823 ms	7.7 MB	10 %	90	Completed	
DocuSign Delivery Process	Delivery Process	DocuSign Test4 - HTML	22 Jul 13:26:28	1,466 ms	8.2 MB	8 %	89	Completed	
DocuSign Delivery Process	Delivery Process	DocuSign Test4 - HTML	22 Jul 12:26:15	844 ms	7.1 MB	13 %	90	Completed	
DocuSign Delivery Process	Delivery Process	DocuSign Test4 - HTML	22 Jul 12:26:13	1,530 ms	8.2 MB	6 %	89	Completed	
Groovy Scheduled Service	Scheduled Service		22 Jul 12:00:00	28 ms	3.6 MB	83 %	90	Completed	
DocuSign Delivery Process	Delivery Process	DocuSign Test4 - HTML	22 Jul 11:25:59	828 ms	7.6 MB	22 %	90	Completed	
DocuSign Delivery Process	Delivery Process	DocuSign Test4 - HTML	22 Jul 11:25:58	1,264 ms	18.9 MB	12 %	89	Completed	
DocuSign Delivery Process	Delivery Process	DocuSign Test4 - HTML	22 Jul 10:25:45	826 ms	7.2 MB	10 %	90	Completed	

The Groovy service log
For each entry, information identifying the service, date and time of the invocation, the result status and server performance information are shown. Additional actions such as links to the submission and error log may also be included.

Scheduled Jobs (Manager v5.0)

This chapter explains what scheduled jobs are, what functions the standard jobs perform in Transaction Manager, and how to manage jobs.

Search

Concepts

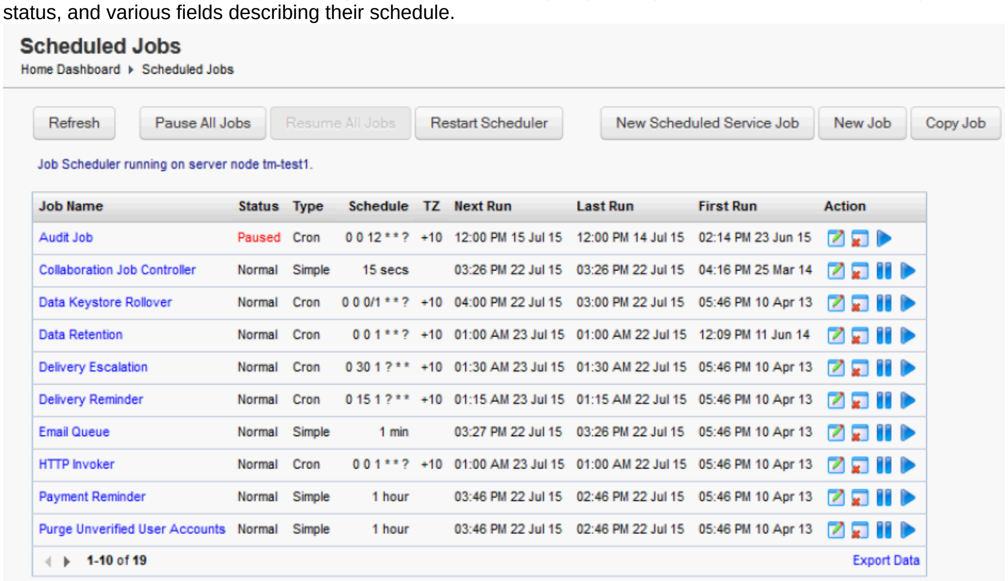
A *scheduled job* in Transaction Manager is functionality that has to run regularly to perform routine operations, such as triggering delivery, generate receipts and temp file cleanup. Often a scheduled job will call the default service of a certain service type to do the actual work, which increases flexibility by allowing you to swap out and reconfigure services. Unlike collaboration jobs (see Collaboration Jobs (Manager v5.0)), scheduled jobs are not necessarily related to one or more submissions. Instead of coming with a fixed set of hard-coded schedules, Transaction Manager allows you to manage existing scheduled jobs, modify their schedules, and control and monitor their execution.
Jobs have a *schedule*, which means they are automatically started by Transaction Manager at regular intervals. You can also pause and resume a scheduled job, as well as trigger it manually if needed.

Transaction Manager provides two scheduling mechanisms: *simple jobs* and *cron jobs*. Simple jobs are characterized by a repeat interval and the number of times a scheduled job should run. This is sufficient for regular scheduled jobs but allows no control exactly when a scheduled job runs. Cron jobs allow finer grained control via a cron expression, which specifies both the run interval and the time the scheduled job should run. This is particularly useful for maintenance jobs that need to run outside of peak times.

Transaction Manager uses the Quartz Job Scheduling Framework to manage scheduled jobs.

Managing Scheduled Jobs

To view the list of scheduled jobs set up in Transaction Manager, go to "System > Scheduled Jobs". As you can see, scheduled jobs have a name, a status, and various fields describing their schedule.



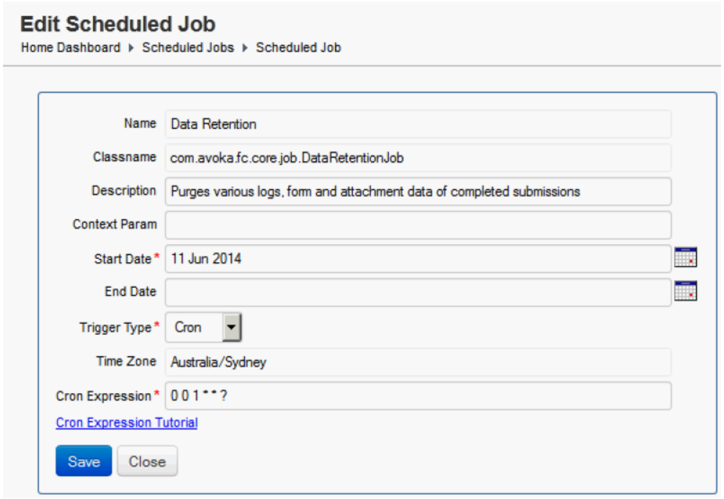
Job Name	Status	Type	Schedule	TZ	Next Run	Last Run	First Run	Action
Audit Job	Paused	Cron	0 0 12 * * ?	+10	12:00 PM 15 Jul 15	12:00 PM 14 Jul 15	02:14 PM 23 Jun 15	
Collaboration Job Controller	Normal	Simple	15 secs		03:26 PM 22 Jul 15	03:26 PM 22 Jul 15	04:16 PM 25 Mar 14	
Data Keystore Rollover	Normal	Cron	0 0 0/1 * * ?	+10	04:00 PM 22 Jul 15	03:00 PM 22 Jul 15	05:46 PM 10 Apr 13	
Data Retention	Normal	Cron	0 0 1 * * ?	+10	01:00 AM 23 Jul 15	01:00 AM 22 Jul 15	12:09 PM 11 Jun 14	
Delivery Escalation	Normal	Cron	0 30 1 ? * *	+10	01:30 AM 23 Jul 15	01:30 AM 22 Jul 15	05:46 PM 10 Apr 13	
Delivery Reminder	Normal	Cron	0 15 1 ? * *	+10	01:15 AM 23 Jul 15	01:15 AM 22 Jul 15	05:46 PM 10 Apr 13	
Email Queue	Normal	Simple	1 min		03:27 PM 22 Jul 15	03:26 PM 22 Jul 15	05:46 PM 10 Apr 13	
HTTP Invoker	Normal	Cron	0 0 1 * * ?	+10	01:00 AM 23 Jul 15	01:00 AM 22 Jul 15	05:46 PM 10 Apr 13	
Payment Reminder	Normal	Simple	1 hour		03:46 PM 22 Jul 15	02:46 PM 22 Jul 15	05:46 PM 10 Apr 13	
Purge Unverified User Accounts	Normal	Simple	1 hour		03:46 PM 22 Jul 15	02:46 PM 22 Jul 15	05:46 PM 10 Apr 13	

Managing scheduled jobs
You can see that Cron jobs are associated with a timezone (the "TZ" column contains the timezone offset), which generally corresponds to the timezone used on the TM server. Whenever TM is restarted, it will adjust the timezones for all Cron jobs if the system time has changed. If you are seeing unexpected timezone values, check the system time settings of the server that Quartz is running on and restart TM if needed.

If there are issues when running scheduled jobs, one option is to pause or trigger individual jobs for debugging purposes. It is also possible to restart the entire job scheduler by clicking "Restart Scheduler". This will reset the scheduler without the need to restart the TM server.
Note that if you are running multiple TM server nodes, the job scheduler will be running only on a single node. The "Scheduled Jobs" page displays the node above the main table. Please ensure you are logged on to the management console on that node when making changes to scheduled jobs.

Adding or Editing a Scheduled Job

To add a scheduled job, click the "New Job" button on the "Scheduled Jobs" page (if the new job will be calling a service definition of type "Scheduled Service", you can click "New Scheduled Service Job"). If you want to edit an existing scheduled job, click the "Edit" icon () next to it. You will be directed to the "Edit Scheduled Job" page.



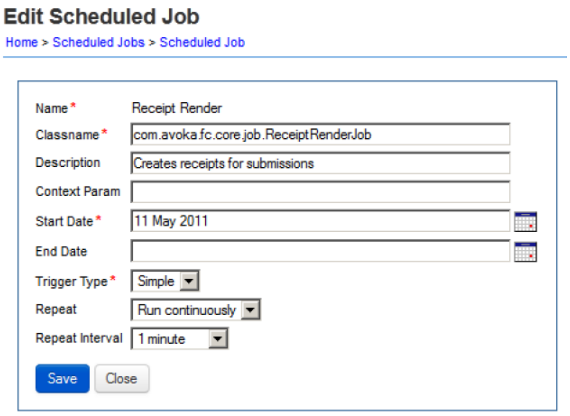
Editing a scheduled job

Enter the unique name of the scheduled job into the "Name" field. Enter the class name of the scheduled job's Java class into the "Classname" field. The class must be one of the classes pre-defined in Transaction Manager (while new job classes can be added in theory, this is beyond the scope of this document).

Optionally, you can enter a description that explains what your scheduled job does.
If the scheduled job can be parameterized, enter the parameter into the "Context Param" field.
Finally, you need to schedule your job. Select the appropriate job type ("Simple" or "Cron") from the "Trigger Type" dropdown. The sections below describe the how to schedule simple and cron jobs.
Once you are done editing the scheduled job, save your changes.

Scheduling a Simple Job

To schedule a simple job, select "Simple" from the "Trigger Type" dropdown. This will cause additional scheduling fields to be displayed.



A schedule for a simple job

The "Repeat" dropdown lets you choose how often the scheduled job should be run (values range from once to continuously). The "Repeat Interval" defines the time interval between two invocations of the job (values range from 1 minute to 24 hours).
The "Start Date" and "End Date" define the date range in which the scheduled job is to run. Leave the end date blank unless you want the scheduled job to expire.

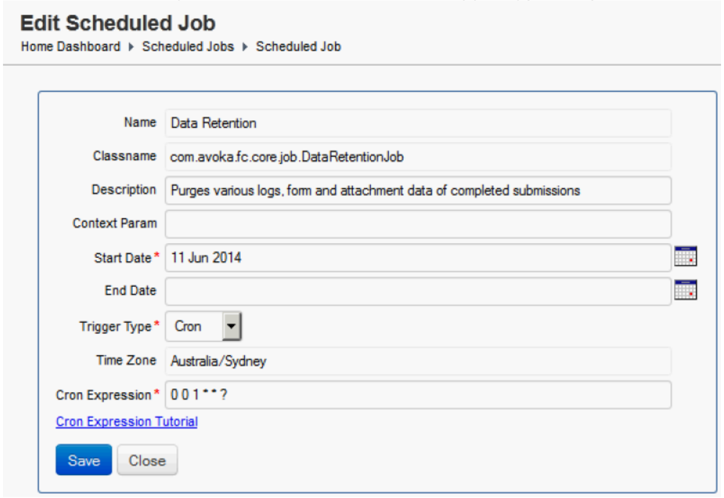
You can have four different types of schedules depending on the settings you choose:

1. Once-off jobsCharacterized by: "Repeat" = "Run Once" Result: Job will run only once on the start date you enter (it will be run at midnight). Use this for non-routine tasks or if you want a job that is triggered manually whenever it needs to be run.
2. Jobs with a fixed number of repetitionsCharacterized by: "Repeat" = "Repeat *<n>*"Result: Job will run n times starting at midnight on the start date you set. Use this for tasks that need to be run a fixed number of times.
3. Jobs that run for a fixed number of daysCharacterized by: "Repeat" = "Run continuously", "End Date" is setResult: Job will run from midnight on the start date you set, repeating after the specified interval. It will not be run after midnight on the end date. Use this for tasks that need to be run for several days.
4. Continuous jobsCharacterized by: "Repeat" = "Run continuously", "End Date" is blank Result: Job will run every n minutes, starting at midnight on the start date you specified. Use this for repeating tasks.

Most of your jobs will likely perform background tasks with no set expiry date and will therefore be set up as continuous jobs. All simple jobs shipped with Transaction Manager also follow this pattern.

Scheduling a Cron Job

To schedule a cron job, choose "Cron" from the "Trigger Type" dropdown. This will cause additional scheduling fields to be displayed.

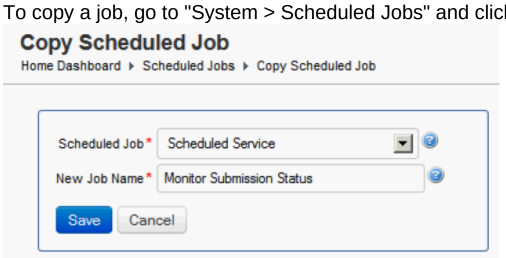


A schedule for a cron job

The schedule is controlled via a cron expression. To learn more about cron expressions, refer to e.g. http://en.wikipedia.org/wiki/CRON_expression or click on the "Cron Expression Tutorial" link at the bottom of the page. The expression in the screenshot above will cause the job to run every day at 1am, and much more sophisticated schedules are possible.

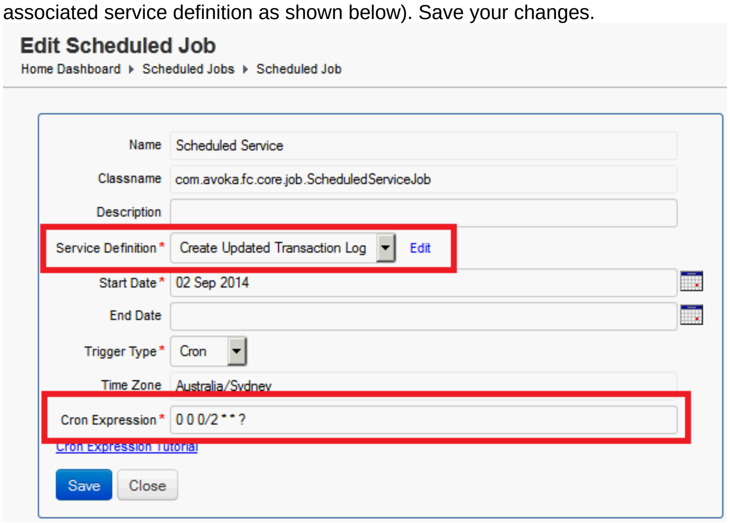
Copying a Scheduled Job

If you would like to add a scheduled job that is based on an existing job, you can simply create a copy of the existing job. This is especially useful to create additional instances of the "Scheduled Service" job, which can be used to execute arbitrary Groovy script on a schedule.
To copy a job, go to "System > Scheduled Jobs" and click the "Copy Job" button. You will be directed to the "Copy Scheduled Job" page.



Copying a job

Choose the scheduled job you want to copy from the "Scheduled Job" dropdown.
Enter the name for the new scheduled job into the "New Job Name" field. Ensure that no other scheduled job with this name already exists.
Click "Save". A new scheduled job with the specified name will be created, and all other settings including the schedule will be based on the job you copied.
Finally, edit the new scheduled job and make any configuration changes you want (for example, for a scheduled service job, change the schedule and associated service definition as shown below). Save your changes.



Adjusting the configuration of a copied job

Pausing and Resuming a Scheduled Job

If you want to temporarily suspend a scheduled job until you are ready to start it again, you can pause it. To do this, go to "System > Scheduled Jobs" and click the "Pause" icon ( ) for the scheduled job in question.
Transaction Manager will then simply not run the scheduled job until you resume it by clicking the "Resume" icon ( ).
If you need to pause or resume all scheduled jobs, you can use the "Pause All Jobs" and "Resume All Jobs" buttons at the top of the page.

Triggering and Interrupting a Scheduled Job

If the time for a scheduled job's next invocation has not yet come but you want it to run anyway, you can manually trigger the job (note: this option is not available for scheduled jobs that are currently paused). This does not affect scheduling at all; subsequent runs will take place as originally scheduled.

To manually trigger a scheduled job, go to "System > Scheduled Jobs" and click the "Trigger" icon ( ) next to the job. The job will be run straight away. If you ever find you need to terminate a running scheduled job (this may be because of bugs in the job or data that the job cannot handle), you can use the "Interrupt" icon ( ). Clicking it will stop the current instance of the scheduled job, if one exists.

Deleting a Scheduled Job

If a scheduled job is no longer needed, you can either disable it by setting an end date (after which it will no longer be run) or delete it entirely.

To delete a scheduled job, go to "System > Scheduled Jobs" and click the "Delete" icon ( ) for the job you want to remove. You will be asked to confirm this action.

Pre-Defined Scheduled Jobs

Transaction Manager comes with a number of scheduled jobs set up and ready to go. They perform useful functions and should not be paused or removed without a reason. Read on to find out more about these scheduled jobs.

The Collaboration Job Controller Job

The Collaboration Job Controller job is responsible for processing current collaboration jobs by calling the Job Controller service associated with the job. Please see Collaboration Jobs (Manager v5.0) for more details.

The Data Keystore Rollover Job

The Data Keystore Rollover job is responsible for creating new keystores for organization if automatic rollover is configured. Please see Operations (Manager v5.0)#_sec_dataretentionorganizationlevel for more information.

The Data Retention Job

The Data Retention job calls the default Data Retention Management service, which is responsible for purging Transaction Manager data such as submissions and error logs. Please see Operations (Manager v5.0)#_sec_dataretention and Services (Manager v5.0)#_sec_dataretentionservice for more information.
Note that other jobs will not run while the data retention job is running, to avoid contention.

The Delivery Escalation Job

The Delivery Escalation job sends out escalation emails for submissions delivered via secure email delivery (see Delivery Configuration (Manager v5.0)#_sec_secureemaildelivery).

The Delivery Reminder Job

The Delivery Reminder job sends out reminder emails for submissions delivered via secure email delivery (see Delivery Configuration (Manager v5.0)#_sec_secureemaildelivery).

The Email Queue Job

The Email Queue job sends out emails placed into the email queue (see Outgoing Email (Manager v5.0)#_sec_emailqueue).

The HTTP Invoker Job

The HTTP Invoker can invoke a HTTP URL. The URL is configured as the context parameter of the job.

The Payment Reminder Job

The Payment Reminder job loads the default payment reminder service and calls it. The service will then send out a reminder email regarding incomplete payments (see Security (Manager v5.0)#_sec_predefinedgroups for the group to which emails will be sent by default).

The Run Reports Job

The Run Reports job goes through all report schedules set up in Transaction Manager. If it finds one that is due to be sent out, it will run the report, extract the result data and send out the report email.

The Scheduled Service Job

The Scheduled Service can be used to invoke a service of type "Scheduled Service". These services can be customized using a Groovy script, and therefore are suitable if you need to create custom logic and run it on a schedule.
The service that the job will invoke can be selected when editing the job.

The Security Policy Manager Job

The Security Policy Manager job is responsible for applying security policies such as user role and group expiry. It also removes user accounts that were registered by users, but for which email verification was not completed (and hence the accounts were never activated or used).

The Submission Email Verification Reminder Job

The Submission Email Verification Reminder job calls the default Submission Email Verification service, which sends out reminders for submissions where the user must verify their email address. For more on submission email verification, please refer to Form Configuration (Manager v5.0)#_sec_submissionemailverification.

The System Monitoring Job

The System Monitoring job monitors other important system jobs (currently the Transaction Processing and Email Queue jobs) and sends out emails to the "Receive System Alerts" group if these jobs do not exist or have not been running as they should.

The Template Version Deployment Job

The Template Version Deployment job is responsible for applying scheduled form version changes (see Form Version Configuration (Manager v5.0)#_sec_changecurrentformversion)

The Transaction Processing Job

The Transaction Processing job calls the default Transaction Processor service, which performs submission processing in the following order: Abandonment, receipting, delivery, task expiry and transaction history creation.

The Virus Scan Job

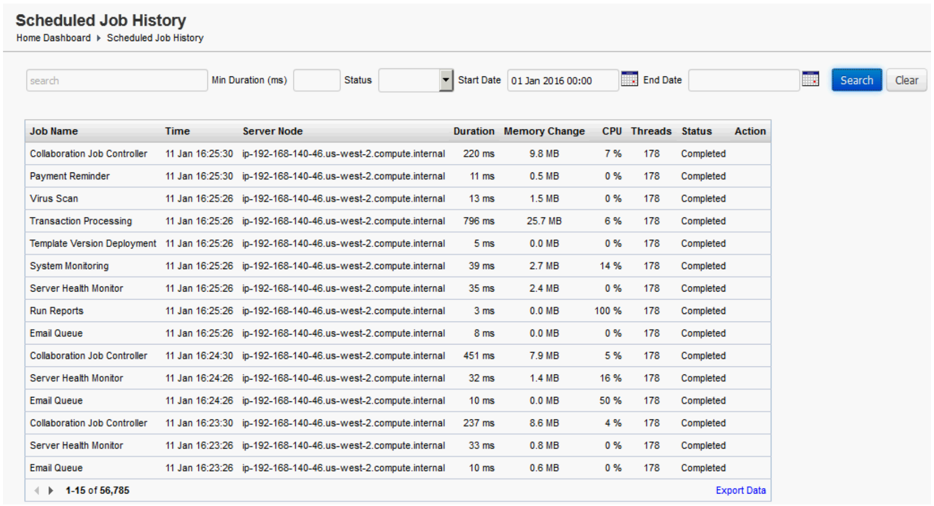
If a virus scan service is defined, the Virus Scan job loads a list of all attachments that need to be virus scanned and then call the default virus scan service for each attachment, setting the attachment's virus scan status appropriately.

Scheduled Job History

Every time a scheduled job runs, TM logs a history event recording which job ran, the result (success/error) and some performance metrics.

To view the scheduled job history, go to "System > Scheduled Job History". You can filter the result list by a substring of the job name, the run duration, the status ("Completed"/"Error") and the run date.

You can also export the search results to Excel by clicking the "Export Data" link at the bottom of the table.


Scheduled Job History
Home Dashboard Scheduled Job History
search Min Duration (ms) Status Start Date 01 Jan 2016 00:00 End Date Search Clear

Job Name	Time	Server Node	Duration	Memory Change	CPU	Threads	Status	Action
Collaboration Job Controller	11 Jan 16:25:30	ip-192-168-140-46.us-west-2.compute.internal	220 ms	9.8 MB	7 %	178	Completed	
Payment Reminder	11 Jan 16:25:30	ip-192-168-140-46.us-west-2.compute.internal	11 ms	0.5 MB	0 %	178	Completed	
Virus Scan	11 Jan 16:25:26	ip-192-168-140-46.us-west-2.compute.internal	13 ms	1.5 MB	0 %	178	Completed	
Transaction Processing	11 Jan 16:25:26	ip-192-168-140-46.us-west-2.compute.internal	796 ms	25.7 MB	6 %	178	Completed	
Template Version Deployment	11 Jan 16:25:26	ip-192-168-140-46.us-west-2.compute.internal	5 ms	0.0 MB	0 %	178	Completed	
System Monitoring	11 Jan 16:25:26	ip-192-168-140-46.us-west-2.compute.internal	39 ms	2.7 MB	14 %	178	Completed	
Server Health Monitor	11 Jan 16:25:26	ip-192-168-140-46.us-west-2.compute.internal	35 ms	2.4 MB	0 %	178	Completed	
Run Reports	11 Jan 16:25:26	ip-192-168-140-46.us-west-2.compute.internal	3 ms	0.0 MB	100 %	178	Completed	
Email Queue	11 Jan 16:25:26	ip-192-168-140-46.us-west-2.compute.internal	8 ms	0.0 MB	0 %	178	Completed	
Collaboration Job Controller	11 Jan 16:24:30	ip-192-168-140-46.us-west-2.compute.internal	451 ms	7.9 MB	5 %	178	Completed	
Server Health Monitor	11 Jan 16:24:26	ip-192-168-140-46.us-west-2.compute.internal	32 ms	1.4 MB	16 %	178	Completed	
Email Queue	11 Jan 16:24:26	ip-192-168-140-46.us-west-2.compute.internal	10 ms	0.0 MB	50 %	178	Completed	
Collaboration Job Controller	11 Jan 16:23:30	ip-192-168-140-46.us-west-2.compute.internal	237 ms	8.6 MB	4 %	178	Completed	
Server Health Monitor	11 Jan 16:23:26	ip-192-168-140-46.us-west-2.compute.internal	33 ms	0.8 MB	0 %	178	Completed	
Email Queue	11 Jan 16:23:26	ip-192-168-140-46.us-west-2.compute.internal	10 ms	0.6 MB	0 %	178	Completed	

1-15 of 56,785 Export Data

The scheduled job history

If an error occurred during job execution, you will be able to view the error log by clicking the "Error Log" icon ( ).

Because TM runs many scheduled jobs as part of normal operations, the scheduled job history can contain a large number of entries. TM purges the scheduled job history as part of its data retention policies. Please refer to Operations (Manager v5.0)#_sec_dataretention for more information on data retention.

Deployment Properties (Manager v5.0)

Much of the functionality implemented in Transaction Manager can be customized without the need to rebuild the application. Among other things, this was accomplished by adding *deployment properties* that control certain behavior.
Deployment properties can be set at runtime the Transaction Manager management console. This section tells you what deployment properties there are and which ones you should modify after installing Transaction Manager.

Page Contents

Search

Current Deployment Properties

The following are the deployment properties set up in Transaction Manager:

Property Name	Description
Alternative Redirect User Agents	Describes a set of user agents requiring alternative HTTP redirect mechanisms
Attachments Max Embedded Size	The maximum size of a single embedded (PDF) attachment
Attachments Max Total Size	The global maximum combined size of all attachments for a single submission.
Cache Timeout	Deployment property cache timeout in seconds
Config Directory	Transaction Manager configuration root directory.
Date Format Long	Long date format.
Date Time Format Long	Long date time format.
Debug Mode	Is system in Debug mode?
Duplicate Submission Rejection Delay	Time in milliseconds after which a duplicate submission is rejected
Email Default Sender	Default sender email address for emails sent by Transaction Manager
Email Duplicate Payment Message	Body text template for emails notifying administrators of a duplicate payment
Email Duplicate Payment Subject	Subject template for emails notifying administrators of a duplicate payment
Email Fraud Payment Message	Body text template for emails notifying administrators of a possibly fraudulent payment
Email Fraud Payment Subject	Subject template for emails notifying administrators of a possibly fraudulent payment
Email Lost Password Message	Lost password email message
Email Lost Password Subject	Lost password email subject
Email Promotion Message	Form promotion notification body
Email Promotion Sender	The email address for the promotion notification messages
Email Promotion Subject	Form promotion notification subject
Email Report Message	Report delivery email message
Email Report Sender	Report delivery sender email address
Email Report Subject	Report delivery email subject
Email Submission Status Message	Submission processing status update email messageNote: Can be overwritten on the organization and form level by using organization and form version properties.
Email Submission Status Subject	Submission processing status update email subjectNote: Can be overwritten on the organization and form level by using organization and form version properties.

Email Task Notification Message	Task notification body
Email Task Notification Subject	Task notification subject
Email Task Sender	The email address shown as the sender for task notification messages
Email Welcome User Message	User welcome email body
Email Welcome User Subject	User welcome email subject
Enable REST/Web Service Delivery	Enable REST and web service delivery for submissions
Enable Script Submissions	Enable test submission scripts by switching off submission request log checks
Enable Web Service Delivery	Enable/disable web service delivery. This can be used to temporarily suspend web service delivery, e.g. during high load or web service maintenance.
Groovy Logging Enabled	Enable logging of Groovy service calls
Groovy Script Timeout	GroovyScript execution timeout in milliseconds
HTTPS Port	Port on which HTTPS operates
LC Form Servers Deployed	If this instance of TM does not use any Adobe LiveCycle servers, set this property to "false"
LC Form Servers Load Balanced	Enable this setting to use load balancing when assigning tasks to Adobe LiveCycle nodes
LiveCycle Repository Domain	Relative location of LiveCycle repository domain for form version publishing
Online Save Enabled	Option to globally disable Online Saving of forms. If set to true, the individual form's setting will be used.
Report Internal URL	Location where reports are run internally by the application (i.e. http://localhost:9080/webreport)
Report Reschedule Delay	Amount of time in minutes to wait before rescheduling a running report.
Server Real Time Health Monitoring	Enable / disable real-time server health monitoring
SMTP Host	SMTP email server Hostname (mail.smtp.host)
SMTP Password	SMTP email server authentication Password (mail.smtp.password)
SMTP Port	SMTP email server Port (mail.smtp.port)
SMTP User	SMTP email server authentication User (mail.smtp.user)
Supported Delivery Methods	Supported delivery methods

Each deployment property is initialized to a default value but can be changed freely. Be aware that changes may impact various functionality in Transaction Manager, so make sure you take care.

Important Deployment Properties

The following deployment properties should be modified after setting up the Transaction Manager management console:

Directories

Transaction Manager uses directories to store various objects such as attachments and report files. By default, these files are stored in a subfolder of the folder where the TM server is installed. You can adjust this by setting the "Config Directory" property to any directory you want. If you do, ensure that the TM service has read/write access to the directory.
Edit the property "LiveCycle Repository Domain" and enter the path (relative to the LiveCycle repository) where Transaction Manager shall publish its forms. If multiple Transaction Manager instances are sharing the same LiveCycle repository, make sure to enter a unique path (e.g. " /tm-test-thirdparty /TransactionManagerRepository/").

SMTP Properties

If you want Transaction Manager to be able to send emails (e.g. for delivery purposes), you have to configure the SMTP settings. This is best done by accessing "System > Email Settings".

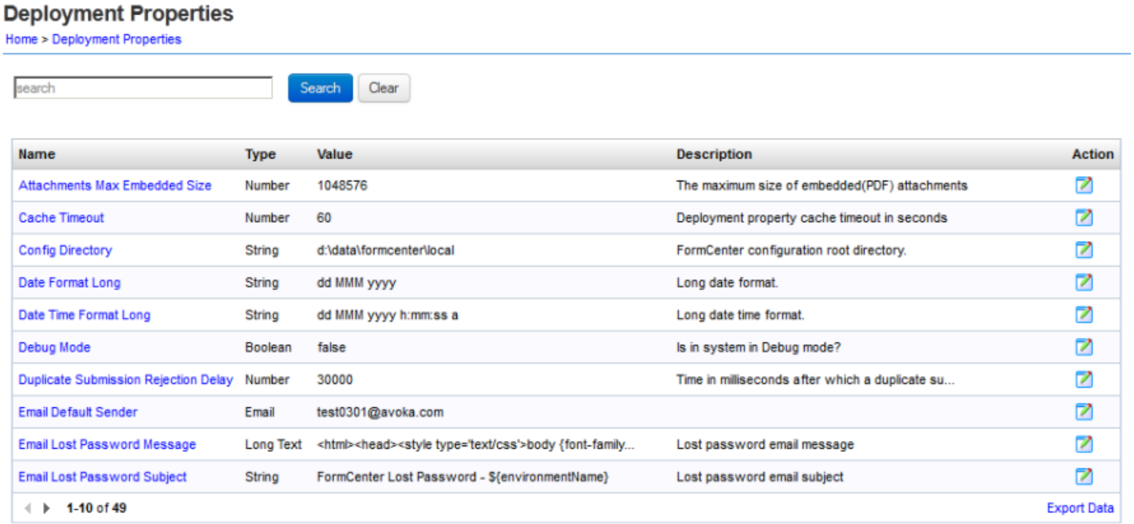
Duplicate Form Submissions

Transaction Manager offers some options to control duplicate submissions, which are multiple submission attempts of a requested form. This can happen either because the user attempts to submit the same form multiple times (by double-clicking the submit button or saving the form offline and then opening multiple copies of it) or commonly in system testing, where scripts are used to automate a number of submissions.

By default, Transaction Manager blocks duplicate submissions when they are more than a certain time interval (controlled by the "Duplicate Submission Rejection Delay" deployment property) apart. If duplicate submissions occur within a certain short period, Transaction Manager assumes that they originate due to accidental double-clicking of the submit button and ignores the duplicates.
If you would like to enable duplicate submissions (e.g. for load testing), change the "Enable Script Submissions" deployment property to "true".
If you would like to customize the message displayed to space users when a duplicate submission is rejected, use the in-built form property named "Submission Duplicate Message".

Browsing Deployment Properties

To view the list of all deployment properties and their current values, go to "System > Deployment Properties".



Name	Type	Value	Description	Action
Attachments Max Embedded Size	Number	1048576	The maximum size of embedded(PDF) attachments	
Cache Timeout	Number	60	Deployment property cache timeout in seconds	
Config Directory	String	d:\data\formcenter\local	FormCenter configuration root directory.	
Date Format Long	String	dd MMM yyyy	Long date format.	
Date Time Format Long	String	dd MMM yyyy h:mm:ss a	Long date time format.	
Debug Mode	Boolean	false	Is in system in Debug mode?	
Duplicate Submission Rejection Delay	Number	30000	Time in milliseconds after which a duplicate su...	
Email Default Sender	Email	test0301@avoka.com		
Email Lost Password Message	Long Text	<html><head><style type='text/css'>body {font-family...	Lost password email message	
Email Lost Password Subject	String	FormCenter Lost Password - ${environmentName}	Lost password email subject	

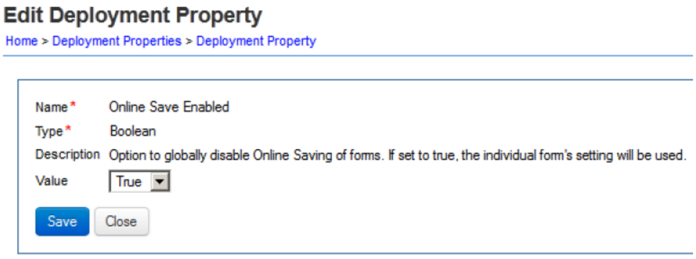
Browsing the list of deployment properties

You can filter the list using a substring of the deployment property's name and/or value, and export the search results to Excel by clicking the "Export Data" link below the table.

You cannot add or remove deployment properties, but you can edit them using the "Edit" icon ( ).

Modifying a Deployment Property

To edit a deployment property's value, go to "System > Deployment Properties" and click the "Edit" icon next to it. You will be directed to the "Edit Deployment Property" page.



Changing a deployment property's value

Enter a new value and save the deployment property. Your changes will take effect almost immediately (by default, deployment properties are cached with a refresh time of 30 seconds)

Appendices (Manager v5.0)

Appendix 1 – Submission Status Values

The table below shows some important combinations of submission status values (form status, attachment status, payment status, receipt status and delivery status) as well as the aggregate "Transaction Status" string displayed on the TM submission search page.
Note that there are more statuses and combinations that are not shown below. Generally the transaction status will indicate the relevant status applicable to the submission and its value (e.g. required, ready, completed, error).

Form	Attachments	Payment	Receipt	Delivery	Tranaction
Saved	-	-	-	-	Form Saved
Submitted	Required	-	-	-	Form Submitted, Attachments Outstanding
Submitted	-	Required	-	-	Form Submitted, Payment Required
Submitted	Required	Required	-	-	Form Submitted, Attachments Outstanding and Payment Required
Completed			Ready		Receipt Render Pending
Completed			Completed	Ready	Delivery Ready, orDelivery Ready, Method Not Configured
Completed			Completed	Completed	Delivery Completed

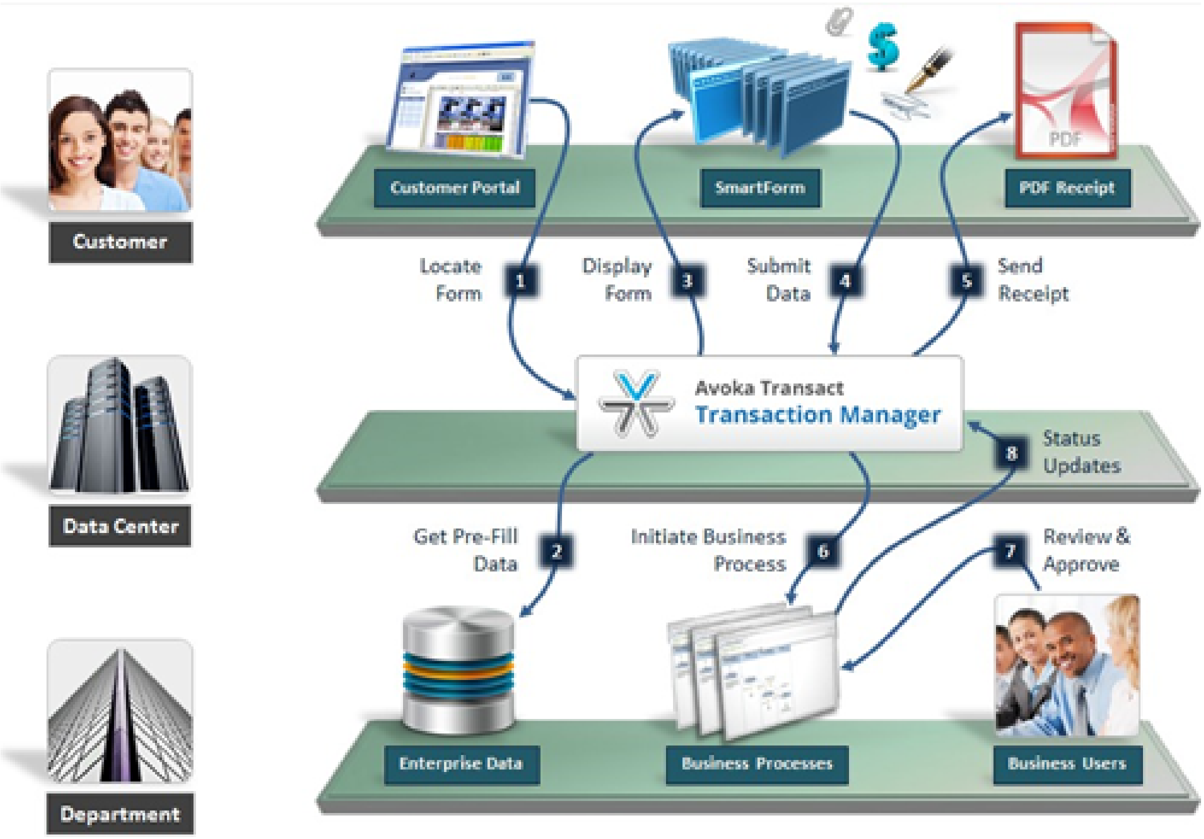
Transact Web Service Guide (Manager v5.0)

Introduction

This document provides a guide to the Avoka Transact integration SOAP Web Services. These web services include:

- Forms Server SOAP Web Service
- Portal SOAP Web Service
- Task and User SOAP Web Service

Using the integration web services you can develop sophisticated form based solutions which include user prefill data, receipt number generation, business process integration and submission processing status updates for customers.



Transaction Manager Web Service Interactions

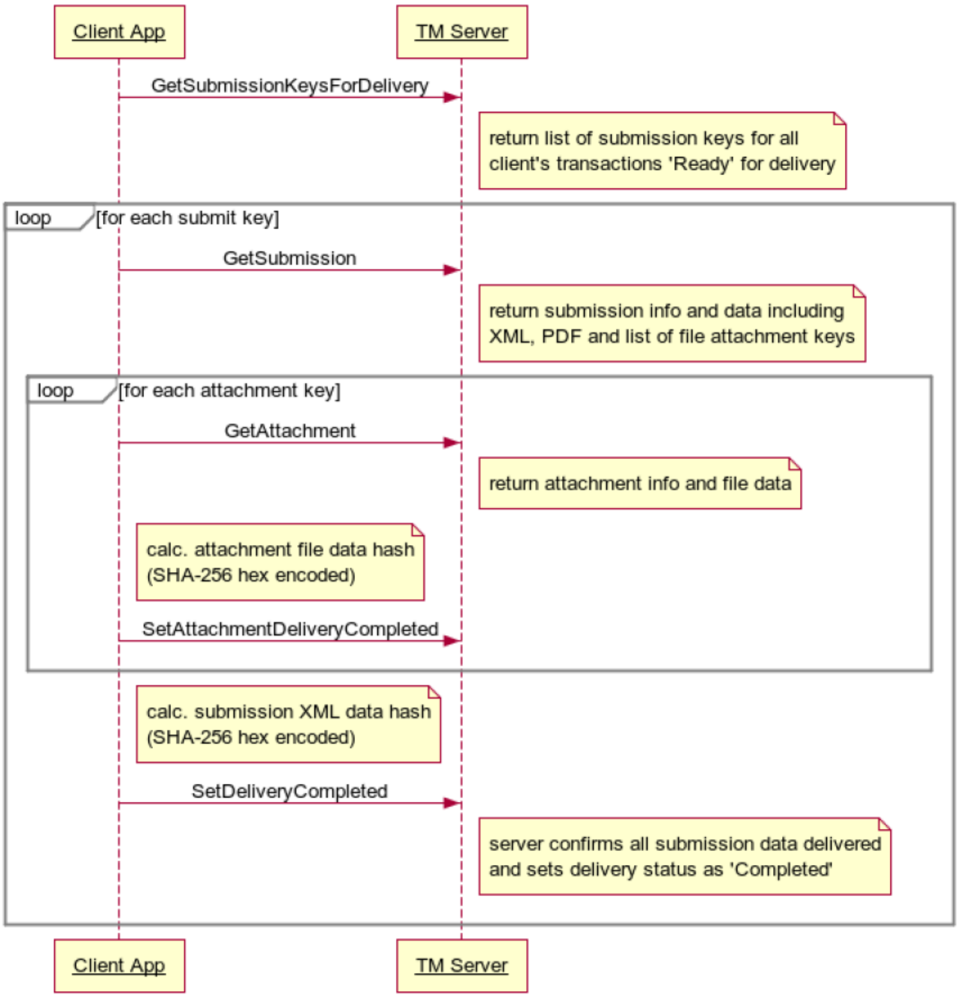
Forms Server SOAP Web Service (Manager v5.0)

Overview

The Form Server web service enables you to retrieve completed form submissions from a remote Transaction Manager server and provide processing status updates.

Submission Delivery

Submission delivery is completed through using a series of web service calls to the Transaction Manager server. These fine grained web service calls enable reliable delivery of submission and attachment data over Wide Area Networks.
The web service invocation sequence is illustrated in the diagram below:



Submission delivery sequence
The pseudo code representation for the submission delivery process is provided below:

```
get list of submission keys for delivery (GetSubmissionKeysForDelivery)
for each submission key in submission key list
                get submission data (GetSubmission)
                for each attachment key is submission attachment keys
                        get attachment data (GetAttachment)
                        confirm delivery of attachment data (SetAttachmentDeliveryComplete)
                confirm delivery of submission data (SetSubmissionDeliveryComplete)
```

When confirming the delivery of the submission XML data or the file attachment data the client needs to provide the Transaction Manager server with a data hash of the received data. The data hash must be a Hex encoded SHA-256 hash of the data. Please see the example code provided later.

Once this delivery has been confirmed the Transaction Manager server can mark the delivery as being completed and the data on the server can be purged.

If the web service client fails during this delivery process before calling *SetSubmissionDeliveryComplete* it can try delivery again at a future point in time. The server will still make the Submission available to the client to retrieve.

Once the web service delivery client has stored the submission in a persistent data store or queue it should call the server *SetSubmissionDeliveryComplete* method as soon as possible. The internal delivery system or processing workflow should not delay this step so that the Transaction Manager server can make this submission data available for purging.

Data Hash Utils

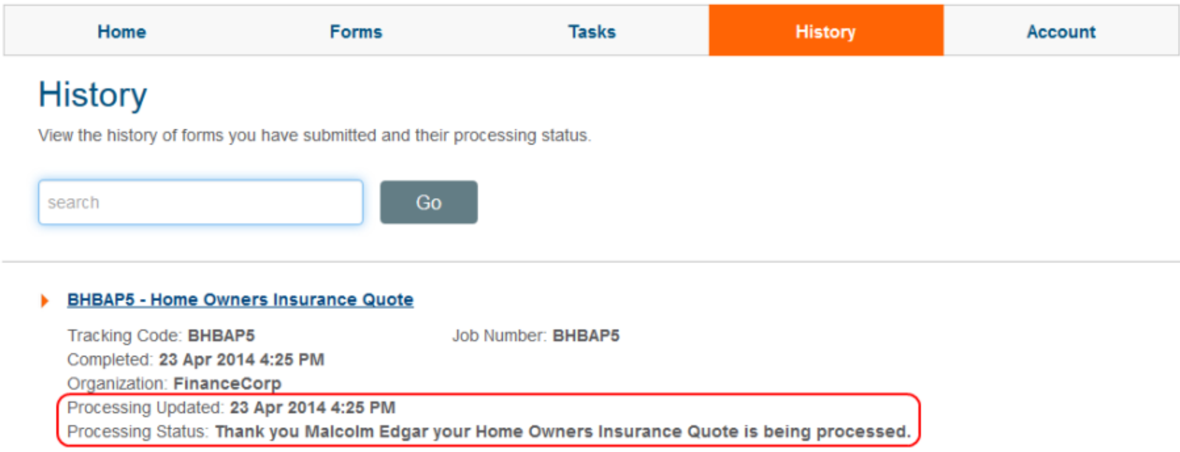
Example data hash configuration Java code is provided below:

```
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
/**
 * Provides Forms API data hashing utility methods.
 *
 * See SHA-2 definition: <a href="http://en.wikipedia.org/wiki/SHA-2">http://en.wikipedia.org/wiki/SHA-
*/
public class DataHashUtils {
        /** Hexidecimal characters for encoding. */
        private static final char[] HEXADECIMAL = { '0', '1', '2', '3', '4', '5','6', '7', '8', '9', 'a', 'b',
'c', 'd', 'e', 'f' };
        private static final String HEXADECIMAL_STRING = "0123456789abcdef";
        /**
        * Return a Hex encoded SHA-256 hash of the given form XML data or attachment file data.
        *
        * @param bytes the byte array to hash
        * @return SHA-256 hash, encoded as a 64 character Hex string
        */
        public static String toSHA256Hash(byte[] bytes) {
        if (bytes == null) {
                throw new IllegalArgumentException("Null bytes parameter");
        }

        try {
                MessageDigest md = MessageDigest.getInstance("SHA-256");
                md.update(bytes);
                byte[] binaryData = md.digest();
                // Hex encode data SHA-256 digest
                char[] buffer = new char[64];
                for (int i = 0; i < 32; i++) {
                        int low = (int) (binaryData[i] & 0x0f);
                        int high = (int) ((binaryData[i] & 0xf0) >> 4);
                        buffer[i * 2] = HEXADECIMAL[high];
                        buffer[i * 2 + 1] = HEXADECIMAL[low];
                }
                return new String(buffer);
        } catch (Exception e) {
                throw new RuntimeException(e);
        }
}

/**
 * Return a Hex encoded SHA-256 hash of the given string value. Please note the string
 * value will be encode as a byte array using the UTF-8 charset before hashing.
 *
 * @param plaintext string value to hash
 * @return SHA-256 hash, encoded as a 64 character Hex string
 */

public static String toSHA256Hash(String plaintext) {
        if (plaintext == null) {
                throw new IllegalArgumentException("Null plaintext parameter");
        }

        try {
                return toSHA256Hash(plaintext.getBytes("UTF-8"));

        } catch (UnsupportedEncodingException uee) {
                throw new RuntimeException(uee);
        }
}

/**
 * Decodes the hex encoded SHA-256 string returned by {@link #toSHA256Hash(String)} into a byte arr
 * @param encodedHash the hash string consisting of 64 hex characters
 * @return a byte array of length 32, representing the encoded hash in binary form
 */
public static byte[] decodeSHA256Hash(String encodedHash) {
        char[] buffer = encodedHash.toCharArray();
        byte[] result = new byte[32];
```

```
        for (int i = 0; i < 32; i++) {
                char lowChar = buffer[i*2 + 1];
                char highChar = buffer[i*2];
                int low = HEXADECIMAL_STRING.indexOf(lowChar);
                int high = HEXADECIMAL_STRING.indexOf(highChar);
                result[i] = (byte) ((high << 4) | low);
        }
        return result;
}
/**
 * Provide a test harness to generate hash values for a given file.
 *
 * @param args the program arguments
*/
public static void main(String[] args) {
        if (args.length != 2) {
                System.out.println("Usage: java -jar DataHashUtils.jar -filename <filename>\n");
                System.out.println("where:");
                System.out.println(" -filename the name of the file to generate the SHA-256 hash for"
                return;
        }
        if (!args[0].equals("-filename")) {
                System.out.println("Usage: java -jar DataHashUtils.jar -filename <filename>\n");
                System.out.println("where:");
                System.out.println("-filename the name of the file to generate the SHA-256 hash for"
                return;
        }

        String filename = args[1];
        File file = new File(filename);
        if (!file.isFile() || !file.canRead()) {
                System.out.println("Cannot read the specified file: " + filename + "\n");
                System.out.println("Usage: java -jar DataHashUtils.jar -filename <filename>\n");
                System.out.println("where:");
                System.out.println("-filename the name of the file to generate the SHA-256 hash for"
                return;
        }
        long length = file.length();
        // Create the byte array to hold the data
        byte[] bytes = new byte[(int)length];
        FileInputStream fis = null;
        try {
                fis = new FileInputStream(filename);
                // Read in the bytes
                int offset = 0;
                int numRead = 0;
                while (offset < bytes.length
                                && (numRead = fis.read(bytes, offset, bytes.length-offset)) >= 0) {
                        offset += numRead;
                }

                // Ensure all the bytes have been read in
                if (offset < bytes.length) {
                        throw new IOException("Could not completely read file " + file.getName());
                }
                System.out.println("Hex encoded SHA-256 data hash\n");
                System.out.println(" File name: " + file.getCanonicalPath());
                System.out.println(" Data hash: " + toSHA256Hash(bytes));

                } catch (IOException ioe) {
                        ioe.printStackTrace();
                } finally {
                        if (fis != null) {
                                try {
                                        fis.close();
                                } catch (IOException e) {
                                }
                        }
                }
        }
}
```

Submission Processing Status Updates

The Form Server web service also enables you to provide submission processing status updates which can be presented to end users and operations staff. This feature is great for providing immediate feedback to customers about the status of their form submission, which in turn reduces calls to your help desk for information.



Submission processing status update in Self Service Portal

Application Programmer Interface

The Web Services Description Language (WDSL) for the Forms Server web service is available on the server URL:

http://<server name>/manager/fcservices/FormsServerWebService?wsdl

The Forms Server web service provides of the following operations:

- GetSubmissionKeysForDelivery
- GetSubmission
- GetAttachment
- SetAttachmentDeliveryComplete
- SetSubmissionDeliveryComplete
- UpdateProcessingStatus
- UpdateProcessingStatusAndEmail

GetSubmissionKeysForDelivery

This operation returns the list of submission key identifiers for submissions with a delivery status of 'Ready' for the given client key.

Input Parameters

- clientKey: the organization client key identifier. This must be a valid client key known to Transaction Manager.

Output Result

- a list of strings containing the submit keys of all submissions ready for delivery to the client organization.

Faults

- Exception: if the organization could not be resolved via clientKey or an unexpected error occurs invoking this method.

GetSubmission

This operation returns the submission object for the given client key and submit key.
To confirm the delivery of the form XML data a Hex encoded SHA-256 hash of the fileDataBytes
element needs to be created for the SetSubmissionDeliveryComplete call.

Input Parameters

- clientKey: the organization client key identifier
- submissionKey: the submission submit key identifier. The submission identified by this key must exist and be associated with the organization identified by clientKey, and be ready for web service delivery.

Output Result

- Submission (Complex Type)

A Submission type provides the following elements:.

  - formCode - a reference to the form that was used for this submission.

- formName - the name of the form used for this submission
- formDataBytes - the XML Data for this submission, this is a base64 encoded text value of the original file bytes data
- formTemplateVersion - the form template version number
- pdfReceiptBytes - a flattened PDF copy of the receipt o receiptNumber - the form receipt submission number o submissionId - the submission id
- submissionKey - the submission surrogate key;
- submissionTime - the time the submission was lodged
- attachmentKeys - a list of the attachment keys that are associated with the submission, which can be used to retrieve the attachments via GetAttachment

Faults

- Exception: if one of the input parameters violates a constraint (e.g. invalid submissionKey) or an unexpected error occurs invoking this method.

GetAttachment

This operation returns the attachment object for the given client key and attachment key.
To confirm the delivery of the attachment a Hex encoded SHA-256 hash of the fileBytes element needs to be created for the SetAttachmentDeliveryComplete call.

Input Parameters

- clientKey: the organization client key identifier

- attachmentKey: the submission attachment key identifier. The attachment identified by this key must exist and be associated with the organization identified by clientKey.

Output Result

- Attachment (Complex Type)

The Attachment type provides the following elements:

- attachmentKey - the attachment key identifier
- attachmentName - the configured attachment name
- documentCode - the configured document code
- documentTypeName - the configured attachment type
- description - the user entered description
- fileName - the name of the file attached by the user
- fileBytes - the attachment file data uploaded by the user, this is a base64 encoded text value of the original file bytes data
- submittedManuallyFlag - specifies whether the user elected to provide the attachment manually
- requiredFlag - specifies whether providing the attachment is required

Faults

- Exception: if one of the input parameters violates a constraint (e.g. invalid attachmentKey) or an unexpected error occurs invoking this method.

SetAttachmentDeliveryComplete

This operation sets the delivery status for an attachment to 'Complete'.

Input Parameters

- clientKey: the organization client key identifier
- attachmentKey: the attachment key identifier. The attachment identified by this key must exist and be associated with the organization identified by clientKey. There must also be a file associated with the attachment.
- dataHash: the Hex encoded SHA-256 hash value of the attachment (fileBytes) data

Output Result

None

Faults

- Exception: if one of the input parameters violates a constraint (e.g. invalid attachmentKey) or an unexpected error occurs invoking this method.

SetSubmissionDeliveryComplete

This operation sets the delivery status for a submission to 'Complete'. Before invoking this method, please invoke SetAttachmentDeliveryComplete for each submission attachment.

Input Parameters

- clientKey: the organization client key identifier
- submissionKey: the submit key identifier. The submission identified by this key must exist and be associated with the organization identified by clientKey.
- dataHash: the Hex encoded SHA-256 hash value of the submission (formDataBytes) data

Output Result

None

Faults

- Exception: if one of the input parameters violates a constraint (e.g. invalid submissionKey) or an unexpected error occurs invoking this method.

UpdateProcessingStatus

This method set the submission processing status to the given status value.

Input Parameters

- clientKey: the organization client key identifier
- submissionKey: the form submission key identifier. The submission identified by this key must exist and be associated with the organization identified by clientKey.
- processingStatus: the current processing status of the form submission by the organization. The maximum length of the processing status is 100 characters, longer values will be truncated.

Output Result

- None: update processing status. If the submission was performed by an authenticated user, and that user has elected to receive notification emails then an notification email will be sent

Faults

- Exception: if an error occurs invoking this method.

UpdateProcessingStatusAndEmail

This method set the submission processing status to the given status value and send out a customized email to the submission user (if known).

Input Parameters

- clientKey: the organization client key identifier
- submissionKey: the form submission key identifier. The submission identified by this key must exist and be associated with the organization identified by clientKey.
- processingStatus: the current processing status of the form submission by the organization. The maximum length of the processing status is 100 characters, longer values will be truncated.
- emailSubject: the email subject value of the notification email
- emailBody: the email notification message body. Email message template variable you can use include:

    ${formDataExtract}
    ${user.fullName}
    ${user.givenName}
    ${user.familyName}
    ${user.email}
    ${user.loginName}
    ${submission.receiptNumber}

Output Result

- None: update processing status. If the submission was performed by an authenticated user, and that user has elected to receive notification emails then an notification email will be sent.

Faults

- Exception: if an error occurs invoking this method.

Portal SOAP Web Service (Manager v5.0)

Overview

This web service enables organizations to integrate Avoka Transact capabilities into their existing Portals and web sites.
Using these web services you can to create the following types of portal pages or views: form landing page
form submission confirmation page form list view / search
form ToDo item list view /search form submission history view / search

Web Service Security

The Portal Web Service now supports invoker authentication and authorization security access control. Web service can be optionally protected using HTTPS BASIC authentication, against Transaction Manager user accounts. When access control is enforced these user accounts must have the role permission "Web Service Invoke" assigned to them.
Web service security can be enabled and disabled using the Transaction Manager portal property "Authenticated Web Services".

Web Service Security Transation Manager Configuration Option

Application Programmer Interface

The Web Services Description Language (WDSL) for the Portal web service is available on the server URL: https://<server name>/manager/secure/services/PortalWebService?wsdl
The Application Programmer Interface (API) to the Portal web service consists of the following operations:

- GetFormDefinition
- GetFormDefinitionList
- GetFormSubmission
- GetFormSubmissionList
- GetFormTodoList
- DeleteFormTodo

GetFormDefinition

This operation returns the FormDefinition matching the specified Portal Name and Form Code.

Input Parameters

- portalName: the name of the Portal (required)
- formCode: the globally unique form code identifier (required)

Output Result

FormDefinition (Complex Type)
The FormDefinition type contains the following elements:

- formCode - the globally unique form code identifer
- formName - the organization unique form name
- formVersion - the currently active form version number
- formURL - the URL to access the form with
- orgName - the owning organization name
- testMode - specifies whether the form is in test mode
- categoryNames - the list of form version category names used to classify the form
- formProperties - the list of form version properties The FormProperties type contains the following elements o name - the form version property name
- value - the form version property value

Faults

- ObjectNotFoundException: if the form specified by the Portal Name or Form Code could not be found.

GetFormDefinitionList

This operation returns a list of FormDefinition items matching the search criteria. The result set will be ordered by form name.

Input Parameters

- portalName: the name of the Portal (required)
- groupNames: the form group name associated with the forms (optional), will filter the result set if specified
- keyword: the free text search keyword (optional), will filter the result set if specified
- formCategories: the list of form category search values (optional), will filter the result set if specified
- fetchLimit: the result set fetch limit, default value is 1,000 items if not specified. If specified it must be in range of 1-10,00 (optional)

Output Result

FormDefinition (Complex Type) list
The FormDefinition type contains the following elements:

- formCode - the globally unique form code identifer
- formName - the organization unique form name
- formVersion - the currently active form version number
- formURL - the URL to access the form with
- orgName - the owning organization name
- testMode - specifies whether the form is in test mode
- categoryNames - the list of form version category names used to classify the form
- formProperties - the list of form version properties The FormProperties type contains the following elements o name - the form version property name
- value - the form version property value

Faults

ObjectNotFoundException: if the specified portal could not be found.

GetFormSubmission

This operation return the FormSubmission matching the specified Portal Name and Submit Key.

Input Parameters

- portalName: the name of the Portal (required)
- submitKey: the globally unique form submit key identifier (required). This is a 32 character Hex encoded UUID value.

Output Result

FormSubmission (Complex Type)
The FormSubmission type contains the following elements:

- submissionId - the globally unique form submission id (submission record primary key)
- submitKey - the globally unique submit key, which provides an alternative key using a UUID value
- userName - the form submitter user login name
- submissionTime - the time the form was submitted

- completionTime - the time the all the required user steps have been completed, e.g. attachments, payments etc. This value will often be the same as the submissionTime.
- receiptNumber - the submission receipt / reference number.
- receiptURL - the URL to access the PDF receipt document
- processingStatus - the latest submission processing status value to be displayed on the submitter
- processingStatusTime - the creation time of the latest submission processing status
- submissionMessage - the submission message displayed to the user on the submission confirmation page, generally used to provide additional instructions
- paymentAmount - the total payment amount associated with the form transaction o numberAttachments - the number of attachments uploaded with form submission o orgName - the name of the organization which owns the form
- formName - the name of the submitted form
- formCode - the globally unique code of the submitted form
- xmlDataExtracts - the list of form submission data XML extracts The XmlDataExtracts type contains the following elements
- name - the submission data extract name
- value - the submission data extract value
- xpath - the submission data extract XPath expression

Faults

- ObjectNotFoundException: if the form submission specified by the Portal Name and Submit Key could not be found.

GetFormSubmissionList

This operation returns a list completed of FormSubmission items matching the search criteria. The result set will be ordered by submission completed time descending.

Input Parameters

- portalName: the name of the Portal (required)
- userName: the form submitter user login name (required)
- groupNames: the submission group names (optional), will filter the result set if specified
- keyword: the search keyword (optional), will filter the result set if specified
- fetchLimit: the result set fetch limit, default value is 1,000 items if not specified. If specified it must be in range of 1-10,00 (optional)

Output Result

FormSubmission (Complex Type) list
The FormSubmission type contains the following elements:

- submissionId - the globally unique form submission id (submission record primary key)
- submitKey - the globally unique submit key, which provides an alternative key using a UUID value
- userName - the form submitter user login name
- submissionTime - the time the form was submitted
- completionTime - the time the all the required user steps have been completed, e.g. attachments, payments etc. This value will often be the same as the submissionTime.
- receiptNumber - the submission receipt / reference number.
- receiptURL - the URL to access the PDF receipt document
- processingStatus - the latest submission processing status value to be displayed on the submitter
- processingStatusTime - the creation time of the latest submission processing status
- submissionMessage - the submission message displayed to the user on the submission confirmation page, generally used to provide additional instructions
- paymentAmount - the total payment amount associated with the form transaction o numberAttachments - the number of attachments uploaded with form submission o orgName - the name of the organization which owns the form
- formName - the name of the submitted form
- formCode - the globally unique code of the submitted form
- xmlDataExtracts - the list of form submission data XML extracts The XmlDataExtracts type contains the following elements
- name - the submission data extract name
- value - the submission data extract value
- xpath - the submission data extract XPath expression

Faults

- ObjectNotFoundException: if the form submission specified by the Portal Name and Submit Key could not be found.

GetFormToDoList

This operation returns a list of FormToDo items matching the search criteria. The result set will be ordered by item submission time and task creation time descending.

Input Parameters

- portalName: the name of the Portal (required)
- userName: the form submitter user login name (required)
- groupNames: the submission group names (optional), will filter the result set if specified
- keyword: the search keyword (optional), will filter the result set if specified
- filterType: the form Todo filter type [Completed | Submitted | Saved | Expired | Assigned], will filter the result set if specified (optional)
- fetchLimit: the result set fetch limit, default value is 1,000 items if not specified. If specified it must be in range of 1-10,00 (optional)

Output Result

FormTodo (Complex Type) list
The FormTodo type contains the following elements:

- todoKey - the globally unique identifier, either the Assigned or Completed Task Key or the Submission Key
- todoType - the ToDo item type: ['Assigned Task' | 'Completed Task' | 'Expired Task' | 'Saved Form' | 'Payment Required' | 'Attachments Required' | 'Wet Signature Required' | 'Authentication Required']
- actionURL - the URL to the open the task, saved form, portal action page URL or the completed task submission PDF receipt
- actionTitle - the user Todo action item short title description
- actionMessage - the user Todo action item long description message
- receiptNumber - the submission receipt / reference number
- formName - the name of the submitted form
- formCode - the globally unique code of the submitted form o orgName - the name of the organization which owns the form o userName - the form submitter user login name
- submissionTime - the time the form was submitted
- taskGroupName - the assigned Task form group name.
- taskCreationDate - the date time the assigned Task was created
- taskScheduledDate - the date time the assigned Task is scheduled to be performed
- taskExpiryDate - the date time the assigned Task will expire
- canDelete - the flag specifying whether the assigned task can be delete by the user
- xmlDataExtracts - the list of form submission data XML extracts The XmlDataExtracts type contains the following elements
- name - the submission data extract name
- value - the submission data extract value
- xpath - the submission data extract XPath expression

Faults

- ObjectNotFoundException: if the form submission specified by the Portal Name and Submit Key could not be found.

DeleteFormTodo

This operation deletes the FormToDo item for matching the ToDo Key.

Input Parameters

- toDoKey: the Task or Submission Key (required), 32 character Hex encoded UUID value (required)

Output Result

- Boolean: Return true if the operation could be completed (e.g. Task deletable or Submission saved) or false otherwise

Faults

- ObjectNotFoundException: if the item specified by the ToDo Key could not be found.

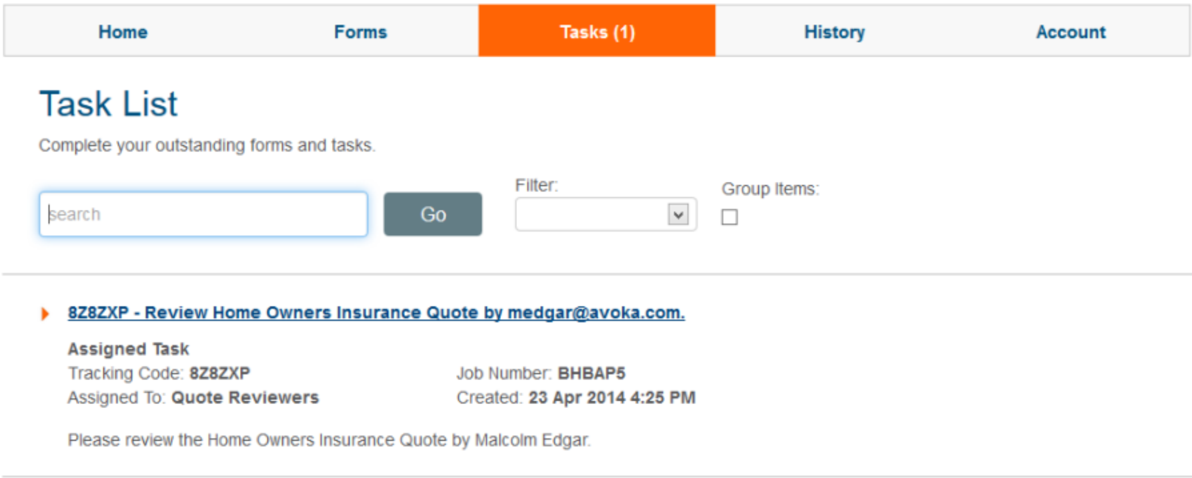
Task and User SOAP Web Service v2 (Manager v5.0)

Overview

The Task and User web service enables you to create form tasks, form groups and user accounts in Transaction Manager. By using the Task and User web service, external applications can assign forms to staff to complete and create multi-step workflows.

In Transaction Manager a task consists of a task description, a reference to a form, optional XML prefill data and some other attributes. Transaction Manager will present the task to the assigned user(s) in the Self Service Portal or Mobile FieldWorker, and will optionally notify users of assigned tasks.

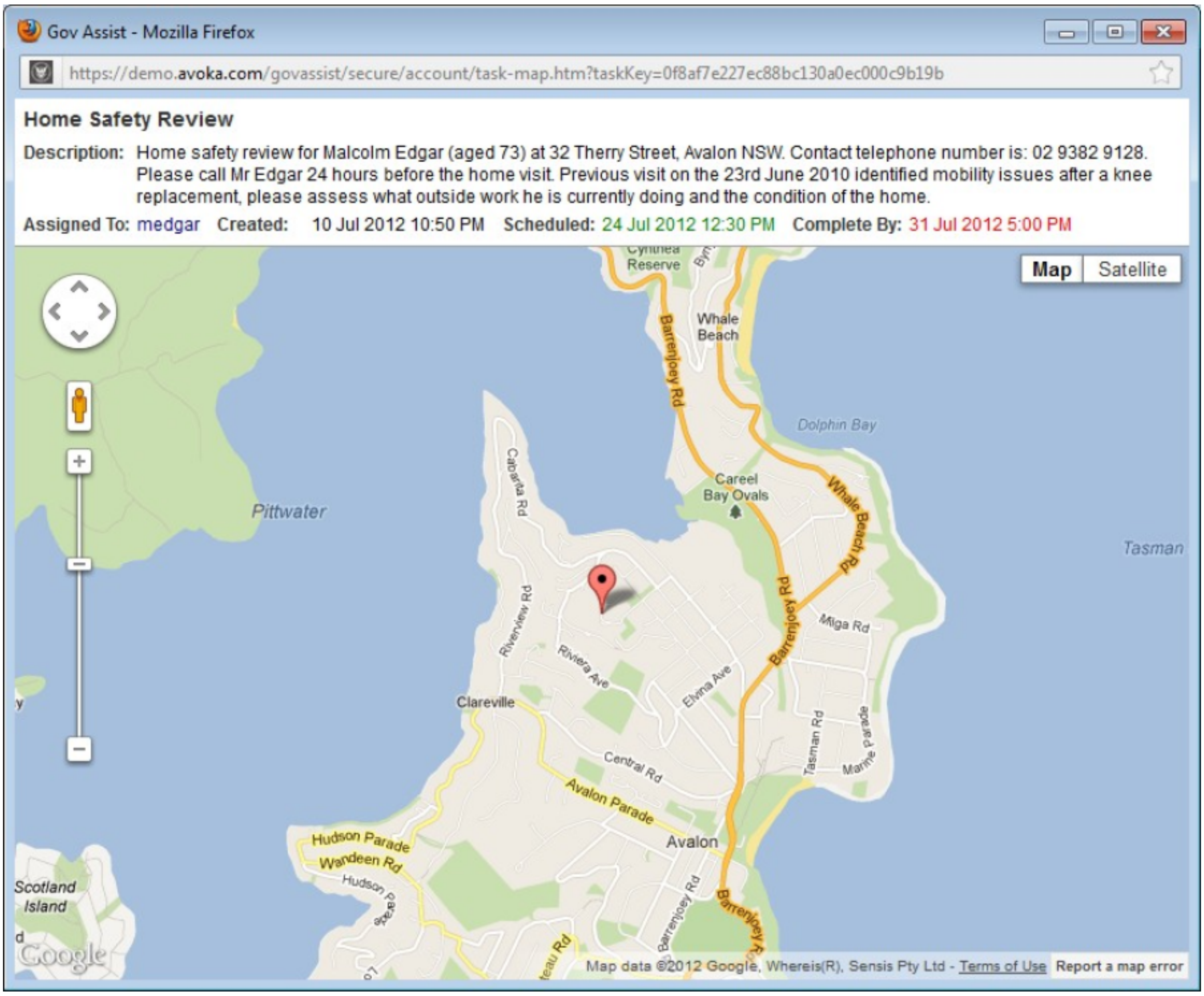
You can manually create tasks and edit tasks in Transaction Manager via the "Operations > Assigned Tasks" screen. This facility is particularly useful when developing web service integration with external systems.



Screenshot: Assigned Tasks screen in Transaction Manager

In Transaction Manager tasks can be assigned to individual users and/or to a group. When tasks are assigned to a group, all the users belonging to that group will be able to view that task.

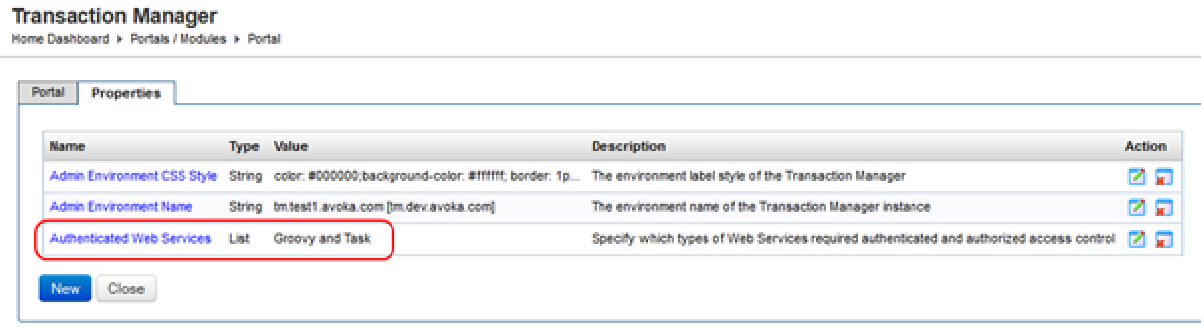
Tasks can also include geo-location latitude and longitude coordinates to support map displays in the Self Service Portal and Mobile FieldWorker applications.



Screenshot: Task Location Display

Web Service Security

The Task and User Web Service supports invoker authentication and authorization security access control. Web service can be optionally protected using HTTPS BASIC authentication, against Transaction Manager user accounts. When access control is enforced these user accounts must have the role permission "Web Service Invoke" assigned to them.
Web service security can be enabled and disabled using the Transaction Manager portal property "Authenticated Web Services".



Web Service Security Transation Manager Configuration Option

Application Programmer Interface

The Web Services Description Language (WDSL) for the Task and User web service v2 is available on the server URL:
https://<server name>/manager/secure/services/TaskUserWebServiceV2?wsdl

The Task and User web service v2 provides of the following operations.

- CreateLocalUser
- CreateFormGroup
- CreateFormTask
- CreateFormReviewTask
- ExpireTask
- AddUserToGroup
- RemoveUserFromGroup
- ReassignTask
- ReassignAllUserTasks

CreateLocalUser

This operation creates a "Local" type Transaction Manager user with the given parameters. Note that LDAP users will be looked up and created in TM automatically by the other operations, so this operation is not needed for LDAP users.
Please ensure the user password conforms to the configured Security Manager password policies. If the password does not meet the password requirements this operation will return a TaskUserWebServiceException fault.

Input Parameters

- userName: the user name (required)
- password: the user password (required, must conform to configured Security Manager password policies)
- email: the user email address (required)
- givenName: the first name (optional)
- familyName: the last name (required)
- portalName: the associated portal (required)
- changePwdAfterLogin: whether the user must change the password after login (required)

Output Result

None

Faults

- InvalidParameterException: if one of the parameters violated a constraint
- PortalNotFoundException: if the portal was not found
- PortalSecurityException: if the portal does not support local users
- UserAlreadyExistsException: if a user with the specified login name already exists
- TaskUserWebServiceException: if other errors occur

CreateFormGroup

This operation creates a form group with the given name.

Input Parameters

- groupName: the form group name (required)
- groupDescription: the form group description (optional)

Output Result

- boolean value: true if a group was created, or false if it already exists.

Faults

- InvalidParameterException: if one of the parameters violated a constraint
- TaskUserWebServiceException: if the group cannot be created (e.g. a non-form group with the same name exists)

CreateFormTask

This operation creates a new form task.

Input Parameters

- subject: the task subject (required)
- message: the task message (optional)
- formCode: the form code (required)
- userName: the assigned user name (optional; must be specified if groupName is not)
- groupName: the assigned form group name (optional; must be specified if userName is not)
- portalName: the user portal name (optional; must be specified if userName denotes an LDAP user)
- formDataXml: the form XML data or schema seed (optional; must not be specified if formPrefillXml is)
- formPrefillXml: the form prefill XML data to map into the form XML (conditional; required when formDataXml specified)
- latitude: the task location latitude (optional; must be between -180 and 180 if specified)
- longitude: the task location longitude (optional; must be between -90 and 90 if specified)
- scheduledTime: the scheduled task time (optional)
- expiryTime: the task expiry time (optional; must be in the future if specified)
- userDeletableFlag: whether the user can delete the task
- sendEmailFlag: whether to notify the assignee(s) of the task via email
- emailSubject: the notification email subject line
- emailMessage: the notification email message body. Email message template variable you can use include:
    - ${format} : Apache Click Format object use for formatting values
    - ${formDataMap} : form data extracts map
    - ${submission} : task submission object
    - ${task} : task submission object (as above)
    - ${environmentName} : server evironment name
    - ${supportEmail} : support email address, defined as a deployment property

Output Result

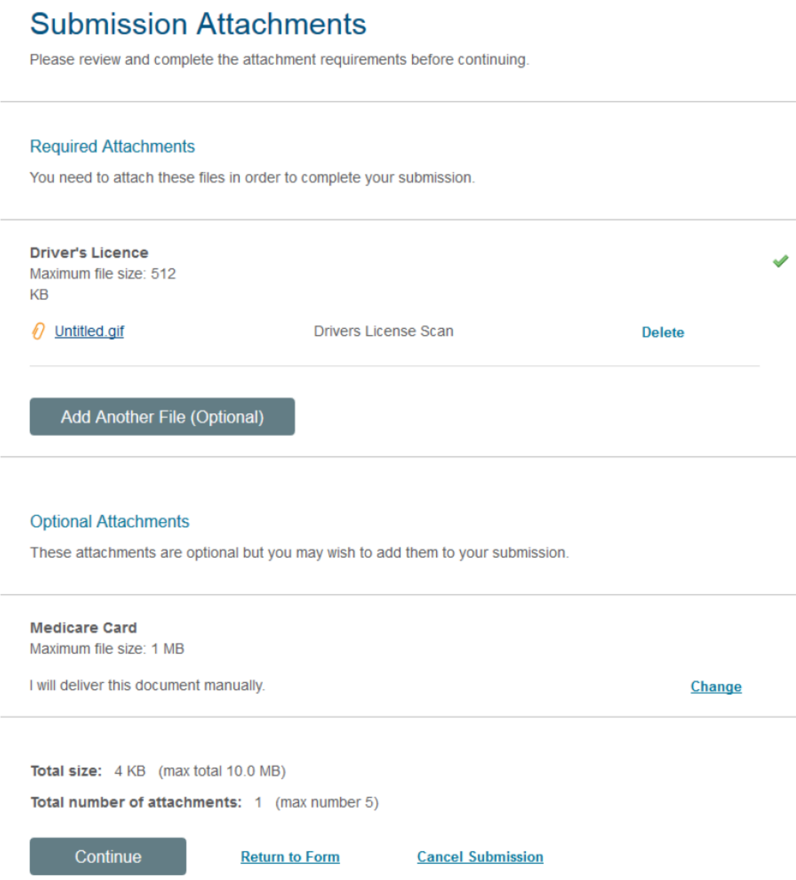
A string containing the new task key identifier.

Faults

- ExpiryDateException: if the expiry date is not in the future
- FormInactiveException: if the form exists but is inactive
- FormNotFoundException: if the form cannot be found
- FormPortalNotSetException: if the form exists but is not exposed on any portals
- GroupNotFoundException: if the group was not found
- InvalidParameterException: if one of the parameters violated a constraint
- PortalNotFoundException: if the portal was not found during LDAP user lookup
- PortalSecurityException: if the portal does not support LDAP users and the user does not exist as a local user
- UserNotFoundException: if the user was not found (including LDAP lookup if applicable)
- TaskUserWebServiceException: if other errors occur

CreateFormReviewTask

This operation creates a new form review task based on the form data and attachments of a previous form submission. The CreateFormReviewTask operation can be used create form review business processes, where the first party submits a form, then a delivery process assigns a form review task to second party (reviewer). The reviewer is then able to check the submitted data, ensure it is correct, make any amendments, check any attachments, and then submit the reviewed form. This reviewed and submitted form would then be delivered to a process to complete the business operation.
With form review tasks any file attachments from the previous submission will be presented to the reviewer upon the submission of the review form. The reviewer can the view the file attachments, delete them or add new attachments.



Screen: Review Attachments From Previous Submission

Form designers often introduce phase variables in the form XML data model to change the forms behavior based on the phase it is in. For example in the review phase, the form may display additional Office Use Only sections where the reviewer can provide additional information.
In comparison to the *CreateFormTask* operation the *CreateFormReviewTask*:

- maintains a relationship to the previous form transaction (submission record)
- you don't have to specify the formDataXml, it will use the data from the previous submission by default. Note some SystemProfile elements are reset at form render time, these include DisplayMode, ReceiptNumber and SubmissionNumber.
- file attachments are copied from the previous form submission on demand to the new form transaction record, so the reviewer can review or modify them in a separate transaction
- you can optionally specify an alternative form template for the reviewer to use, this could include additional sections for the reviewer to complete

Input Parameters

- submitKey: the submission to base the task on (required)
- subject: the task subject (required)
- message: the task message (optional)
- formCode: the form code of the task form (optional; if not specified, the submission's form will be used)
- userName: the assigned user name (optional; must be specified if groupName is not)
- groupName: the assigned form group name (optional; must be specified if userName is not)
- portalName: the user portal name (optional; must be specified if userName denotes an LDAP user)
- formDataXml: the form XML data or schema seed (optional; if not set, the XML data associated with the submission will be used)
- latitude: the task location latitude (optional; must be between -180 and 180 if specified)
- longitude: the task location longitude (optional; must be between -90 and 90 if specified)
- scheduledTime: the scheduled task time (optional)
- expiryTime: the task expiry time (optional; must be in the future if specified)
- userDeletableFlag: whether the user can delete the task
- sendEmailFlag: whether to notify the assignee(s) of the task via email
- emailSubject: the notification email subject line
- emailMessage: the notification email message body. Email message template variable you can use include:
    - ${format} : Apache Click Format object use for formatting values
    - ${formDataMap} : form data extracts map
    - ${submission} : task submission object
    - ${task} : task submission object (as above)
    - ${environmentName} : server evironment name
    - ${supportEmail} : support email address, defined as a deployment property

Output Result

A string containing the new task key identifier.

Faults

- ExpiryDateException: if the expiry date is not in the future

- FormInactiveException: if the form exists but is inactive
- FormNotFoundException: if the form cannot be found
- FormPortalNotSetException: if the form exists but is not exposed on any portals
- GroupNotFoundException: if the group was not found
- InvalidParameterException: if one of the parameters violated a constraint
- PortalNotFoundException: if the portal was not found during LDAP user lookup
- PortalSecurityException: if the portal does not support LDAP users and the user does not exist as a local user
- SubmissionNotFoundException: if the submission was not found
- UserNotFoundException: if the user was not found (including LDAP lookup if applicable)
- TaskUserWebServiceException: if other errors occur

ExpireTask

This operation immediately expires a task.

Input Parameters

- taskKey: the task key (required)

Output Result

- boolean value: true if the task was expired, false if it could not be expired (because it is either already expired or has been completed).

Faults

- InvalidParameterException: if one of the parameters violated a constraint
- TaskNotFoundException: if the task was not found

AddUserToGroup

This operation adds a user to a form group.

Input Parameters

- userName: The login name of the user (required)
- groupName: The name of the form group (required)
- portalName: the user portal name (optional; must be specified if userName denotes an LDAP user)

Output Result

- boolean value: true if the user was added to the group; false if they already were a member.

Faults

- GroupNotFoundException: if the group was not found
- InvalidParameterException: if one of the parameters violated a constraint
- PortalNotFoundException: if the portal was not found during LDAP user lookup
- PortalSecurityException: if the portal does not support LDAP users and the user does not exist as a local user
- UserNotFoundException: if the user was not found in LDAP and does not exist as a local user
- TaskUserWebServiceException: if other errors occur

RemoveUserFromGroup

This operation removes a user from a form group.
Note: This works only for Transaction Manager defined group membership; if the user is an LDAP user, group membership may also be imported from the LDAP directory when the user logs in.

Input Parameters

- userName: the login name of the user (required)
- groupName: the name of the form group (required)

Output Result

- boolean value: true if the user was removed from the group; false if they were not a member.

Faults

- GroupNotFoundException: if the group was not found
- InvalidParameterException: if one of the parameters violated a constraint
- UserNotFoundException: if the user does not exist in Transaction Manager
- TaskUserWebServiceException: if other errors occur

ReassignTask

This operation reassigns a specific task.

Input Parameters

- taskKey: the task key (required)
- userName: the login name of the user to assign the task to (optional; needs to be specified if groupName is not)

- groupName: the name of the forms group to assign the task to (optional; if not specified, the group assignment will be preserved, i.e. NOT nullified)
- portalName: the user portal name (optional; must be specified if userName denotes an LDAP user)

Output Result

- boolean value: true if the task was reassigned, false if the task was completed or expired.

Faults

- GroupNotFoundException: if the group was not found
- InvalidParameterException: if one of the parameters violated a constraint
- TaskNotFoundException: if the task was not found
- PortalNotFoundException: if the portal was not found during LDAP user lookup
- PortalSecurityException: if the portal does not support LDAP users and the user does not exist as a local user
- UserNotFoundException: if the user was not found (including LDAP lookup if applicable)
- TaskUserWebServiceException: if other errors occur

ReassignAllUserTasks

This operation reassigns all tasks assigned to a specific user.

Input Parameters

- existingUserName: the login name of the existing user (required)
- newUserName: the login name of the user to assign the tasks to (optional; needs to be specified if groupName is not)
- newGroupName: the name of the forms group to assign the tasks to (optional; if not specified, the current group assignment will be preserved, i.e. NOT nullified)
- portalName: the user portal name (optional; must be specified if userName denotes an LDAP user)

Output Result

- boolean value: true if any tasks were reassigned; false if no eligible tasks were found (completed or expired tasks are ignored).

Faults

- GroupNotFoundException: if the group was not found
- InvalidParameterException: if one of the parameters violated a constraint
- PortalNotFoundException: if the portal was not found during LDAP user lookup
- PortalSecurityException: if the portal does not support LDAP users and the user does not exist as a local user
- UserNotFoundException: if the user was not found (including LDAP lookup if applicable)
- TaskUserWebServiceException: if other errors occur

Task and User SOAP Web Service (Manager v5.0)

Overview

The Task and User web service enables you to create form tasks, form groups and user accounts in Transaction Manager. By using the Task and User web service, external applications can assign forms to staff to complete and create multi-step workflows. This web service has been replaced by the newer Version 2 service, which provides additional operations and access to the richer Task API.
This Task and User Web Service is provided for backward compatibility.

Application Programmer Interface

The Web Services Description Language (WDSL) for the Task and User web service is available on the server URL:
http://<server name>/manager/fcservices/TaskUserWebService?wsdl

The Task and User web service provides of the following operations:

- FindOrCreateUserAccount
- CreateTaskGroup
- AssignUserTask
- AssignGroupTask
- ExpireTask

FindOrCreateUserAccount

This operation attempts to find a user by login name. If unsuccessful, it creates a user for the specified portal.

Input Parameters

- loginName: the login name identifying the user (required)
- email: the user's email address (required)
- givenName: the user's given name(s)
- familyName: the user's family name (required)
- portalName: the name of the portal which the user should be associated with (required)

Output Result

A string containing the user key (a UUID) of the existing or newly created user.

Faults

- PortalNotFoundException: if the portal with the name specified in portalName cannot be found
- UserInactiveException: if the user exists but is inactive
- TaskUserWebServiceException: if loginName is blank or an unexpected error occurs invoking this method

CreateTaskGroup

This operation will create a user group with a given name. If such a group already exists, no action is taken.

Input Parameters

- groupName: the name of the new group (required)
- portalName: the name of the portal to which the group belongs (currently unused)

Output Result

- boolean value: true if a group was created, or false if it already exists.

Faults

- TaskUserWebServiceException: if groupName is blank or an unexpected error occurs invoking this method

AssignUserTask

This operation will create and assign a task to a single user.

Input Parameters

- userKey: he user key of the assignee (required)
- portalName: the name of the portal which the task will be associated with (required)
- formCode: the form code of the form on which the task is based (required)
- formDataXml: the xml seed data to pass directly into the form (mutually exclusive with formPrefillXml)
- formPrefillXml: the xml prefill data to map into the form's schema seed (mutually exclusive with formDataXml)
- taskMessage: the task message to identify and explain the task to the user (required)
- datetimeExpiry: the date and time the task will expire, if it has not been completed
- userDeletableFlag: whether the assignee will be able to delete the task (required)

- sendEmailFlag: whether an email should be sent to the assignee to notify them of the task (required)
- emailSubject: the subject of the email to the user
- emailMessage: the notification email message body. Email message template variable you can use include:
  - ${format} : Apache Click Format object use for formatting values
  - ${formDataMap} : form data extracts map
  - ${submission} : task submission object
  - ${task} : task submission object (as above)
  - ${environmentName} : server evironment name
  - ${supportEmail} : support email address, defined as a deployment property

Output Result

a string containing the task key (a UUID) of the new task.

Faults

- ExpiryDateException: if the datetimeExpiry parameter contains a date in the past
- FormNotFoundException: if the form specified by formCode does not exist
- FormInactiveException: if the form specified by formCode is inactive
- FormPortalNotSetException: if the form specified by formCode is not associated with a portal
- PortalNotFoundException: if the portal specified by portalName does not exist
- PortalMismatchException: if the form specified by formCode is not associated with the portal specified by portalName
- UserNotFoundException: if the user specified by userKey does not exist
- UserInactiveException: if the user specified by userKey is inactive
- TaskUserWebServiceException: if required input parameters are not set, both *formDataXml* and *formPrefillXml* are set or an unexpected error occurs invoking this method.

AssignGroupTask

This operation will create and assign a task to an existing group of users.

Input Parameters

- groupName: the name of the group of assignees (required)
- portalName: the name of the portal which the task will be associated with (required)
- formCode: the form code of the form on which the task is based (required)
- formDataXml: the xml seed data to pass directly into the form (mutually exclusive with formPrefillXml)
- formPrefillXml: the xml prefill data to map into the form's schema seed (mutually exclusive with formDataXml)
- taskMessage: the task message to identify and explain the task to the users (required)
- datetimeExpiry: the date and time the task will expire, if not completed before then
- userDeletableFlag: whether the assignees will be able to delete the task (required)
- sendEmailFlag: whether an email should be sent to the assignees to notify them of the task (required)
- emailSubject: the subject of the email to the users
- emailMessage: the message body of the email to the users

Output Result

- a string containing the task key (a UUID) of the new task

Faults

- ExpiryDateException: if the datetimeExpiry parameter contains a date in the past
- FormNotFoundException: if the form specified by formCode does not exist
- FormInactiveException: if the form specified by formCode is inactive
- FormPortalNotSetException: if the form specified by formCode is not associated with a portal
- GroupNotFoundException: if the group specified by groupName does not exist
- PortalNotFoundException: if the portal specified by portalName does not exist
- PortalMismatchException: if the form specified by formCode is not associated with the portal specified by portalName
- TaskUserWebServiceException: if required input parameters are not set, both *formDataXml* and *formPrefillXml* are set or an unexpected error occurs invoking this method.

ExpireTask

This operation will expire an existing task. Please note that the task must have a state of "Assigned".

Input Parameters

- taskKey: the valid task key (required)

Output Result

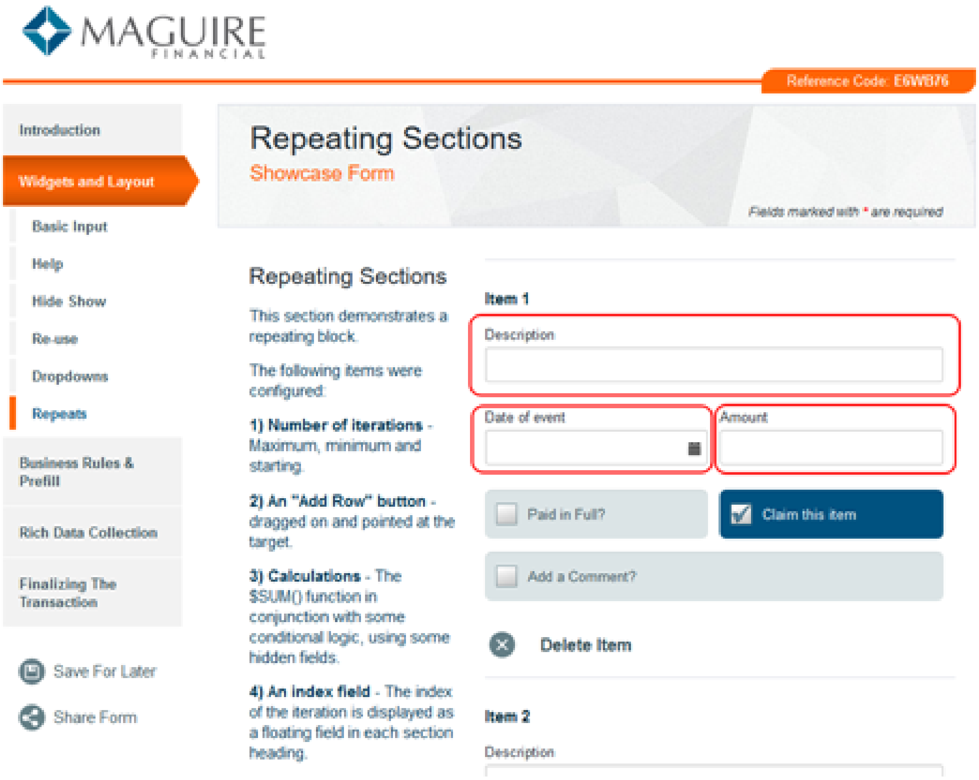
None

Faults

- TaskNotFoundException: if no task with this task key was found
- TaskAlreadyCompletedException: if a task exists but has already been completed
- TaskAlreadyExpiredException: if the task has already been expired
- TaskUserWebServiceException: if an unexpected error occurs invoking this method

Data Extract Example - Repeats (Manager v5.0)

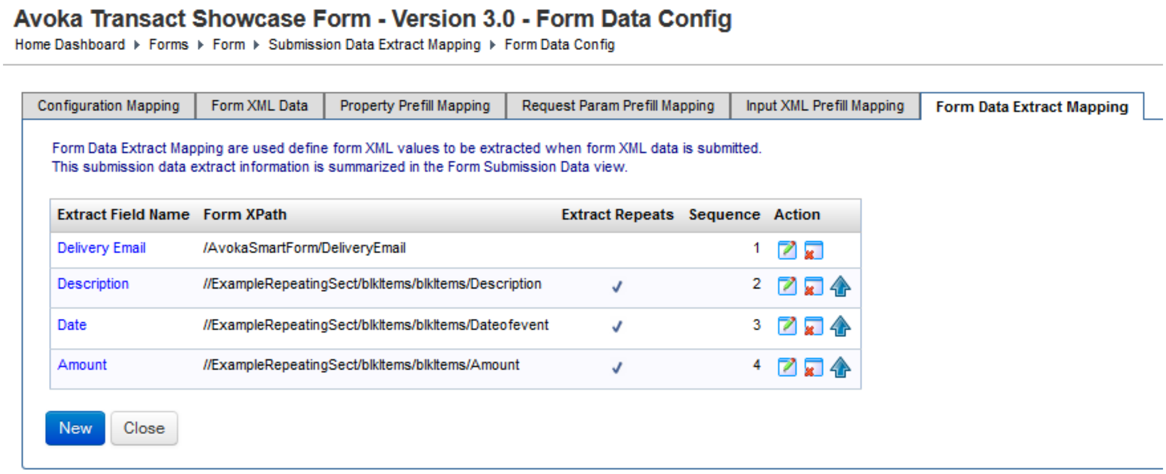
Below is an example of repeating data extracts for the Avoka Transact Show Case form.



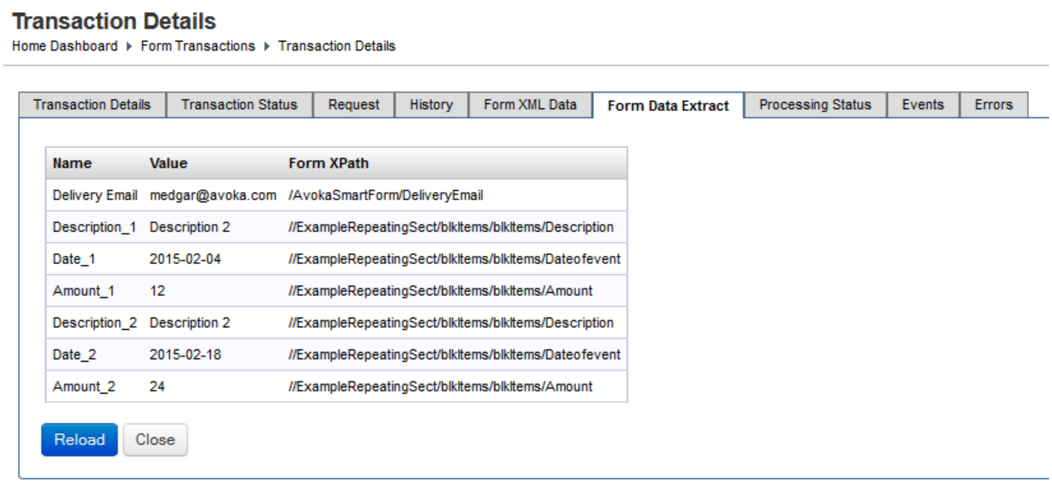
Example form XML data structure:


```
<RepeatingSections>
    <ExampleRepeatingSect>
        <blkItems>
            <blkItems>
                <Description>Description 2</Description>
                <Dateofevent>2015-02-04</Dateofevent>
                <Amount>12</Amount>
                <PaidinFull>true</PaidinFull>
                <Claimthisitem>true</Claimthisitem>
                <AddaComment>false</AddaComment>
                <Comment/>
                <AmountPaid>12</AmountPaid>
                <AmountClaimed>12</AmountClaimed>
            </blkItems>
            <blkItems>
                <Description>Description 2</Description>
                <Dateofevent>2015-02-18</Dateofevent>
                <Amount>24</Amount>
                <PaidinFull>true</PaidinFull>
                <Claimthisitem>true</Claimthisitem>
                <AddaComment>false</AddaComment>
                <Comment/>
                <AmountPaid>24</AmountPaid>
                <AmountClaimed>24</AmountClaimed>
            </blkItems>
        </blkItems>
        <Totals>
            <Totalamountpaid>36</Totalamountpaid>
            <TotalAmountClaimed_1>36</TotalAmountClaimed_1>
        </Totals>
    </ExampleRepeatingSect>
</RepeatingSections>
```

Form Data Configuration for repeating data extracts, please note the Extract Repeating Data option is selected. Please also note that you need to specify the parent XPath, so TM can determine what elements are related. For example, don't do //Description otherwise TM may not be able to associate grouped data.



Example transaction data result:



Security Manager (Manager v5.0)

Contents

Active Directory Federated Services (ADFS) (Manager v5.0)

Avoka Transact supports 2 different authentication and authorisation integration with **Active Directory Federated Services 2.0,** these are Federation Services (SSO) & WS-Trust.

ADFS allow a Cloud Hosted Transact server to authenticate and authorise users for an organisation Active Directory.

The table below explains the main features and differences between the 2 Transact Security Managers that to Federation Services (SSO) & WSTrust.

Type	Desc	SSO	Space Support
Federation Services (SSO)	A user connects to a Transact Web secure page. The security Manager redirects the Users Browser to the ADFS Server: • If the user is running Internet Explorer and the user is logged onto the domain the Users browsers can pass the Windows Authentication to the ADFS server this is the (SSO). • With other browsers the ADFS server will prompt for the users credentials. After validating the user via Windows Authentication: • ADFS encrypts a SAML token using the Public Key to to get the user profile information such as first name, last name, email, groups etc. This will update or create the user record. • automatically redirects the users browser back to the Web Spaces home page. Passing the encrypted SAML Token Transact Web • Decrypts the SAML Token using to get the user profile information such as first name, last name, email, groups etc. This will update or create the user record. The Federation Services (SSO) Sequence Diagram (Manager v5.0) shows the connection between the Users Browser, Transact Web server and the ADFS server. Note: Each Transact Web Space in each environment requires its own Public / Private Key as well as its own Security Manager.	**Yes**	**Web Only**
WS-Trust	The user logs into TranactWeb or TransactField with their AD credential (Username and password). The Transaction Manager - Security Manager then calls the ADFS - WS-Trust Endpoint. The ADFS service checks if the user credentials. If valid, the user details are passed back in the web service response to the Security Manager. The SAML token is decrypted using the security managers Private key to get the user profile information such as first name, last name, email, groups etc. This will update or create the user record. WS-Trust Sequence Diagrams (Manager v5.0) page shows how TransactField and TransactWeb connect to WS-Trust using the sequence diagram. Note: • WS-Trust requires a different Public and Private key for each environment. • The WS-Trust Security Manager can be shared between multiple TransactWeb and TransactField Spaces.	**No**	**T-Field** and **Web**

Installation

Installation involves configuring either TM Security Manager created from the following templates: **Microsoft ADFS Security Manager** (Federation Services SSO) or **Microsoft WS-Trust Security Manager**

The installation process for WS-Trust and Federation Services SSO have a lot of common installation and configuration steps. The table below summarises the process:

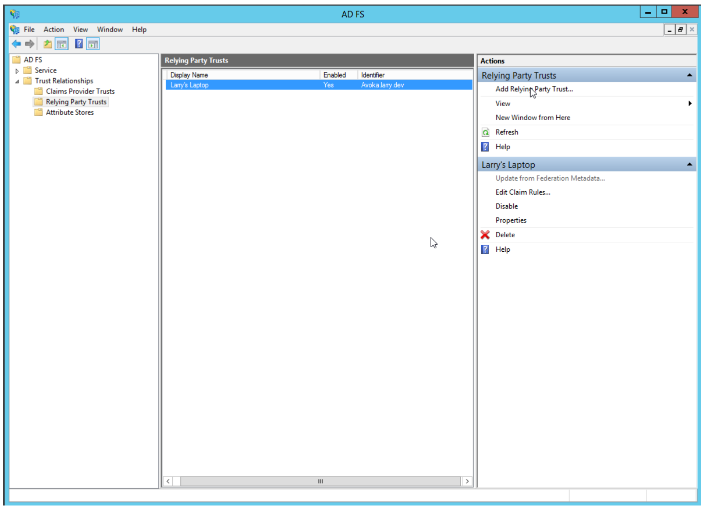
Step	Desc	Who	Federation SSO	WSTrust
Creating Private / Public Keys and Trusting SSL Certificates (Manager v5.0)	This step involved using the keytool to generate and store the steps in a command prompt / terminal Note: • Keys are usually created with the Java that is supplied with Transact. • Java that comes with later versions of Transact 4.1.6 + can produce RSA256 key pairs. This means SHA-256 can be used instead of SHA1 • The key stores can be uploaded to a Transact Manager however its not possible to download them from Transaction Manager • The storage of Private Keys is something that is covered under Avoka's security practise. For Hosted System this is an Avoka IT function	Avoka IT or Client IT	Common	Common
Creating the _SP.xml (Manager v5.0)	The _SP.XML file is provided to the client's (ADFS specialist) for the purpose of creating a Relying Party Trust. This process involves the temporary use of a spring-security-saml2-sample war file. This is deployed to the server that and Apache settings must be updated to allow the context to go through. The IT administrator then connects to the we page to copy field contents to a newly created _SP.XML file. Finally some of the URL's in the _SP.XML file needs to be.	Avoka IT or Client IT	Common	Common
Create the ADFS Relying Party Trust (Manager v5.0)	The _sp.xml file created in the above steps is sent to the client. The process for adding a new Relying Party Trust is detailed here. It involves the following • Wizard is run to setup the Relying Party Trust from the sp_file supplied. • Claim Rules are setup • Algorithm are checked and Endpoints are updated. • WS Trust requires a new identifier to be setup.* Note:	Client IT Admin ADFS Admin	*Common	*Common

	• All new installations should use the SHA-256 algorithm. Old installs may have used SHA-1.			
Microsoft ADFS Security Manager Configuration (Manager v5.0)	This is the Security Manager for the ADFS Federated SSO. It has a 1 to 1 association with a Relying Party Trust. Federated SSO uses browser redirecting from Transact Web Space to ADFS as well as from ADFS back to the web space's context. This Security Manager can only be associated with a single Relying Party Trust and a single Web Space	Avoka Installer Client IT Admin	Yes	No
Microsoft WS-Trust Security Manager (Manager v5.0)	This is the Security Manager for the WS-Trust.	Avoka Installer Client IT Admin	No	Yes

Create the ADFS Relying Party Trust (Manager v5.0)

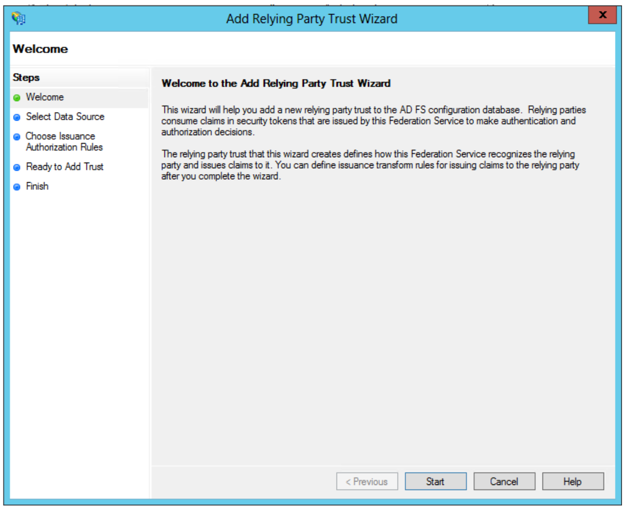
Open AD FS Manager on the Windows Server.

Add Relying Party Trust by Importing the _sp.xml File



The Add Relying Party Trust Wizard comes up

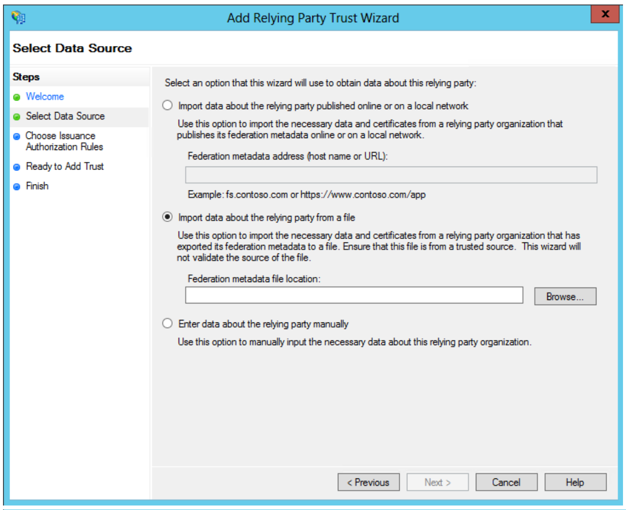
Select Start

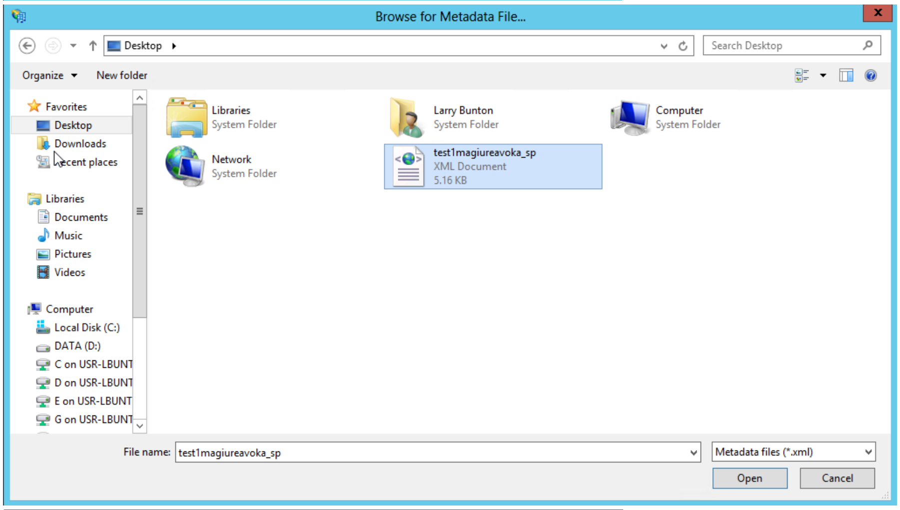


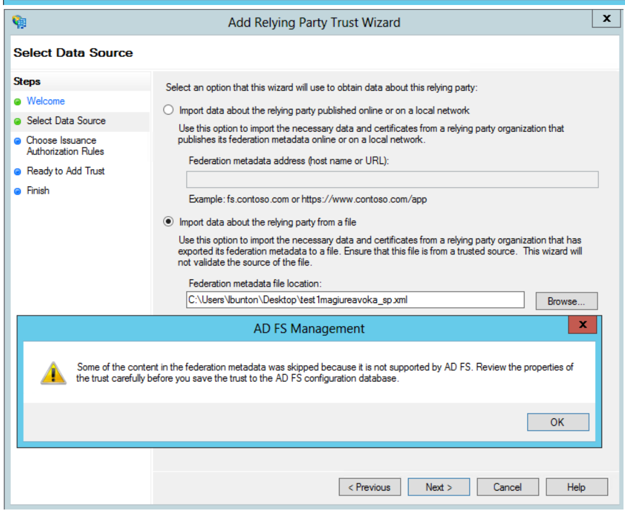
Select "Import data about the relying party from a file".

Browse and select the test1maguireavoka_sp.xml file created above.

Press Next A warning dialog will appear press OK.







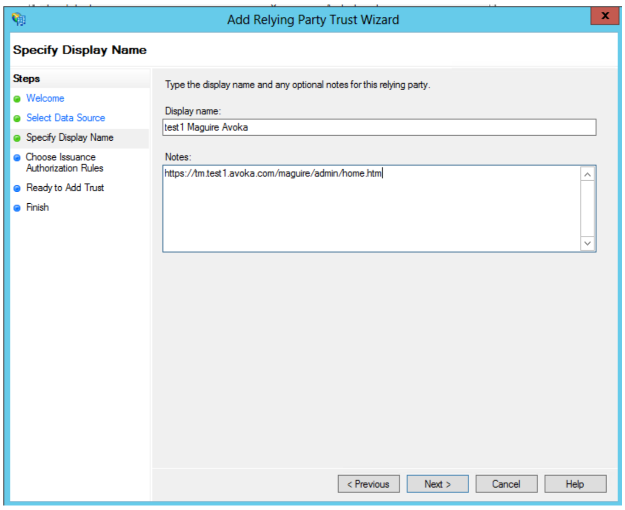
Type in the the Display Name and Notes. Note the diagram differs from the recommended values. These can be updated later.

Display Name: {environment}.{web space context}

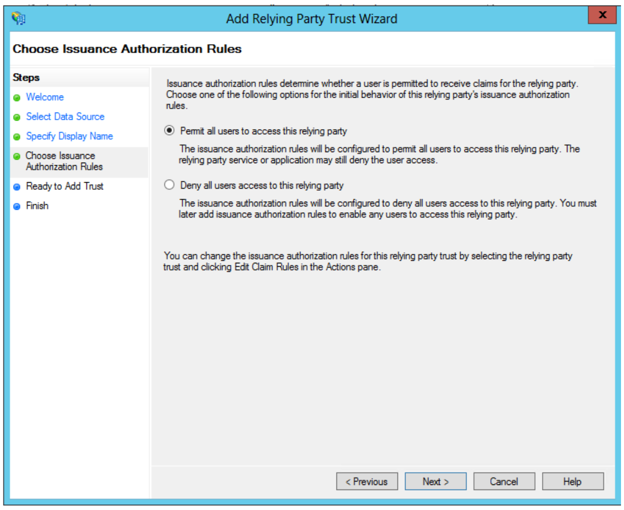
e.g. test1.maguireavoka

Notes: https://{server domain name}/{web space context}/secure/account/home.htm

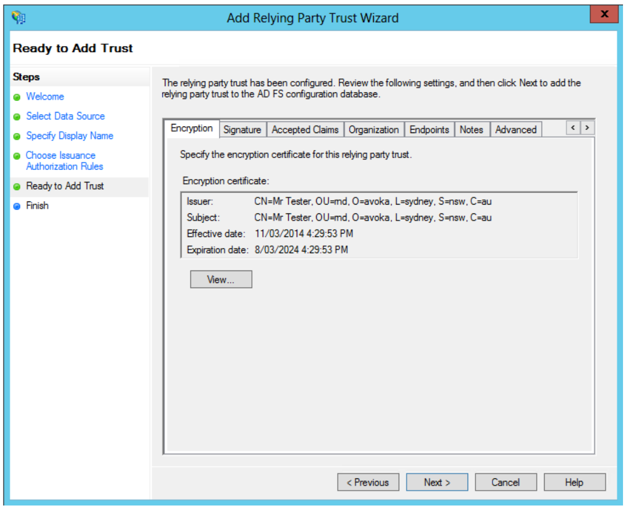
e.g. https://tm.test1.avoka.com/maguireavoka/secure/account/home.htm



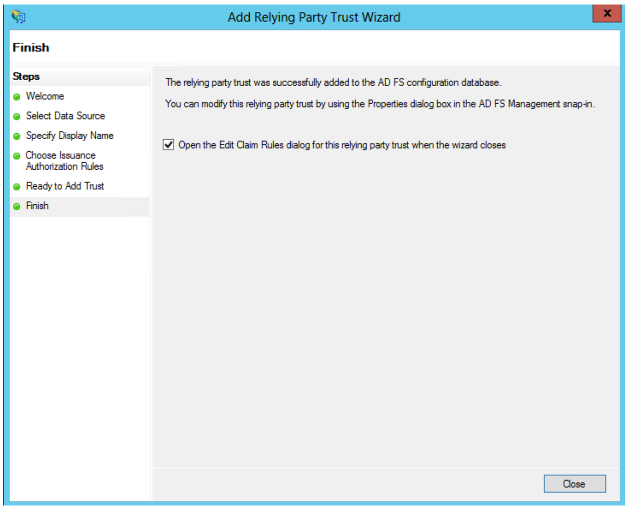
Select Permit all users to access this relying party. Then click Next.



Click Next



select the Open the Edit Claim Dialog check box and click **Close**

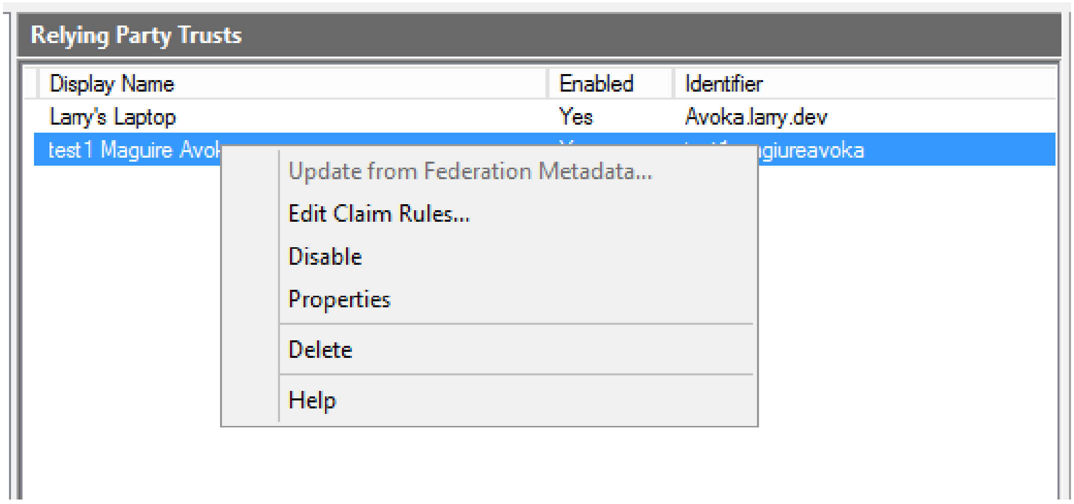


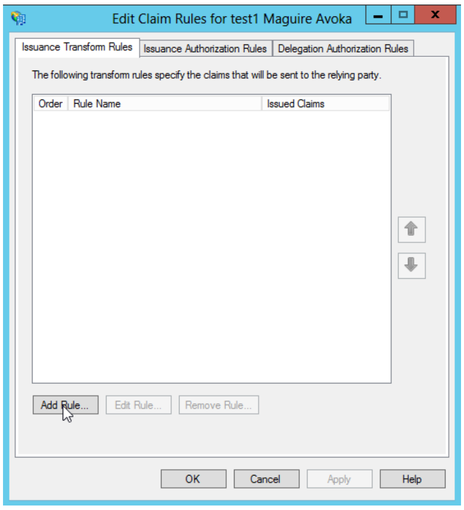
Claim Rules

Right Click on the Relying Party Trust, select **Edit Claim Rules**...

You will get a Dialog **Edit Claim Rules for <Environment>.<webspace>**.

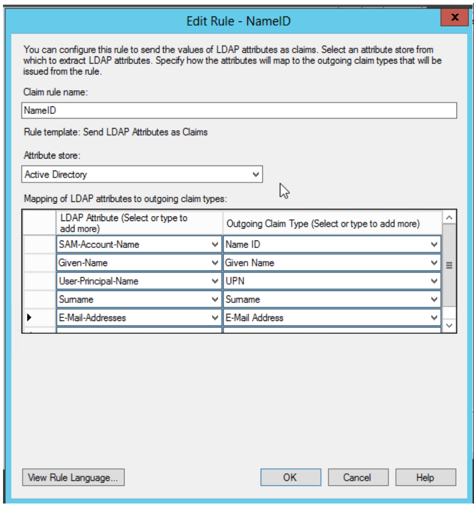
Select **Add Rule.**

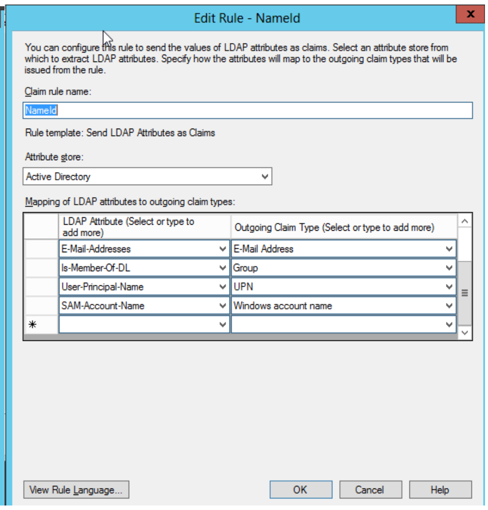




Setup the **Claim Rule**, **Attribute Store** and **Mappings** as follows.

Note: The second screen capture shows the 3 additional fields that don't fit on the first screen





Select Ok then OK

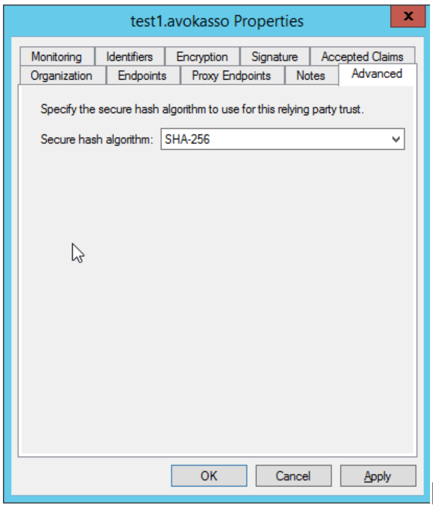
Update Algorithm

From the main screen select the relying trust. Right click and select Properties.

Next Click on the **Advanced tab**.

Select Secure hash algorithm as **SHA-256**

Note: Older systems used SHA-1 algorithm, all new 4.1.6+ clients should use SHA-256



Federated SSO Update Endpoints

From the main screen select the relying trust. Right click and select Properties.

It is possible to reuse a Relying Party Trust for 1 Federated SSO with a portal and WS-Trust. In which can you can perform both this

The following instructions are different for the WS-Trust and Federated SSO

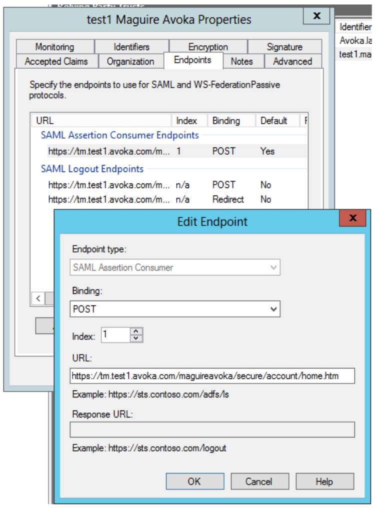
Select the Endpoint tabs

Make the POST SAML Assertion Consumer Endpoint the Default

Double click to get the Edit Endpoint

Update the URL: https://{server environment name}/{}web space name}/secure/account/home.htm

eg. https://tm.test1.avoka.com/maguireavoka/secure/account/home.htm



Adding WS-Trust Identifier

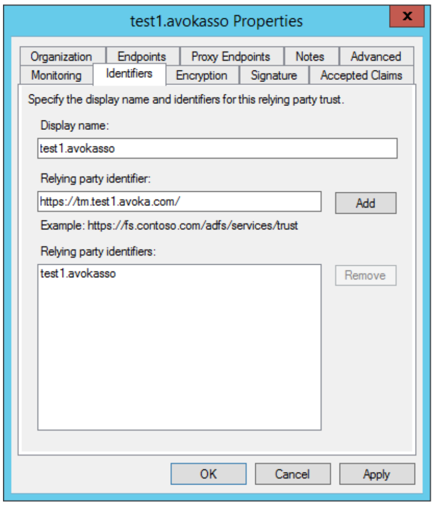
From the main screen select the relying trust. Right click and select Properties.

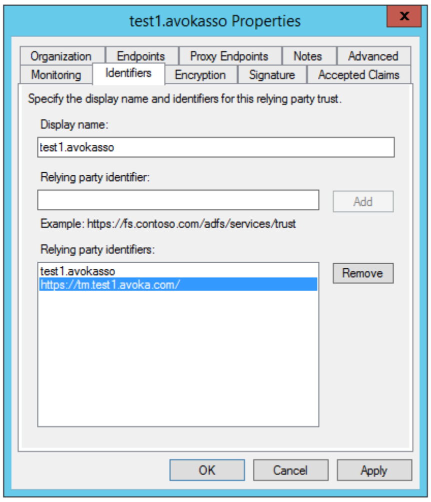
Select the **Identifiers** tabs

Update the Relying party identifier: https://{server environment name}/

 https://tm.test1.avoka.com/

Click Add, then Apply





Creating Private / Public Keys and Trusting SSL Certificates (Manager v5.0)

Overview

Public and private key need are generated and stored in a Java Keystore. The keystore that holds the private and public key will be uploaded to the Transaction Manager - Security Manager. It is recommended that the generation of the private / public key pairs be done on a secure server.

ADFS using Federation SSO, and WS-Trust and will only communicate works over HTTPS / SSL. In addition the Transact Server must trust the SSL certificate of the ADFS Server and visa versa. It is better to use certificates generated by a Trusted Certificate Authority than a self signed certificate, Even so Transact may have to trust the ADFS servers SSL certificate.

Note: The processes of trusting the ADFS certificate writes to the Java cacerts file. The process may need to be repeated if t

Prerequisites

Java

This is installed with Transact the default locations are here

C:\avoka\transact\manager\jdk1.7.0_80 (Windows)

/data/avoka/transact/manager/jdk1.7.0_80 (Linux)

Generate the Private / Public Key Pair into a Keystore (.jks)

The following uses keytool located here"C:\avoka\transact\manager\jdk1.7.0_80\jre\bin\keytool.exe"

It creates:

- A keystore named **samlKeystore.jks**
- The private and public Key with RSA-2048 key size which is valid for 10 years.
- **devmaguire** is the matching keystore **alias** for the public / private key. This alias can be any name, I have used the following naming convention <environment><clientname>
- The alias password is also set to **mysecret**. This should be replace with a more secure password.
- The **keystore** password is set to the same as the **alias** password

Keytool

```
 #> keytool -genkeypair -alias devmaguire -keypass mysecret -keyalg RSA -keysize 2048 -validity 3650 -keystore
samlKeystore.jks
Enter keystore password:
Re-enter new password:
What is your first and last name?
 [Unknown]: maguire
What is the name of your organizational unit?
 [Unknown]: avoka-it
What is the name of your organization?
 [Unknown]: avoka technologies
What is the name of your City or Locality?
 [Unknown]: manly
What is the name of your State or Province?
 [Unknown]: nsw
What is the two-letter country code for this unit?
 [Unknown]: au
Is CN=maguire, OU=avoka-it, O=avoka technologies, L=manly, ST=nsw, C=au correct?
 [no]: yes
#>
```

Trusting the SSL certificates

<Yo can you please complete this>

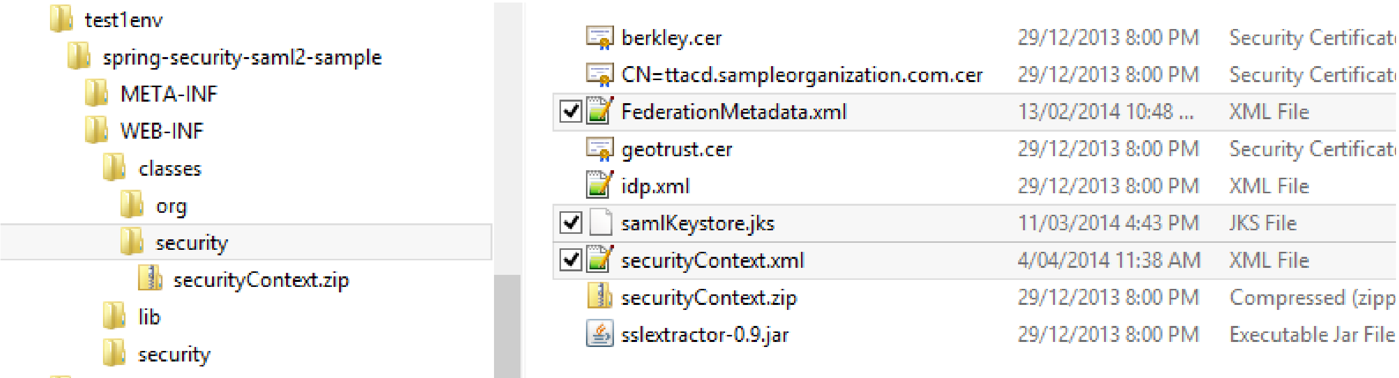
Creating the _SP.xml (Manager v5.0)

Extract a spring-security-saml2-sample.war file

Using 7Zip extract the contents of the base spring-security-saml2-master into the C:\avoka_adfs\spring-security-saml2-sample\

note C:\avoka_adfs is a temporary directory used for editing and updating files.

We will be updating the files highlighted in this directory.



Connect to the ADFS server to get FederationMetadata.xml

The following is required if you are connecting to a new ADFS server. This step has already been completed for Manly-2013

https://<yourservername>/FederationMetadata/2007-06/FederationMetadata.xml

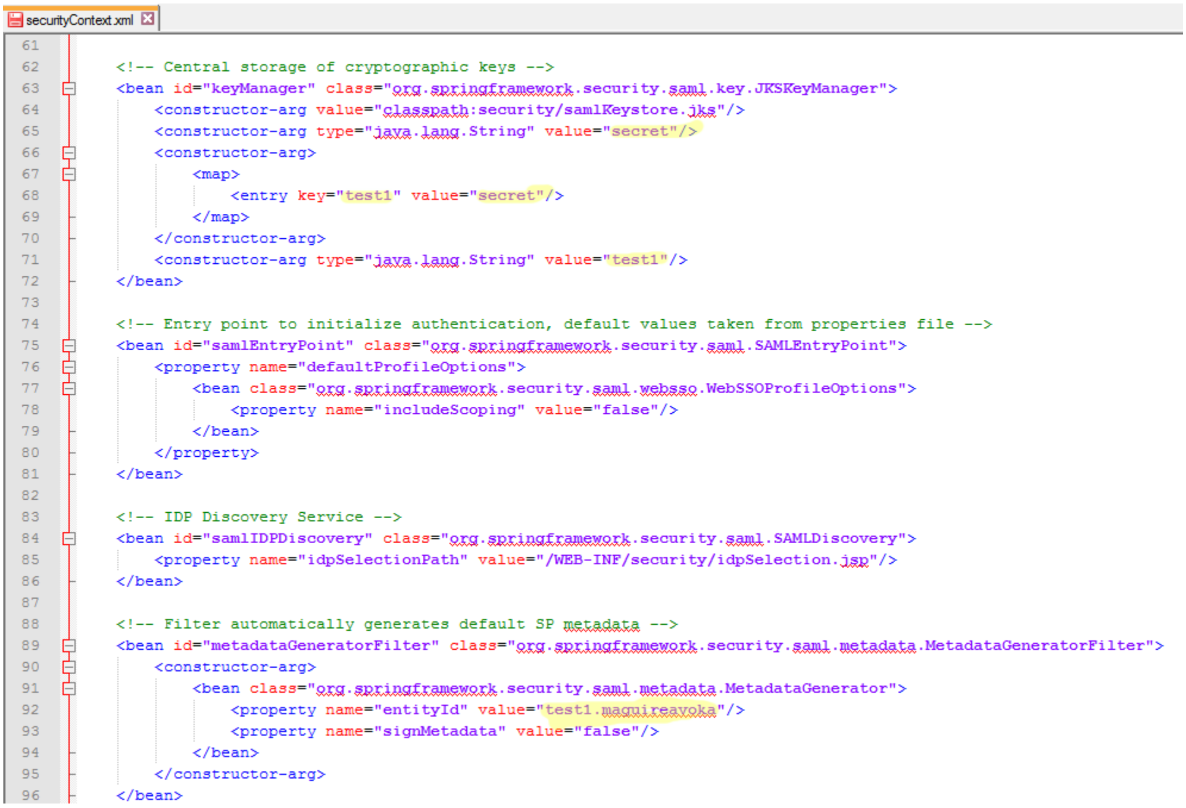
save to file

C:\avoka\cert\{environment}\spring-security-saml2-sample\WEB-INF\classes\security\FederationMetadata.xml

Edit securityContext update your keystore alias and password.

C:\avoka\cert\{environment}\spring-security-saml2-sample\WEB-INF\classes\security\securityContext.xml

For the above credentials.




```xml

<!-- Central storage of cryptographic keys -->
<bean id="keyManager" class="org.springframework.security.saml.key.JKSKeyManager">
    <constructor-arg value="classpath:security/samlKeystore.jks"/>
    <constructor-arg type="java.lang.String" value="secret"/>
    <constructor-arg>
        <map>
            <entry key="test1" value="secret"/>
        </map>
    </constructor-arg>
    <constructor-arg type="java.lang.String" value="test1"/>
</bean>

<!-- Entry point to initialize authentication, default values taken from properties file -->
<bean id="samlEntryPoint" class="org.springframework.security.saml.SAMLEntryPoint">
    <property name="defaultProfileOptions">
        <bean class="org.springframework.security.saml.websso.WebSSOProfileOptions">
            <property name="includeScoping" value="false"/>
        </bean>
    </property>
</bean>

<!-- IDP Discovery Service -->
<bean id="samlIDPDiscovery" class="org.springframework.security.saml.SAMLDiscovery">
    <property name="idpSelectionPath" value="/WEB-INF/security/idpSelection.jsp"/>
</bean>

<!-- Filter automatically generates default SP metadata -->
<bean id="metadataGeneratorFilter" class="org.springframework.security.saml.metadata.MetadataGeneratorFilter">
    <constructor-arg>
        <bean class="org.springframework.security.saml.metadata.MetadataGenerator">
            <property name="entityId" value="test1.maguireavoka"/>
            <property name="signMetadata" value="false"/>
        </bean>
    </constructor-arg>
</bean>
```

Create a spring-security-saml2-sample.war file

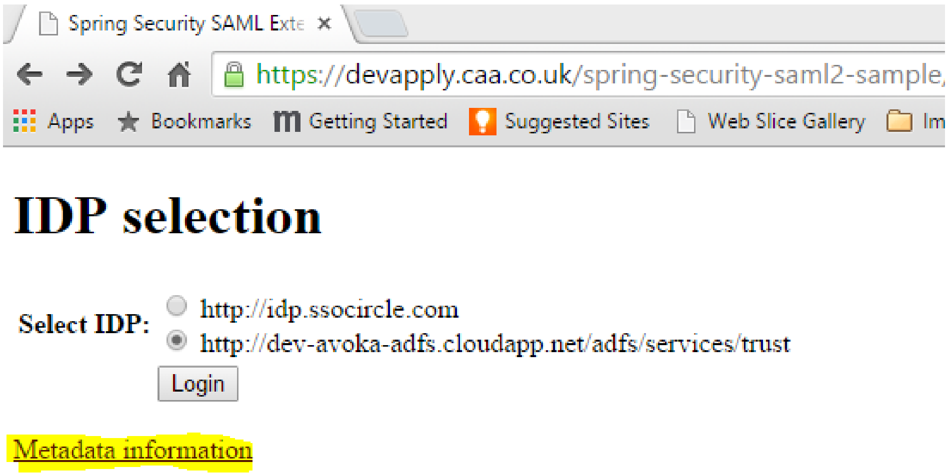
Using 7Zip the contents of the C:\avoka\cert\{environment}\spring-security-saml2-sample\ directory into spring-security-saml2-sample.war

Deploy spring-security-saml2-sample.war file to The TM Server

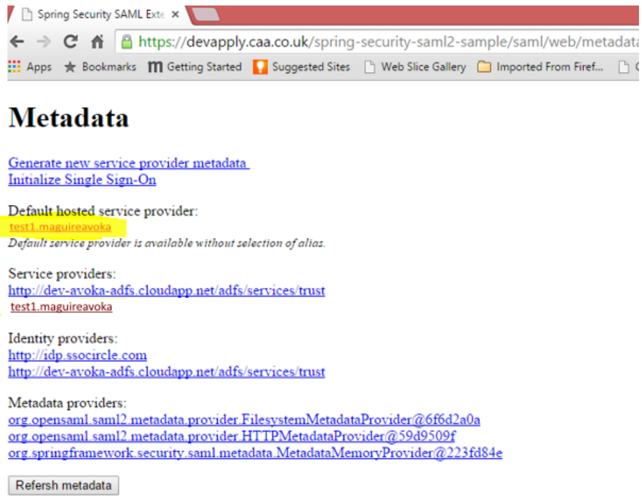
1. Update Apahce proxy setting for the war file context spring-security-saml2-sample.

2. https://<servername>/spring-security-saml2-sample/

- check the server.log for errors
- You will get to a IDP selection page.



- click on the "Metadata information" Link



- Under the "Default hosted service provider" click on the
- {environment}.{portal} link eg test1.maguireavoka

Follow the instructions to store the metadata in file **{environment}{portal}_sp.xml** e.g. test1maguireavoka_sp.xml

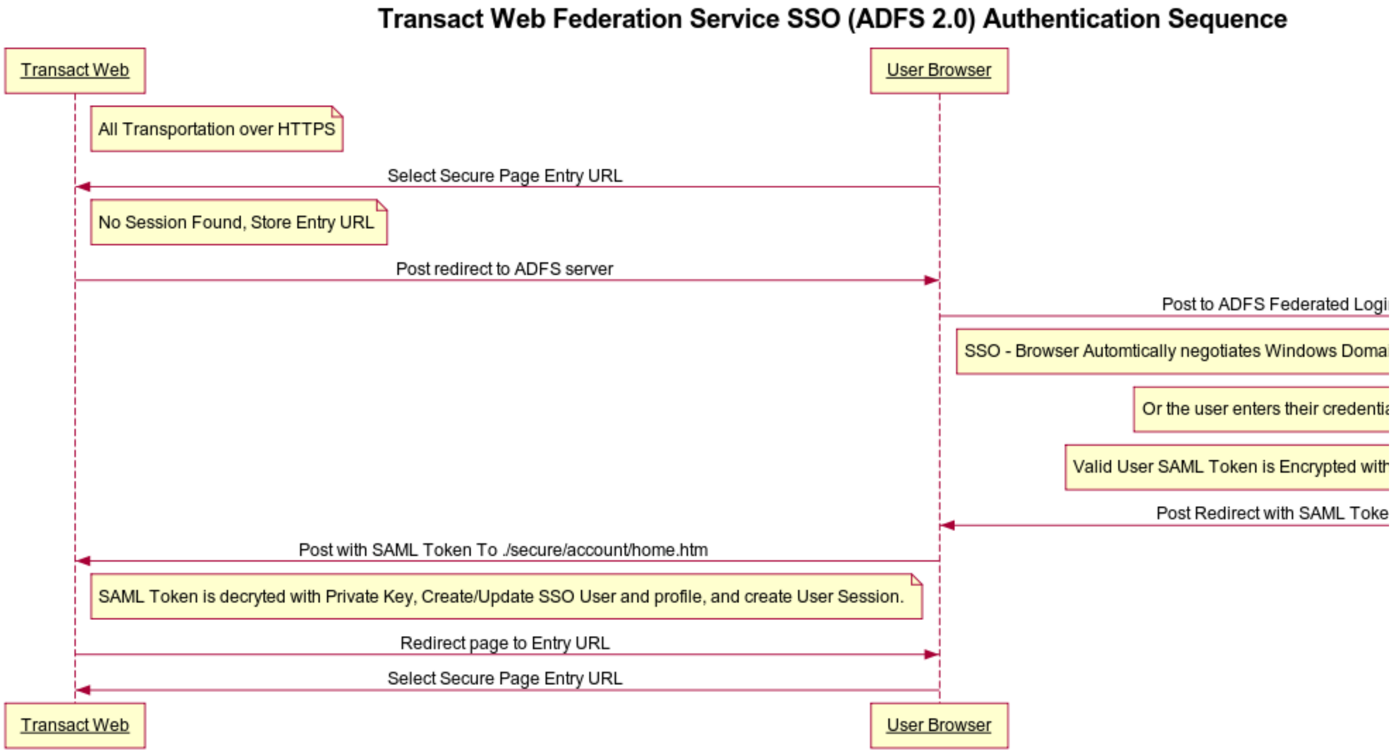
This metadata includes the exported public keys as well as other settings used by ADFS. It is used by the next section to which details creating the Relying Party Trust in ADFS.

Remove the spring-security-saml2-sample.war

Delete the file from standalone\deployments

Remove Apache settings for the spring-security-saml2-sample

Federation Services (SSO) Sequence Diagram (Manager v5.0)

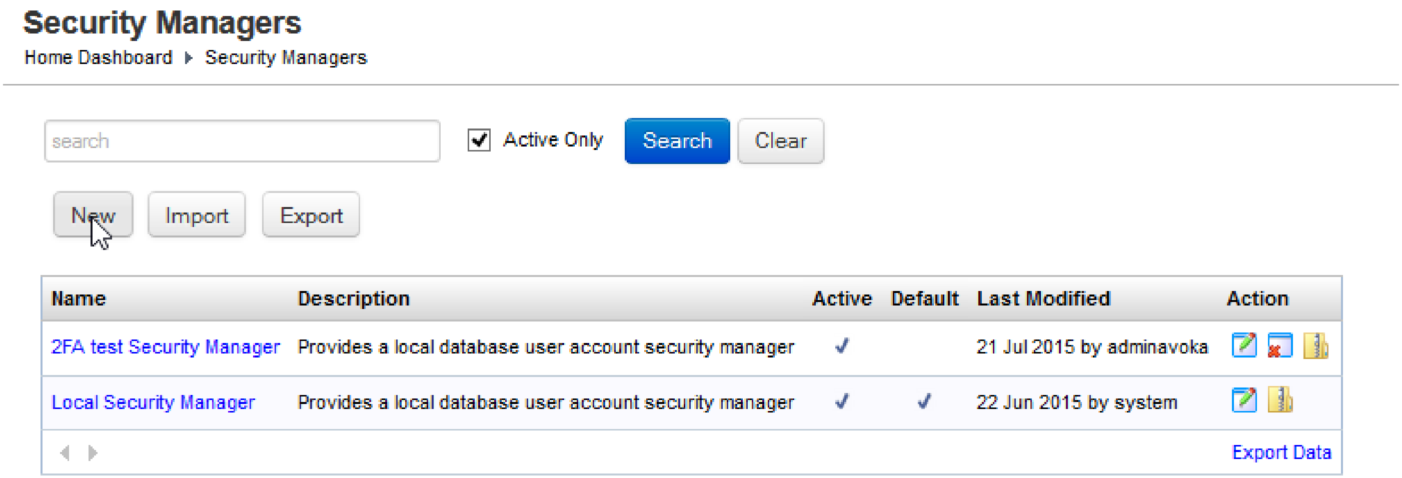


Microsoft ADFS Security Manager Configuration (Manager v5.0)

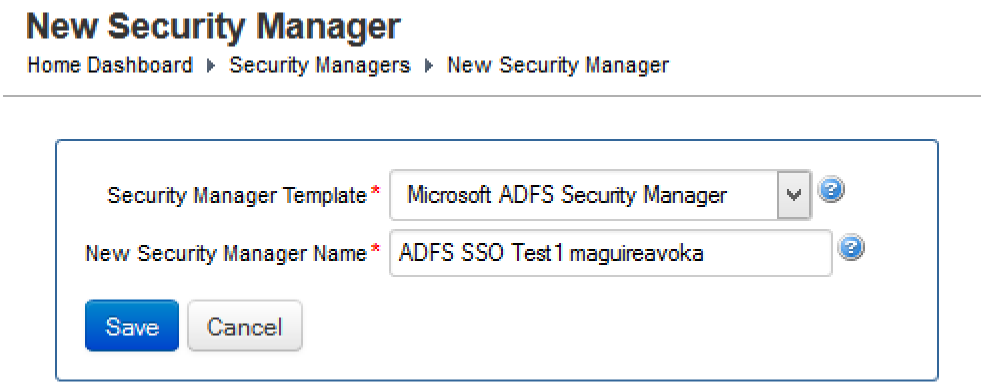
Create a new Security Manager

Select Security Managers from the Security Menu

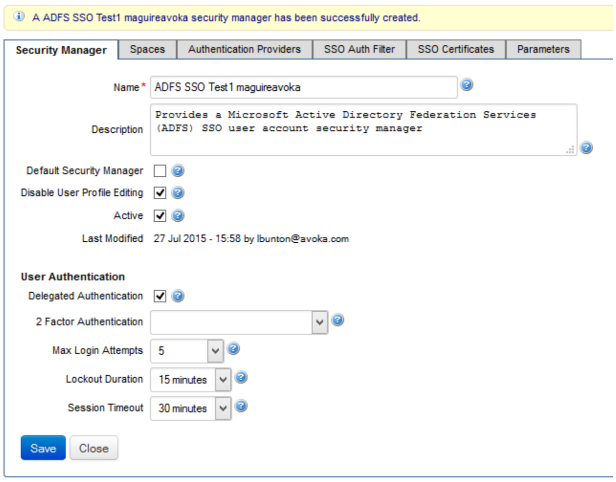
Click New



Enter New Security Manager Name

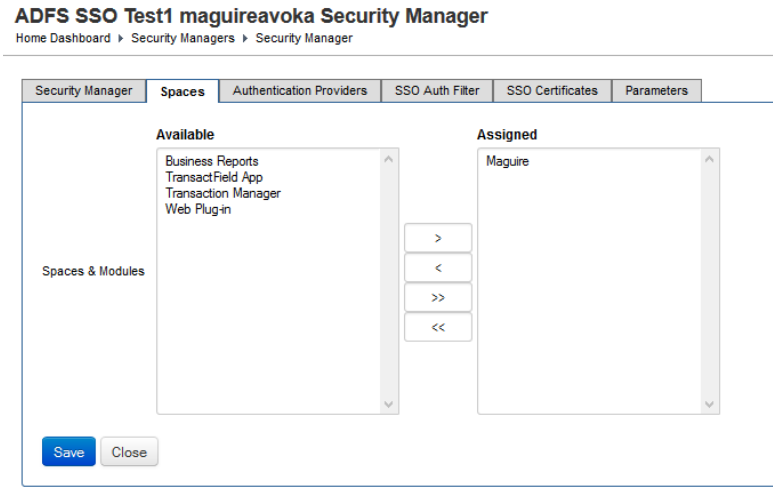


Security Manager Tab



Spaces Tab

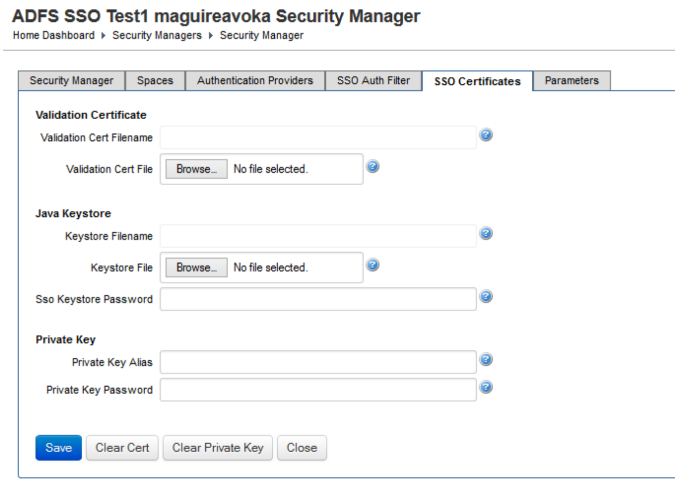
There should only be 1 Web Space selected for the 1 ADFS Security Manager.



SSO Certificates Tab

Use the keystore, password, alias and alias password used in the Create the Private / Public Key step

- **Keystore File**: Upload the samlKeystore.jks
- **Sso Keystore Password**
- **Private Key Alias**
- **Private Key Password**



SSO Auth Filter Tab

Login URL: https://[SERVER_ADDRESS]/adfs/ls/IdpInitiatedSignon.aspx?logintoRP=[RELYING_PARTY_IDENTIFIER]

https://adfs.maguire.com/adfs/ls/IdpInitiatedSignon.aspx?logintoRP=test1.avokasso

Logout Chain URL: ./index.htm

This will redirect the user to the

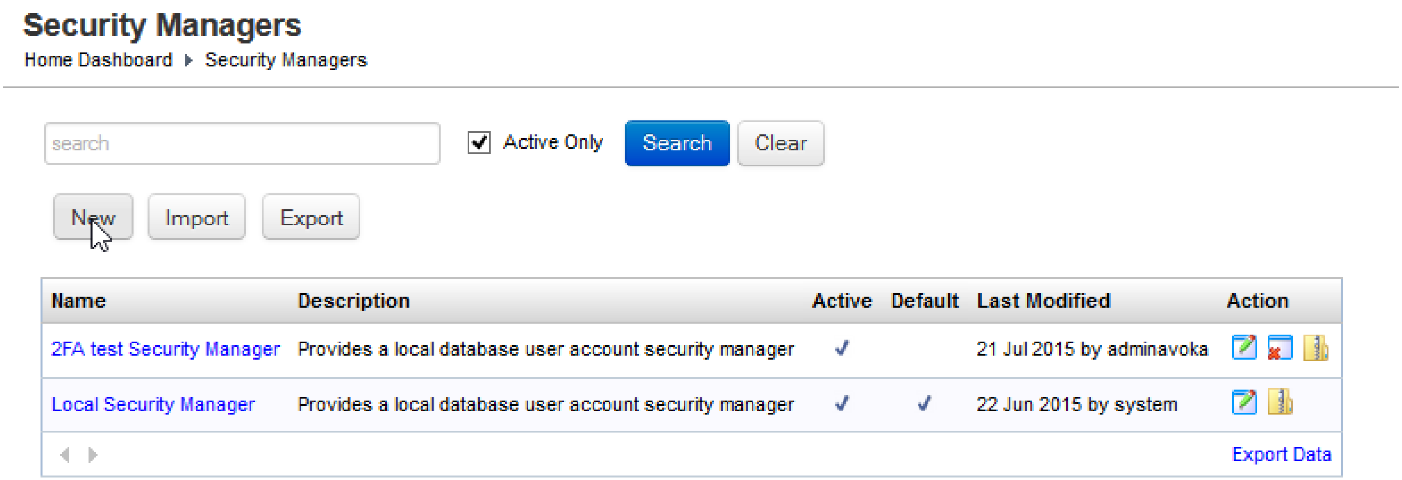

ADFS SSO Test1 maguireavoka Security Manager
Home Dashboard
Security Managers
Security Manager
Security Manager
Spaces
Authentication Providers
SSO Auth Filter
SSO Certificates
Parameters
Enable SSO Filter
Get SSO Auth Token Script
/* Provides a Groovy script to get an SSOAuthenticationToken from a request.
The returned SSOAuthenticationToken will then be processed by the configured AuthenticationProvider(s).
Script parameters include:
request : javax.servlet.http.HttpServletRequest
portal : com.avoka.fc.core.entity.Portal
securityManager : com.avoka.fc.core.entity.SecurityManager
Script return:
SSO auth token, or if null then other Authentication filters will be executed : com.avoka.fc.core.security.SSOAuthenticationToken
Script throws:
redirect exception to redirect to an external login page : com.avoka.fc.core.servlet.RedirectException
*/
import com.avoka.fc.core.util.RedirectUtils
import com.avoka.fc.core.entity.SecurityManager
import com.avoka.fc.core.security.SAML2AttributesParser
import com.avoka.fc.core.security.SSOAuthenticationToken
SAML2AttributesParser saml2AttributesParser = new SAML2AttributesParser(request);
// Stores the Entry URL into the session which is used by the Auth Ok Response Script.
if (!saml2AttributesParser.hasSamlToken()) {
RedirectUtils.storeSessionEntryUrl(request)
return null
}
// Sets the validator cert, the keystore, using supplied alias and password from SSO Certificate Page.
saml2AttributesParser.setFromSecurityManager(securityManager)
// Set the group name attribute, only set if groups are to be imported by ADFS.
saml2AttributesParser.setGroupAttributeName("http://schemas.xmlsoap.org/claims/Group")
// If successful returns the SSOAuthenticationToken or otherwise null.
return saml2AttributesParser.getSSOAuthenticationToken()
Auth OK Response Script
/** Provides a Groovy script successful authentication event handler which can optionally
write to the Servlet response and return false to signal that the request processing has been completed.
Script parameters include:
request : javax.servlet.http.HttpServletRequest
response : javax.servlet.http.HttpServletResponse
portal : com.avoka.fc.core.entity.Portal
securityManager : com.avoka.fc.core.entity.SecurityManager
Script return:
true to continue processing, or false to signal request processing has completed : boolean
Script throws:
redirect exception to redirect to an external page : com.avoka.fc.core.servlet.RedirectException
*/
import com.avoka.fc.core.util.RedirectUtils
import com.avoka.fc.core.servlet.RedirectException
import org.apache.commons.lang.StringUtils
// Then on redirect to ADFS for login, after the token validation, The Auth OK Response Script is used to redirect to the initially saved Entry URL.
def redirectURL = RedirectUtils.getSessionEntryUrl(request)
if (StringUtils.isNotEmpty(redirectURL.toString())) {
throw new RedirectException(redirectURL.toString())
}
return true
Groovy Services Guide
Avoka TM Javadoc API
Login URL
https://[SERVER_ADDRESS]/adfs/ls/IdpInitiatedSignon.aspx?logintoRP=[RELYING_PARTY_IDENTIFIER]
Logout Chain URL
Save
Close

Microsoft WS-Trust Security Manager (Manager v5.0)

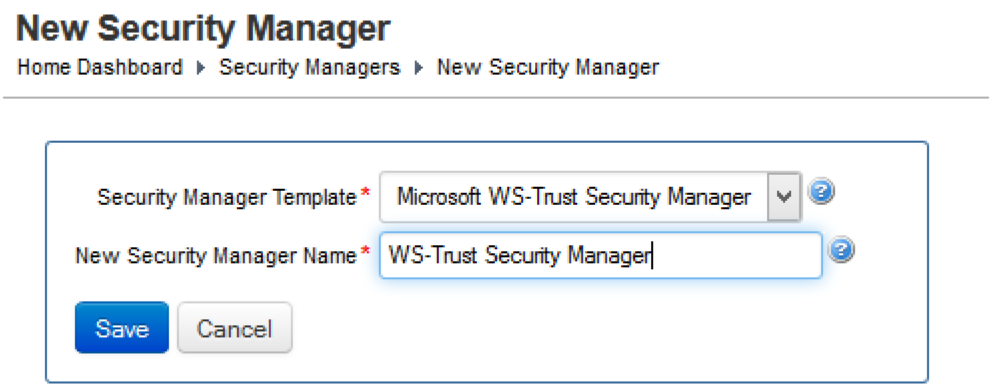
Create a new Security Manager

Select Security Managers from the Security Menu

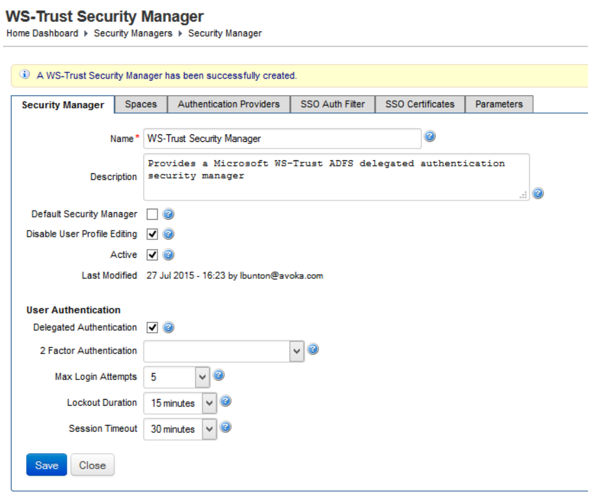
Click New



Enter New Security Manager Name

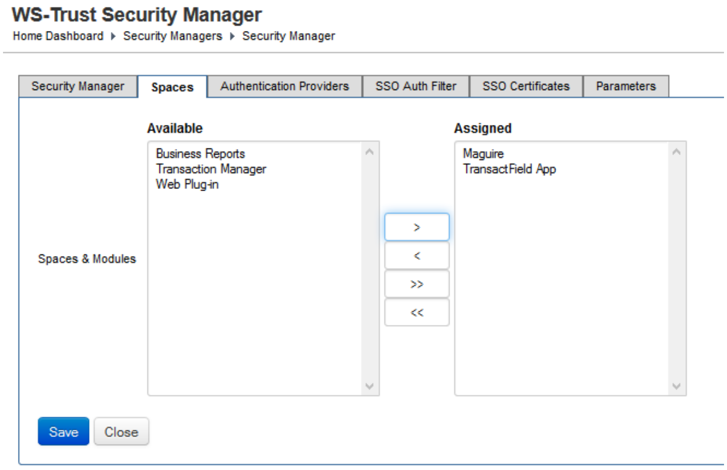


Security Manager Tab



Spaces Tab

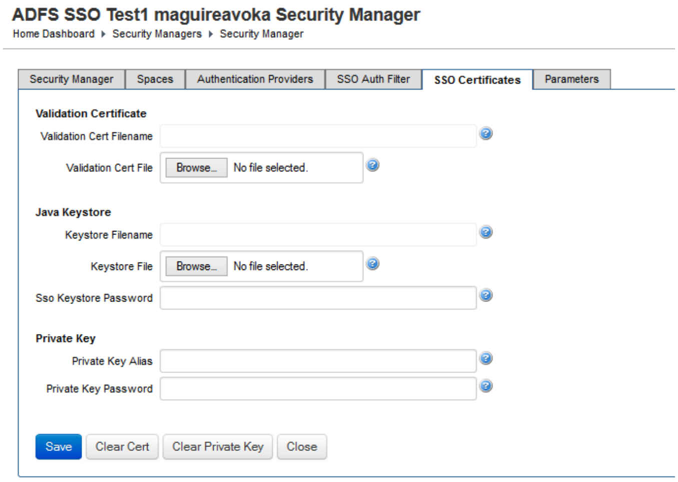
WS-Trust can be assigned to as many TransactWeb and TransactField spaces.



SSO Certificates Tab

Use the keystore, password, alias and alias password used in the Create the Private / Public Key step

- **Keystore File**: Upload the samlKeystore.jks
- **Sso Keystore Password**
- **Private Key Alias**
- **Private Key Password**



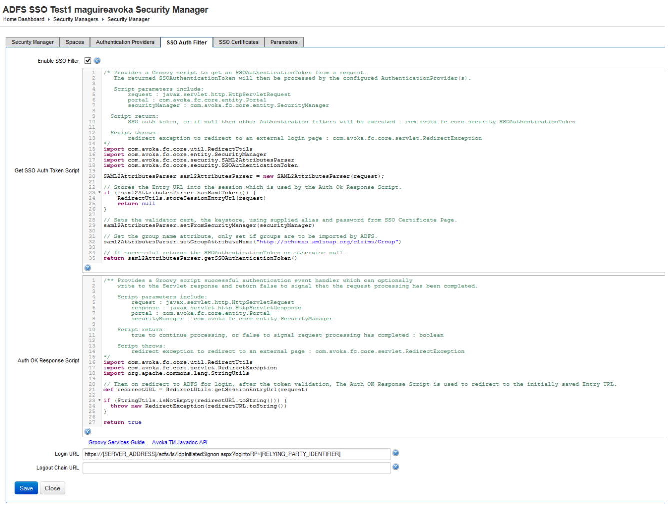
SSO Auth Filter Tab

Login URL: https://[SERVER_ADDRESS]/adfs/ls/IdpInitiatedSignon.aspx?logintoRP=[RELYING_PARTY_IDENTIFIER]

https://adfs.maguire.com/adfs/ls/IdpInitiatedSignon.aspx?logintoRP=test1.avokasso

Logout Chain URL: ./index.htm

This will redirect the user to the



Parameters Tab

WS-Addressing Endpoint Reference:

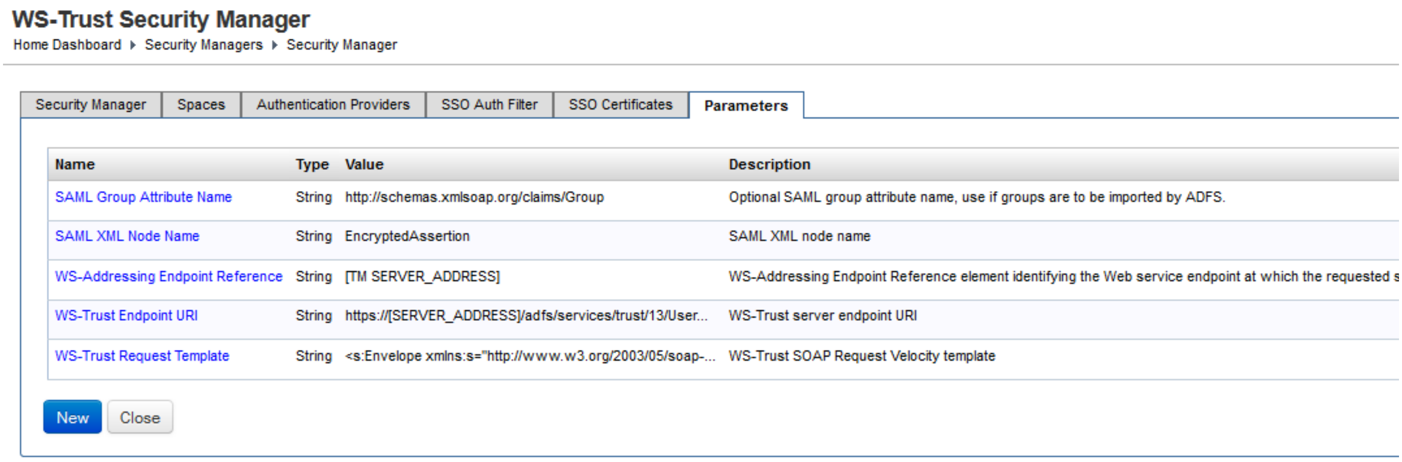
[TM SERVER_ADDRESS]

e.g. https://tm.test1.avoka.com/

WS-Trust Endpoint URI:

https://[SERVER_ADDRESS]/adfs/services/trust/13/UsernameMixed

e.g. https://adfs.maguire.com/adfs/services/trust/13/UsernameMixed

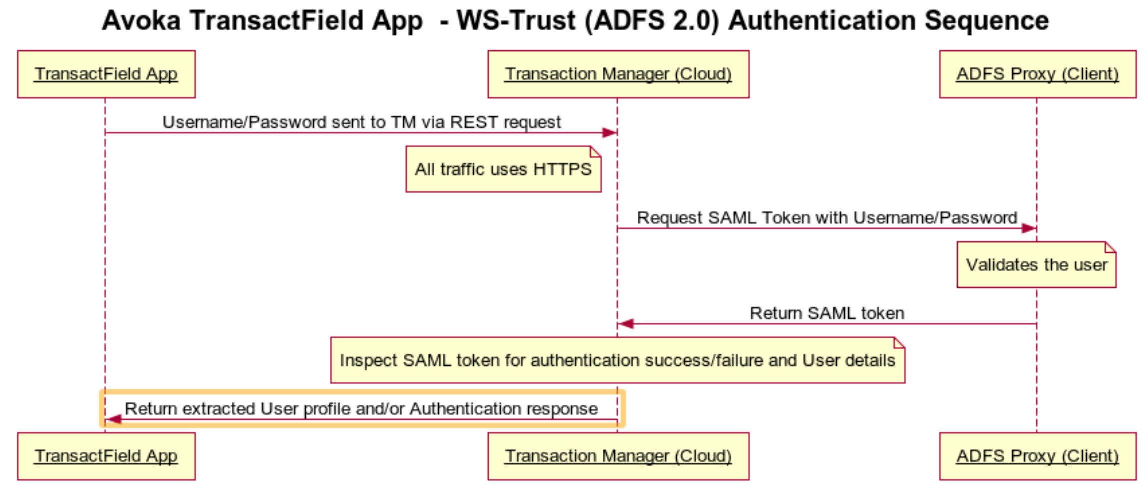


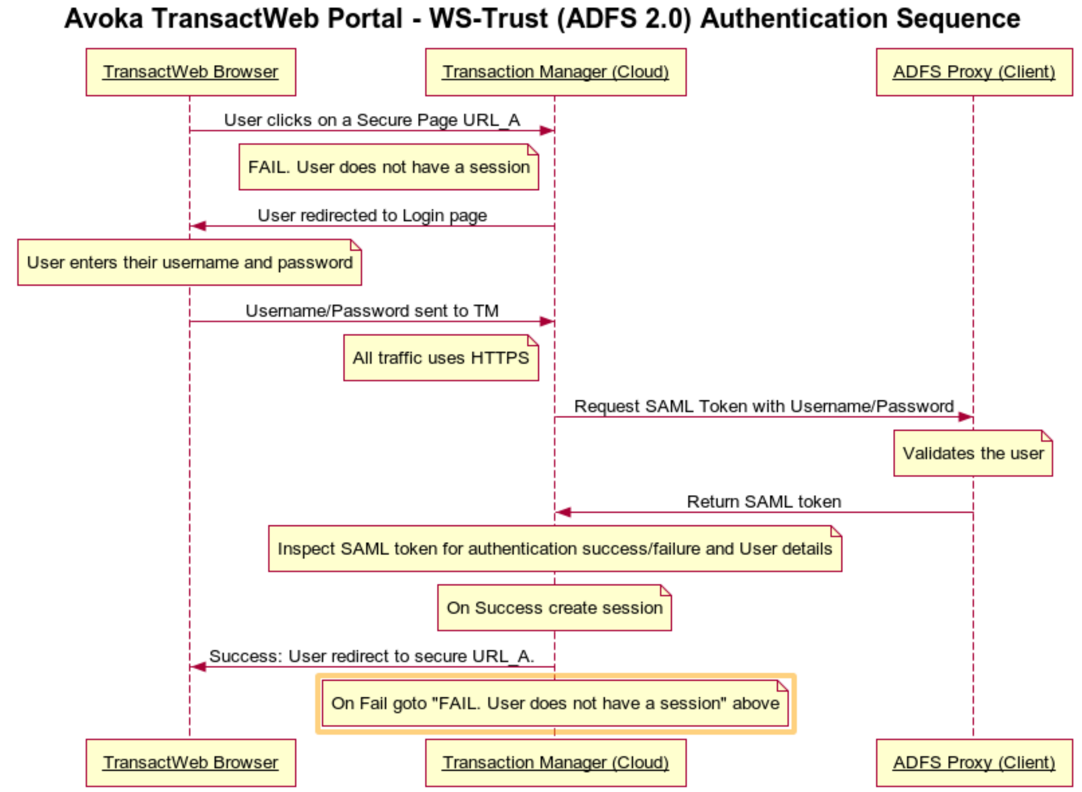
Name	Type	Value	Description
SAML Group Attribute Name	String	http://schemas.xmlsoap.org/claims/Group	Optional SAML group attribute name, use if groups are to be imported by ADFS.
SAML XML Node Name	String	EncryptedAssertion	SAML XML node name
WS-Addressing Endpoint Reference	String	[TM SERVER_ADDRESS]	WS-Addressing Endpoint Reference element identifying the Web service endpoint at which the requested s
WS-Trust Endpoint URI	String	https://[SERVER_ADDRESS]/adfs/services/trust/13/User...	WS-Trust server endpoint URI
WS-Trust Request Template	String	<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-...	WS-Trust SOAP Request Velocity template

WS-Trust Sequence Diagrams (Manager v5.0)

Below it the WS-Trust Flow for TransactField App and TransactWeb.

Note Both use the Same Security Manager which is the Transaction Manager component that communicates with the ADFS WS-Trust Endpoint to authenticate the user.





2 Factor Authentication (Manager v5.0)

Transaction Manager has supported 2 factor authentication since 4.X

This document details how to setup this for the Transaction Manager module.

Access Control Options (Manager v5.0) contains details on how to setup the service.

Work in Progress.

Form A/B Testing (Manager v5.0)

Introduction to Transact A/B Testing

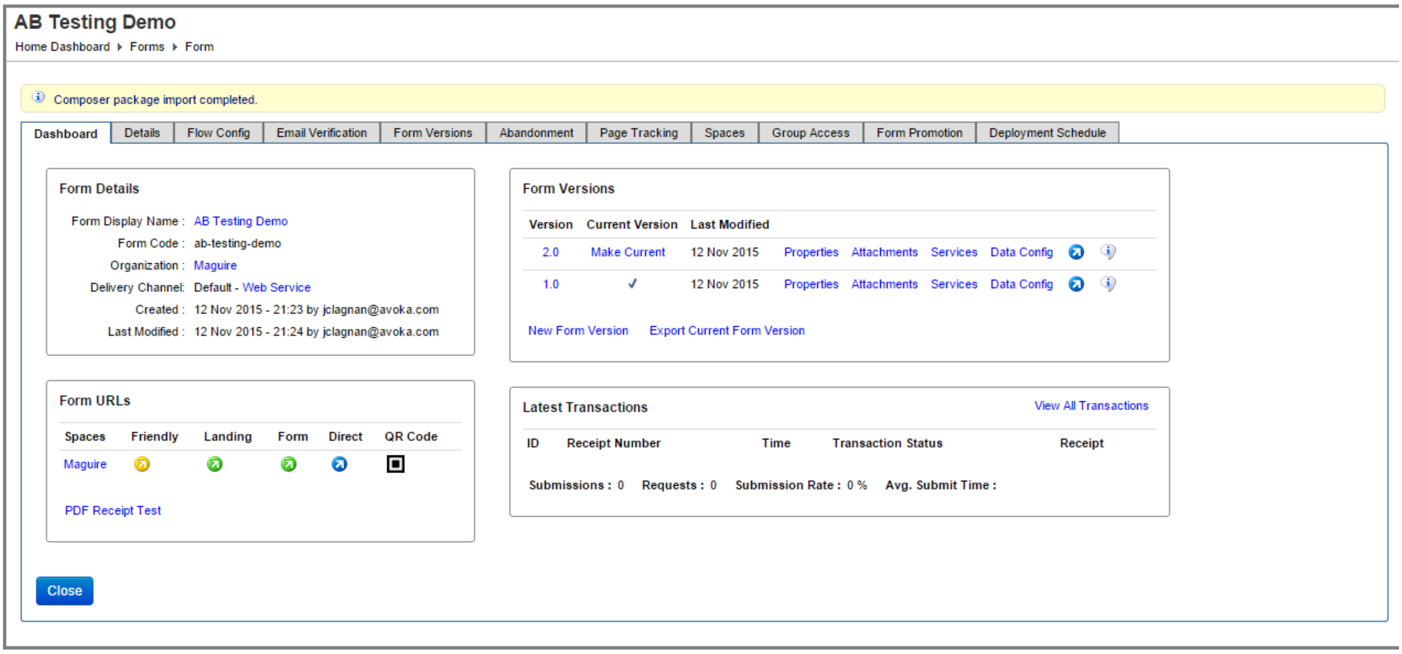
Avoka Transact supports dynamic form version selection. The key use of this new capability is to enable A/B testing of applications for transaction conversion rate optimization. Business teams can now incrementally change application forms, making individual changes and test the results quickly. By testing multiple versions in parallel, you can isolate the changes you have made from other variables that would be introduced if comparing different versions running at different times. By rapidly iterating and testing new changes, organizations can quickly and dramatically improve transaction conversion rates.

A/B Testing analytics in Transact supports time-series analysis of multiple form versions, with filtering by device type.

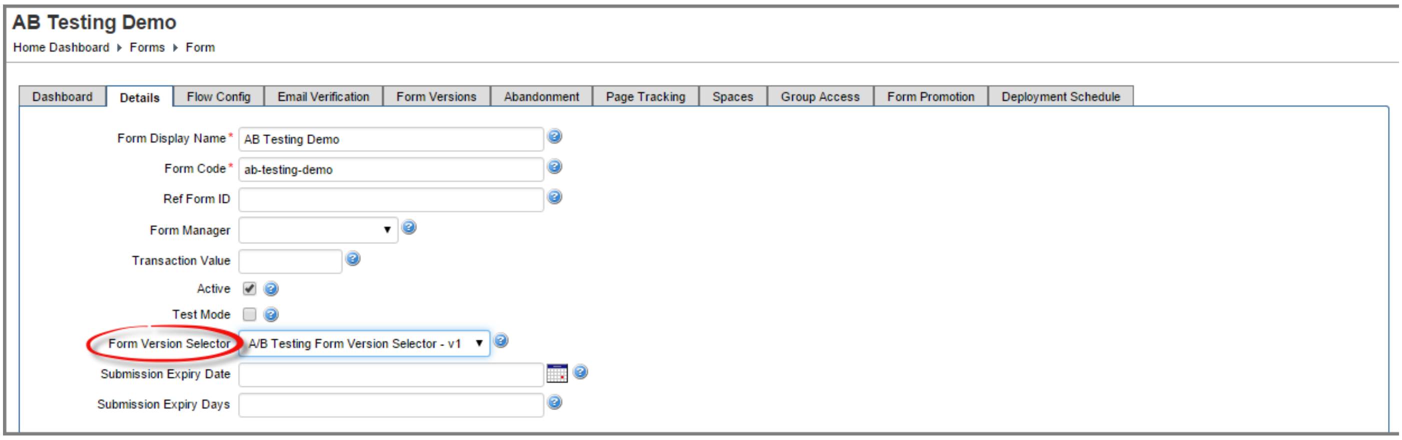
Form version selection is provided using a randomized distribution for A/B Testing, however it is possible to devise a customized version using a scriptable Form Version Selection service. This capability also supports dynamic form version selection for market segment targeting. For instance, if you have a different type of product for a specific geography you can use form version selection to choose the correct product. If you want to provide an alternative product for an existing customer, you can also use form version selection for this purpose.

How to set up A/B Testing.

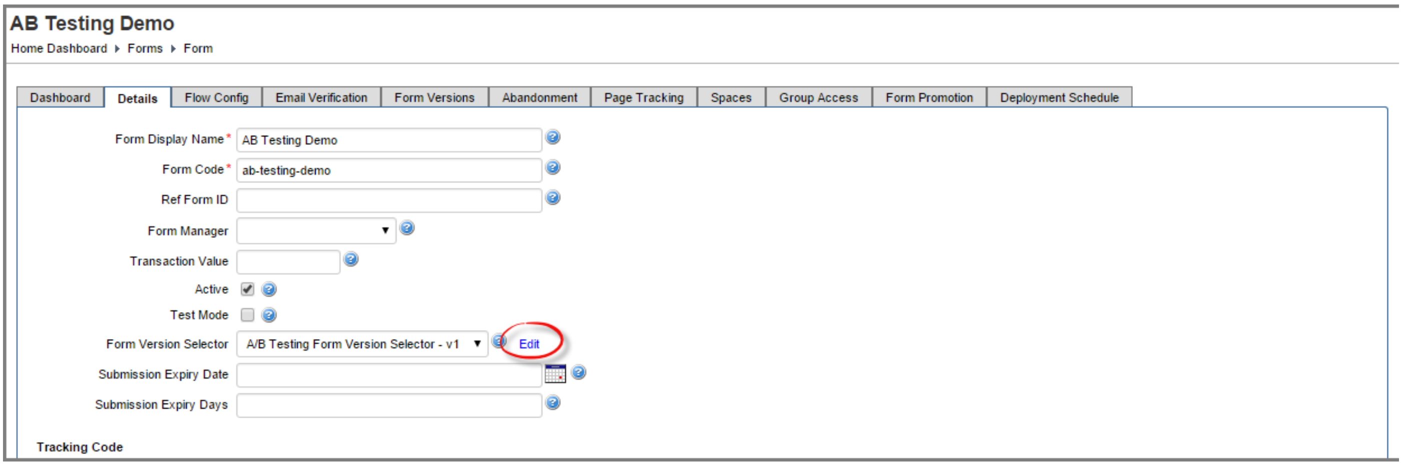
The following example shows 2 form versions of the form "AB Testing Demo". It is possible to have more than this include them in the A/B testing. In the screen below, under normally conditions only the current version v1.0 of the form would be offered to the user. The steps below show how you can include multiple versions to be shown to users at random.



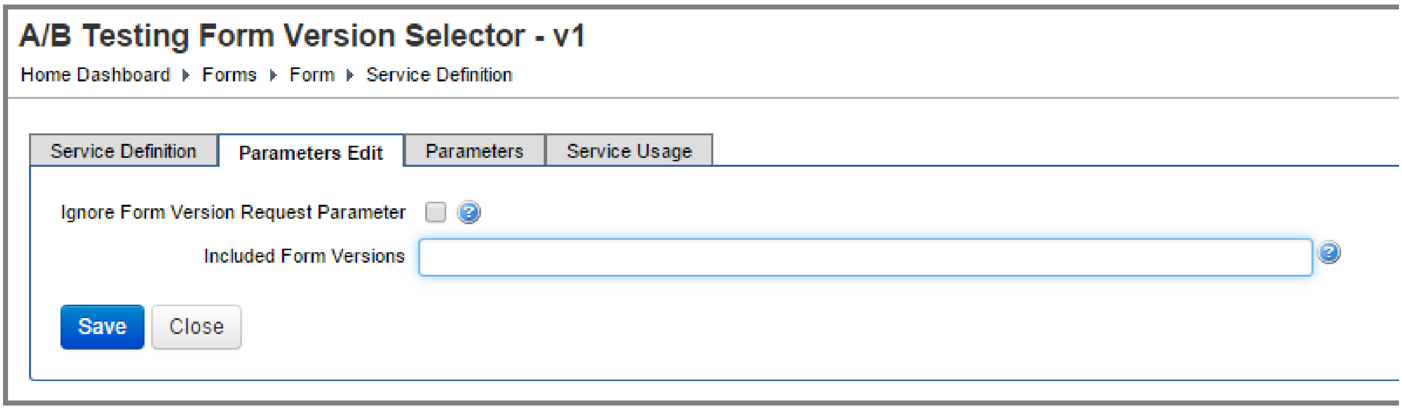
The Details Tab has the option to use a Form Version Selector. The dropdown menu has the form selectors that are available for use. The standard TM form selector will provide a randomized distribution for the A/B testing. It is possible to customize form selectors based on client requirements.



It is possible to modify the service by using the edit facility that navigates to the service definition for the Form Version Selector.



Within the service Definition for the selector it is possible to specify Form Versions you want included in the AB testing. This is especially useful when multiple versions exist but only some of them are relevant to the test.

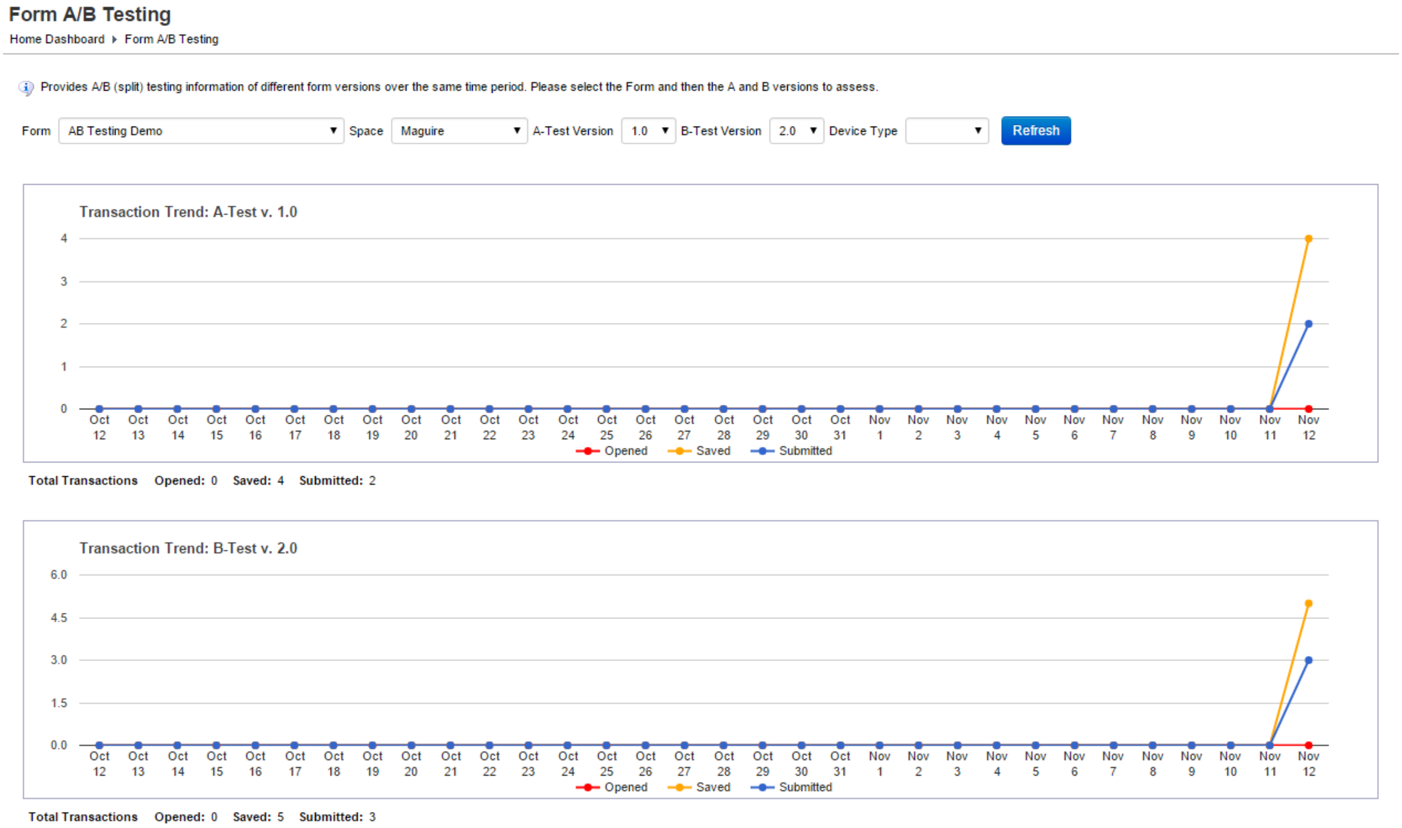


Once the Selector has been set up and parameters have been defined, the form versions are ready for A/B testing. It will mean that any one of the eligible versions will be rendered to the user requesting the form.

Unknown macro: 'pdf_pagebreak'

Analyzing the A/B testing results.

Transact provides out of the box analytics for the A/B testing. It allows comparison between any two form versions at one time. The versions to compare are selected via a dropdown list that includes all eligible versions. Statistics are gathered on Form Open, Save and submitted, further refinement by device type is also available. In the screen grab below results are being compared for form versions 1.0 vs 2.0 on desktop devices.



Transact Database Queries

The underlying Transact database queries used for the Form A/B Testing include:

- Opened - number of transaction "Opened" events. This query runs of the submission_history table.
- Saved - number of transaction "Saved" events excluding background saves. This query runs off the submission_history table.
- Submitted - number of transaction "Submitted" and "Completed" events. This query runs off the submission_history table.

Work Space Help Desk (Manager v5.0)

This is permission driven. Two permissions that drive this functionality are: "Help Desk View" and "Help Desk Authenticated Edit".

Having the Help Desk View Permission will create a new tab on the portal called Help Desk.

This tab shows all the forms submitted, saved or abandoned on the portal. The Help Desk agent can open any of these forms

Managing Form Spaces (Manager v5.0)

Introduction

This topic discusses how to create and manage Form Spaces in Transaction Manager. In Avoka Transact a "Form Space" is a public web module which is used to support Form Transaction Applications

In previous versions of Avoka Transact "Form Spaces" were called Portals, however to avoid confusion with customer's web sites and older Portal technologies, we now use the term "Form Spaces".

In Java Enterprise Edition (JEE) terms a "Form Space" is a Java Web Application which is deployed using a Web Application Archive (WAR) file. Unlike traditional JEE applications in Avoka Transact "Form Spaces" WAR files are used to initialize the Transact database with configuration settings and HTML content. In Avoka Transact the Java application code is provided by the Transact Application server modules and the HTML content is rendered from the Transact database. This model is used to make application upgrades easier. "Form Space" WAR files contain content and default configurations, and don't contain code.

Space Types

In Avoka Transact there are 3 type of of "Form Spaces" available.

Form Space Type	TM Installer	Default Context	Description
TransactField App	Default Server Install	field-worker	This is the Form Space that all TransactField App devices communicate with.
Web Plug-in	Default Server Install	web-plugin	The web plug-in is used to host forms that are embedded in a client website page.
Maguire	Maguire Examples Selected	maguire	This is an example responsive portal that is loaded with example forms and services. It should only be deployed to DEV and TEST environments.

Exclusions to this article

TransactField systems which have more than just the default TransactField App Space.

This has been done to cater for:

- clients with different business structures requiring separate security managers.
- clients who operate in multiple geographic regions requiring multiple Languages or styling.
- shared environments used by Client Services / Pre Sales

Avoka Hosted Services has instructions for creating additional TransactField App Spaces

Form Spaces / Portal with Custom Java Code

Portals created prior to TM 4.2 included java classes. These may have been customised by client services. The Form Spaces created by the Avoka Transact 4.2 onwards do not contain the portal library code. The java source files are kept in a separate portal library.

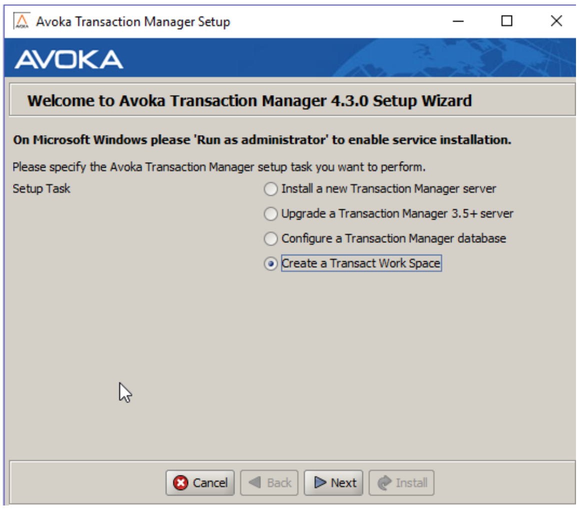
- Avoka Escalations have Linux BASH scripts or Windows batch files that can identify portals that may have been modified in this way.

- Avoka Hosted Services can run these on the Linux hosted clients.
- Engineering has Eclipse projects for form spaces that contain the java source code for TM 4.2.5+. These are ONLY for upgrading older clients that require java source customisations.

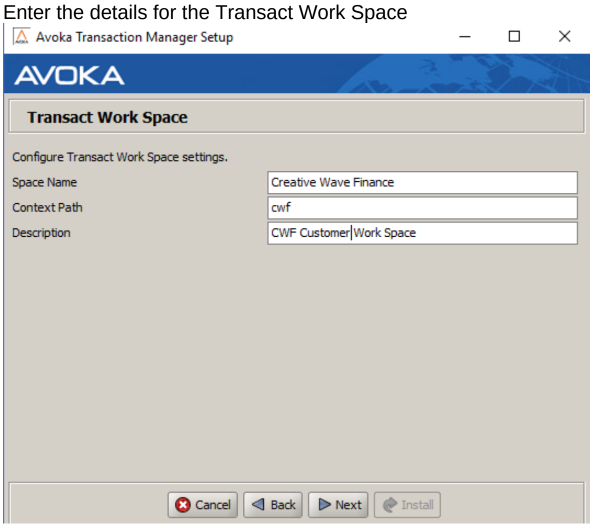
Creating a Form Space WAR File

The following instructions use the Avoka Transaction Manager Installer ***Avoka-TM-Setup.jar.***

1. Double Click the ***Avoka-TM-Setup.jar*** file
   Select Create a Transact Work Space



2. Enter the details for the Transact Work Space



Note

Space Name:

- **IMPORTANT -** Do NOT use the ampersand '&' character in the space name.
- This is the name that Transaction Manager uses as the primary key when a space is imported.
- It should be unique to a customer's environment

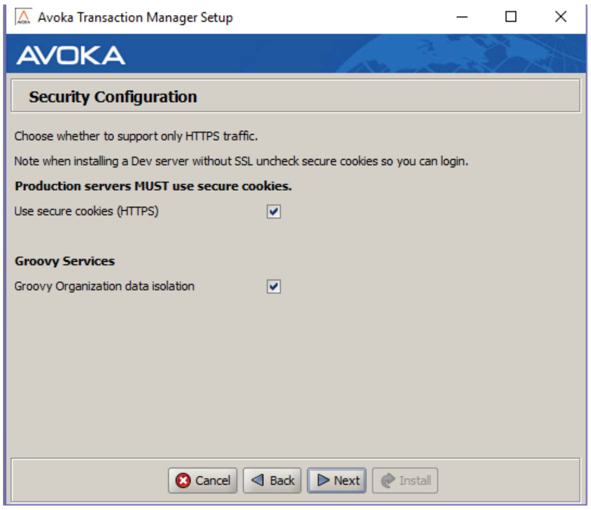
Context Path:

- **IMPORTANT -** Do NOT use the ampersand '&' character in the context path.
- This is a component of the URL as follows https://{Server Domain Name}/{context path}. It's the base path to access the Web Application Archive WAR file.
- This should be unique per environment.
- For the configuration below the URL could be https://creativewave.transactcentral.com/**cwf**
- Transaction Manager modules and web context path like the Transaction Manager Administration module is manager. The URL is https://creativewave.transactcentral.com/manager

Description:

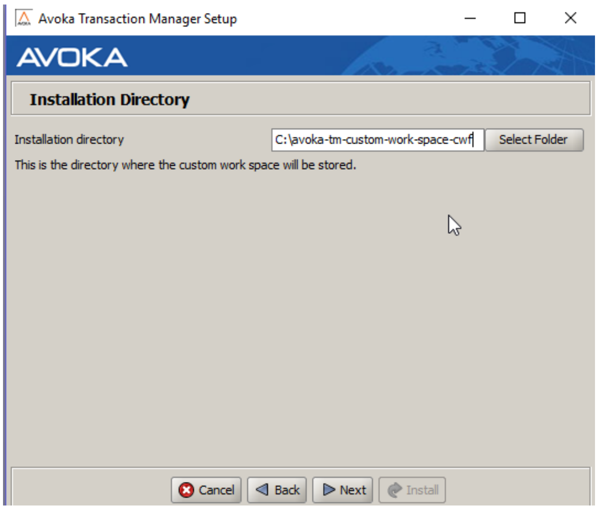
- This is an optional description of the work (form) space

3. Enter the Security Configuration.



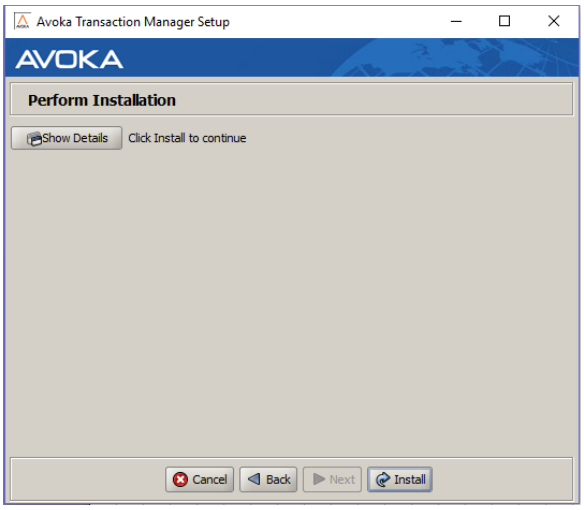
> **Note**
>
> When building a new form space you should accept the default.
>
> The only time you should uncheck **use secure cookies (HTTPS)** is if you are building a test form space to run on your own PC. **Do not provide test form spaces to a client.**

4. Transact Manager Setup



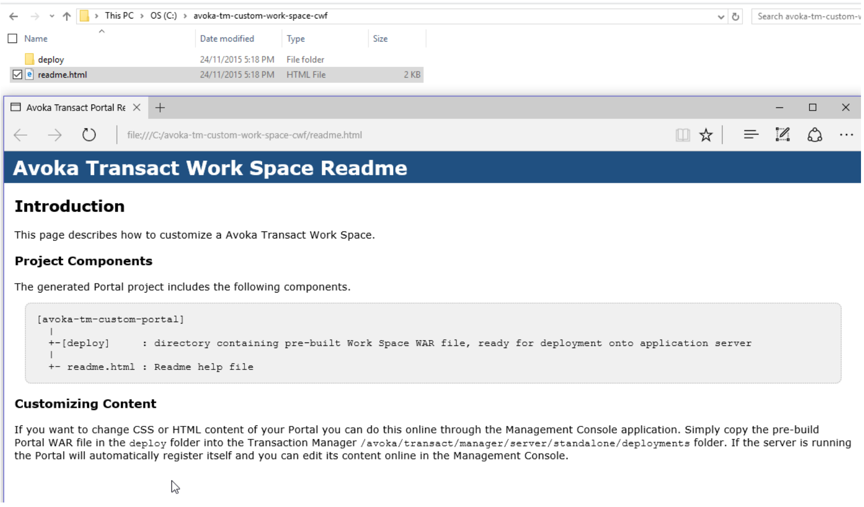
> **Note**
>
> By default on a Windows PC it will install to a directory ***c:\avoka-tm-custom-work-space***
>
> If you are intending to make a number of form spaces its good practice to make the install directory unique.
>
> In the screenshot above I have appended the context ***c:\avoka-tm-custom-work-space-cwf***

5. Perform installation

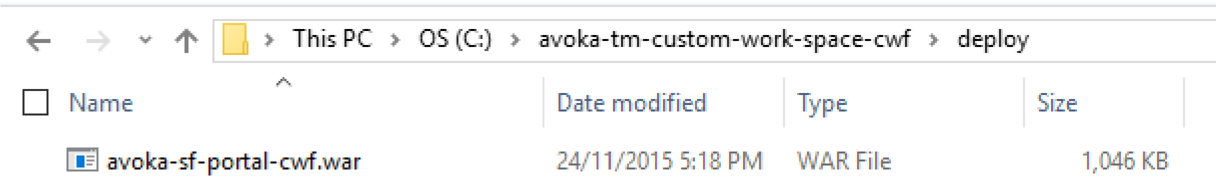


This creates the formspace war file in the specified directory in 4.

Look at the readme.html file



The WAR file **avoka-sf-portal-*{context}*.war** in located the deploy subdirectory.



Importing Form Space War file into Transaction Manager

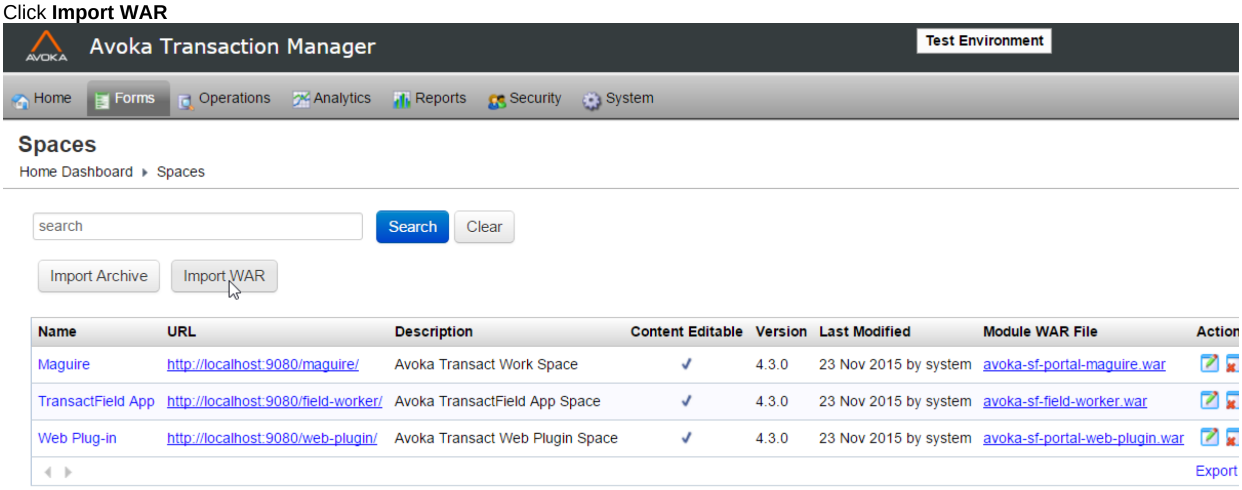
There are 2 ways to install the WAR file into transaction Manager.

1. Copy the pre-build Work Space WAR file in the `deploy` folder into the Transaction Manager `/avoka/transact/manager/server/standalone/deployments` folder

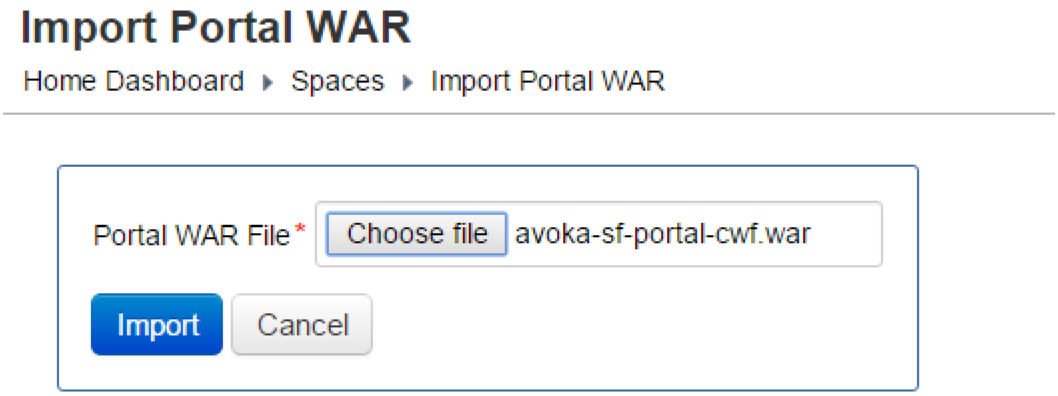
> **Note**
>
> This requires access to the transaction manager directory for the server that you are installing to.
>
> Once deployed you can go to the Form > Form Space menu to see the Spaces search page as per step 2.c. below.

2. Import the WAR file using the Transaction Manager Console. This requires you to have TM Administrator with Global Access.

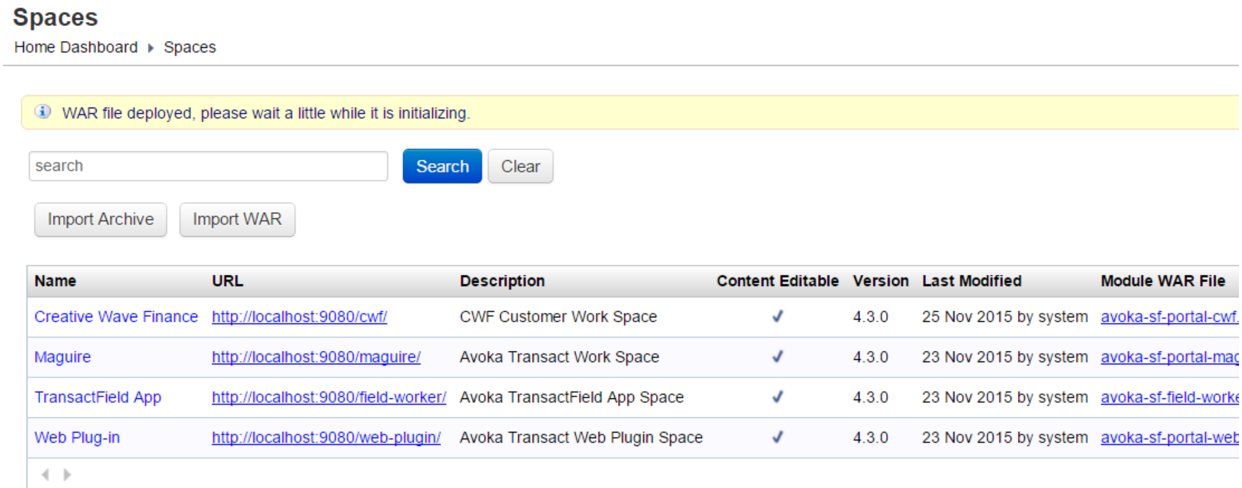
   a. Select the Menu **Forms > Forms Spaces**
      This brings up the Spaces search page as per below
      Click **Import WAR**



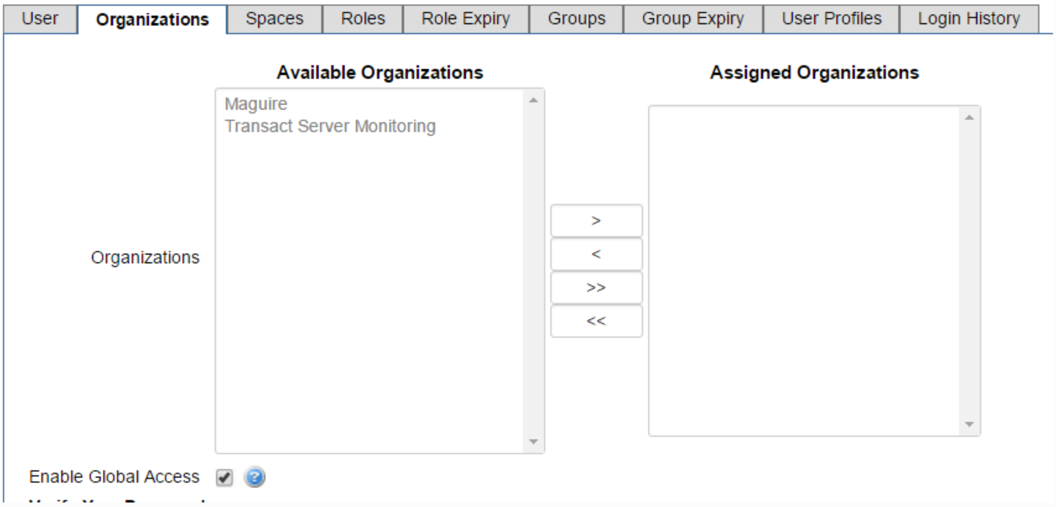
   b. Choose the WAR file and click **Import**



   c. You will be returned to the Spaces search page. You may have to click search after 5-10 seconds for the work space to appear.



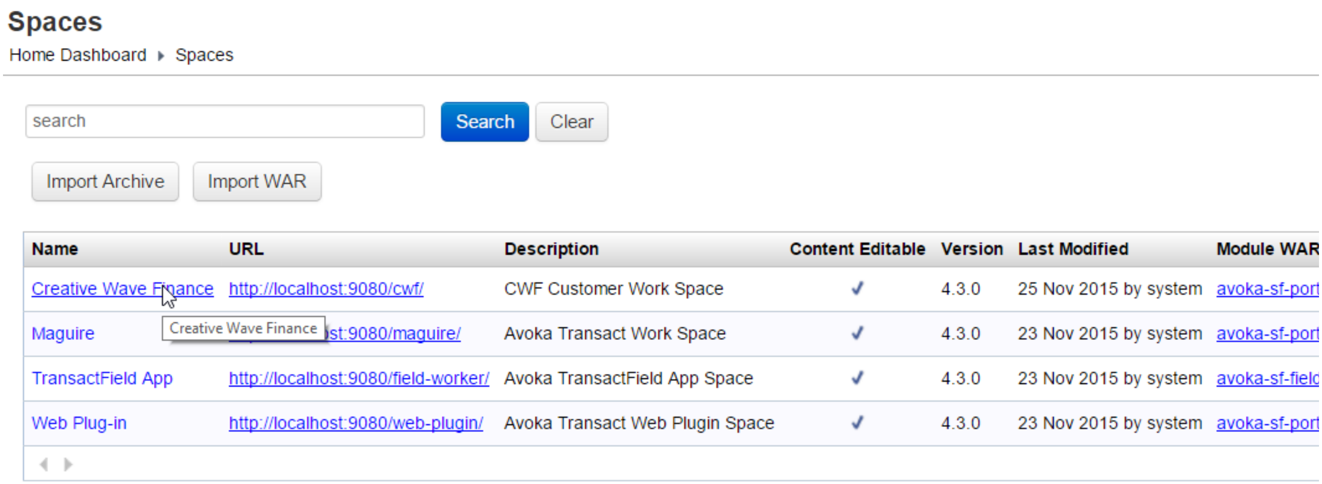
> **Note**
>
> If you don't see the new work space in the list it could mean that there is a problem with your user account configuration.
>
> TM versions prior to 4.3.0 allow an organisation to be assigned for users with Global Access.



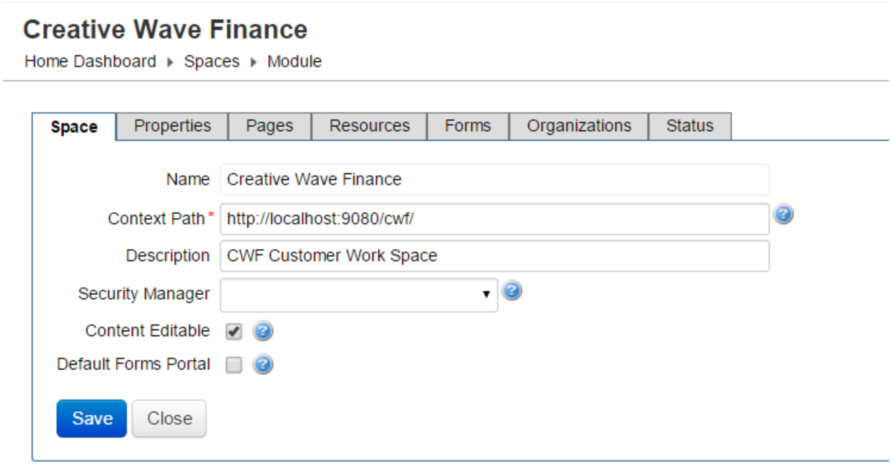
This is an issue as you can only see the spaces that belong to the assigned organisations. When a new space is created it doesn't have an organisation assigned and hence it may be hidden from you.

Setup Form Space in the Transaction Manager Console

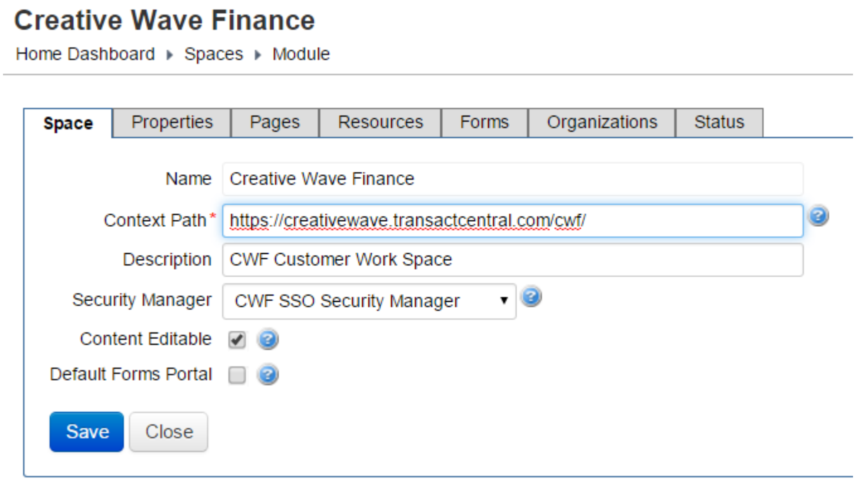
1. Click on the Form Space



2. Change the context path from localhost
From:



To:



Setup Apache Mappings

Apache Server Backgroud

Apache Server sits between Transaction Manager and the Network. It is used to:

- TM offloads the SSL Termination onto Apache. SSL certificates are loaded into Apache.
- Serves cached static content
- Reverse Proxy restring access to the Application Manager running Transaction Manager. Only certain context paths are exposed to the network. This is configured in the **mod-proxy.conf**

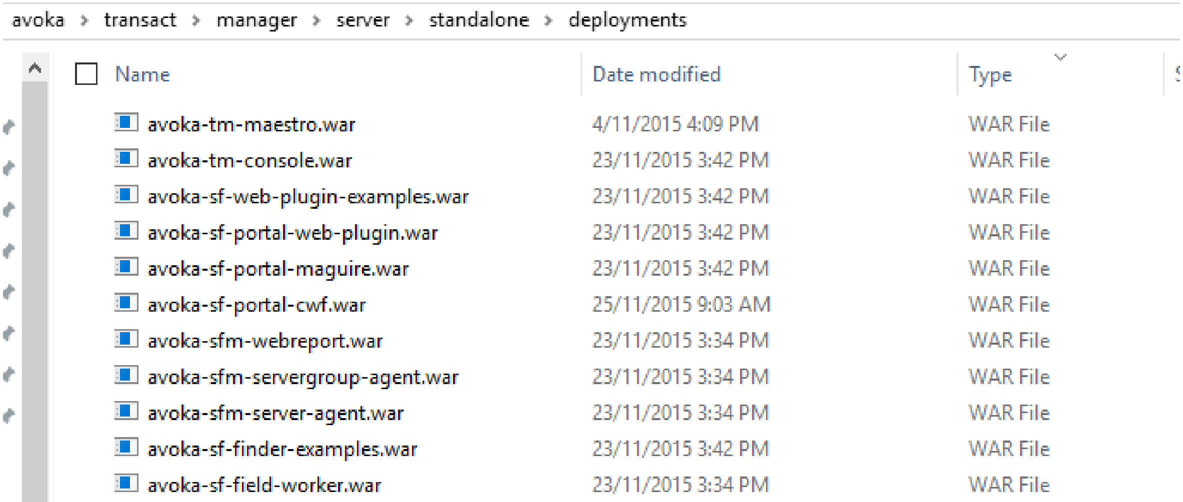
After installing a new form space we need to update the Apache Configuration to expose that to the network. If you need help please see Avoka Hosted Services team.
In the code block below we have setup the Creative Wave Financial Space, by first cloning the TM Example Maguire Portal settings.

extract from /Apache2/bin/mod-proxy.conf

```
# TM Example Maguire Portal
ProxyPassMatch ^/maguire(.*)$ ajp://localhost:9009/maguire$1
ProxyPassReverse /maguire ajp://localhost:9009/maguire

# TM Creative Wave Financial Space
ProxyPassMatch ^/cwf(.*)$ ajp://localhost:9009/cwf$1
ProxyPassReverse /cwf ajp://localhost:9009/cwf
```

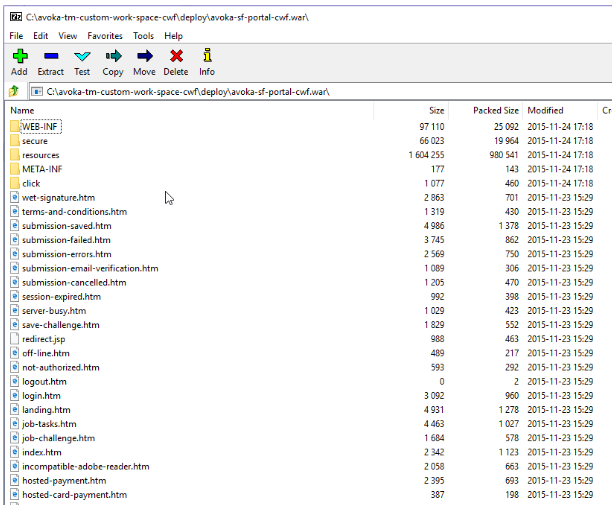
WAR File Components

Transaction Manager is hosted on a J2EE application server runtime. Each form space is built into its own WAR file which was deployed as per the instructions above. The screenshot below show a listing for the war files in the deployments directory. This included modules and example application that come with TM. The form spaces start with "avoka-sf-portal-" + <portal context> + ".war". We can see avoka-sf-portal-cwf.war file has been deployed.



WAR (Web Application Archive)

The WAR file is a type of JAR file defined by the J2EE application server specifications. WAR and JAR files can be opened in tools such as 7Zip. The contents of the war file can be seen below.



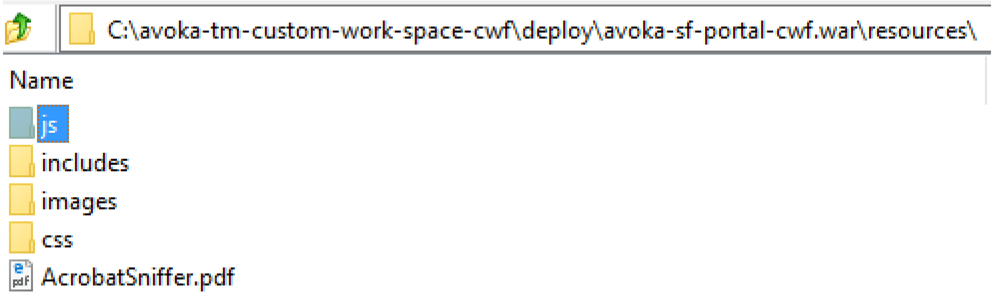
Work Space WAR files contains

Page Templates (.htm files) which are held in the

- root directory (Unsecured can be accessed anonymously)
- secure directory (required an authentication user session) These contain the pages users seen in the form space after a user logs in such as the Tasks or To Do page.

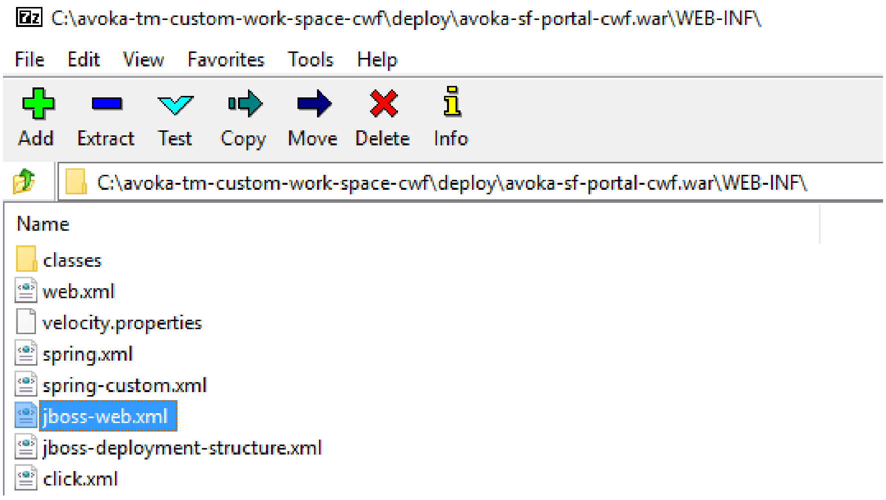
Resources (See image below) this include items used to style the form space.

- **css**: Cascading Style Sheets
- **images**:
- **includes**: html snippets
- **js**: JavaScript files



WEB-INF

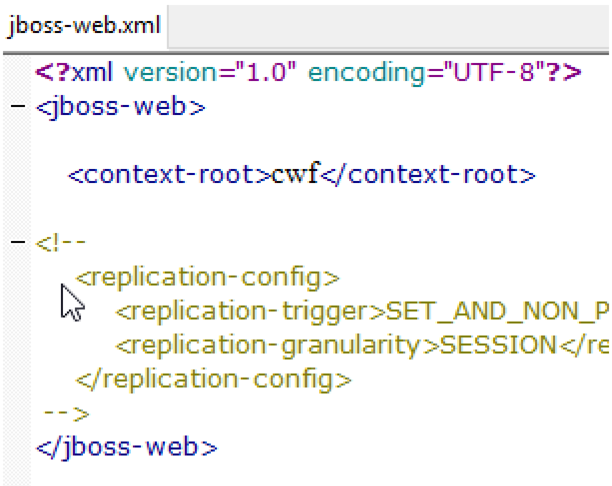
This directory contains the list of xml files and the classes subdirectory. When creating the WAR file (ref Creating a Form Space WAR File) the installer does string substitutions on the context path, and space name into the portal files.



The important files are the:

web.xml: The web application deployment descriptor file describes how the web application should be deployed to the application server. It is required for every Web Application

jboss-web.xml: The installer substitutes the context path here. Note the context path **cwf** we specified in the installer has been written into the context-root element.



Important

Always use the TM Installer to create your Form Space WAR file.

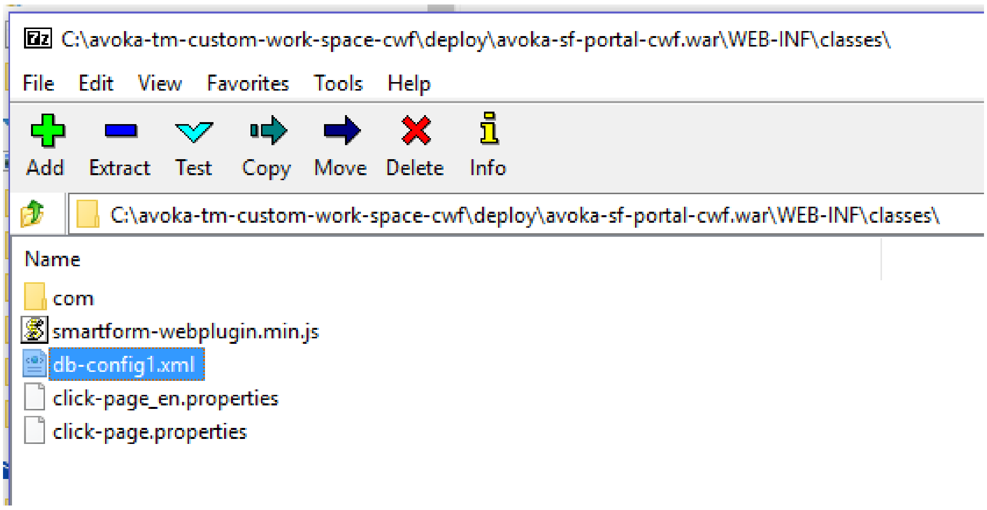
Creating a copy of an existing Form Space WAR file can result in problems with deployment, if all the required substitutions are not made.

If the copying has resulted in 2 WAR files with the same context, the J2EE application server will only deploy 1 WAR file.

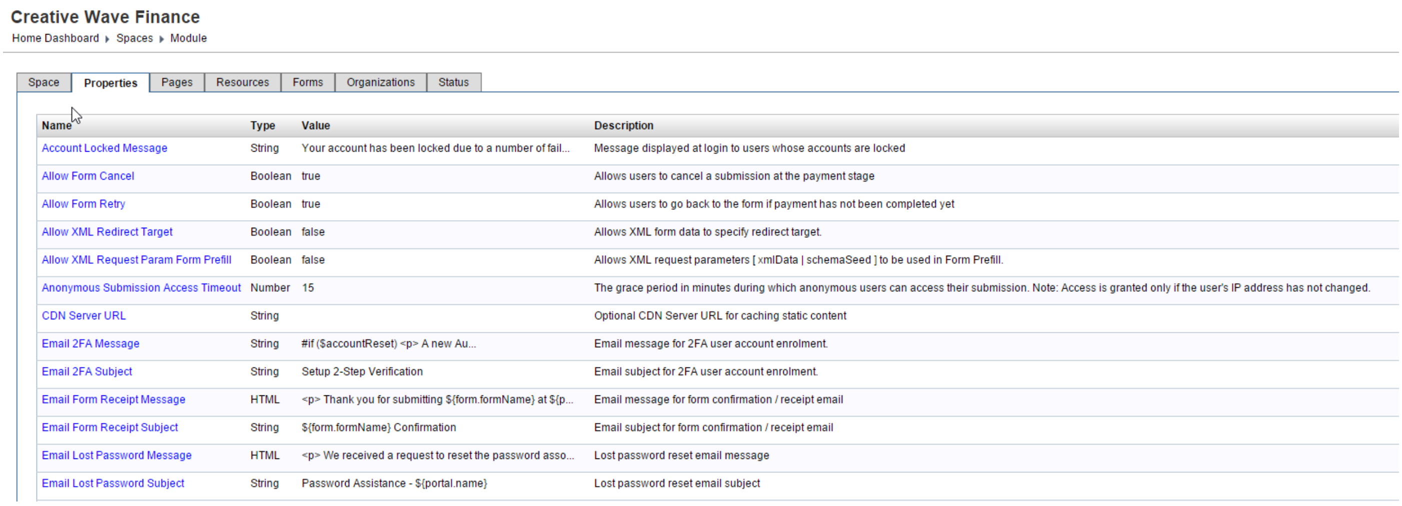
Each WAR file must have a unique context path per TM environment (DEV, UAT, PROD).

WEB-INF/classes

This directory contains the compile java packages and classes.



db-config.xml: This is the important file in this directory. The db-config.xml has had the context path space name substitutions applied. It is used to create all the new form space properties when the WAR file is deployed. The Roles for the Form Space are also defined here.



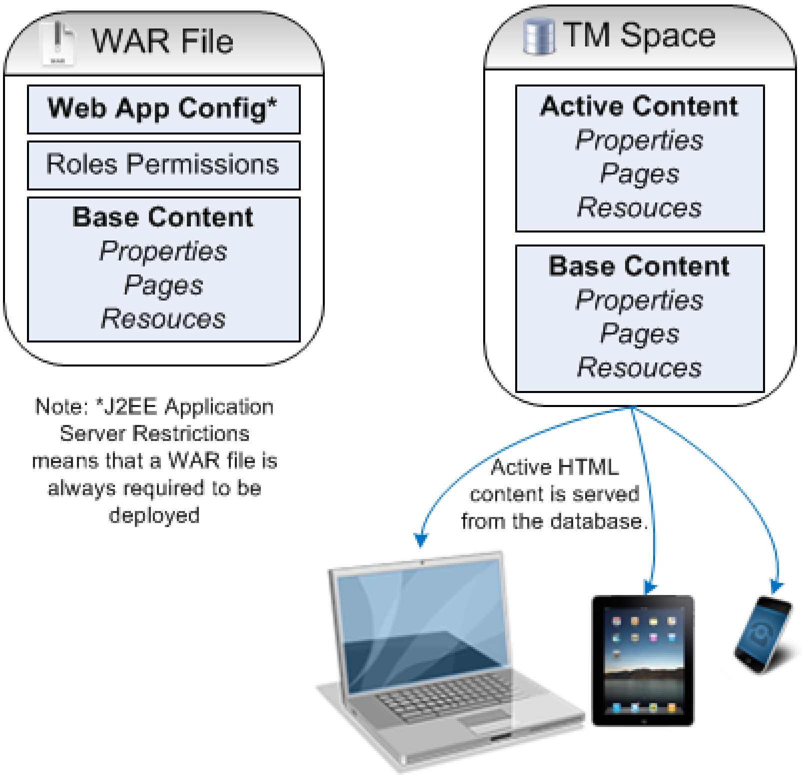
Name	Type	Value	Description
Account Locked Message	String	Your account has been locked due to a number of fail...	Message displayed at login to users whose accounts are locked
Allow Form Cancel	Boolean	true	Allows users to cancel a submission at the payment stage
Allow Form Retry	Boolean	true	Allows users to go back to the form if payment has not been completed yet
Allow XML Redirect Target	Boolean	false	Allows XML form data to specify redirect target.
Allow XML Request Param Form Prefill	Boolean	false	Allows XML request parameters [xmlData | schemaSeed] to be used in Form Prefill.
Anonymous Submission Access Timeout	Number	15	The grace period in minutes during which anonymous users can access their submission. Note: Access is granted only if the user's IP address has not changed.
CDN Server URL	String		Optional CDN Server URL for caching static content
Email 2FA Message	String	#if ($accountReset) <p> A new Au...	Email message for 2FA user account enrolment.
Email 2FA Subject	String	Setup 2-Step Verification	Email subject for 2FA user account enrolment.
Email Form Receipt Message	HTML	<p> Thank you for submitting ${form.formName} at ${p...	Email message for form confirmation / receipt email
Email Form Receipt Subject	String	${form.formName} Confirmation	Email subject for form confirmation / receipt email
Email Lost Password Message	HTML	<p> We received a request to reset the password asso...	Lost password reset email message
Email Lost Password Subject	String	Password Assistance - ${portal.name}	Lost password reset email subject

Transact Architecture Space Architecture

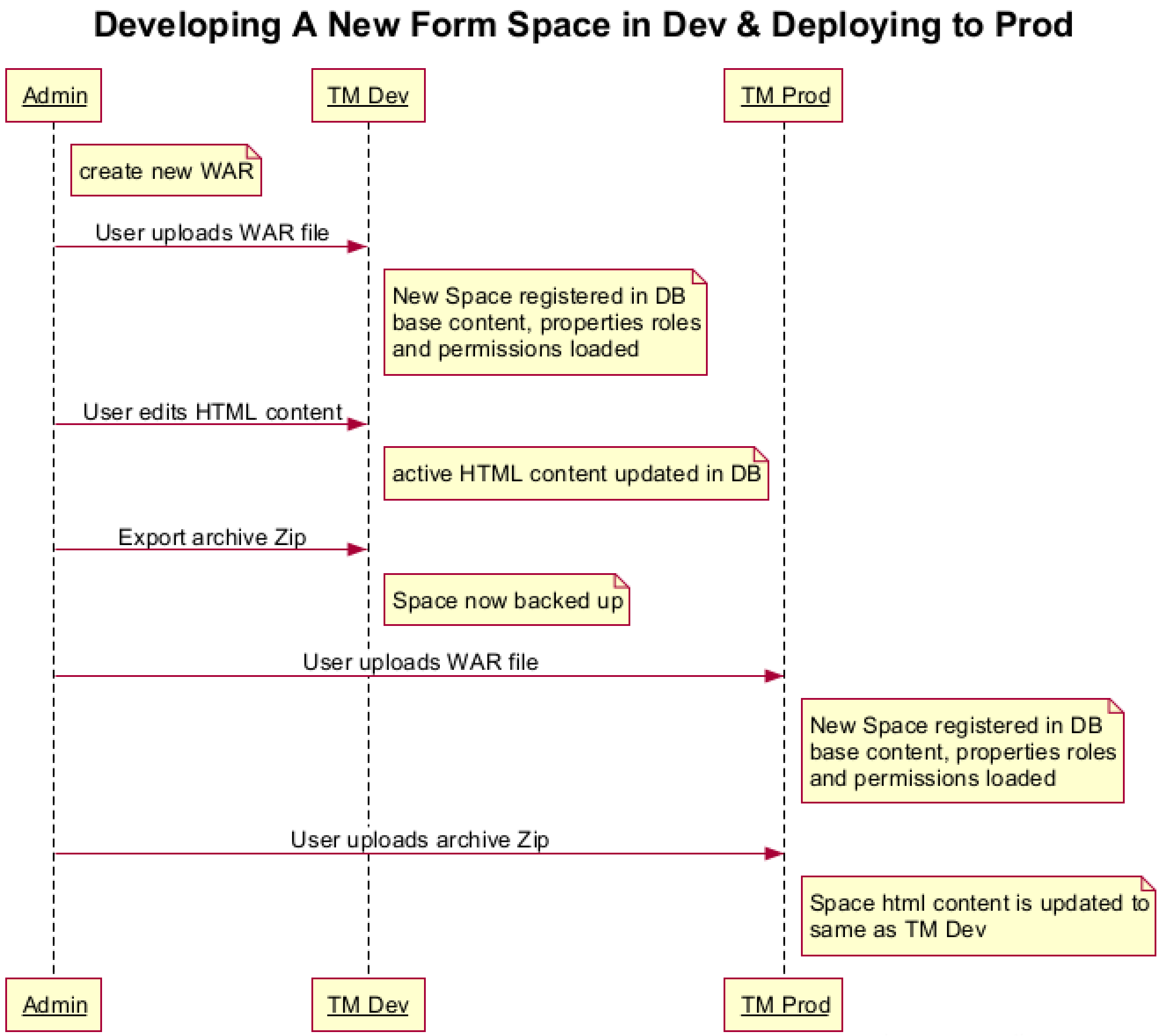
The following diagram explains the relation ship between the WAR File and the the Space.

- It is necessary to upload a war file
- The war files content becomes the base content in the database.

- The active content in the space is either edited in the admin console or imported in an archive zip file.



The sequence diagram shows developing a new form space and then deploying it to production



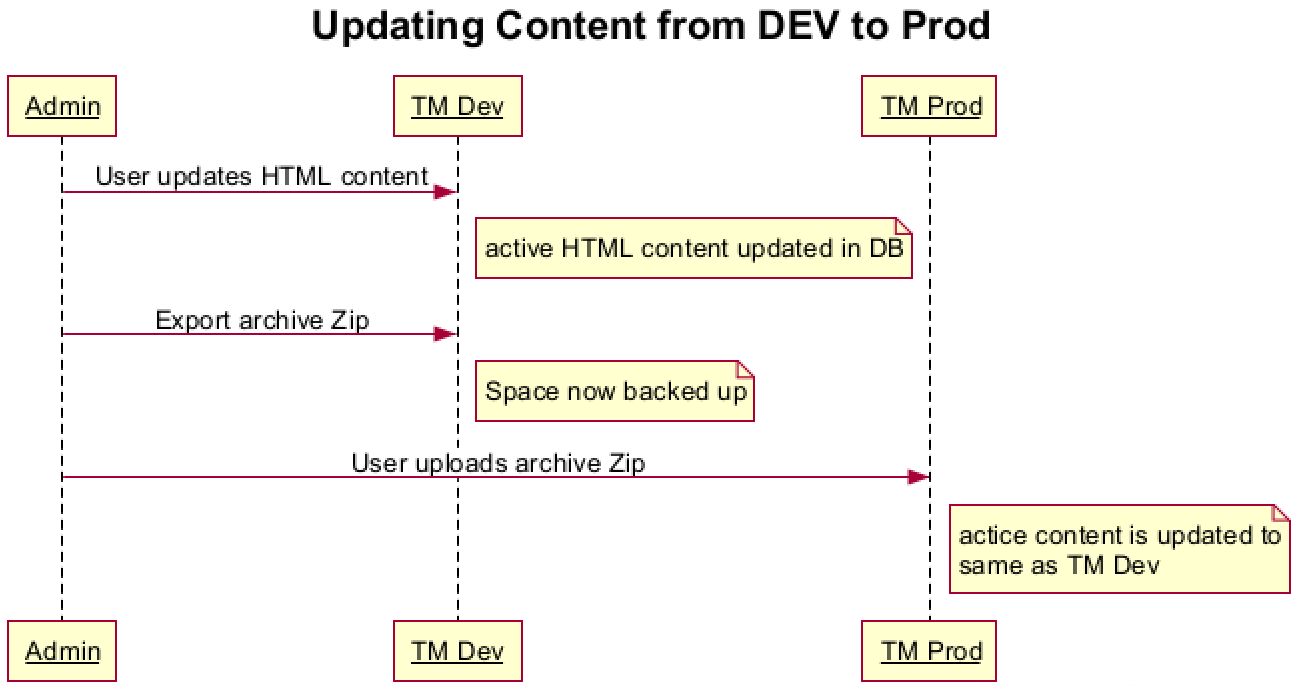
Editing Portal Content - Skinning

The portal skinning has been documented by Client Services

TODO ADD LINK Unknown User (jbasey)

Updating and Deploying HTML Content

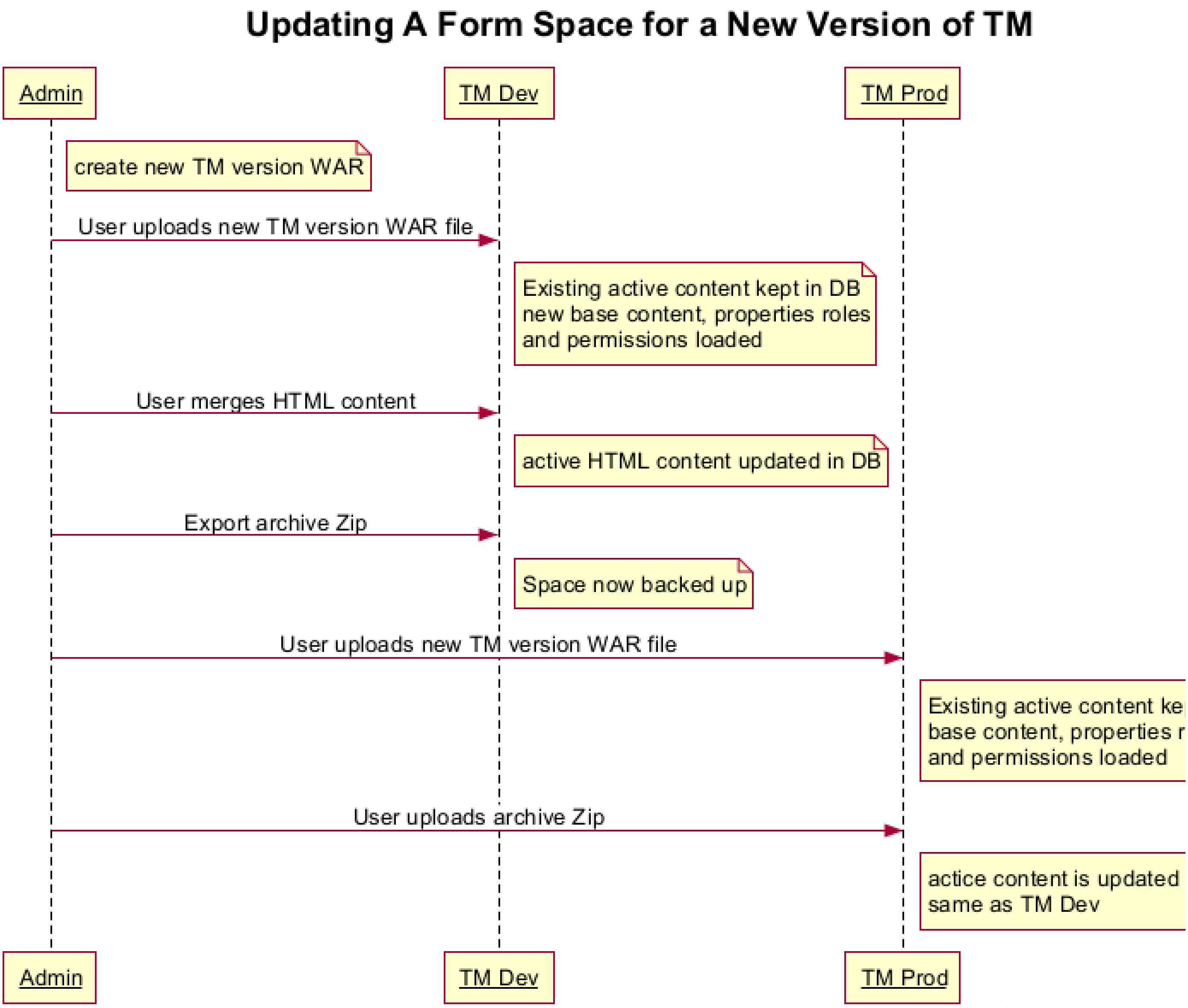
The following sequence diagram shows the steps required to update content from Dev to Prod.



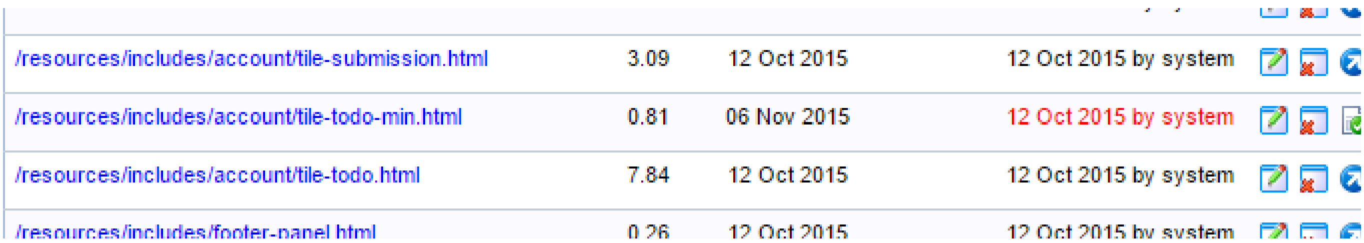
Merging Active HTML Content

The following sequence diagram shows the steps required when a new version of TM is installed.

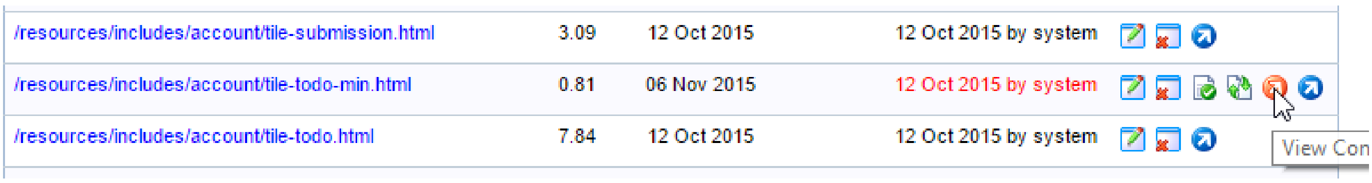
When upgrading a new war file, it is important to review all the pages and resources that were customised. This is required as new versions of Transact may introduce new functionality and bug fixes which update the pages. By default when a Form Space is upgraded the existing active html content (pages and resources are kept). The Transaction Manager Console - Spaces Page enables the reviewing of pages and resources with similar principals found in source control systems.

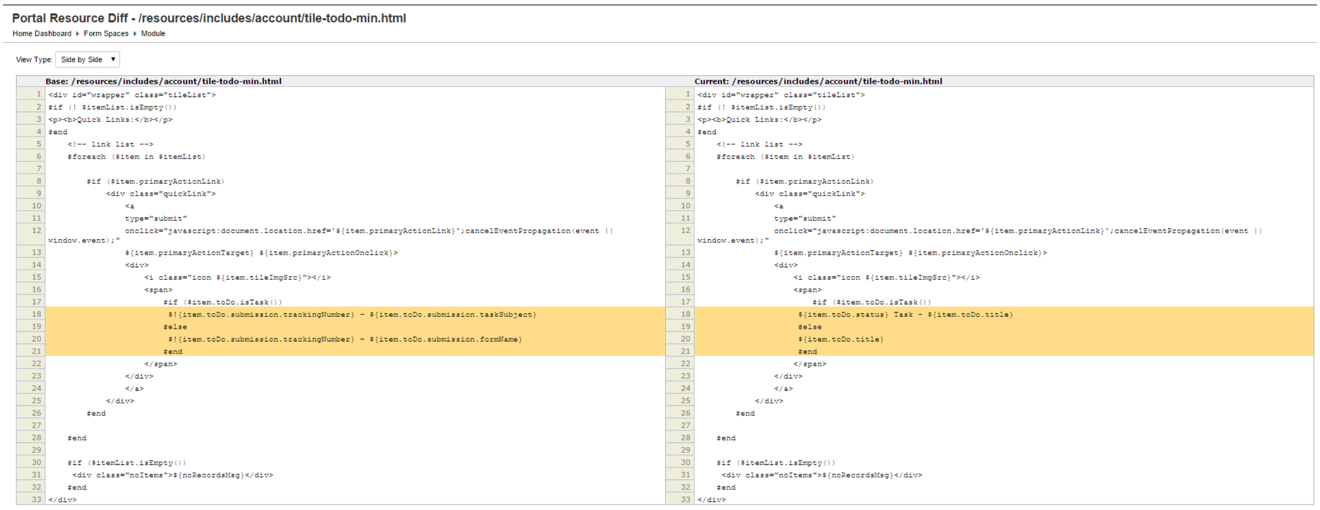


In this example the **tile-todo-min.html** was update in the 4.3.0 form space as part of a bug fix. Note this is highlight and has more action items.



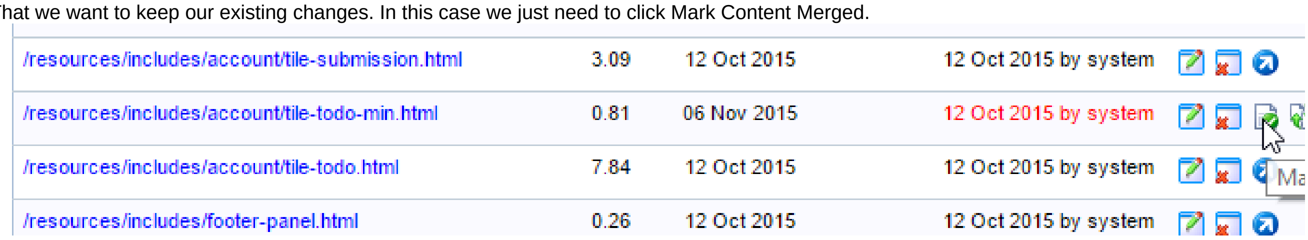
Click on the View Content Diff to Base Resource and use this to assist in merging content.





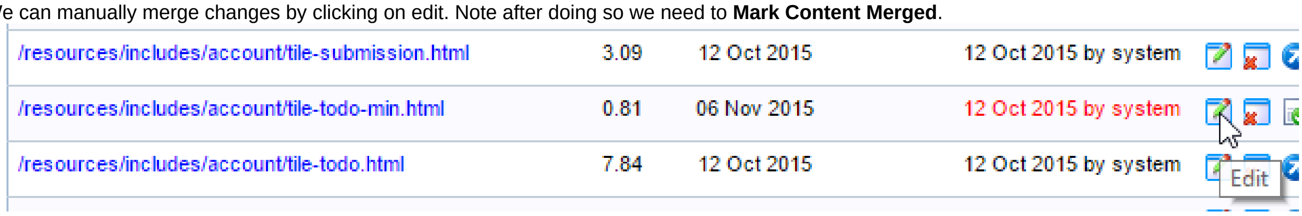
We may decide:

- That we want to keep our existing changes. In this case we just need to click Mark Content Merged.



- We can replace the content with the the base content. By clicking first on "Restore all Portal Resource with theie Base Content"

- We can manually merge changes by clicking on edit. Note after doing so we need to **Mark Content Merged**.



BIRT Reporting Guide Manual (Manager v5.0)

Introduction

This document is aimed at users wanting to create or modify Avoka Transaction Manager BIRT business reports.
BIRT stands for Business Intelligence and Reporting Tools, and is an open source data visualization and reporting platform.

http://www.eclipse.org/birt/

Transaction Manager includes the BIRT runtime as well as a number of standard reports you can use out of the box. This guide assumes you are already familiar with BIRT and covers only specifics you need to be aware of when developing TM reports.

Operational Transaction Summary

Avoka Transact

This report includes only transactions that have been finalized (completed or abandoned). Transactions in test mode are not shown.

Organization(s): All
Start date: 12/11/2014 24:00
End date: 12/12/2014 24:00

Form Name	Form ID	Total	Completed	System Abandoned	Admin Abandoned	User Abandoned	Completion Rate	Last Completed	Median Time to Submit	Payments	Amount Paid
Administrative Services		620	219	334	24	43	35%	11 Dec 14	<1 min	0	$0.00
401 Test - HTML	3070	4	2	2	0	0	50%	08 Dec 14	<1 min	0	$0.00
754 Tfield test case - Default	2878	6	5	0	0	1	83%	10 Dec 14	<1 min	0	$0.00
ASC-2952 Dropdown Test - HTML	2713	2	2	0	0	0	100%	02 Dec 14	<1 min	0	$0.00
ASC-3049 Maguire - Default Rec	3216	6	6	0	0	0	100%	11 Dec 14	<1 min	0	$0.00
ASC-3062 Signature Widget test	3131	2	0	2	0	0	0%			0	$0.00
ASC-3062 Signature Widget test	3132	5	0	5	0	0	0%			0	$0.00
ASC-3062 Signature Widget test	3130	1	0	1	0	0	0%			0	$0.00
ASC-3080 DOB issue Maguire - H	3114	2	0	2	0	0	0%			0	$0.00
ASC-3080 DOB issue Maguire 41v	3116	1	0	1	0	0	0%			0	$0.00
ASC-3095 Calendar date selecti	3119	2	0	2	0	0	0%			0	$0.00
ASC-3095 Calendar date selecti	3118	2	0	2	0	0	0%			0	$0.00
ASC-3100 Address widget T-Fiel	3213	3	3	0	0	0	100%	11 Dec 14	<1 min	0	$0.00
ASC-3100 Address widget T-Fiel	3214	1	1	0	0	0	100%	11 Dec 14	<1 min	0	$0.00
Address receipt41 - Default Re	3171	1	1	0	0	0	100%	05 Dec 14	<1 min	0	$0.00
Address widget T-Field 41- HTM	2336	24	12	6	0	6	50%	09 Dec 14	6 min	0	$0.00
Address widget T-Field41SP1 -	3170	15	4	11	0	0	27%	09 Dec 14	<1 min	0	$0.00
Asset Management Form - HTML	2780	1	0	0	0	1	0%		1303 min	0	$0.00
Attachment - HTML [Phantom Rec	1874	57	35	6	0	16	61%	10 Dec 14	2 min	0	$0.00
AttachmentV2 - Default Receipt	3051	1	0	1	0	0	0%			0	$0.00
Avoka Transact Showcase Form f	1757	3	1	0	1	1	33%	18 Nov 14	<1 min	0	$0.00
Avoka Transact Showcase Form4.	2332	14	8	1	1	4	57%	21 Nov 14	<1 min	0	$0.00
Avoka Transact Showcase with M	2600	2	0	2	0	0	0%			0	$0.00
Avoka Transact Showcase with M	2333	21	18	1	0	2	86%	05 Dec 14	<1 min	0	$0.00
Barcode New Widget - HTML	2712	1	0	0	0	1	0%		2 min	0	$0.00
Calendar - HTML	3056	4	1	3	0	0	25%	02 Dec 14	<1 min	0	$0.00
Calendar and Date Test - Defau	2871	18	15	2	0	1	83%	01 Dec 14	<1 min	0	$0.00
Calendar and Date Test SP3 - s	3071	25	0	25	0	0	0%			0	$0.00
Calendar4.0SP3 - HTML [Phantom	3055	16	1	15	0	0	6%	26 Nov 14	<1 min	0	$0.00
CalendarMaguire - Default Rece	2682	14	2	11	0	1	14%	18 Nov 14	2 min	0	$0.00

Developing Reports (Manager v5.0)

This chapter goes into specifics on how to develop a BIRT report that will work with TM.

BIRT Development Environment

TM currently uses BIRT version 4.3.0. We recommend that you use the same version of BIRT to develop your reports. Otherwise it is up to you to ensure they run without problems on the version of TM that you are using.
Download and install the designer of your choice from http://download.eclipse.org/birt/downloads/

TM Database Setup

To develop and test your reports, you need to have access to a TM database. We highly recommend that you use a database that has realistic TM data. This can be done by obtaining a database backup from a TM server running the version of TM you are targeting and restoring this backup to a local database you can access.

Please refer to the TM Installation Guide for the set of supported database versions and instructions on how to set them up.

Note that TM supports three databases: MySQL, SQL Server and Oracle. Always make sure to use the same database (including version) for report development as what is used by your TM servers because each database requires subtly different syntax and possibly other configuration such as BIRT report parameter types. If you are developing reports for multiple databases, you need to test them on each database separately and it may be necessary to maintain different versions of the report design.

Data to Report on

This guide focuses on a very specific subset of the TM database, namely the transaction history entity. The transaction_history table contains historical data for submissions that have been finalized (either by being completed successfully or being abandoned). The transaction history contains a large amount of information about each transaction, including form and organization name, payment and attachment information and time taken to complete the transaction. Please refer to section The Transaction History Table for a detailed list of columns in the transaction_history table.

While using the transaction history as a basis for reports allows you to create reports for a large number of use cases, there are some cases where it is not sufficient: Reports requiring live submission data (i.e. submissions that have not yet been completed or abandoned), reports requiring submission information that is not present on the transaction history (e.g. filtering by submission properties or data extracts) and reports requiring data from other tables (e.g. audit log, request log, form). If you think that the transaction history is not sufficient for your use case, please contact your Avoka representative to discuss how your report can be implemented.

Installing Database Driver(s)

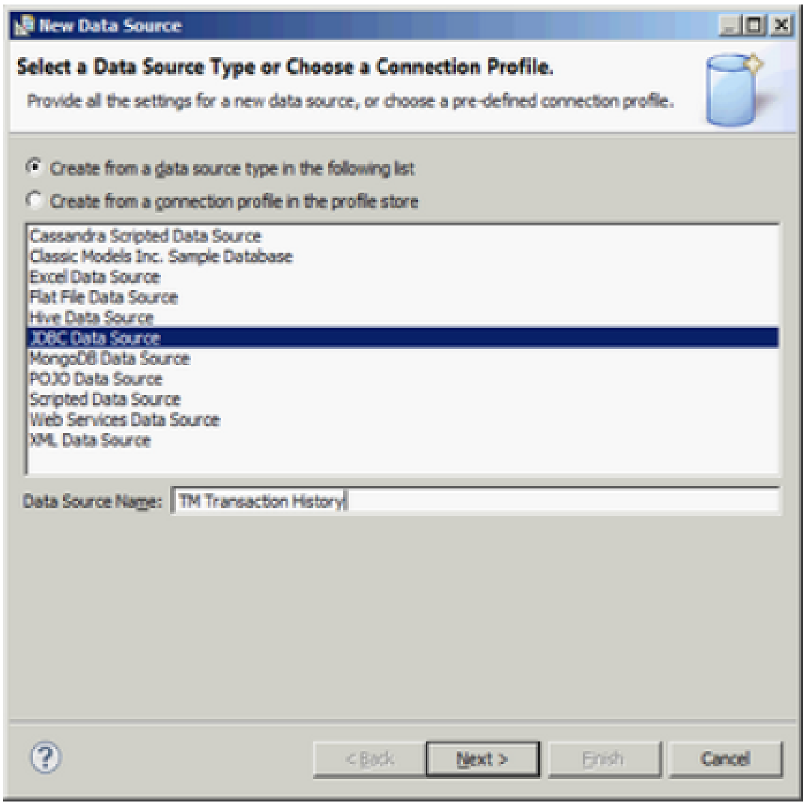
In order to connect to your TM database during report development, you'll need to ensure that BIRT has access to the appropriate JDBC drivers.

The drivers can be copied from the TM server (or installation artefacts) directory in the /modules subfolder:
MySQLThe current MySQL driver is at /modules/com/mysql/main.
SQL ServerThe current SQL Server driver is at /modules/com/microsoft/sqlserver/main.
OracleThe current Oracle driver is at /modules/com/oracle/main.

Copy the driver JAR file to the appropriate directory in your BIRT designer (generally [designer root folder]/plugins/org.eclipse.birt.report.data.oda.jdbc_4.3.0.v201306041519/drivers) and restart BIRT designer. All driver JARs in the directory should automatically be loaded by BIRT designer and will be available when creating a data source.

Setting up a Data Source

TM currently uses two data sources, one for the transaction history table (which can be situated in its own database schema) and another for all other tables. This guide focuses on the transaction history table, so for our purposes you just need to set up the report with a single data source.
In your BIRT report, add a JDBC data source named (for example) "TM Transaction History".

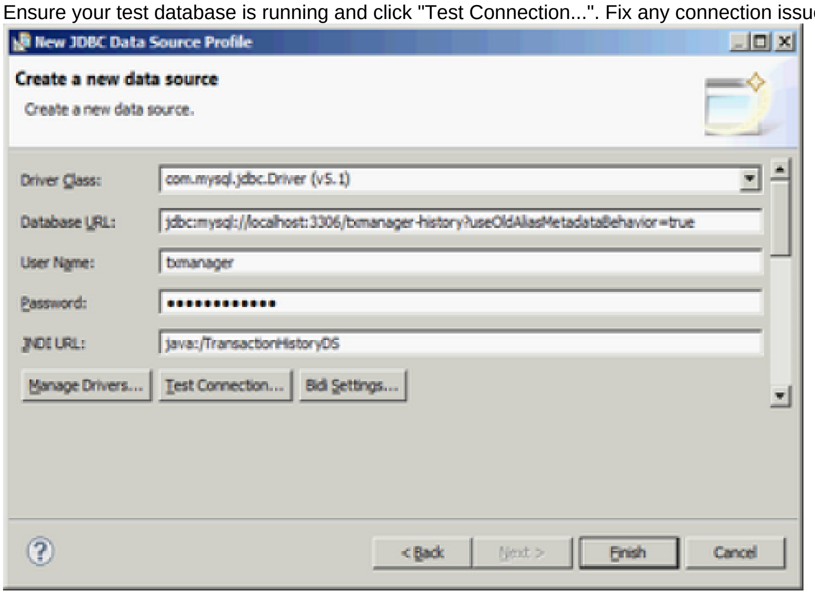


Next, fill in the JDBC connection parameters as follows:

Set the JNDI URL to "java:/TransactionHistoryDS". This is very important since when your report is deployed to a TM server, it will be looking up the data source using the JNDI URL. Only if this lookup fails (e.g. when you are working inside BIRT designer) will BIRT fall back to the hard coded connection parameters below.

Set the driver class to the appropriate driver for the database you will be using ("oracle.jdbc.OracleDriver" for Oracle, "com.microsoft.sqlserver.jdbc. SQLServerDriver" for SQL Server and "com.mysql.jdbc.Driver" for MySQL).
Set the database URL to the appropriate URL for accessing your test database, and the username and password to the values needed to successfully connect to your test database.

Ensure your test database is running and click "Test Connection...". Fix any connection issues that come up.



You can use this data source in all reports utilizing the transaction history table.

Supported BIRT Parameters

TM supports a certain set of report parameters and will pass them into the report whenever a report schedule is triggered if the report has parameter(s) with one of the following names:

- client_oidThis parameter will be populated with the organization ID (a number), if the report is being run for a specific organization at the time (this is the case when running / rendering a client report schedule). In BIRT, this should be modelled as an "Integer" parameter.
- start_dateThis parameter will be populated with the start date and time of the reporting period. This should be modelled as a "Date/Time" parameter in BIRT

- end_dateThis parameter will be populated with the end date and time of the reporting period.
- batch_idThis parameter is specific to payment processing and does not apply to the transaction history data set. It represents the number representing a batch (corresponding to a 24 hour period) of payments. This should be modelled as an "Integer" parameter in BIRT.

At this stage, TM does not support passing arbitrary parameters to a report at runtime. If this is something you need, please contact your Avoka representative to discuss.

Please note that if a user is rendering the report on-screen (as opposed to an automated report schedule being triggered by a background job), TM will not pass in any parameters except for client_oid (if available). Therefore you can configure these parameters in BIRT to be required / not hidden to solicit input from the user viewing the report (or make parameters hidden and use default values if they should not be exposed). You can use additional, arbitrarily named parameters for onscreen report viewing.

If you would like to see examples of these parameters being used and passed into queries, please see the TM standard reports (specifically "Client Operational Summary" (client-operational-summary-*.rptdesign) for start_date, end_date and client_oid and "Payments Daily" (payments-daily-all.rptdesign) for batch_id).

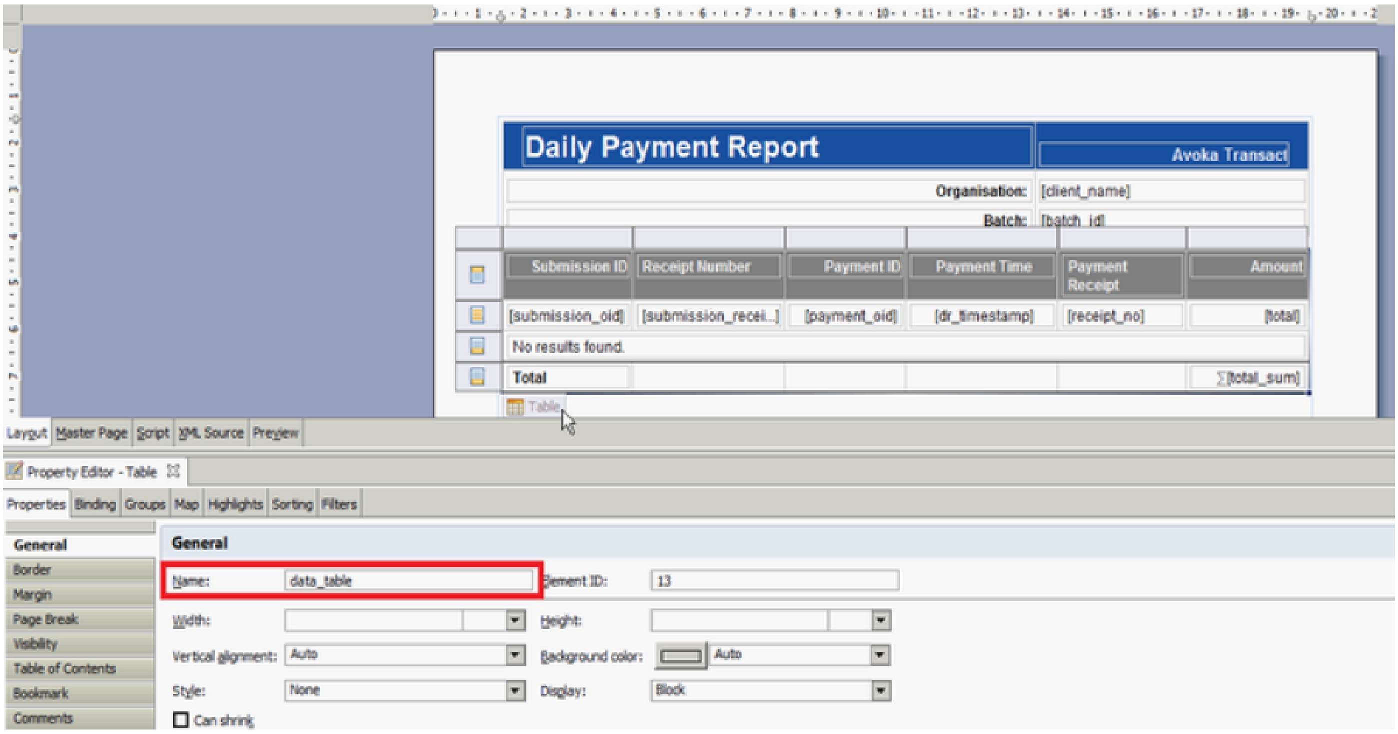
Supported Output Formats

TM uses multiple output formats when rendering reports: For on-screen viewing of reports via the TM management console, the reports are rendered as HTML. When sending out reports as part of a schedule, the output format can be configured to be PDF, XLS or CSV when creating the report schedule (see the TM Administration Guide, chapter "Reports" for details).

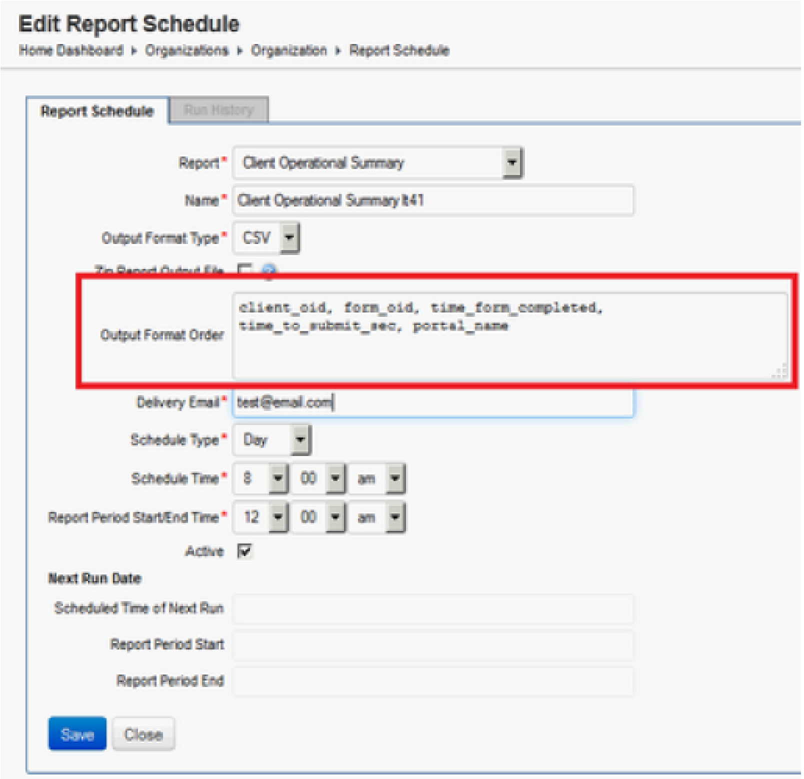
When choosing PDF or XLS as the output format, the report is rendered by a BIRT emitter and appears similar to the HTML rendition. However, when choosing CSV as the output format, the TM reporting module generates the CSV file such that it is easier to be processed automatically by backend applications. Read on for more information on how the CSV report is generated and how to design your report for usage with CSV report schedules.

CSV Reports

The CSV output format that can be used when setting up report schedules in TM is designed to produce pure CSV reports that can be programatically processed by backend applications.
When creating a CSV report, TM uses data from a single table in the BIRT report and ignores all other elements and formatting. Specifically, TM looks for a table named "data_table" and extracts the data from the report design.Therefore, you need to name the table containing the data that you want to use for CSV export "data_table" (see screenshot).



TM allows you to filter the set of columns that will be included in the CSV file by specifying the set of column names in the report schedule configuration. This also allows you to reorder the columns. All other filtering, as well as sorting, has to be done in the report design.



If you would like to see an example of a CSV compatible report, look at the TM report named "Payments Daily" (payments-daily-all.rptdesign).

Deploying Reports to TM

For more information on how to deploy reports to TM and set up report schedules, please refer to the TM Administration Guide, chapter "Reports".

Ongoing Maintenance

TM is a living product that is continually being enhanced. New versions are released regularly, and often contain changes to the database schema that may affect BIRT reports.

Therefore, for each release, the standard TM BIRT reports are reviewed and modified as needed. While substantial changes to the submission data model are rare, it is still necessary to review all custom BIRT reports whenever a new version of TM is to be deployed. To do so, review the TM release notes and upgrade instructions for all versions between the current version and the version you will be deploying. In addition, perform testing of the reports on a server running the new TM version.

Reporting Tables (Manager v5.0)

Transaction History Table

The transaction history table contains extensive data about finalized (completed or abandoned) submissions. It is lightweight as it does not store the submission XML data or other large CLOB/BLOB fields and does not contain sensitive user-entered information. It allows you to write a number of reports on transaction metrics and is used by the TM reports "Operational Summary" and "Client Operational Summary".

By default, transact_history records are kept for 5 years, while submission records are kept for 1 year. These data retention settings are configurable with the Data Retention Management service.

Submission records are published to this table once they progress to a form_status of "Completed" or "Abandoned". This is performed by a background job which runs every hour.

The transaction_history table can be located in a separate history database, or can be located in the TM transaction database. It uses a separate JDBC connection pool.

The TM server supports a Groovy Service which can be used to publish data from the transaction_history table to an external data warehouse (DWH). The transaction_history table provides 2 columns to support this (dwh_publish_status and dwh_publish_time).

Columns

Avoka Transaction Manager provides a transaction history table (transaction_history) which is designed specifically for reporting and analytics. By providing a separate reporting table, this helps ensure system online transaction processing (OLTP) performance is not effected by online analytics processing (OLAP).

Data in the transaction history table is de-normalized (including related data from the form, client, template_version and portal tables) to make report development easier for report writers.
Data from the main transaction table (submission) is rolled into the transaction_history table, about 30 minutes after a transaction has been completed by the user. A transaction is considered to be completed when the form_status = 'Completed' or form_status = 'Abandoned'. The copying of transaction records is performed by the background TransactionProcessor job which runs every 5 minutes.

The transaction history table is very extensive and contains a large number of columns you can report on. Here is a breakdown of the columns and the data they typically contain. Please note the available indexed columns when creating your queries:

- **abandonment_form_status** Type: varchar(20) nullIndexed: **Yes** The form status of the submission before it was abandoned. Generally set only if form_status is "Abandoned". Common values: "Opened", "Saved", "Submitted".
- **abandonment_reason** Type: varchar(2000) nullA reason entered for why the submission was abandoned. Generally set only if form_status is "Abandoned", and only if the abandoning user wanted to provide a reason.
- **abandonment_timestamp** Type: timestamp nullThe date and time the submission was abandoned. Generally set only if form_status is "Abandoned". Please see the next section for tips on date-based filtering.
- **abandonment_type** Type: varchar(4) nullA value indicating by which actor the submission was abandoned. "User" means that the submission user abandoned it, "Administrator" refers to a TM administrator abandoning the submission, and "System" refers to a background job in TM itself abandoning the submission, usually because it was left uncompleted for too long. Generally set only if form_status is "Abandoned".
- **adobe_version** Type: varchar(40) nullThe version number of Adobe Acrobat or Reader (contained in the HTTP request header "acrobat-version") used by the submission user. A typical values would be "11.0.2" or the like. Generally set only if the form is PDF-based rather than a pure HTML form.
- **attachment_count** Type: integer nullThe number of electronic attachments associated with the submission (not counting manual attachments, where the user does not attach a concrete file and instead promises to submit the attachment through channels outside TM). Can be null or a numeric value.
- **attachments_status** Type: varchar(20) nullThe attachment status identifier for the submission. Common values are null (no attachments were ever required by TM), "Completed" if the attachment step was fully completed, "Required" if attachments required by TM are missing and "Optional" if all required attachments are present but the user has not confirmed the set of attachments. The latter two statuses are present only if the form_status is "Abandoned" and are not valid for completed submissions.
- **attachments_total_size** Type: integer nullThe total size in bytes of all electronic attachments associated with the submission. Can be null or a numeric value.
- **authenticated_flag** Type: integer nullIndicates whether the submission user was authenticated (whether as a user account in TM or an external user). If so, the value will be 1; any other value indicates that the user was anonymous.
- **authentication_status** Type: varchar(20) nullThe authentication status value of the submission. This is relevant only if the form requires authentication (a setting in TM), in which case the user must authenticate prior to being able to submit. Possible values are null (authentication was not required), "Completed", "Cancelled" (the user opted to reject the submission rather than proceed) and "Required". The latter two statuses are present only if the form_status is "Abandoned" and are not valid for completed submissions.
- **client_code** Type: varchar(20) not null A short, globally unique string identifying the organization associated with the submission. The code is assigned by TM administrators. Typical values are "doha" or "fincorp".
- **client_name** Type: varchar(100) not nullThe name of the organization associated with the submission. Typical values are "Department of Health" or "FinanceCorp".
- **client_oid** Type: bigint not nullIndexed: **Yes** The database OID of the organization associated with the submission. This is a numeric value. Note that even if organizations are deleted and recreated, records in the transaction history table are unaffected.
- **contact_age** Type: integer nullThe age of the user or their contact as extracted from submitted data. This will be set only if the form data configuration setting "Contact Age XPath" was defined at the time the submission was created. Possible values are null or an integer.

- **contact_gender** Type: char(1) nullThe gender of the user or their contact as extracted from submitted data. This will be set only if the form data configuration setting "Contact Gender XPath" was defined at the time the submission was created. Possible values are null or a single character representing the gender as contained in submitted data.
- **contact_postcode** Type: vachar(20) nullThe postcode of the user or their contact as extracted from submitted data. This will be set only if the form data configuration setting "Contact Postcode XPath" was defined at the time the submission was created. Possible values are null or a the postcode (represented as a string value) as contained in submitted data.
- **content_length** Type: varchar(20) nullThe content length obtained from the HTTP servlet request when the user last saved/submitted the form. The content length is numeric but represented here as a string value.
- **duration_client_response** Type: integer nullThe time (in milliseconds) it took to stream the rendered form to the browser when the user last opened the submission. The value may be null or an integer.
- **duration_render** Type: integer nullThe time (in millicseconds) it took to render the form on the server when the user last opened the submission. The value may be null or an integer.
- **duration_total** Type: integer nullThe total time (in millicseconds) it took to process the form rendering request when the user last opened the submission. The value may be null or an integer.
- **dwh_publish_status** Type: varchar(50) nullThe status value that records whether this transaction history entry has been published to an external data warehouse. This column is not used or written to by TM itself, but if a Groovy-based data warehouse publishing service exists on the TM instance, it may have set a status value.
- **dwh_publish_time** Type: timestamp null The date and time when this transaction history entry was published to an external data warehouse. This column is not used or written to by TM itself, but if a Groovy-based data warehouse publishing service exists on the TM instance, it may have set a value.
- **email_verif_status** Type: varchar(20) nullThe submission email verifcation status. Submission email verification can be enabled for individual forms. If enabled, users must click on a link inside a confirmation email to progress the submission. Possible values are null (email verification was not required for this submission), "Completed", "Required" and "Error" (if an error occurred while sending the confirmation email) and "Abandoned" (if the user has not completed email verification in time). Submissions with a form_status of "Completed" can have email verification status values of null, "Completed" or possibly "Abandoned" (depending on form configuration, the submission may be processed even if email verification has not been completed).
- **field_worker_flag** Type: integer nullIndicates whether the submission was last saved or submitted using TransactField App. If so, the value will be 1; any other value indicates that a different portal was used.
- **form_code** Type: varchar(20) not nullA short, globally unique string identifying the form associated with the submission. The code is assigned by TM administrators. Typical values are "transact-showcase" or "customer-enquiry".
- **form_name** Type: varchar(120) not nullThe name of the form associated with the submission. This is the name that users will see on the portal when filling in forms. Typical values are "Transact Showcase Form" or "Customer Enquiry Form".
- **form_oidType: bigint not nullIndexed: *Yes** The database OID of the form associated with the submission. This is a numeric value. Note that even if forms are deleted and recreated, records in the transaction history table are unaffected.
- **form_status** Type: varchar(20) nullIndexed: **Yes** The status of the rendered form that this submission is based on. Possible values are "Completed" (the user submitted the form and completed all necessary post-submission steps) and "Abandoned" (the user or an administrator explicitly abandoned the submission, or it was abandoned by a TM background job if the user did not complete the form after a configurable amount of time). This column is very important when you need to distinguish between "successful" and "failed" submissions.
- **form_version_number** Type: varchar(10)Indexed: YesThe version number string identifying the version of the form associated with the submission. Each form may have multiple versions with their own configuration, and the version number can be assigned by TM administrators. Typical values are "1.0" or "2.0".
- **ip_address** Type: varchar(50) nullThe IP address of the user when they last saved / submitted the submission. The IP address is obtained from the HTTP request header "X-Forwarded-For" (if present) or the remote address in the HTTP servlet request. Typical values are "11.12.13.14".
- **ln_status** Type: integer null A numeric value used to record various metrics related to TM transaction licensing. This column is not intended for use in custom reports.
- **payment_dr_receipt_no** Type: varchar(20) nullThe payment receipt identifier received from the payment provider. This column is set only if payment was completed for the submission and can be used to reconcile records kept by the payment provider with TM submissions. Typical values vary by payment provider.
- **payment_dr_timestamp** Type: timstamp nullThe date and time when payment was completed for this submission. If payment was not required or not successfully completed, the value will be null.
- **payment_service_code** Type: varchar(20) nullThe unique identifier of the payment provider service in TM. This column is set only if payment was completed for the submission and can be used to identify which payment service was used for this submission. Typical values are "PayPal" or "Westpac".
- **payment_status** Type: varchar(20) nullThe payment status of the submission. Possible values are null (no payment was required), "Completed", "Required" (user has not completed payment) and "Error" (the user attempted but did not successfully complete payment). The latter two status values can occur when form_status is "Abandoned" but are not valid for completed submissions.
- **payment_total** Type: decimal(15,5) nullThe total payment amount. If payment was not required for the submission, the value will be null. A typical value might be 12.58.
- **payment_type** Type: varchar(20) nullThe payment type describes how payment was made. Possible values are null (payment was not required), "Credit/Debit Card", "Hosted" (payment was made via a 3 party payment provider), "BPAY" and "Manual" (an administrator marked the payment as completed). Please refer to the Transaction Manager Administration Guide, chapter "Online Payment" for more information on the different payment types and payment providers supported by TM.
- **pdf_submit_flag** Type: integer nullIndicates whether the submission was PDF based. If so, the value will be 1; any other value indicates that the submission was not PDF based (e.g. HTML form).
- **percentage_completed** Type: integer nullThe percentage to which the form has been completed as an integer value. This value is set only if specified by the form.
- **portal_name** Type: varchar(60) nullIndexed: **Yes** The name of the TM portal on which the submission was last saved/submitted. Typical values are "TransactField App" and "Maguire".
- **receipt_number** Type: varchar(100) nullThe receipt number assigned to the submission. Receipt numbers are generated in TM using several possible approaches (see the Transaction Manager Administration Guide, chapter "Form Configuration", section "Tracking Code and Receipt Number"). Note that the tracking code is not currently present in the transaction_history table.
- **receipt_render_duration** Type: integer nullThe time (in millicseconds) it took to render the receipt for this submission on the server. The value may be null or an integer.
- **referer** Type: varchar(255) nullIndexed: **Yes** The value of the HTTP request header "referer" when the user started the submission.
- **request_cookie** Type: varchar(2000) nullThe value of the HTTP request header "Cookie" when the user last saved/submitted the submission.
- **request_key** Type: varchar(50) nullIndexed: **Yes** A UUID identifying the TM request log entry that was created when the user started the submission. A typical value would be "00b765d4ebf81232f67069a6115047d5".
- **request_url** Type: varchar(1000) nullThe value of the request URL in the HTTP servlet request when the user started the submission.
- **revision_number** Type: integer nullThe submission revision number, which is used for conflict resolution and is incremented every time the submission is saved/submitted. Possible values are null or positive integers.

- **signature_completed_timestamp** Type: timestamp nullThe date and time when the user completed signatures for this submission. This column will be set only if the submission included a signature step, which the user completed.
- **signature_status** Type: varchar(40) nullThe signature status for this submission. Wet ink or digital signatures can be set up in TM as a post-submission step (see the Transaction Manager Administration Guide chapter "Signatures"). Possible values for the status are null (signatures were not required for this submission), "Completed", "Completed - Upload Scanned Copy" (the user printed and signed the receipt, then uploaded it to TM), "Completed - Deliver Manually" (the user downloaded the receipt and will send it back outside of TM), "Required" and "Cancelled". The latter two status values can occur if form_status is "Abandoned" but are not valid for completed submissions.
- **signature_type** Type: varchar(20) nullThe signature type describes what kind of signatures were requested. Possible values are null (signatures were not required), "Wet" (wet ink), "Digital" (in-form digital signatures) and "ESignature" (the receipt is signed with the organization's certificate and presented to the user). Please refer to the Transaction Manager Administration Guide, chapter "Signatures" for more information on the different signature types.
- **submission_expiry_date** Type: timestamp nullThe date and time when the submission would expire unless completed by the user. This column is set only if the form was configured with an expiry time when the submission was started.
- **submission_oid** Type: bigint null The database OID of the submission. This is a numeric value. Note that even if submissions are deleted, records in the transaction history table are unaffected.
- **submit_key** Type: varchar(60) nullA UUID identifying the submission. A typical value would be "00b765d4ebf81232f67069a6115047d5".
- **submitted_offline_flag** Type: integer nullApplies to PDF submissions only. Indicates whether the submission was saved offline and completed after having been reopened in a PDF reader. If so, the value will be 1; any other value indicates that the submission was not submitted offline.
- **task_flag** Type: integer nullIndicates whether the submission is based on a task assigned to a user or group, rather than a user opening a form of their own accord. If so, the value will be 1; any other value indicates that the submission is not based on a task.
- **time_email_verif_completed** Type: timestamp nullThe date and time when submission email verification was completed. This column is set only if submission email verification was required.
- **time_form_completed** Type: timestamp nullThe date and time when all user steps were completed for the submission. Generally this column is set only if the form_status is "Completed".
- **time_request** Type: timestamp nullIndexed: **Yes** The date and time when the user started the submission.
- **time_submission** Type: timestamp nullIndexed: **Yes** The date and time when the user last saved/submitted the submission.
- **time_to_submit_sec** Type: integer nullThe number of seconds the user actually spent inside the form up to the most recent save/submission. May not be set if form_status is "Abandoned".
- **trans_ref_number** Type: varchar(100) nullA transaction reference string as extracted from submitted data. This will be set only if the form data configuration setting "Transaction Ref Number XPath" was defined at the time the submission was created. Possible values are null or the transaction reference as contained in submitted data.
- **transaction_history_oid** Type: bigint not nullIndexed: **Yes** The database OID of the transaction history entry. This is used as the primary key for this table.
- **transaction_score** Type: integer nullA numeric score describing the value of the submission on some kind of scale. This value is not assigned by TM; instead, the form itself can store a transaction value in the submitted XML, and TM will store it. Possible values are null or an integer.
- **transaction_value** Type: decimal nullA number describing the value (monetary or otherwise as defined by the form developer) of the transaction as extracted from submitted data. This will be set only if the form data configuration setting "Transaction Value XPath" was defined at the time the submission was created. Possible values are null or a number describing the transaction value as contained in submitted data.
- **user_agent** Type: varchar(200) nullThe value of the HTTP request header "User-Agent".
- **user_agent_browser_type** Type: varchar(100) nullIndexed: **Yes** The name of the browser used by the submission user. This information is extracted from the user_agent value. Example values are "Mozilla" and "Chrome Mobile".
- **user_agent_browser_version** Type: varchar(100) nullIndexed: **Yes** The major version identifier of the browser used by the submission user. This information is extracted from the user_agent value. Example values are "33" and "Unknown".
- **user_agent_device_type** Type: varchar(20) nullIndexed: **Yes** The device type used by the submission user. This information is extracted from the user_agent value. Possible values are "Desktop", "Tablet", "Phone", "Digital Media Rec.", "Game Console" and "Unknown".
- **user_agent_os_type** Type: varchar(100) nullIndexed: **Yes** The name of the operating system used by the submission user. This information is extracted from the user_agent value. Example values are "Windows" and "Android".
- **user_agent_os_version** Type: varchar(100) nullIndexed: **Yes** The version identifier of the operating system used by the submission user. This information is extracted from the user_agent value. Example values are "10" and "Vista".

Tips and Tricks

Here are some tips to keep in mind when writing reports against the transaction history table.

Filtering by Date

When writing reports, you often need to filter or group by date. There are three key date columns to consider for the transaction_history table:

- time_requestUse this column when you want to filter on the time the submission was initiated. This column applies to all transaction history entries.
- time_submissionUse this column when you want to filter on the time the submission was completed. This column applies to all transaction history entries.
- abandonment_timestampUse this to filter on when submissions with a form_status of "Abandoned" were abandoned. Depending on system settings and how the submission was abandoned, abandonment may happen a long time after the user last worked on the submission.
- time_form_completedUse this column to filter on when submissions (generally with a form_status of "Completed") were completed by the user.

Filtering by Organization/Form

If you want to group transaction history data by form and/or organization, you have a few options:

- client_oid / form_oidThe database ID of the organization or form. This value cannot change until the organization/form is deleted and is, therefore, useful to find all submissions for a specific organization/form, even if its name has changed.
- client_code / form_codeA short globally unique string identifying the organization / form. This value can be edited; therefore, the same organization /form may be represented in the transaction_history table under several client/form codes (but they would all have the same client_oid / form_oid). This column can be used as a display value that is shorter than the organization/form name.
- client_name / form_nameThe name of the organization / form as displayed on TM portals. This value can get fairly long and may need to be truncated for display (or you can use the organization/form code). This value can be edited; therefore, the same organization/form may be represented in the transaction_history table under several client/form names (but they would all have the same client_oid / form_oid).

Note that all the columns above are guaranteed to not contain null values.

Other Filters

There are other columns that are commonly used for filtering, such as:

- form_statusUse this to distinguish between completed and abandoned submissions [Completed | Abandoned].
- authenticated_flagWhether or not the user filled in the form anonymously or was logged in to TM or an external authentication provider.
- abandonment_typeUse this to distinguish how a submission was abandoned [System | User | Administrator].
- field_worker_flagWhether the submission was made on a TransactField App portal.
- task_flagWhether the submission was based on a task assigned to a user or group of users, as opposed to a user starting a submission of their own accord.

Transaction Lookup Table

The transaction lookup table (transaction_lookup) provides a fast way to provide a list of organization names, organization codes, form names or form codes in select lists.
This table is populated when the transaction_history table data is updated. When performing organization or form based filtering of the transaction_history records, use the client_oid and form_oid values from the transaction_lookup table, as these values are indexed in the transaction_history table.

Columns

The transaction_lookup table column definitions are provided below:

- **client_code** Type: varchar(20) not null - A short, globally unique string identifying the organization associated with the submission. The code is assigned by TM administrators. Typical values are "doha" or "fincorp".
- **client_oid** Type: bigint not null - The database OID of the organization associated with the submission. This is a numeric value. Note that even if organizations are deleted and recreated, records in the transaction history table are unaffected.
- **form_code** Type: varchar(20) not null - A short, globally unique string identifying the form associated with the submission. The code is assigned by TM administrators. Typical values are "transact-showcase" or "customer-enquiry".
- **form_name** Type: varchar(120) not null - The name of the form associated with the submission. This is the name that users will see on the portal when filling in forms. Typical values are "Transact Showcase Form" or "Customer Enquiry Form".
- **form_oid** Type: bigint not nullThe database OID of the form associated with the submission. This is a numeric value. Note that even if forms are deleted and recreated, records in the transaction history table are unaffected.
- **lookup_oid** Type: bigint not nullIndexed: **Yes** The database OID of the transaction lookup entry. This is used as the primary key for this table.

Transact Integration Guide (Manager v5.0)

Introduction

This document covers a range of topics around designing and hosting forms using Avoka Transact, such as form prefill, saving and task assignment. In contrast to the TM Administration Guide, this document is structured around common administration tasks and explains how to perform them.
The topics are roughly following the order of the form rendering and submission process where applicable.

Who should read this document?

This document is intended for Avoka Transact administrators and form developers who want to configure various advanced features for their forms. Its main focus is on Transaction Manager, but it also contains some references to Transact Composer.

This document explains these features from a user's perspective and should be read in conjunction with the Transaction Manager Administration Guide, which is more of a reference guide to the management console. Therefore, the Administration Guide will often discuss individual pages in great detail, whereas this document focuses on the steps necessary to complete specific tasks.

Access Control Options (Manager v5.0)

Transaction Manager comes with an extensive security model. Access to the management console is controlled using permission and organization based filtering, whereas access to forms and content on the user space is controlled using groups, permissions, and user account based filtering.
Much of the security model including the full list of permissions in the management console is discussed in the TM Administration Guide.

Can TM use two-factor authentication?

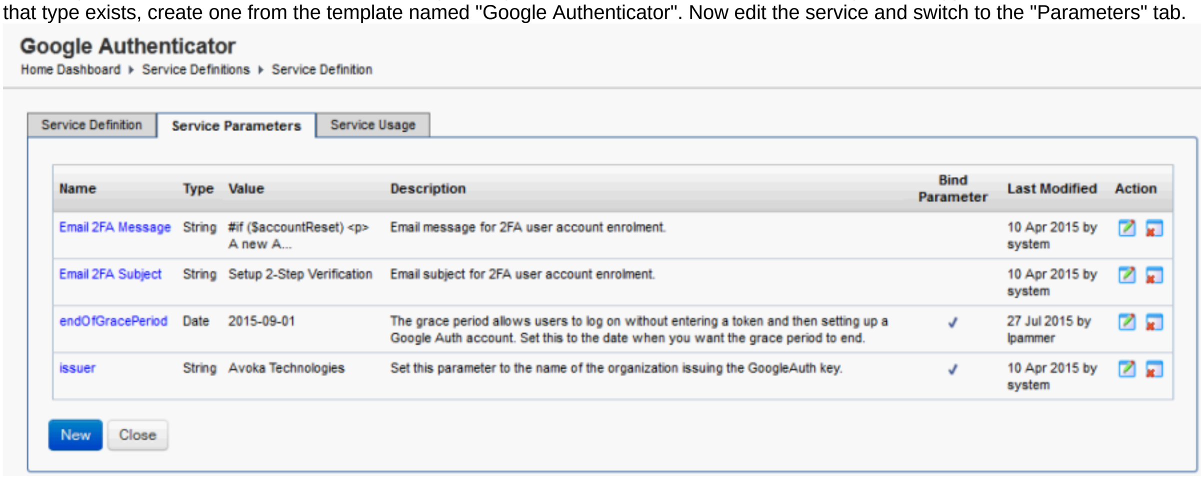
In addition to default password-based authentication, TM can be configured to use two-factor authentication, where the user is asked to provide an additional secret when entering their credentials (e.g. a time-based token). This is much more secure and defends against common attacks on user accounts such as password enumeration. If two-factor authentication is set up, TM will also not increase lockout attempts unless a valid token was presented, which makes it harder for attackers to lock a user account by repeatedly providing invalid credentials.

Two-factor authentication implementations are set up as service definitions of type "Two Factor Authentication" in TM. Out of the box, TM comes with a service definition providing two-factor authentication using the Google Authenticator one-time password service ({+}https://en.wikipedia.org/wiki/Google_Authenticator+).

When two-factor authentication is turned on, users must provide a token generated by the Google Authenticator app in addition to the login name and password. In order to get the token, users have to install the Google Authenticator app and set up a key generated for them on which the tokens will be based. TM provides enrollment functionality for new and existing users, generating keys and emailing out instructions. Administrators can also enroll users and reset their keys if requested. Note that two-factor authentication keys are never exposed to administrators and only sent to the user's email address when needed.

One important issue with two-factor authentication is the cutover period if TM was previously configured to use single-factor (password) authentication. TM supports the setup of a grace period, during which existing users without a two-factor authentication key can log in using their password, and are then able to securely set up multi-factor authentication. Once the grace period ends, users can no longer log in with their password alone; however, administrators can manually trigger the two-factor authentication setup email for them (or the grace period can be extended if need be).

To configure a two-factor authentication service, go to "System > Service Definitions" and filter by service type "Two Factor Authentication". If no service of that type exists, create one from the template named "Google Authenticator". Now edit the service and switch to the "Parameters" tab.



Name	Type	Value	Description	Bind Parameter	Last Modified	Action
Email 2FA Message	String	#if ($accountReset) <p> A new A...	Email message for 2FA user account enrolment.		10 Apr 2015 by system	
Email 2FA Subject	String	Setup 2-Step Verification	Email subject for 2FA user account enrolment.		10 Apr 2015 by system	
endOfGracePeriod	Date	2015-09-01	The grace period allows users to log on without entering a token and then setting up a Google Auth account. Set this to the date when you want the grace period to end.	✓	27 Jul 2015 by lpammer	
issuer	String	Avoka Technologies	Set this parameter to the name of the organization issuing the GoogleAuth key.	✓	10 Apr 2015 by system	

Service parameters for the Google Authenticator service
Adjust the value of the "issuer" parameter (this will denote the organization issuing the Google Authenticator key and will be visible to the user).
Also, consider how long the grace period should be and set the "endOfGracePeriod" parameter to the date when the grace period should end.

You can also customize the contents of the enrolment/reset email sent out to your users. However, take care not to remove key content that the user will need to set up successfully.
Once the service has been configured, you can activate it for certain spaces. This is done by configuring a security manager to use a two-factor authentication service. Go to "Security > Security Managers" and edit the security manager you want to set up with two-factor authentication. Choose the service you configured above from the "2 Factor Authentication" dropdown and save.

Important note: Once a security manager is configured to use a two-factor authentication service, this takes effect immediately and for all spaces associated with the security manager (possibly including the management console). To avoid locking yourself or other users out of the system, always test your service carefully first (possibly using a security manager associated with a single test space) and use a grace period so your users can get set up with two-factor authentication keys. Also, remember that a security manager that is marked as the system default security manager is used by all spaces who are not explicitly set to use a specific security manager.

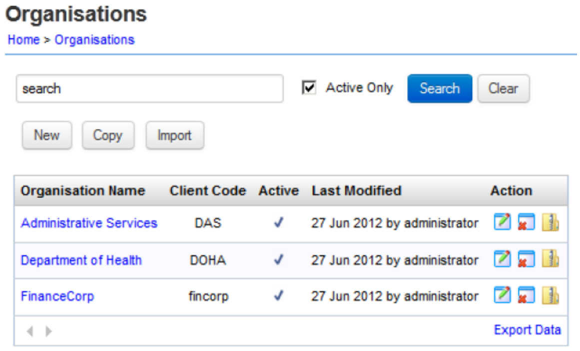
If you would like to disable two-factor authentication, edit the security manager again, blank out the "2 Factor Authentication" dropdown and save. From then on, users can once again log on just using their login name and password.
If two-factor authentication is enabled, administrators can generate/reset authentication keys for users by editing the user and clicking the "Create 2FA Key"/" Reset 2FA Key" button at the bottom. This will trigger a key to be generated for the user and the two-factor authentication email to be sent to the user's email address (exactly as if the user had completed enrolment themselves).

Organizations (Manager v5.0)

An organization is an entity that owns forms. In Transaction Manager, each organization can be configured in a variety of ways, influencing all forms created for the organization. In addition, administrators can be assigned to an organization, causing them to be unable to view or modify data belonging to other organizations.

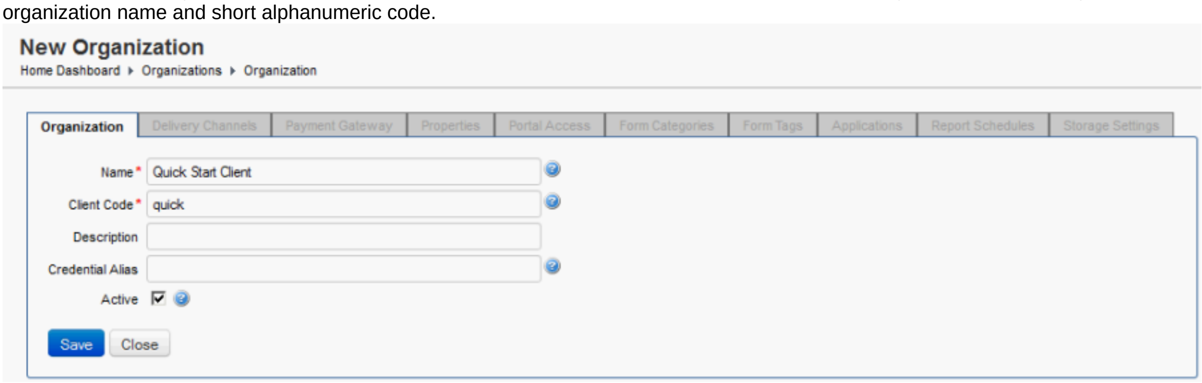
How do I create an organization?

To create an organization in Transaction Manager, use the menu item "Forms > Organizations". The screenshot below shows the page you should be seeing.



The list of organizations set up in TM

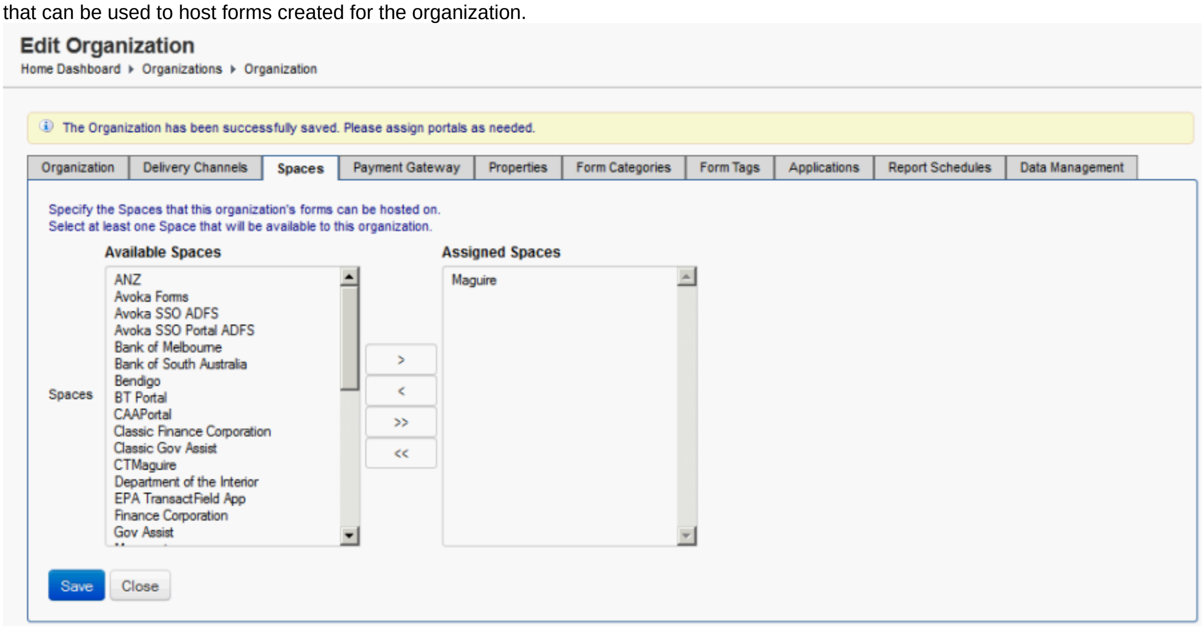
To create a basic organization, click the "New" button above the table. You will see a form into which you can enter the organization's details. Enter a unique organization name and short alphanumeric code.



Creating an organization

Note: You cannot create an organization that has the same name or client code as an existing organization. You can use the "Organizations" page to see which names and codes are already taken.

Save the organization. You will now be taken to the "Spaces" tab. Assign all spaces that should be available to the organization. This will affect the spaces that can be used to host forms created for the organization.



Assigning portals to the organization

Save your changes.

For detailed information about all the options available when configuring an organization, please refer to the Transaction Manager Administration Guide, chapter "Organization Configuration".

Configuring Forms (Manager v5.0)

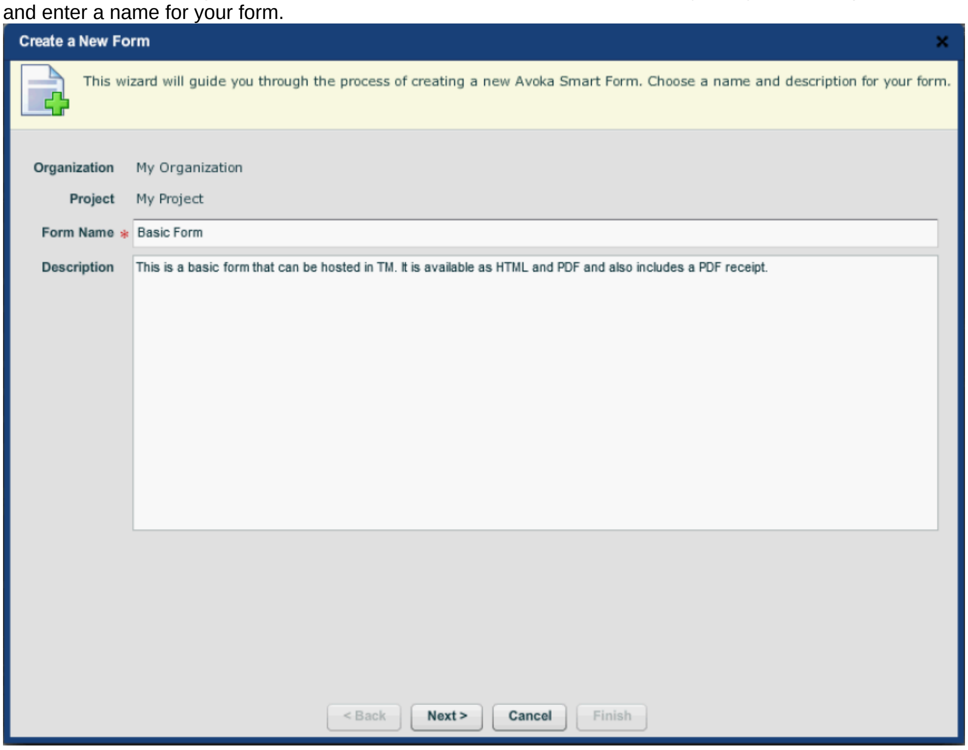
Forms are the central entities in Avoka Transact. This section deals with general topics around form configuration that are not directly related to the user-facing form fill-in flow, such as how to deploy forms to Transaction Manager.

How do I create and deploy a Composer SmartForm?

You can easily publish Composer forms to Transaction Manager, or alternatively export a ZIP file from Composer in case the two servers cannot communicate. The ZIP file can be imported to TM, which will automatically host the form on the space of your choice.
This section will show you how to create a very simple form in Composer, publish it to Transaction Manager and render and submit the hosted form. It assumes you have already created an organization and project in Composer, and an organization in Transaction Manager. This process of creating and hosting a basic form can be done in a matter of minutes.

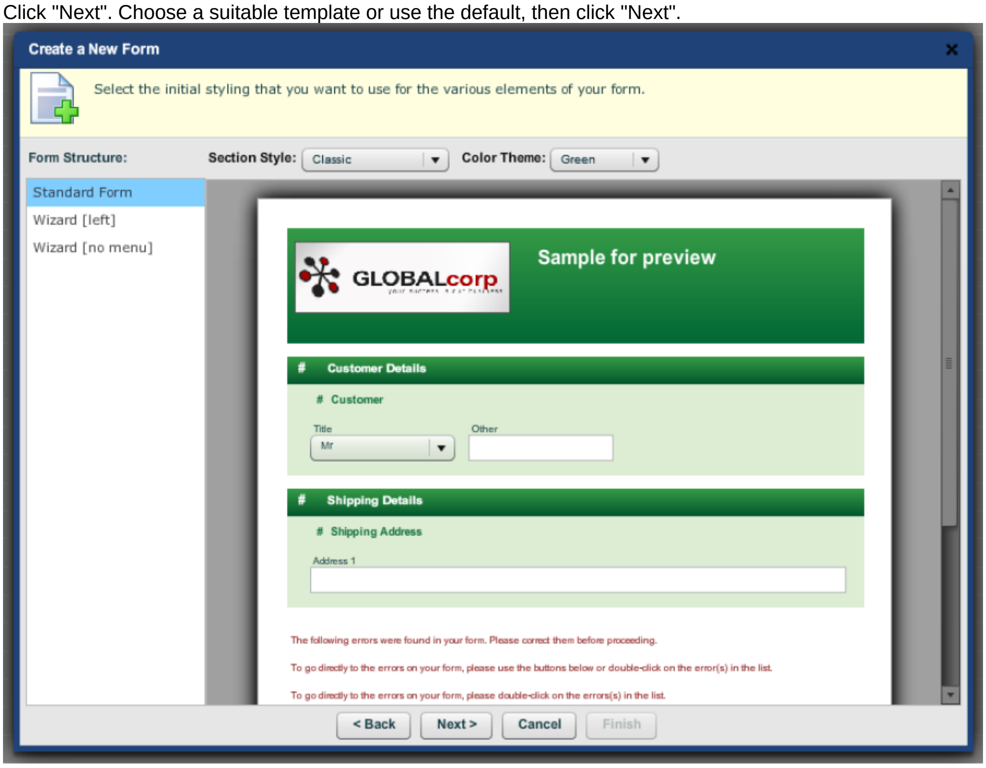
Creating a Composer form

To create a basic Composer SmartForm suitable for use in Transaction Manager, log on to Composer and navigate to your project. Click "Create New Form" and enter a name for your form.



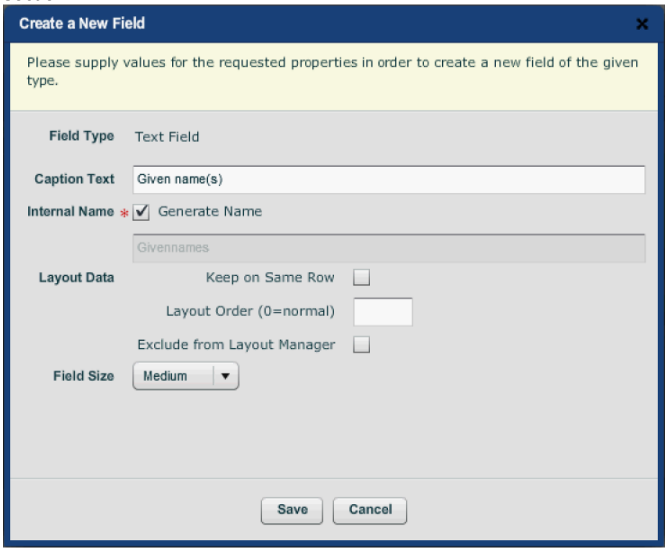
Creating a form in Transact Composer

Click "Next". Choose a suitable template or use the default, then click "Next".



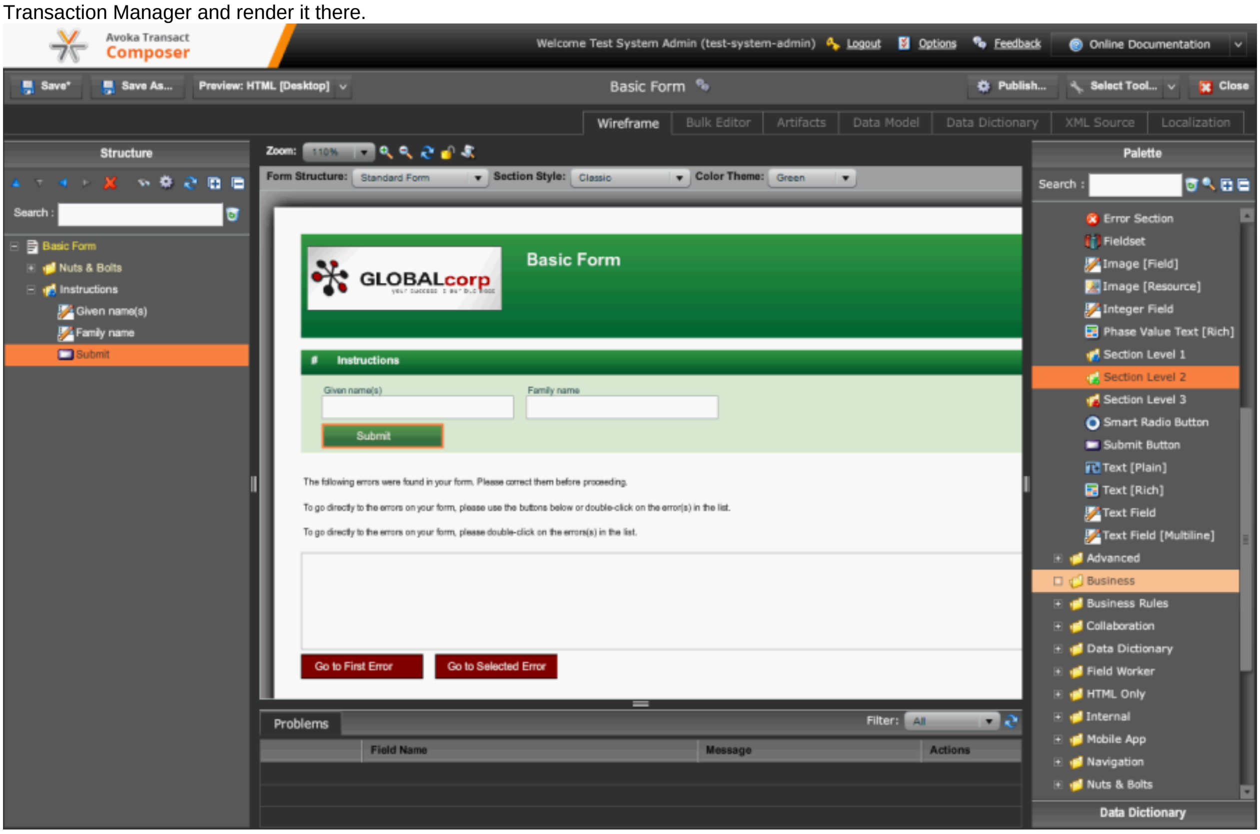
Configuring the styling for the form

We will use a standard form with a green color theme. When you change the settings, the sample will change to reflect your choices. Click "Next" when you are satisfied. Leave the settings on the final screen unchanged and click "Finish" to create the form.
Composer will automatically open the form for you to edit. The form contains a basic logo and a section as well as the standard error section, all of which we could customize. However, for the purposes of this walkthrough, we will simply add a few fields and a submit button.
To add the fields, expand "Field Types > Common" in the palette area. Drag and drop a widget of type "Text Field" into the blue rectangle in the "Instructions" section.



Adding a text field to the form

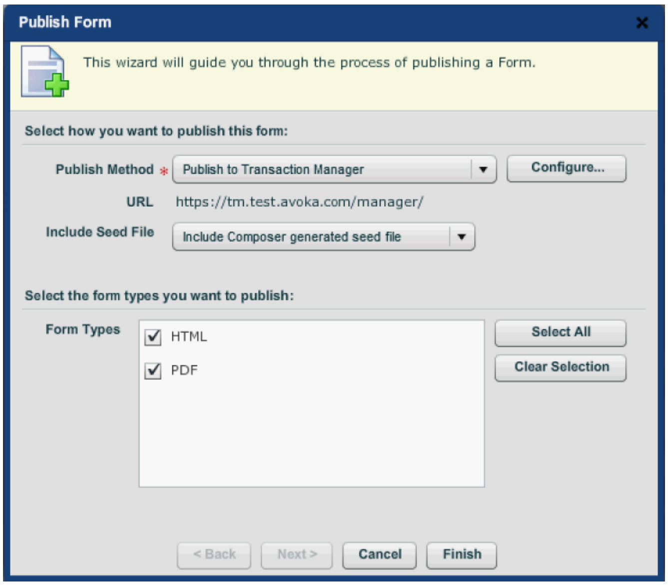
For our basic form, we will add two text fields for given and family names. The screenshot above shows the creation dialog for the first text field. Fill in a label (e.g. "Given name(s)" and save. Drag and drop in another text field to the right of the first field, fill in the label (e.g. "Family name") and tick "Keep on same line", then save.
Finally, drag and drop a "Submit Button" widget to the right of the fields. Accept the default values in the dialog and save.
The screenshot below shows the form we created. You can preview the form in Composer as PDF or HTML, but we will simply go on to publish the form to Transaction Manager and render it there.



The basic form in Composer

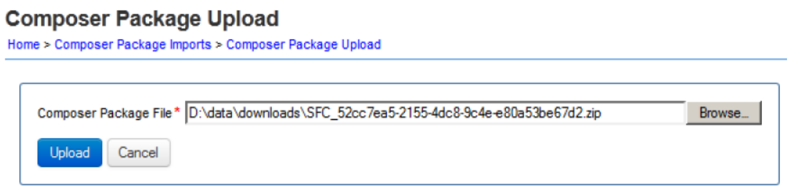
Publishing the form to Transaction Manager

To publish the basic form we just created to Transaction Manager, ensure the form is open in Composer and click the "Publish..." button above the form area. You may be prompted to save changes to your form before the publish dialog is shown.



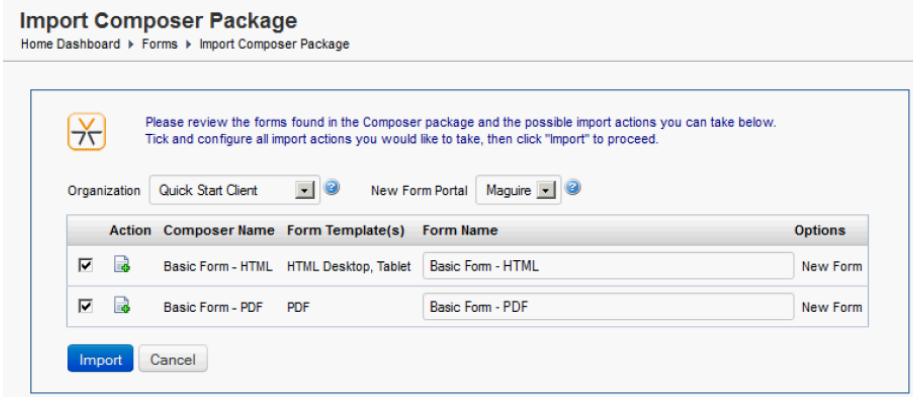
Publishing the basic form

Assuming the Composer and TM servers can communicate with each other, you can use the publish method "Publish to Transaction Manager". If this has been set up for your Composer instance already, choose this method and ensure the Transaction Manager URL matches the target environment.
Otherwise please select "Publish as zip file" as the publish method.
Click "Finish".
If you published the file as a ZIP file, navigate to and log in to the TM management console. Go to "Forms > Composer Package Imports" and click "Import Composer Package".



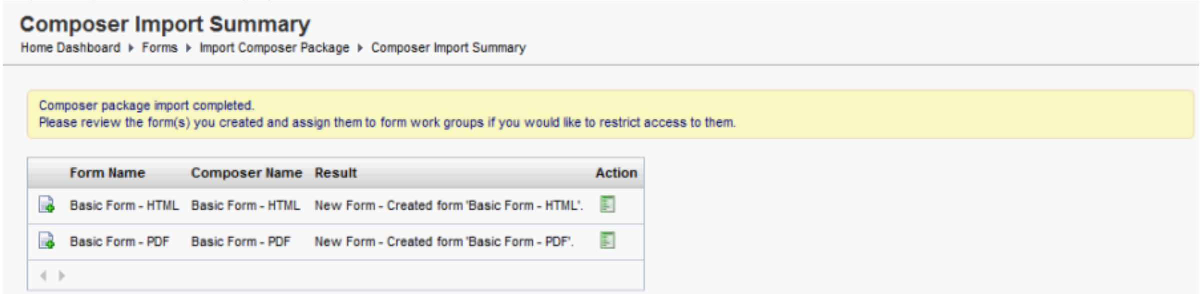
Uploading the ZIP file containing the basic form into Transaction Manager

Click "Upload" to get to the "Import Composer Package" page.
If you published directly from Composer to Transaction Manager, you should arrive at the "Import Composer Package" page after you log in to TM.



Importing the basic form

Please select the target organization in Transaction Manager from the "Organization" dropdown. Transaction Manager will then attempt to find the forms published by Composer, and will allow you to update the existing forms if they are found. In our case, we are publishing for the first time, and so the only option is to create new forms.
As you can see, Composer has published our basic form as two forms. This is the same Composer form in two versions, HTML and PDF (both using a PDF receipt). The checkbox next to each row allows you to select which of the forms you would like to import to Transaction Manager. For this walkthrough, we will import both forms.
Choose a suitable space from "New Form Space" (we are using the "Gov Assist" example space) and click "Import". The import may take a few seconds to complete.
Finally, a summary of the actions taken during import will be shown. If any errors should occur, they will be displayed in the summary as well. If you imported only a single form, the edit page for the form is shown instead.

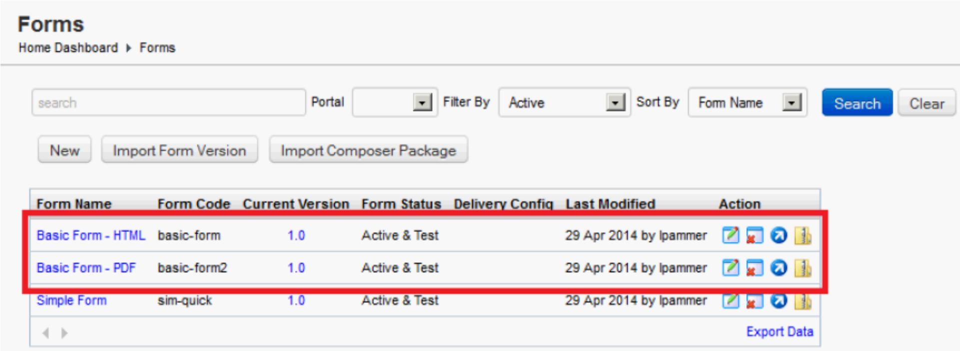


The summary displayed after importing a form

Now that we have successfully deployed the basic form to Transaction Manager, we can test it on the space.

Testing the form

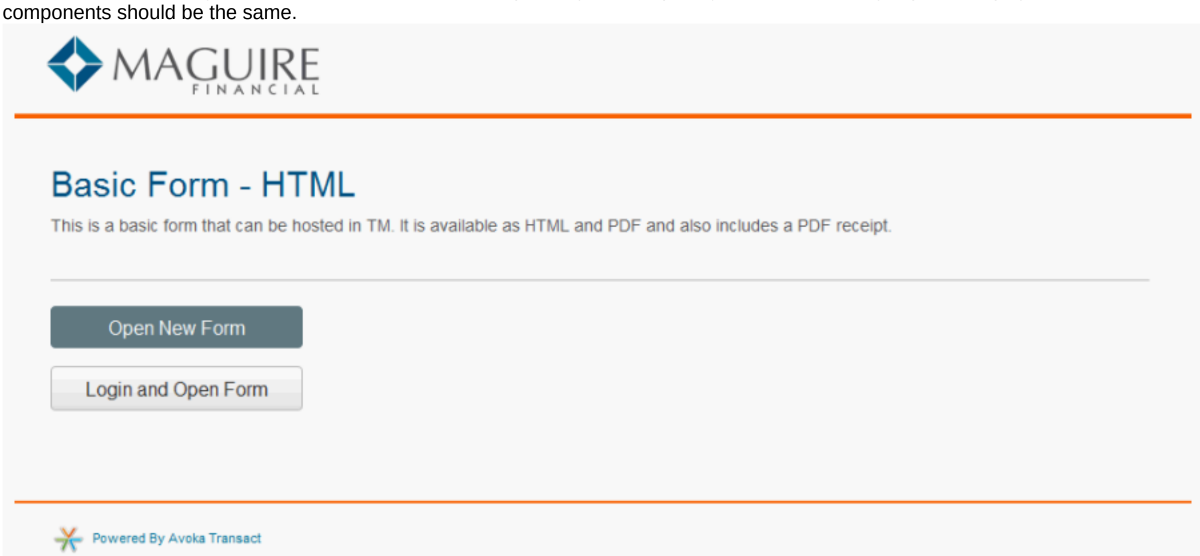
To test the basic form, go to "Forms > Forms" and filter by the organization you have imported the forms to. You should see both the HTML and PDF versions of the basic form.



The basic forms in the TM management console

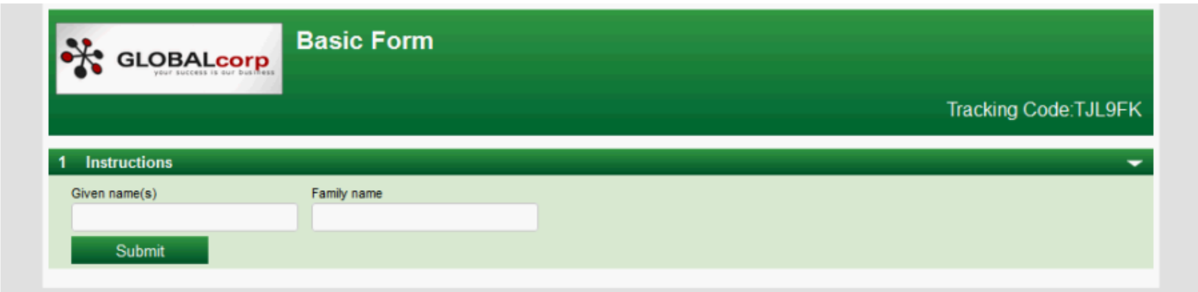
The basic form is now hosted on Transaction Manager and can be filled in and submitted. Note that new forms start out in test mode and hence will not be offered on the form search page on the space. This is intentional to avoid accidentally exposing forms publicly before they are ready. Forms can be changed to production mode easily if desired. For our walkthrough, you can start the fill-in process directly via the TM management console.

Edit one of the forms (we will be using the HTML form here on) and click on the icon under the "Landing Page" heading in the "Form URLs" section. From here on, you will interact with the space and the basic form just like one of your end users would. There is a lot of content you can add to the landing page, but for the moment we just want to test the basic form. Depending on the space you selected during import, this page will look different, but the basic components should be the same.



The landing page for the basic form

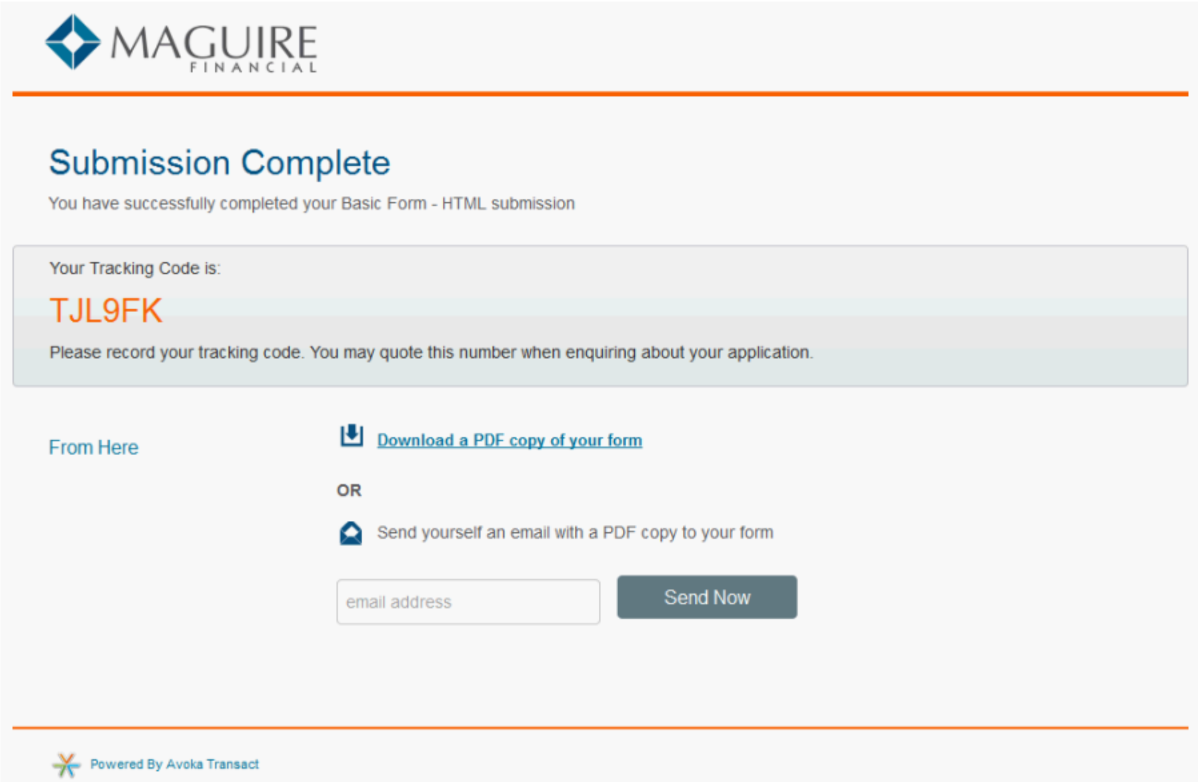
Click "Open New Form" to proceed to the form without logging on as a space user. You will see the basic form with the two text fields and the submit button, rendered as HTML.



The basic form rendered on a TM user portal

Fill in a given and family name, then click "Submit".

Transaction Manager will now process the form submission and display the submission confirmation page, generating a unique receipt number for your reference and offering the receipt to be downloaded or emailed out to the applicant.



The submission confirmation page for the basic form

Have a look at the receipt by clicking "Save a PDF copy of your form". Your browser will download a PDF file containing the receipt (this PDF file was generated automatically by Composer). The given and family names you filled in should be shown, but none of the fields should be editable. The receipt will generally be black and white only so it can be printed easily if needed.

There is much more you could build into and configure for your forms using Avoka Transact. The focus of this walkthrough was simply to show the basics of designing a SmartForm from scratch and hosting it on Transaction Manager. The remainder of this document will provide details on many common questions around form and space customization.

How do I deploy a Designer PDF form?

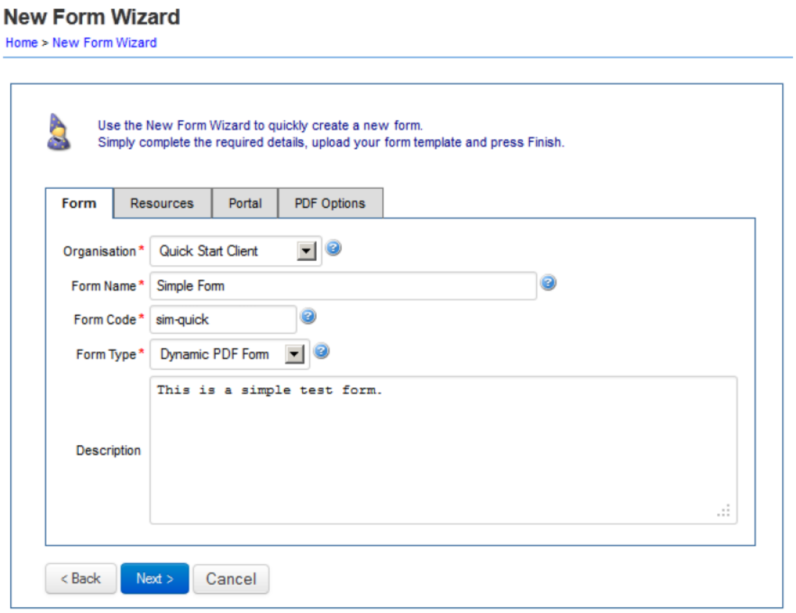
If you have developed an XDP or PDF form using Adobe Designer, you can easily deploy it to Transaction Manager. This section assumes you already have a suitable form in XDP or PDF format.

In addition to the form, you will need an administrator account on the TM management console. The organization that will host the form needs to be set up in Transaction Manager (see "How do I create an organization?").

Deploying the form

The easiest and quickest way to set up a new form is to use the "New Form" wizard provided by Transaction Manager.

To start the wizard, choose the menu item "Forms > New Form Wizard". The screenshot below shows example values.



Creating a new form

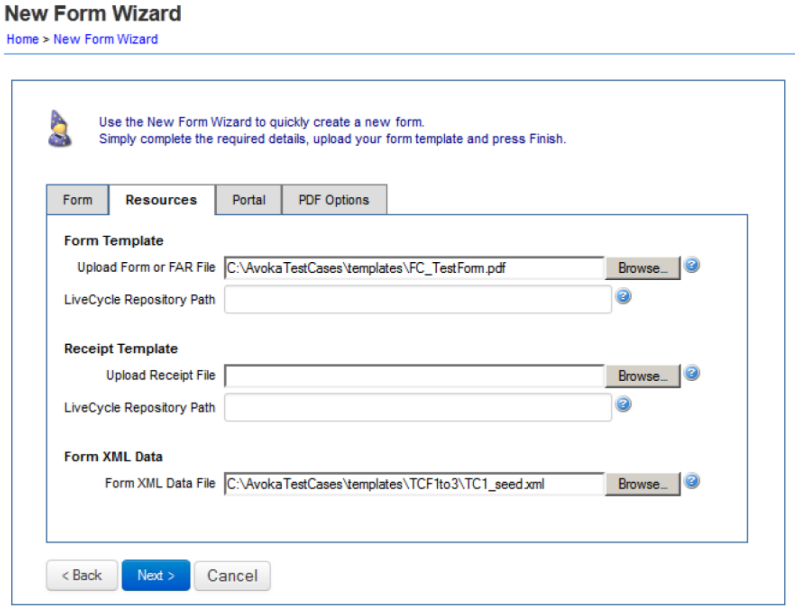
Choose your organization from the "Organization" dropdown, if it is shown (this depends on your assigned permissions).

Enter the full form name as well as a short, unique form code.

Choose "Dynamic PDF Form" from the "Form Type" dropdown.

Optionally enter a description, which can be displayed to space users accessing the form.

Next, switch to the "Resources" tab.

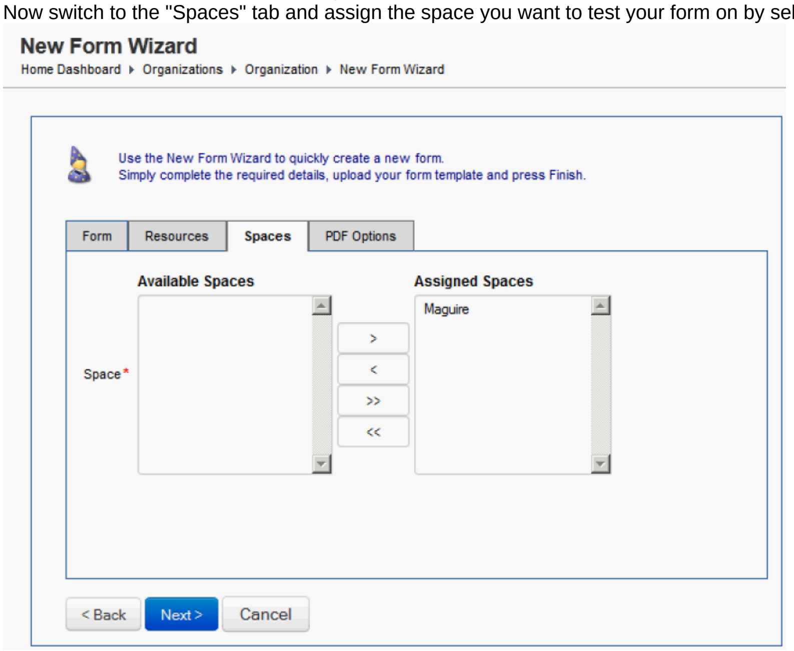


Defining form resources

You will have to upload a file in the "Form Template" and (optionally) "Form XML Data" sections. Upload the XDP or PDF form in the "Form Template" section, and the XML seed file in the "Form XML Data" section. If the form you have chosen does not have an XML data configuration, leave the "Form XML Data" section blank.

Note: Many features in Transaction Manager such as attachments, signatures and payment require that you upload a seed file, so that Transaction Manager can locate and write to or read from special sections in the form data.

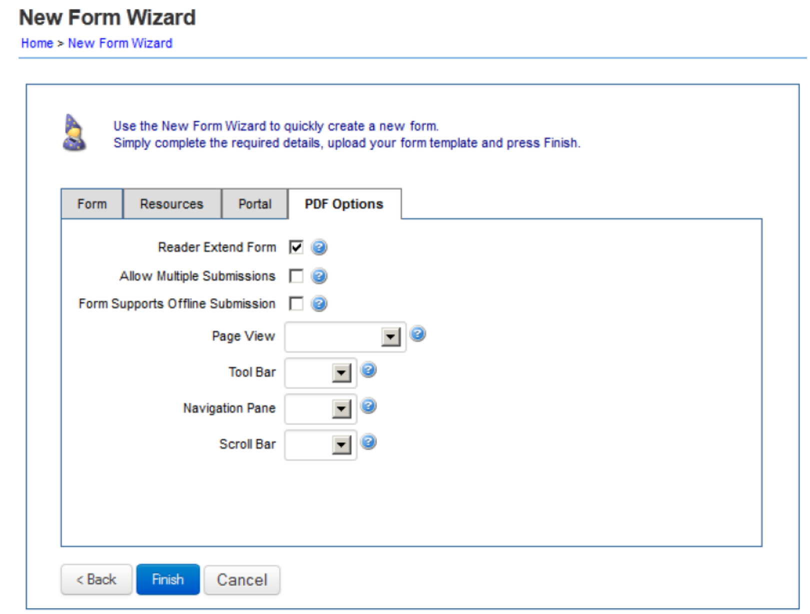
Now switch to the "Spaces" tab and assign the space you want to test your form on by selecting the space and using the ">" button.



Assigning the portal that will host the form

If the form you uploaded does not need to be reader-extended before it can be filled in, or you do not have a valid Reader Extension certificate, change to the "PDF Options" tab and untick the "Reader Extend Form" checkbox. If it is left ticked, make sure that your Transaction Manager system has been set up with the appropriate Reader Extension certificate first. You can find instructions on how to set up a Reader Extension certificate in the Transaction Manager Installation Guide.

You can also modify how the form is displayed in the PDF viewer by using the dropdown lists.

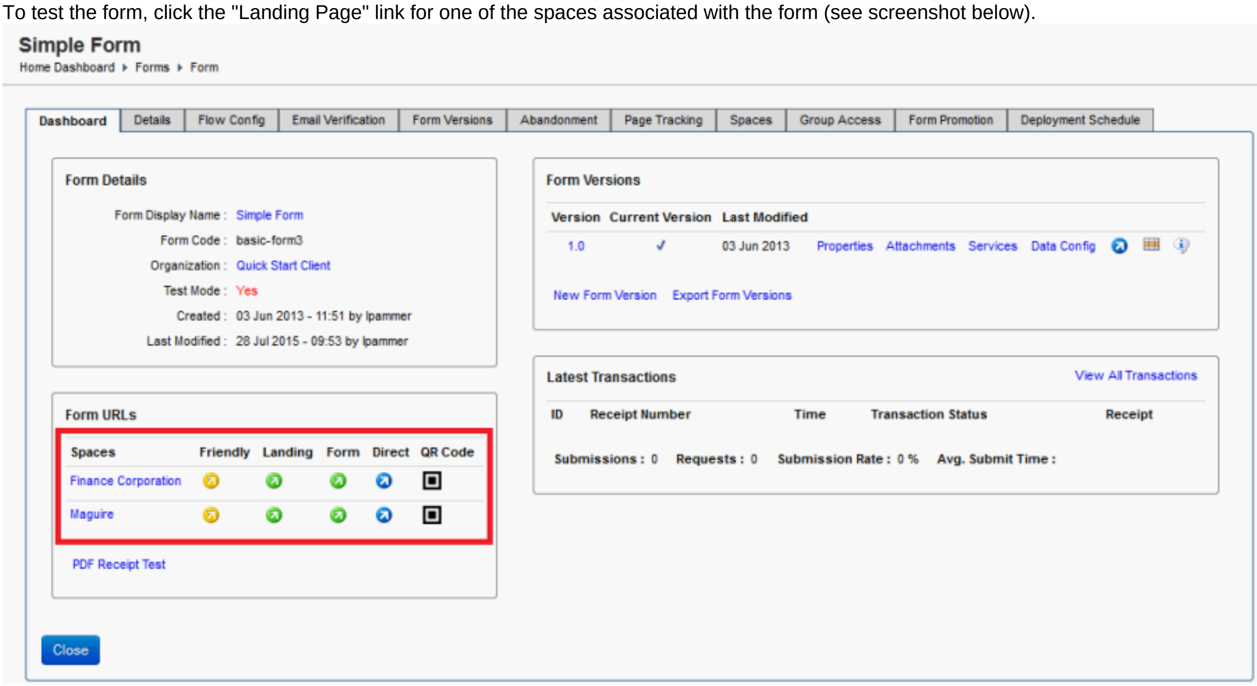


Configuring PDF options

Save the new form by clicking the "Finish" button. It is now ready for you to test.

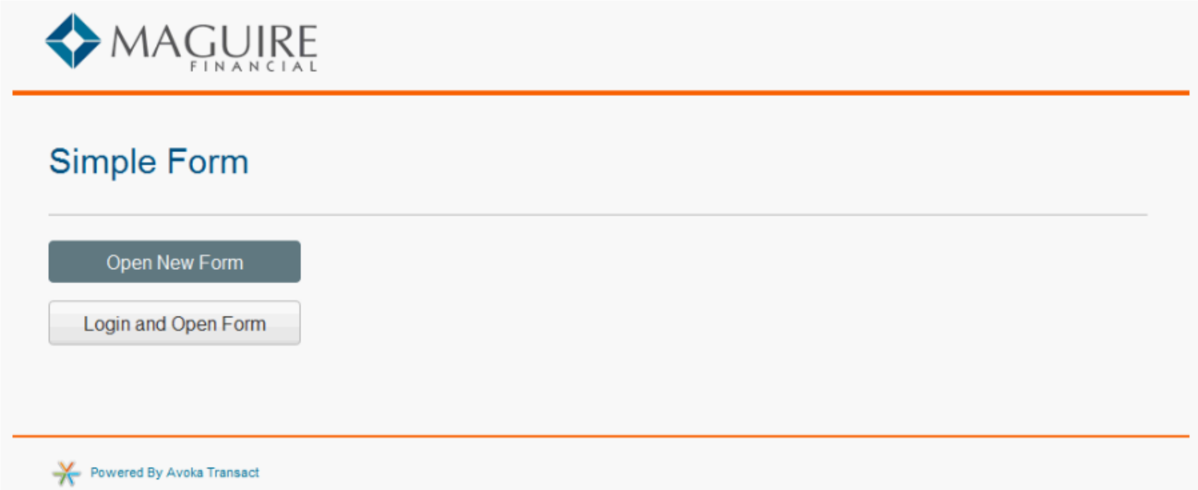
Testing the form

After setting up a form, you should test it to make sure it can be displayed, filled in and submitted without problems.
To test the form, click the "Landing Page" link for one of the spaces associated with the form (see screenshot below).



Rendering a new form

If you have already navigated away from that page, go to "Forms > Forms", find your new form in the list and edit it. **Note:** Transaction Manager will split search results over several pages, so use the "Next" link at the bottom if you cannot see your form on the first page.
When you access the landing page, a new browser window will be displayed containing the landing page for the form. This page is actually part of the Transaction Manager user space. From now on until after you submit the form, you will interact with the space just like your users would.
If your user space is configured to allow account self-registration, the landing page will look similar to the one shown in the screenshot below (which is showing one of the example spaces). While the landing page looks very simple at this point, you can add a wealth of content (description, instructions for the user, organization logo etc.) using TM.

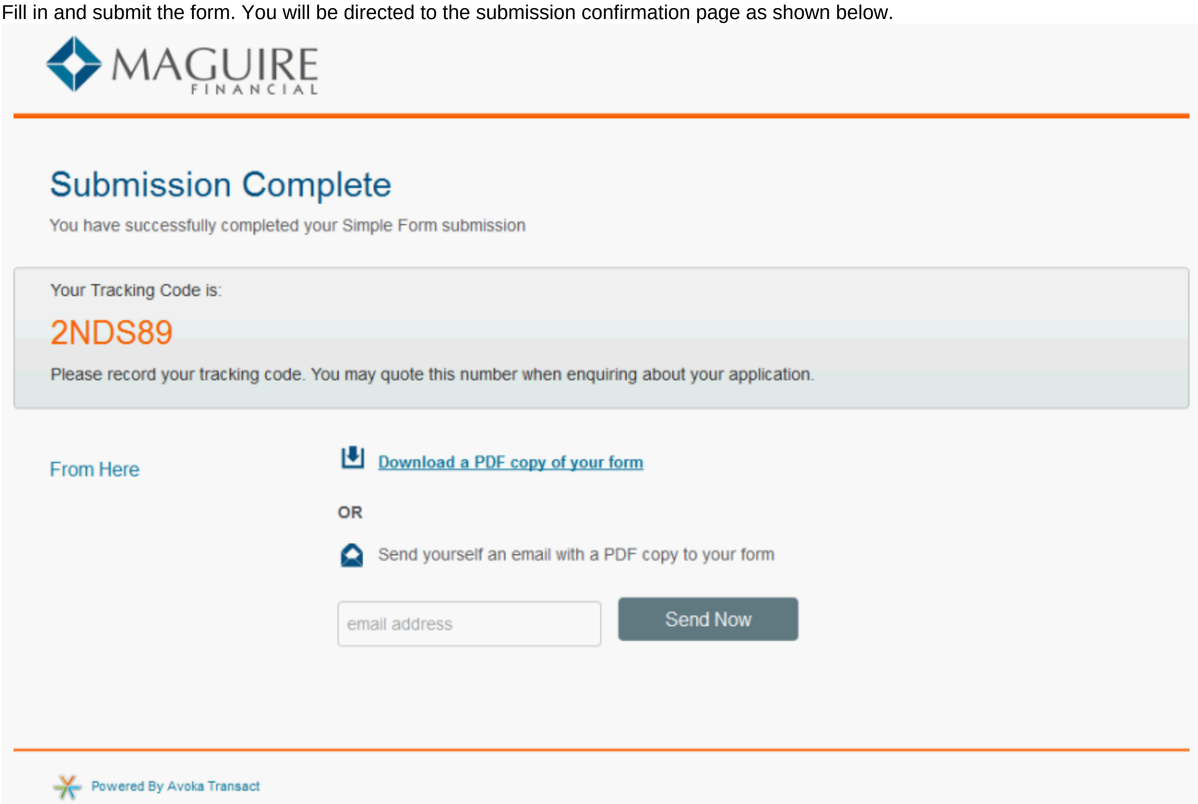


The landing page for your new form

On the TM example spaces, you can register a new user space account or log in if you already have one. If you want to test the form anonymously, simply select the option "Open New Form".

The form should load up inside the same browser window and behave as designed. If there are problems displaying the forms, ensure you have Adobe Reader 8 or higher installed on your system and that it is configured to display PDF files in your web browser.

Fill in and submit the form. You will be directed to the submission confirmation page as shown below.



The submission confirmation page

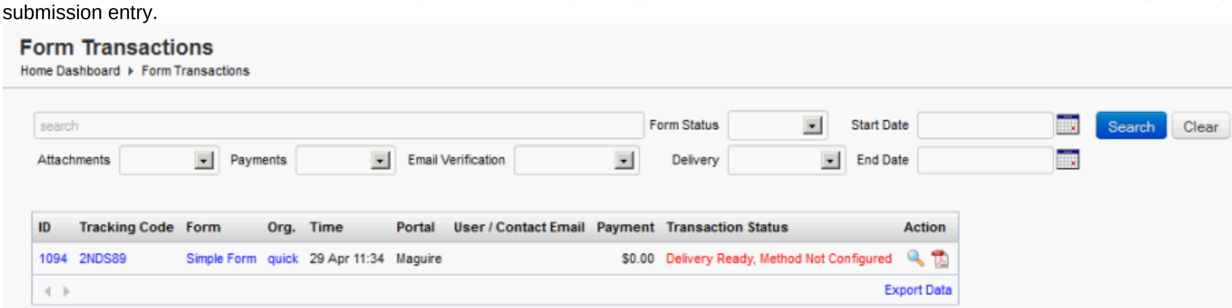
Now that you have rendered and submitted the form, you can check on the submission to make sure everything worked correctly.

You can email yourself a copy of the form receipt by entering your email address on the confirmation page and clicking the "Send email" button (or equivalent). For this to work, Transaction Manager has to have been configured for SMTP email delivery. You should get the email within a few minutes. It will include the receipt in PDF format.

Checking the submitted form

To see whether Transaction Manager has correctly processed the form submission, a good first step is to render the form receipt. To do this, simply open the PDF file included in your receipt email or click the "Save a PDF copy of your form" link on the confirmation page.

Alternatively, you can switch back to the management console, go to "Operations > Form Transactions" and click the "View Receipt" icon ( ) next to the submission entry.



Inspecting the submission log

The receipt should render just as the form did; however, all fields will be read-only in the receipt, and dynamic behavior will be disabled (e.g. you should not be able to submit the form again).

After inspecting the receipt, look at the submission log entry as well (refresh the page if you had navigated to it before). The "Transaction Status" column should indicate that the submission is ready for delivery but that no delivery method has been configured.

User Spaces (Manager v5.0)

Transaction Manager offers a variety of options to expose forms to your target audience. You can use the TM self-service space, which provides a complete solution for end-to-end form fill-in. If you already have a web portal into which you would like to integrate form fill-in, you can easily add calls to the Web Plugin module to render forms. If you wish to provide your users with sophisticated form lookup as well as offering existing resources, you can use the Finder module to provide custom form and resource finding.

What kind of user space should I use?

Transaction Manager can expose multiple user spaces with a different set of forms on each one. This means you can combine one or more self-service spaces with the Finder and/or the Web Plugin modules, potentially for different target audiences and with different branding.
Generally, if you already have your own branded web portal and want to embed forms inside it without styling your own self-service space, you can use the Web Plugin.

If you want to take advantage of the stand-alone self-service space, you can change the styling and page content to suit your needs. The self-service space provides the full set of form-related flows, including authentication against the Transaction Manager user base (or external directories) and a personalized secure area showing the user's submission history, to-do list and available forms.

How can I use LDAP to authenticate users?

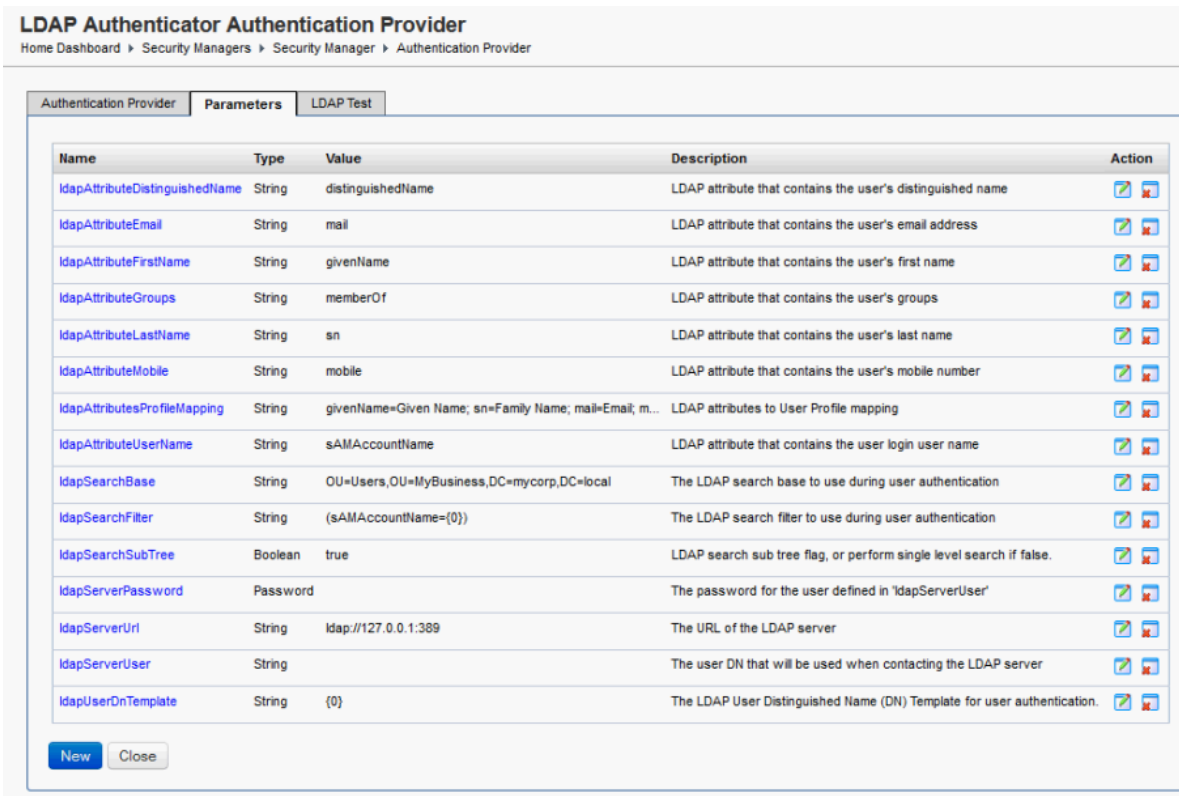
In TM, authentication is handled by a security managers. Security managers are configurable entities associated with one or more spaces, and also handle user enrollment and security-related functionality such as automated lockout and password constraints.
Security managers authenticate either against the TM user database (local authentication) or against an externally maintained user base (delegated authentication, e.g. LDAP or SSO). Delegating security managers can have multiple authentication providers, which handle the authentication process itself (non-delegating security managers would have a single authentication provider operating against the TM user base).

Authentication providers are called sequentially until one of them is able to authenticate the user. There are pre-defined authentication provider templates including authentication against the TM user base and authentication against an LDAP directory. You can configure your security manager to use one or more authentication providers and define the order in which they are to be called.

TM already provides an LDAP-based security manager named "LDAP Security Manager", which you can configure for your LDAP directory.

Configuring LDAP authentication parameters

Once you have an LDAP-based security manager, you need to configure it such that it can connect to your specific LDAP directory. To do so, edit your security manager and switch to the "Authentication Providers" tab. Edit the LDAP authentication provider and switch to the "Parameters" tab (see the image below).


LDAP Authenticator Authentication Provider
Home Dashboard ▸ Security Managers ▸ Security Manager ▸ Authentication Provider
Authentication Provider | Parameters | LDAP Test

Name	Type	Value	Description	Action
ldapAttributeDistinguishedName	String	distinguishedName	LDAP attribute that contains the user's distinguished name	
ldapAttributeEmail	String	mail	LDAP attribute that contains the user's email address	
ldapAttributeFirstName	String	givenName	LDAP attribute that contains the user's first name	
ldapAttributeGroups	String	memberOf	LDAP attribute that contains the user's groups	
ldapAttributeLastName	String	sn	LDAP attribute that contains the user's last name	
ldapAttributeMobile	String	mobile	LDAP attribute that contains the user's mobile number	
ldapAttributesProfileMapping	String	givenName=Given Name; sn=Family Name; mail=Email; m...	LDAP attributes to User Profile mapping	
ldapAttributeUserName	String	sAMAccountName	LDAP attribute that contains the user login user name	
ldapSearchBase	String	OU=Users,OU=MyBusiness,DC=mycorp,DC=local	The LDAP search base to use during user authentication	
ldapSearchFilter	String	(sAMAccountName={0})	The LDAP search filter to use during user authentication	
ldapSearchSubTree	Boolean	true	LDAP search sub tree flag, or perform single level search if false.	
ldapServerPassword	Password		The password for the user defined in 'ldapServerUser'	
ldapServerUrl	String	ldap://127.0.0.1:389	The URL of the LDAP server	
ldapServerUser	String		The user DN that will be used when contacting the LDAP server	
ldapUserDnTemplate	String	{0}	The LDAP User Distinguished Name (DN) Template for user authentication.	

New Close

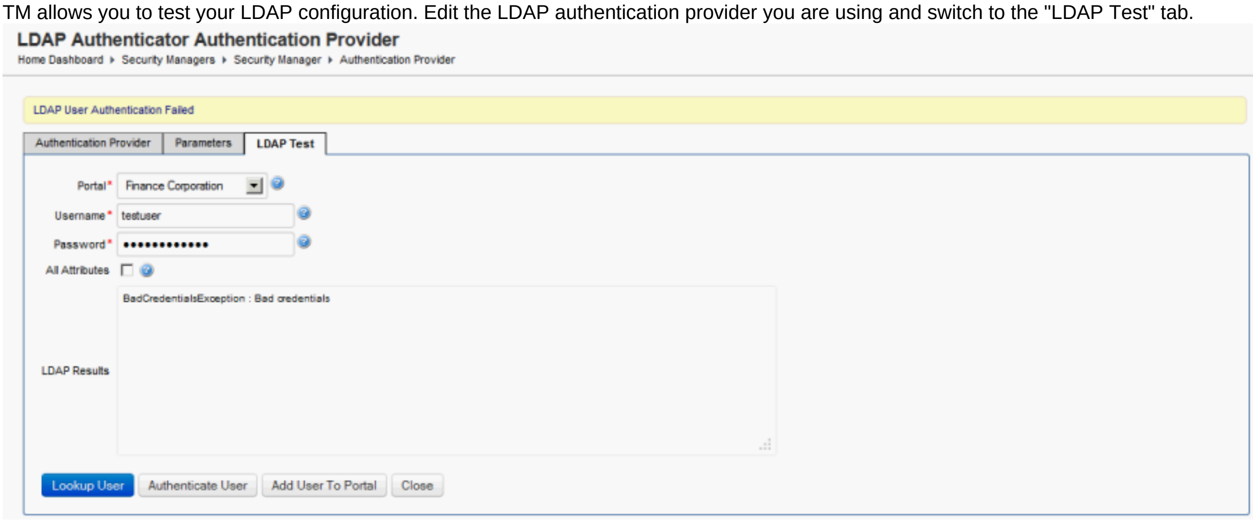
The list of LDAP authentication parameters
You will need to adjust the parameters to suit the LDAP directory you are using. Apart from connection parameters, there are also several parameters named "ldapAttribute*". These parameters are used to load information about the user from the LDAP directory. Basic attributes are user name, first name, last name, email address and mobile phone number. The set of groups a user belongs to can also be retrieved by using the "ldapAttributeGroups" attribute. These groups are used by TM to determine form access, in addition to group membership in TM.

In addition, you can retrieve custom attributes and store them in the user profile by using the "ldapAttriutesProfileMapping". The format you need to follow is:
[LDAP attribute name 1]=[user property name 1]; [LDAP attribute name 2]=[user property name 2]
For example country=Nationality; ssn=Social Security Number

When the user is authenticated, these attributes will be retrieved and mapped into the user record maintained in TM. If you map values into the user profile, these values can be prefilled into forms.
Always make sure to test LDAP authentication after making any changes. TM provides a test harness you can use to test the current configuration (see the following section).

Testing LDAP authentication

TM allows you to test your LDAP configuration. Edit the LDAP authentication provider you are using and switch to the "LDAP Test" tab.



Testing LDAP authentication
Select the space you want to test against and enter the user name of the user.
You can either look up a user, authenticate a user or add a user to the space. Depending on what you want to do, you'll need to fill in different fields.
You need to enter the user's password if testing user authentication.
You can tick "All Attributes" if you want to do a lookup and would like to see all user attributes, not just the ones for which you've defined mappings.
Finally, click the appropriate button to perform the test. The authentication provider will contact the LDAP directory using the currently configured parameters and perform the desired operations. The results will be displayed in the "LDAP Results" field. If the invocation is not successful, you may have to adjust the LDAP parameters as described in the previous section and try again.

Using LDAP groups in TM

In TM, form work groups can be defined to control access to forms in a sophisticated manner (for example, you can define a form work group whose users can see submissions completed by other users in the group). Users must belong to a form work group to gain access to the associated forms. For local users, this is achieved by assigning the users to the form work group via the TM management console. While this is also possible for externally managed users (e.g. LDAP), it is impractical because LDAP user accounts for external users are created on the fly, i.e. when they first log in.
To enable you to maintain group associations in the LDAP directory (or re-use existing LDAP groups for form access purposes), TM can load the groups a user belongs to from the LDAP directory at login time. Whenever the user subsequently attempts to access a form belonging to a form work group, the groups loaded from the LDAP directory are searched and access is granted if a group with the same name (after processing) is found.

The way TM loads the group associations from the LDAP directory and parses out group names is as follows:
The LDAP authentication provider must have the "ldapAttributeGroups" parameter set to the name of an attribute (e.g. "memberOf"). When the user record is retrieved from LDAP, all values for this attribute are loaded and each of them is parsed for a group name as follows: Simple strings (e.g. "mygroup" or "Human Resources") are left as is, and will be compared to group names in TM directly. For distinguished names (DNs), only the first element (usually the CN) is used; for example, if the group name in LDAP is "CN=mygroup,OU=Distribution Groups,OU=MyBusiness,DC=mycompany", the resulting group name after parsing would be "mygroup".

Can I integrate TM with my existing SSO system?

Transaction Manager forms are often integrated with an existing web site or web application. In many cases, a user logs into the existing web application, and it is undesirable to force them to log in a second time into Transaction Manager when they need to fill in a form. To avoid signing in twice, a technique known as Single Sign On can be used. The idea is that once a user has logged in to one application, they are "pre-authenticated" for all applications, including Transaction Manager.

Transaction Manager allows you to integrate with your SSO system of choice by providing a customizable SSO security manager. You can customize its authentication logic via Groovy scripts.

Configuring an SSO security manager

TM includes a configurable SSO security manager named "SSO Security Manager". To customize it, go to "System > Security Managers" and edit the SSO security manager.
You can provide custom Groovy scripts for three extension points:

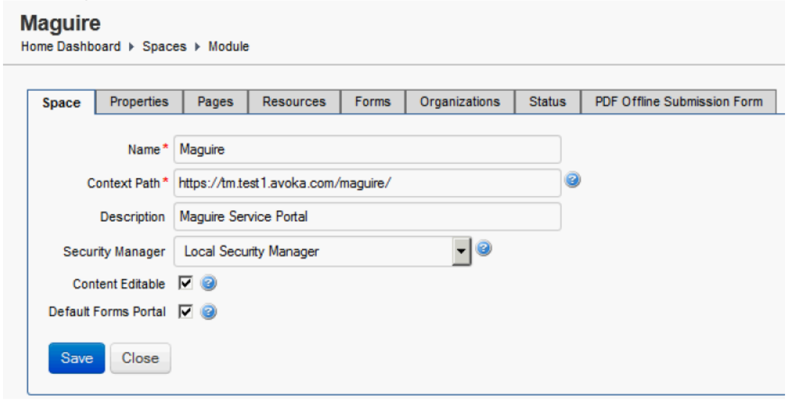
- Get SSO authentication token ("SSO Auth Filter" tab) This script creates or loads an SSO authentication token from the user request. If the user is not yet authenticated, it can elect to redirect the user to an external login site.
- Authenticate (authentication provider parameter) The SSO authentication provider performs authentication based on the SSO token. The scripts can throw various exceptions to communicate to TM that login was unsuccessful.

- Authentication OK response ("SSO Auth Filter" tab) After authentication has succeeded, another script is called to perform optional processing steps.

Please refer to the TM Groovy services guide ("System > Groovy Services Guide") for more information on these extension points.

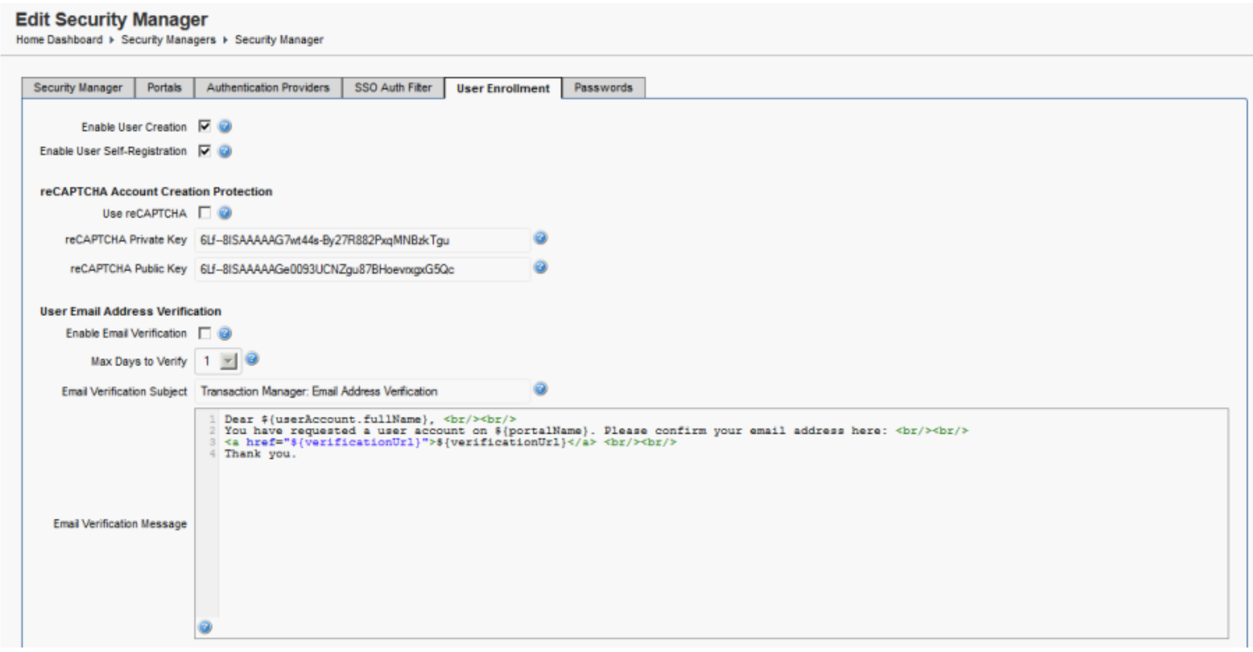
How can I verify email addresses for new users?

An email verification step is often part of the user registration step for secure systems. Transaction Manager supports email verification for self-registered user accounts. If configured, users will be directed to a separate page after submission, notifying them that they need to click on a link in an email sent to them.
To enable email verification for a space, you need to reconfigure the security manager used by the space. The user enrollment service also contains a number of other useful settings around user registration.
Important note: A security manager may be used by multiple spaces. When you make changes to the configuration, make sure you understand which spaces this will affect to avoid unintentional side effects.
To find out which security manager is associated with your space, go to "System > Modules" or "Forms > Form Spaces" in the TM management console and edit the space.



Finding out the security manager for a space
The security manager is displayed in the "Security Manager" dropdown. If nothing is selected, the space is using the system default security manager.

To modify the email verification settings, go to "Security > Security Managers". Find the security manager configured for the space (or if none was selected, the security manager with a tick in the "Default" column) and edit it. Switch to the "User Enrolment" tab.
Section "User Email Address Verification" related to email verification (see image below). **Note:** This section is enabled only if "Enable User Creation" is ticked.



Email verification settings in the security manager
Tick "Enable Email Verification" to enable an email verification step at account registration time.

Choose the time in days the user has to complete email verification from the "Max Days to Verify" dropdown. If the user does not complete email verification within that period, their account may be purged by TM.
Finally, you can configure the verification email that will be sent to the user. You can edit the "Email Verification Subject" and "Email Verification Message" fields to customize the email.
The Velocity variables you can use in the email subject and body are as follows:

- userAccount Has five properties describing the user: "loginName", the login name of the new user account; "fullName", the full name of the user; "givenName", the given name of the user; "familyName", the family name of the user; "email", the email address of the user.
- verificationUrl The URL the user will have to access to complete email verification. Make sure this is contained in the email body; otherwise users won't be able to pass email verification.
- portalName The name of the user space that the user self-registered on.
- portalHref The base URL of the user space that the user self-registered on.

Values for these variables are automatically provided by TM. An example email body template might be:
"Dear ${userAccount.fullName},

You have requested a user account on ${portalName}. Please confirm your email address here:

${verificationUrl}

Thank you."
Once you have configured email verification, test user enrollment by registering a user on the user space(s) you modified.

How do I include a form in my web CMS using the Web Plugin?

After form submission, how do I redirect a user to a different page in my web CMS?

How do I prefill user details in a form on the Web Plugin?

Prefill (Manager v5.0)

Prefilling a form, usually with information that can be obtained about the user, is a highly desirable feature for most organizations. It reduces the effort required by the user at the same time as improving data quality. There are a number of ways that prefill can be implemented.

What prefill options does TM support?

Form prefill is a crucial capability for SmartForms, and there are many possible ways to obtain prefill data (web services, HTTP request parameters and many others). To give you maximum flexibility, Transaction Manager supports a wide range of prefill options.
This section gives an overview of the prefill capabilities provided in Transaction Manager. The next section talks about how prefill happens and how multiple ways to obtain prefill data might be combined.
There are 4 different prefill methods supported in TM:

- Seed file This is the simplest form of prefill - it is really just directly modifying the seed XML file. Usually this is used only for simple testing purposes.
- Property prefill This maps a property value defined in TM into a field on the form. TM properties can be maintained by TM administrators (organization and form version properties) and space users themselves (user profile properties). Examples would be an interest rate table which needs to be updated on a weekly basis without changing the form design, or information about the user that is being maintained in their TM user profile.
- Request parameter prefill This approach can be used when the form is invoked from an external web site, passing through some information in the URL that should be displayed in the form. A Transaction Manager administrator would configure the mapping from the URL parameter to a field in the form. Apart from request parameters, session attributes and cookie values can also be used to pass prefill data. Example: A URL that contains the user's client number.
- Service-based prefill This is the most flexible prefill method where a TM service definition is used to obtain prefill XML data, which is then mapped into the form. Depending on the service implementation, prefill data can be obtained from external services (e.g. web services). By default, TM provides three implementations: a web service prefill service, a prefill service calling an Adobe LiveCycle process and a Groovy prefill service, which is based on a Groovy script that can be adjusted at runtime. Example: The client-number passed in the Request Parameter is passed as a parameter to a web service that has been exposed in the organization's CRM system. This web service returns an XML document that represents information about the client with that client-number. This data is mapped to fields in the form.

The following sections describe how these prefill methods are applied when the form is rendered, and talk about how to configure each method in greater detail.

What prefill options are available on the TransactField App?

Forms rendered on the TransactField App can mostly be prefilled like other forms, but there are some restrictions and caveats as follows:
Because TransactField App forms are filled in using a dedicated client application instead of as part of a browser session, request parameter prefill is not available, since there is no referring page to set these values.
In addition, forms synchronized down to the TransactField App receive a seed file that has most forms of prefill already applied to it (this is necessary because the TransactField App client, unlike other user spaces, does not have access to Transaction Manager entities such as property values and service definitions). This means that property values are written to the seed file at synchronization time as opposed to at render time, and service-based prefill is also done at synchronization time. User property prefill is done by the TransactField App at render time, using the user profile stored in the TransactField App (which is also synchronized with TM).

How does the prefill and rendering pipeline work?

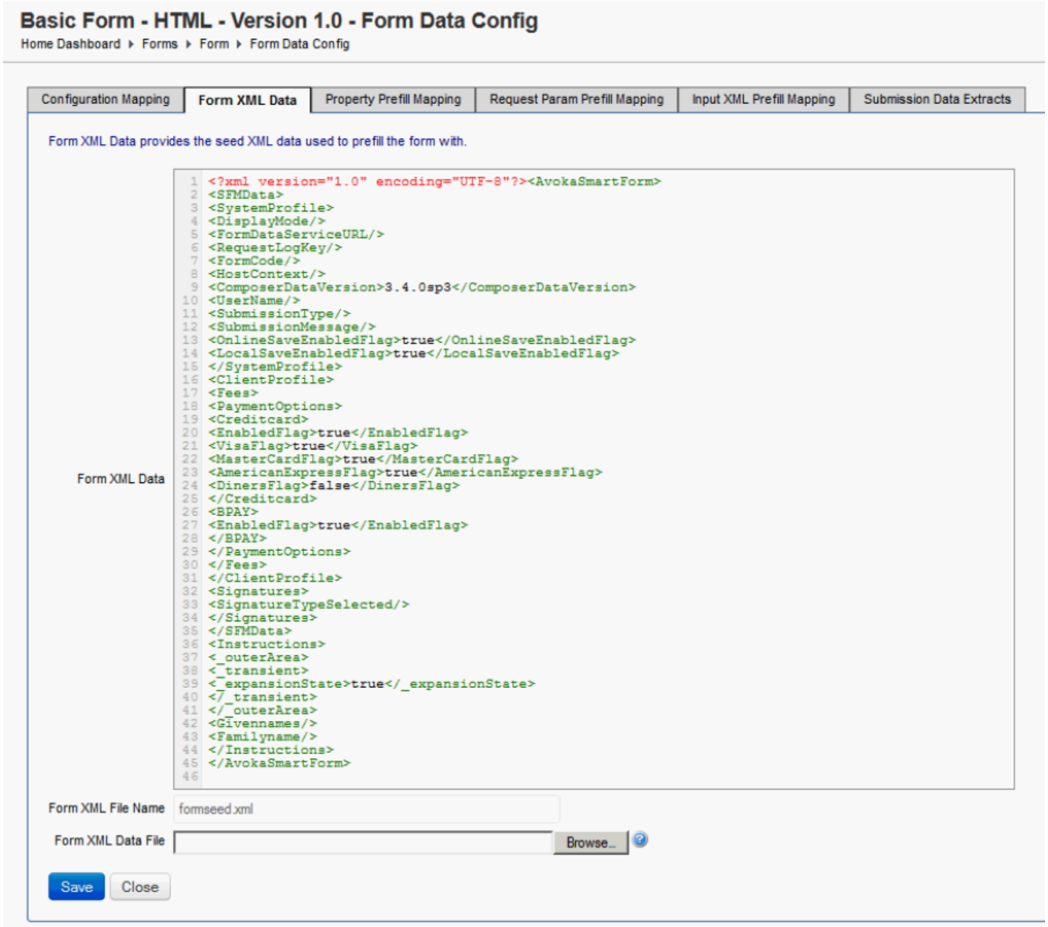
Before the form is rendered, the various prefill methods are applied in the following order:

1. Seed file
2. Property prefill (organization, form)
3. Request parameter prefill
4. Property prefill (user profile)
5. Service-based prefill

Note that while the prefill methods can be combined, prefill data obtained in later stages may overwrite prefill data from earlier stages if multiple mappings to the same form field exist. For each form version, only one prefill service definition can be used for service-based prefill at the same time.
Please read on to learn how to configure individual prefill methods for a form.

How can I configure seed file prefill?

Transaction Manager provides an XML seed file that is used as the basis for the XML payload. Composer will generate a default seed file which contains the structure of the seed file, with no data.
You may edit and change the seed file used for a particular form in Transaction Manager. To edit the seed file for a form, edit the form in the TM management console, switch to the "Form Versions" tab and click on the "Form Data Configuration" icon ( ) for the current version. Switch to the "Form XML Data" tab, where you can edit the seed file or upload a new seed file.



Changing the XML seed file for a form version

To apply seed file prefill, you can simply fill in values for some elements (as shown in the screenshot above) and save your changes. Render the form to test the new seed file.

Make sure to use seed file prefill carefully and reset the seed file when you are done to avoid accidental use of test data. Also, be extremely careful when changing values in the "SystemProfile" section. This section is used by Transaction Manager to store important information about the form, and the form may fail to render or submit if the system profile is modified. No prefill values should be entered into the system profile unless absolutely necessary.

Note: Seed file prefill uses hard coded data. It is generally only used in form testing.

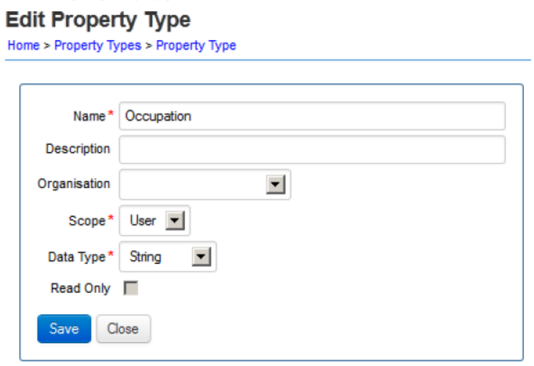
How can I configure property prefill?

Property prefill is used to prefill property values maintained in Transaction Manager into fields on the form. Properties can be declared on one of three levels: Organization, where every form belonging to the organization will inherit a snapshot of the organization properties; form, where every form version can define its own property value; and user, where properties are maintained as part of the user profile and can be edited directly by space users.

Adding TM property types

To view and add to the set of property types set up in Transaction Manager, go to "Forms > Property Types". You will see the set of property types that are currently set up, along with their scope.

To add a property type, click "New". The screenshot below shows an example.



Creating a user property type

Fill in a unique name for the property.

Choose the appropriate scope from the "Scope" dropdown ("Client" for properties that should be set on an organization level, "Form" for properties where values are specific to each form version, and "User" for properties exposed in the space user profile).

Choose the appropriate data type and save.

Ensure there is a property type for each property you would like to map into the form for prefill.

Setting organization and form property values

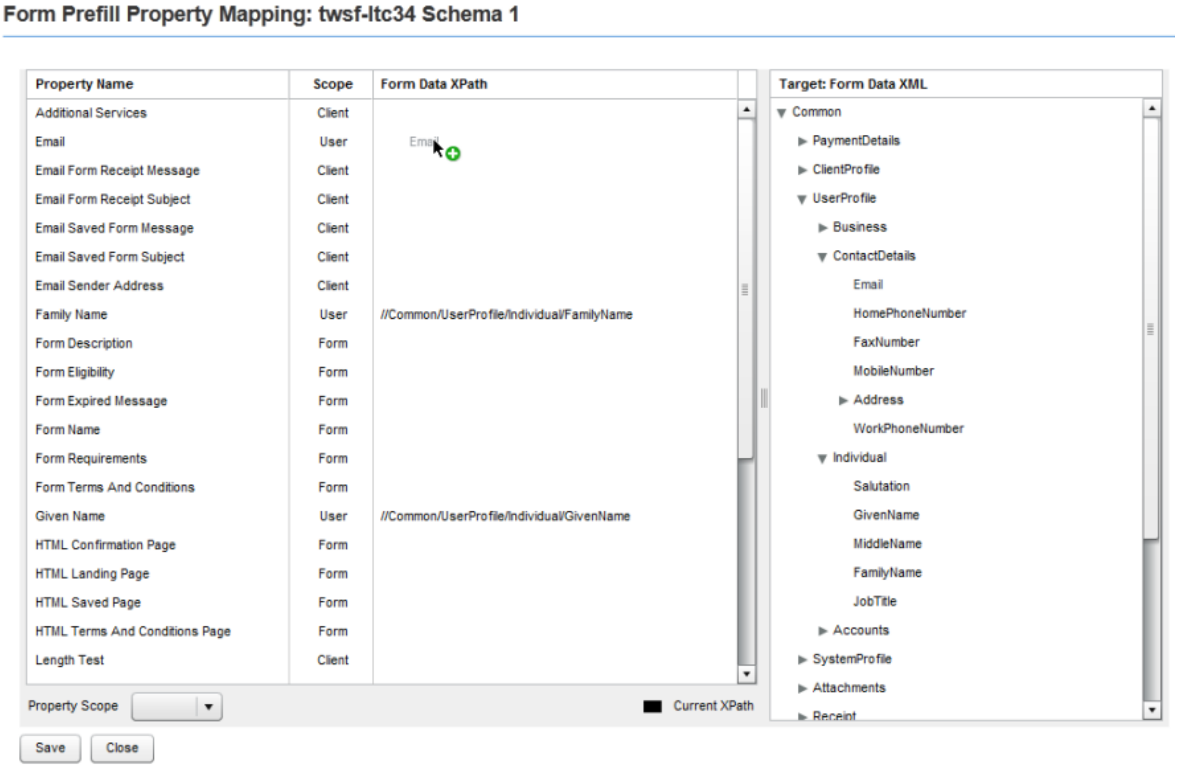
After you have created your set of property types, you need to define the prefill data values for your form(s) by assigning values to these properties.
To understand how property values are maintained, it is important that you are aware of the form versioning system used in Transaction Manager. As it aims to preserve a form version independently, most of the data that affects a form version is stored along with the form version entity. This is normally straightforward, because much of the configuration is done on the form version level anyway. However, there are exceptions, and organization property values are among these. While properties with a scope of "Client" are set on the organization level, each form version contains a copy of the set of organization property values, which are initialized with the current property values configured for the organization when the form version is created.
This means that once a form version exists, changes to organization property values do not automatically affect the form version. Therefore, if you would like to change an organization property value used by a form, you have two options:

1. **Modify the property value on the organization level and then synchronize the organization property values stored for your template version.** This approach is appropriate if the change should be applied for new form versions as well. For example, if an organization's website URL changes, you would want to change the value of the organization property to the new URL, and then apply this change to all form versions that use the URL. To use this approach, edit your organization and switch to the "Properties" tab. Add a new property value if none exists, or edit the existing value for the property type in question. Afterwards, edit your form, switch to the "Form Versions" tab and edit the current version. Switch to the "Properties" tab. You can either click "Sync Org Properties" to reload all organization properties from the organization configuration, or if you want to modify only one organization property value, simply edit the existing value or create a new value by clicking "New" and selecting the same property type.
2. **Edit the property value stored for the form version, leaving the organization property value unchanged.** This approach is appropriate if the new value needs to be overwritten just for this one form version. For example, if the support phone number is configured as an organization property and is consistent for all forms but one, you can simply edit the form version and modify the phone number for just this form version. Newly created forms would still be initialized with the globally defined support phone number. Note: If you find that many forms need to overwrite an organization level property, you should consider using a property of scope "Form" instead. To use this approach, edit your form, switch to the "Form Versions" tab and edit the current version. Switch to the "Properties" tab. Edit the existing value for the property type or create a new value by clicking "New" and selecting the desired property type.

If you create a new form, the organization property values will be copied from the current organization configuration. If you create a new form version for an existing form, all property values (scopes "Client" and "Form") are copied from the previous form version.

Define property prefill mappings

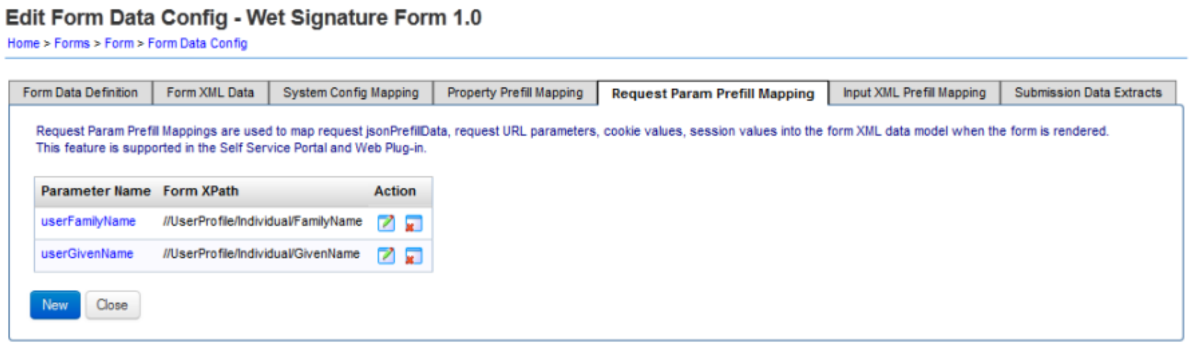
To map TM properties to fields in your form, edit the form and switch to the "Form Versions" tab. Click on the "Form Data Configuration" icon ( ) for the current version and switch to the "Property Prefill Mapping" tab. Click "Edit" to access the mapping page (see screenshot below).



Configuring property prefill mappings
All available properties are shown on the left (whether or not a value is assigned to them). To map a property to a form field, expand the nodes representing the seed file structure on the right to find the location you want to map to, and drag a node into the "Form Data XPath" column next to the property. The XPath of the node should be filled in automatically when you release the node.
Create mappings for all properties you would like to use for prefill, then save your changes.
When a form is rendered, the current set of property values defined for the form version and the set of user properties (if the user is logged in) are retrieved, and are prefilled into the seed file according to the mappings.

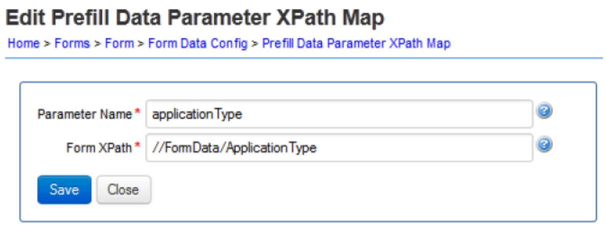
How can I configure request parameter prefill?

Request parameter prefill is used to map prefill data provided in the HTTP request, session or cookies into fields on the form. In addition, you can pass in a request parameter named "jsonPrefillData", which should be set to a JSON string containing the map of prefill parameters as name-value pairs.
To set up request parameter prefill mappings for a form version, edit the form and switch to the "Form Versions" tab. Click on the "Form Data Configuration" icon ( ) for the current version and switch to the "Request Param Prefill Mapping" tab. You will see the list of request parameter prefill mappings currently defined for the form version (see screenshot below for an example).



The set of request parameter prefill mappings for a form version

To add a request parameter prefill mapping, click "New".



Setting up a request parameter prefill mapping

Enter the name of the parameter containing the prefill data into the "Parameter Name" field. When the form is rendered, TM will search the set of HTTP session attributes, cookies, request parameters and the special request parameter "jsonPrefillData" for an entry with this name in this order. If a match is found, the value of the entity will be used as the prefill data. If multiple matches are found, the later entry takes precedence (e.g. request parameters over session attributes).

Enter the XPath to the element associated with the target form field into the "Form XPath" field. If prefill data exists, TM will write the value to the XPath specified. If specifying a potentially ambiguous XPath (e.g. using "//"), make sure it is specific enough to resolve to the correct element. If you enter an XPath that does not resolve to a node in the current seed file, you will not be able to save the mapping.

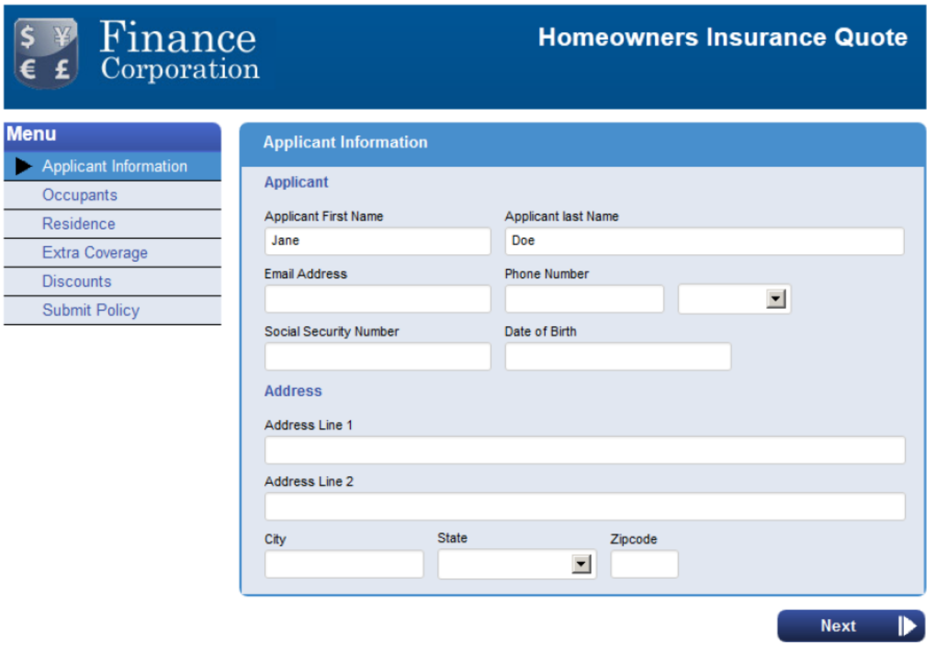
Save the request parameter prefill mapping. The change will take effect immediately.

Simple request parameter prefill example

Edit a test form and set up two property prefill mappings from "userGivenName" to a field in your form, e.g. "//UserProfile/Individual/GivenName", and from "userFamilyName" to another form element. Choose elements that will be displayed when the form opens.

Access your form via the landing page, e.g. https://tmserver.com/govassist/landing.htm?formCode=testform and add a given and family name parameter to the URL: https://tmserver.com/govassist/landing.htm?formCode=testform&userGivenName=Jane&userFamilyName=Doe

Access the new URL and continue to the form. You should see the given and family name populated according to the mappings you defined.



A form prefilled using request parameters

Keep in mind that you can also pass data as session attributes and cookie values instead of request parameters.

How can I configure service-based prefill?

Service-based prefill utilizes a service definition set up in Transaction Manager to obtain XML prefill data, which is then mapped into the seed file or even directly used as the seed file. The service definition encapsulates the logic used to obtain the XML prefill data. Therefore, this is the prefill method of choice when integration with third party systems is required for prefill and it is not practical to use request parameter prefill.

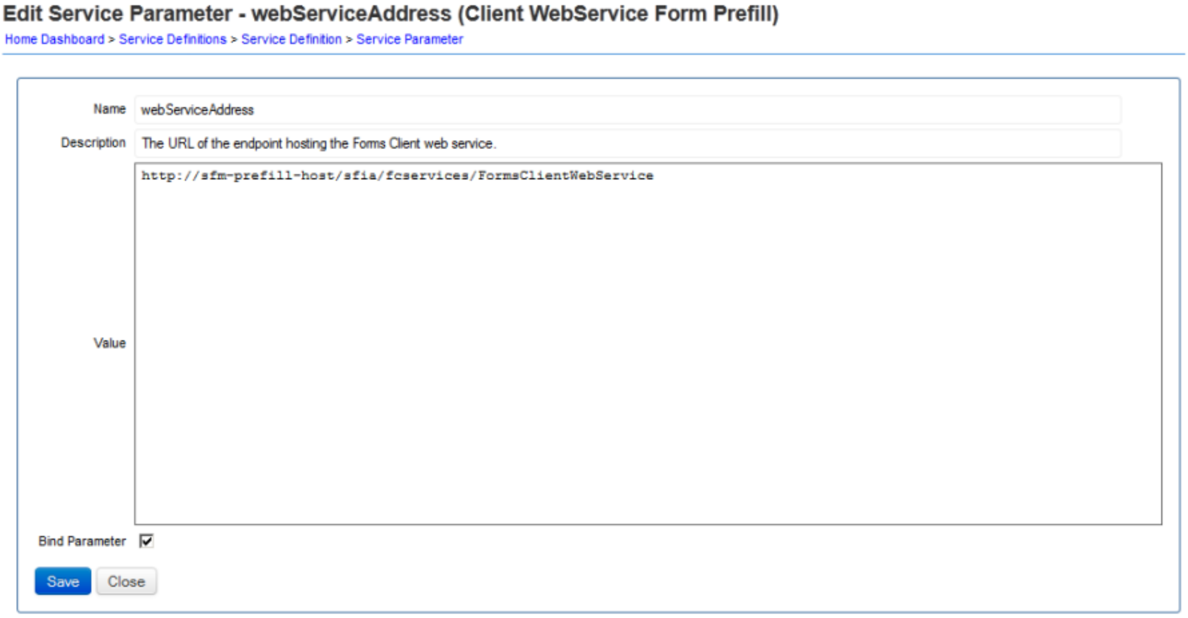
Note: For security reasons, XML prefill is disabled by default. It can be enabled by setting the space property "Allow XML Request Param Form Prefill" to "true".

Transaction Manager provides two prefill service definitions out of the box: A service that invokes an Adobe LiveCycle process to obtain prefill data; and a service that executes custom Groovy code (which itself can call web services or access other sources of prefill data), which can be customized at runtime.

Additional service definitions with custom implementations can be added; please contact Avoka for more details.
The following sections explain how to configure the prefill service definitions, how to define the mappings from the prefill XML data into the form and how to configure a form version to use a specific prefill service.

Configuring the client web service prefill service

The client web service prefill service connects to a web service to obtain prefill data. The web service must conform to the Forms Client web service API specified in the Transaction Manager Web Service Guide. If you need to call a web service that does not conform to this API, you may be able to use the Groovy prefill service to accomplish this.
To configure the client web service prefill service, log in to the TM management console and go to "System > Service Definitions". Filter the list of service definitions by service type "Form Prefill" and edit the service named "Client WebService Form Prefill". Switch to the "Service Parameters" tab.
Edit the "webServiceAddress" parameter and enter the URL of the Forms Client web service endpoint that shall be used to obtain prefill data.



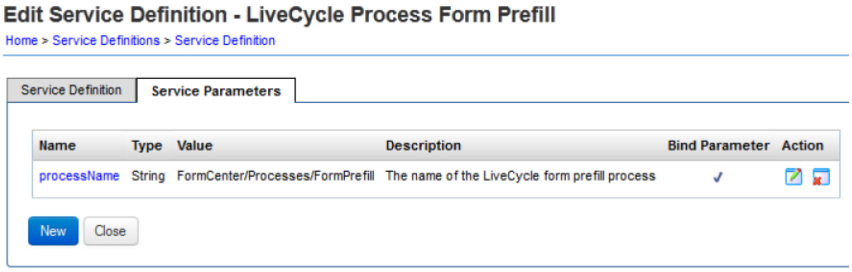
Configuring the URL of the Forms Client web service endpoint
Save your changes and configure your form(s) to use the client web service prefill service as described in "Configuring the prefill service for a form version".

Configuring the LiveCycle process prefill service

The LiveCycle process prefill service calls an Adobe LiveCycle process to obtain XML prefill data.
The input parameters passed to the process are as follows:

- clientCode The organization code of the organization associated with the form.
- formCode The form code of the form.
- requestLocale The locale specified in the HTTP servlet request.
- requestHeaders The HTTP servlet request headers.
- requestParams The HTTP servlet request parameters.
- sessionAttributes The HTTP session attributes.
- seedXml The seed file XML content.
- userAgent The value of the "user-agent" header in the HTTP servlet request.
- userLogin The login name of the current user as provided by the space's authentication service. May be empty if user is not logged in.
- userKey The user key (a UUID) stored for the user, if they are logged in using an TM user account.
- userEmail The email address stored for the user, if they are logged in using an TM user account.

The output parameter containing the XML prefill data string is "prefillXml".
The LCA file supplied with TM contains a prefill process that you can use, or you can provide your own process. To configure the process used by the LiveCycle prefill service, go to "System > Service Definitions" and find the service definition named "LiveCycle Process Form Prefill". Edit the service definition and switch to the "Service Parameters" tab.

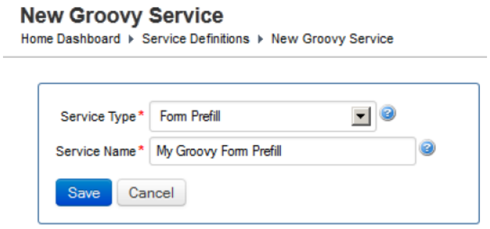


Configuring the LiveCycle process prefill service
Edit the "processName" parameter and enter the desired process name. Save your changes.
Ensure the process you are calling is deployed to the LiveCycle instance(s) used by Transaction Manager.

Configuring the Groovy prefill service

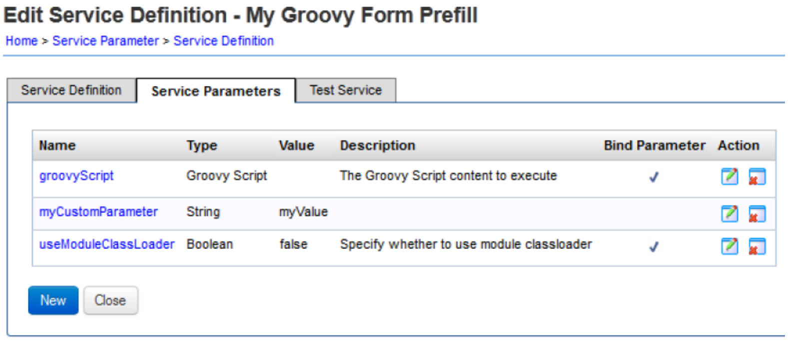
The Groovy prefill service allows you to define and change the logic by which prefill data is obtained at runtime, by modifying a Groovy script exposed as a service parameter. You can define multiple Groovy prefill services with different implementations. This can be very useful when integrating with other systems.

This section assumes you want to add a new Groovy prefill service and preserve existing services. If you just want to edit an existing service, skip the step to create a service definition and just edit the desired service directly.
To configure a new Groovy prefill service based on an existing one, go to "System > Service Definitions" in the management console and click "New Groovy Service".
Select "Form Prefill" from the "Service Type" dropdown.
Enter a name for the new service into the "Service Name" field and save (the screenshot below shows an example).



Creating a new Groovy prefill service based on a template

This will create a new Groovy prefill service based on a template defined in Transaction Manager. You can now customize the new service by editing it and switching to the "Service Parameters" tab.



The service parameters for a Groovy prefill service

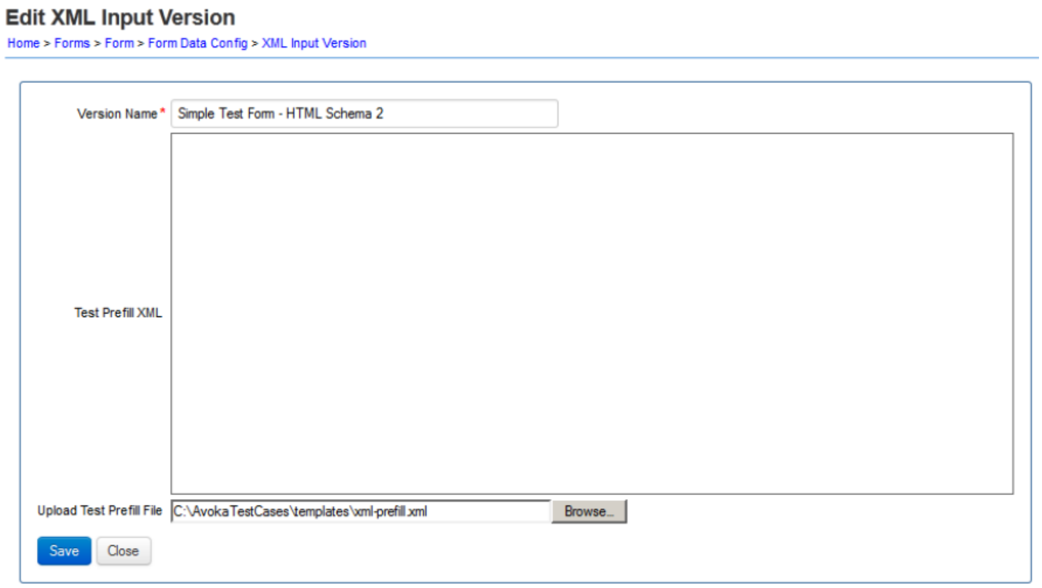
Customize the "groovyScript" parameter with Groovy code to obtain prefill data (e.g. call a web service). The parameters provided to the script are as follows:

- userAccountId The login name of the current user, as provided by the space's authentication service. Can be empty for anonymous users.
- form The form entity for which to obtain prefill data.
- formRequestParams The contents of the session attribute "FormRequestParams". This is a Map<String, Object> which is generally populated by the landing page and contains the request parameters that were passed to the landing page..
- request The HTTP servlet request instance.
- schemaSeed The form seed file XML document.
- serviceDefinition The Groovy prefill service definition entity.
- serviceParameters A Map<String, String> of the service parameters for the Groovy service definition.

The script should return a String containing the XML prefill data.
You can add custom service parameters that your Groovy script can reference. If you do so, make sure to untick the "Bind Parameter" checkbox when you create the parameter (the "Bind Parameter" checkbox should be set only for service parameters where the service class has a named getter and setter for a parameter with the same name.

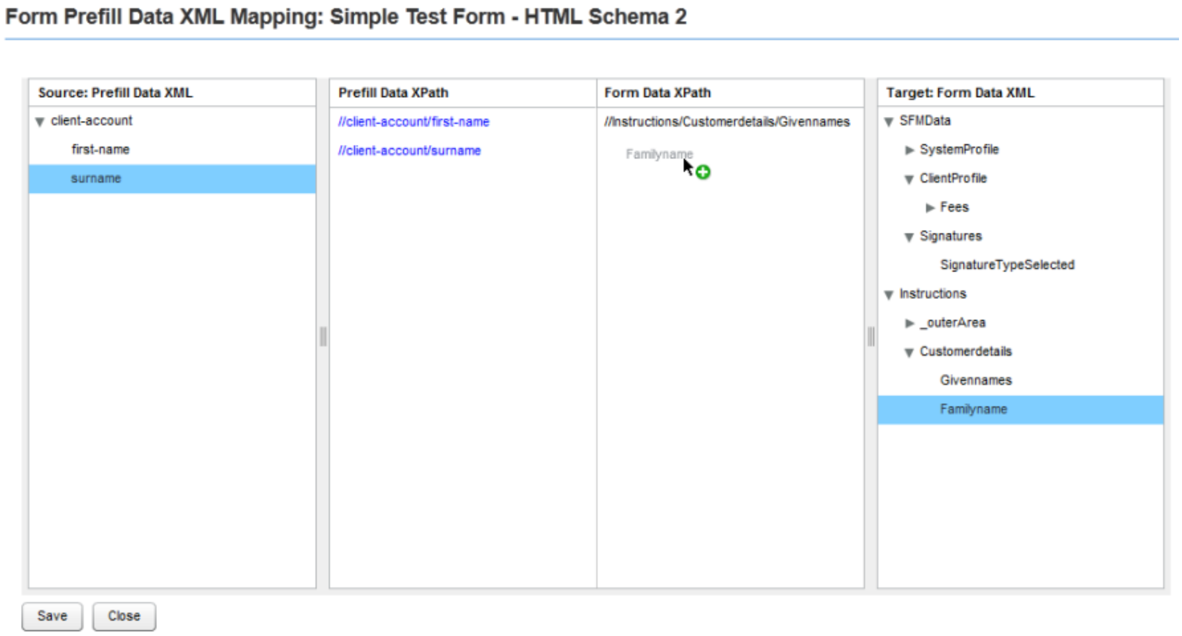
Adding prefill mappings

There are two ways that prefill services can be used: They can either return an XML file with prefill data, which is then populated into the form via explicitly defined mappings (as described in this section); alternatively, the prefill service can return the modified seed file directly. Transaction Manager will assume that the returned XML represent the seed file if no XML prefill mappings are defined for the form; otherwise, TM will treat the XML returned by the service as prefill XML data, and the seed file will be prefilled according to the XML prefill mappings.
Using prefill mappings and returning XML prefill data introduces an additional level of indirection. This gives you greater flexibility, allows you to reuse prefill services across different forms and allows you to make changes to the mappings at any time.
To define XML prefill mappings for your forms, you need a template XML file for your prefill data (the structure must match, but actual prefill content is not needed in the sample).
Once you have the sample XML file, edit your form in the TM management console, switch to the "Form Versions" tab and edit the form data configuration for the current version ( icon). Switch to the "Input XML Prefill Mapping" tab.
Click on "New Input XML". Upload your sample prefill XML file and save.



Uploading the XML file containing the prefill data structure

After uploading the prefill XML file, you can add mappings from the prefill data into the form. To do this, click "Edit XML Mapping" on the "Input Prefill XML Mapping" tab.



Creating XML prefill mappings

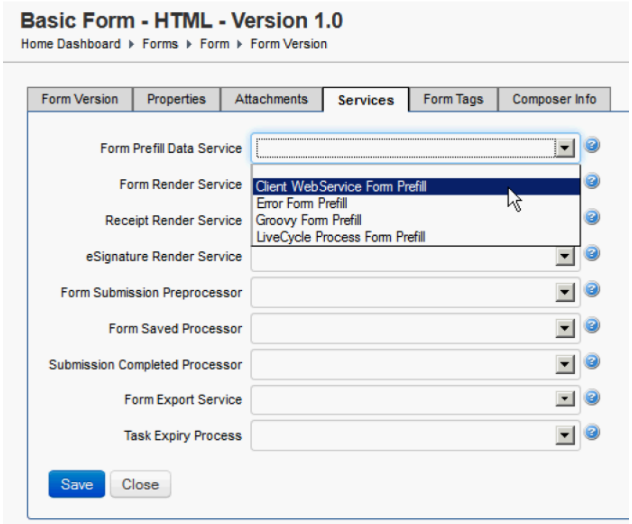
You will see the structure of the XML prefill file on the left, and the structure of the form's seed file on the right.

The middle area displays existing mappings and allows you to define new ones. Simply drag and drop elements from the "Prefill Data XML" area into a new row under "Prefill Data XPath", and drag and drop the target element in the seed file to the same row under "Form Data XPath". The screenshot above shows the "Familyname" element being dragged into place.

Create a mapping for each element from the prefill data that you want to populate into the form. When you are done, save your changes.

Configuring the prefill service for a form version

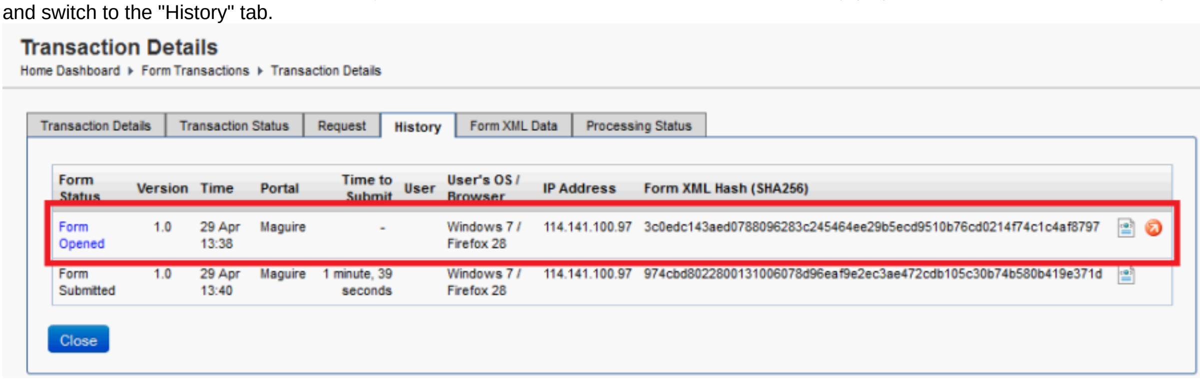
To configure a form for service-based prefill, edit the form and switch to the "Form Versions" tab. Edit the current version and switch to the "Services" tab. Select the desired prefill service from the "Form Prefill Data Service" dropdown and save.



Configuring service-based prefill for a form version

How can I see what data was passed into the form?

To see the data that was passed into the form, you can inspect the submission in TM. Transaction Manager creates a submission object as soon as the form is opened, and adds a submission history entry containing the data that was passed into the form.
Once the user submits or saves the form, you can find the submission in the TM submission log ("Operations > Form Transactions"). View submission details and switch to the "History" tab.



Viewing data passed into a form at render time

Find the entry with a form status of "Form Opened" (there could in theory be multiple such entries if the user saved and reopened the form multiple times).

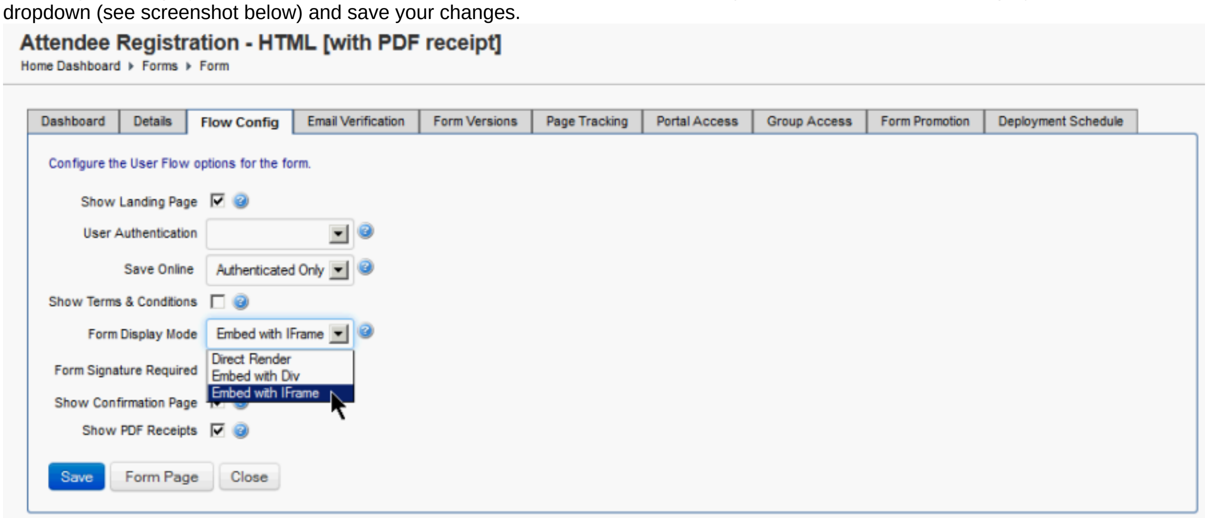
Use the "View XML" ( ) link next to the submission history entry to see the XML data passed in. This includes all prefill data that was mapped into the seed file.

Form Display and Layout (Manager v5.0)

Due to the multitude of devices users employ to fill in online forms, there are a multitude of considerations around form display on screens of various sizes and varying user input and output modalities. Transaction Manager provides responsive spaces and multiple options to control form display and layout. Composer forms likewise support responsive layout for various device types.

What options do I have around form display inside the page?

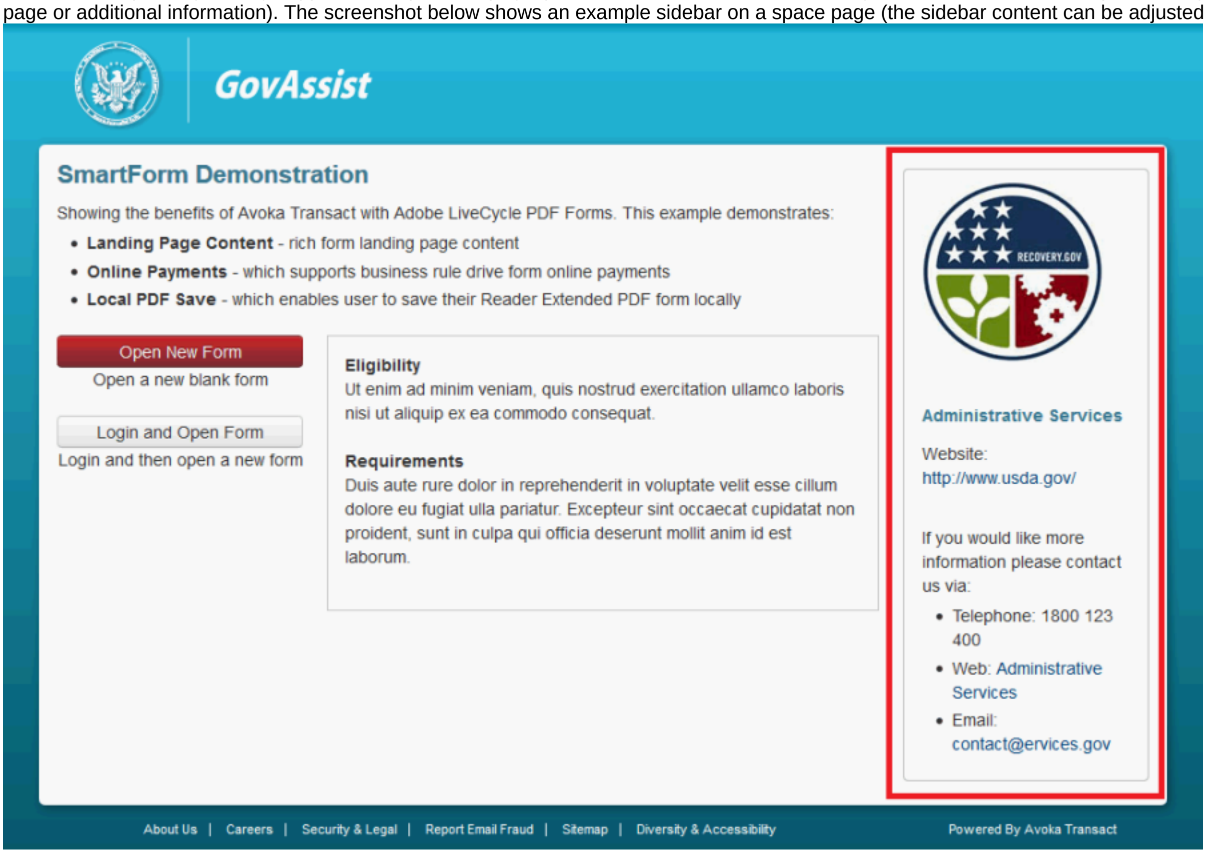
For each form, you can choose from three display modes: direct render, embed with div and embed with iFrame.
XXX what are the caveats and recommendations?
To change the display mode for a form, edit the form and switch to the "Flow Config" tab. Choose the desired display mode from the "Form Display Mode" dropdown (see screenshot below) and save your changes.



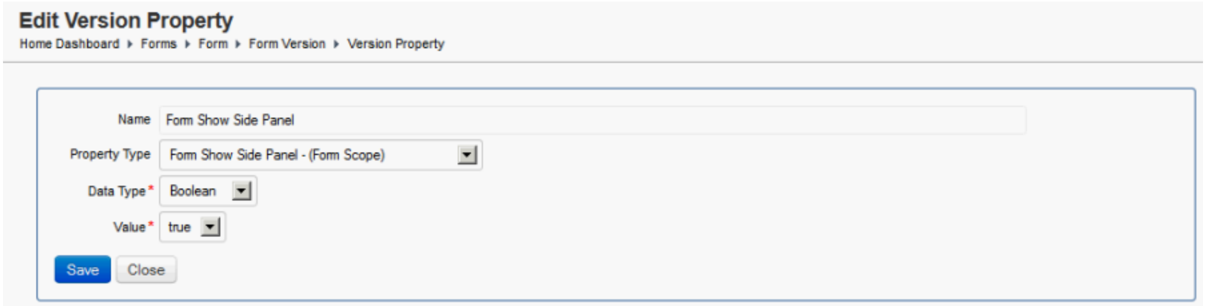
Choosing the display mode for a form
Make sure the form renders correctly on all target browsers and devices.

Can I show the side panel during form fill-in?

The TM user spaces generally support a side panel displaying additional information about the organization and/or the form (such as logo, link to the web page or additional information). The screenshot below shows an example sidebar on a space page (the sidebar content can be adjusted at runtime).



The sidebar (highlighted in red)
By default, this panel is not shown while the user is filling in the form to provide the maximum possible amount of screen real estate to the form and avoid distracting the user. However, you can override this setting for each form and show the side panel throughout the form fill-in process. This is supported only for HTML forms on responsive spaces. If enabled, the sidebar will behave responsively as on all other space pages and will be hidden on small screens.
To enable the side panel for a form, edit the form's current version and switch to the "Properties" tab. Add a property of property type "Form Show Side Panel" and set the value to "true" (see screenshot below). Save the property to apply the change.



Enabling the side panel for a form version

Dynamic Data Retrieval (Manager v5.0)

While a user is filling in the form, the form may want to retrieve data on the fly to respond to a user's selections (for example address lookups or availability checks). It is not always feasible to prefill all data the form might need; instead it is better if the form simply could look up the data that is needed. Transaction Manager exposes endpoints for the form to call, and routes the call on to a configurable service definition that will perform the data lookup. The results returned by the service are passed back into the form.
Composer includes widgets for making dynamic data calls and populating the results into your form.
This section covers common questions around dynamic data retrieval.

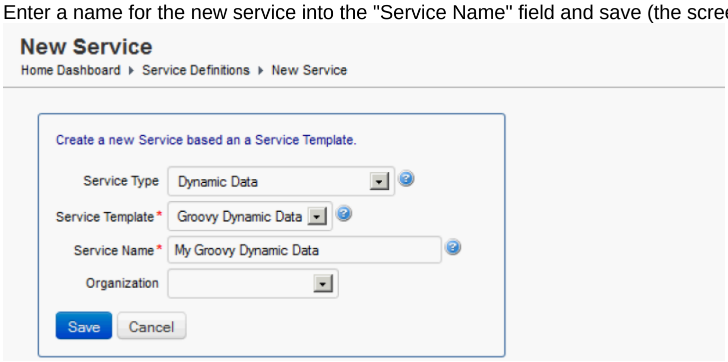
How can my form retrieve data from external sources at runtime?

Your form can dynamically call Transaction Manager to obtain data at runtime. Transaction Manager will route the request to a service definition you provide, which will handle the request and pass result data back. Your form can then handle the result. HTML forms can call a servlet to obtain dynamic data, whereas PDF forms make a web service call. The URL of the TM endpoint is automatically populated into the system profile.
Note: Dynamic data retrieval works only if the form is online and has connectivity to the TM server. Dynamic data retrieval is not available for PDF forms that have been saved offline and will not work on TransactField App forms if connectivity is lost.
If you would like to implement dynamic data retrieval, there are two implementation steps:

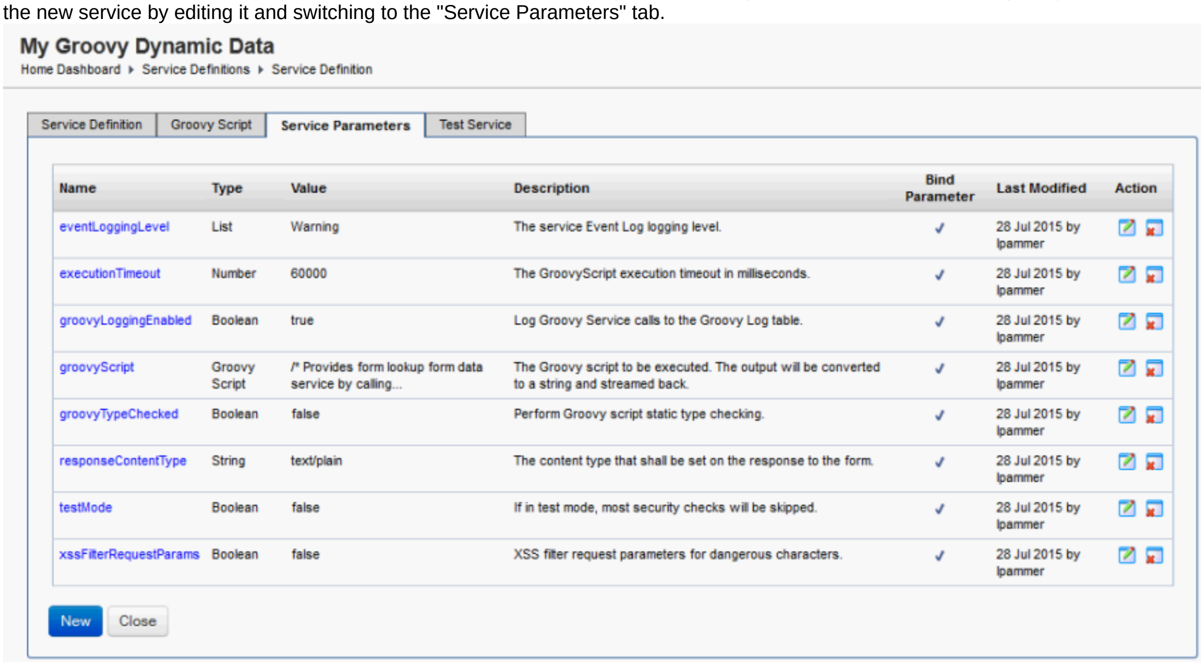
- Provide the logic to handle a dynamic data request as a Transaction Manager service definition
- Enhance your form to make a dynamic data call with the appropriate input parameters, and handle the output as well as any errors that might occur.

Configuring a dynamic data service in TM

Dynamic data services in TM are configured as service definitions of type "Dynamic Data". The logic to retrieve data based on the input parameters provided by the form is encapsulated in a configurable Groovy script. You can maintain multiple dynamic data services in your Transaction Manager system, and each of them can be reused by multiple forms.
This section assumes you want to add a new Groovy dynamic data service and preserve existing services. If you just want to edit an existing service, skip the step to create a service definition and just edit the desired service directly.
To configure a new Groovy dynamic data service based on an existing one, go to "System > Service Definitions" in the management console and click "New".
Select "Dynamic Data" from the "Service Type" dropdown.
Select one of the templates from the "Service Template" dropdown (depending on your server, there may be one or more to choose from).
Enter a name for the new service into the "Service Name" field and save (the screenshot below shows an example).



Creating a dynamic data service based on a template
This will create a new Groovy dynamic data service with all the settings and service parameters based on the template you selected. You can now customize the new service by editing it and switching to the "Service Parameters" tab.



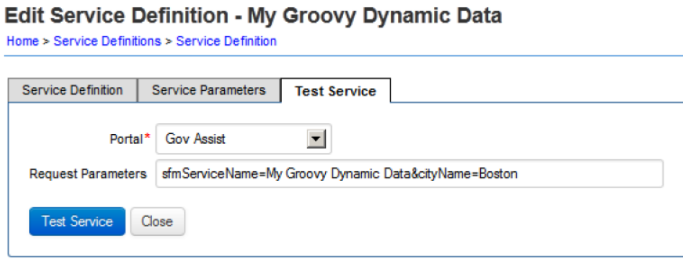
Name	Type	Value	Description	Bind Parameter	Last Modified	Action
eventLoggingLevel	List	Warning	The service Event Log logging level.	✓	28 Jul 2015 by lpammer	
executionTimeout	Number	60000	The GroovyScript execution timeout in milliseconds.	✓	28 Jul 2015 by lpammer	
groovyLoggingEnabled	Boolean	true	Log Groovy Service calls to the Groovy Log table.	✓	28 Jul 2015 by lpammer	
groovyScript	Groovy Script	/* Provides form lookup form data service by calling...	The Groovy script to be executed. The output will be converted to a string and streamed back.	✓	28 Jul 2015 by lpammer	
groovyTypeChecked	Boolean	false	Perform Groovy script static type checking.	✓	28 Jul 2015 by lpammer	
responseContentType	String	text/plain	The content type that shall be set on the response to the form.	✓	28 Jul 2015 by lpammer	
testMode	Boolean	false	If in test mode, most security checks will be skipped.	✓	28 Jul 2015 by lpammer	
xssFilterRequestParams	Boolean	false	XSS filter request parameters for dangerous characters.	✓	28 Jul 2015 by lpammer	

Configuring a dynamic data service
You can change several security settings, chief among them the "testMode" parameter, which can be used during development of the service to turn off most security checks, so you can call the service outside of a form for testing purposes. You can also set the content type of the response that is sent back to the form.
For debugging purposes, you can change the "eventLoggingLevel" parameter. Depending on the log level, additional information may be logged to the TM event log.

The logic to obtain results based on parameters passed through by the form is encapsulated in a Groovy script stored in the "groovyScript" parameter. Edit the parameter and provide your own Groovy script (which might call a web service or access some other external data source).
The parameters that TM passes to the Groovy script are as follows:

- request The HTTP servlet request. For PDF forms calling the dynamic data web service, the web service input parameter containing a JSON representation of the input parameters provided by the form, a mock request object is passed in, containing each parameter in the JSON string as a request parameter.
- form The form entity.
- serviceDefinition The dynamic data service definition entity.
- serviceParameters A Map<String, String> of the service parameters for the dynamic data service definition.

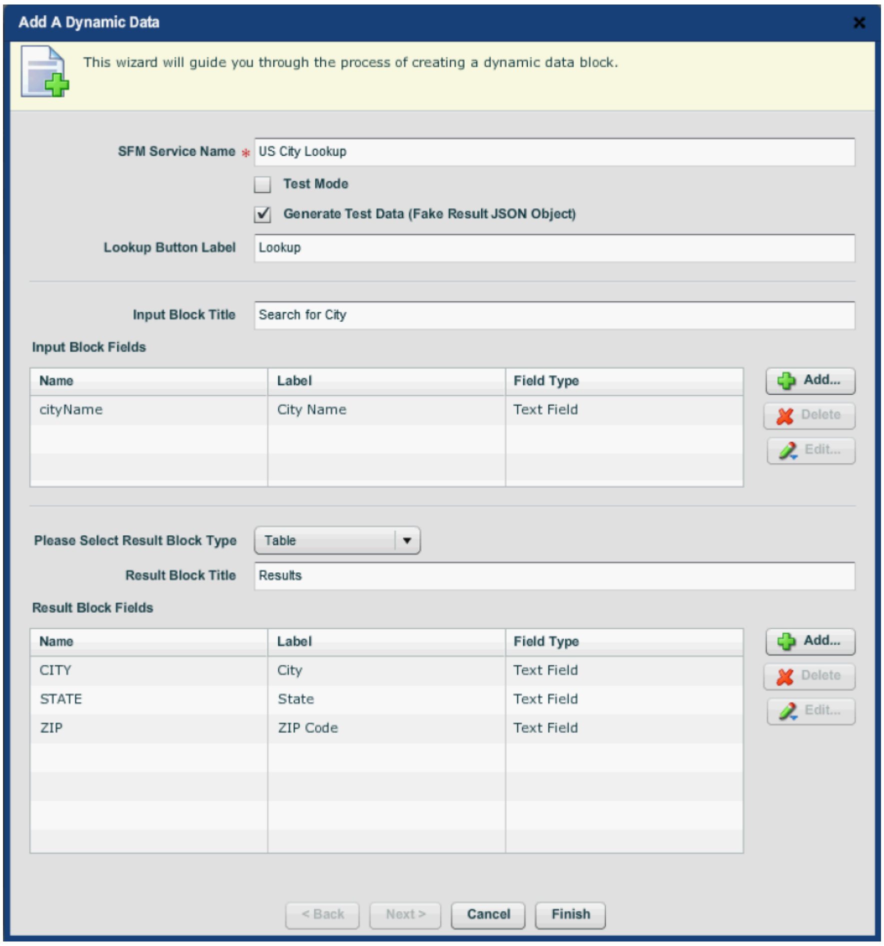
For this example, we will be using the following script:
import wslite.soap.* import net.sf.json.* import net.sf.json.xml.* def cityName = request.getParameter('cityName') def client = new SOAPClient('http://www.webserviceX.NET/uszip.asmx') def response = client.send(SOAPAction:'http://www.webserviceX.NET/GetInfoByCity') { body { GetInfoByCity ('xmlns':'http://www.webserviceX.NET') { USCity(cityName) } } } def root = new XmlSlurper(false,true).parseText(response.text) def dataset = root.'**'.findAll{ it.name() == "NewDataSet" }[0].children(); def result = [] dataset.each { def str = "{" it.childNodes().each { str = str + "'${it.name()}' : '${it.text()}'," } str = str[0..-2] + "}" result += new JSONSerializer().toJSON(str) } println result return result
This service queries the parameter "cityName" passed in by the form, passes it into a city lookup web service operation, and passes back the response as a JSON string. Because the script is defined as a service parameter in TM, it can be adjusted at any time without requiring a server restart.
Ensure you test your dynamic data service thoroughly. TM allows you to test your service directly - edit the service, turn on test mode by editing the "testMode" service parameter and switch to the "Test Service" tab.



Testing a dynamic data service
You can select the space you would like to test on and enter the service parameters in the format "<parameter name>=<parameter value>&<parameter name 2>=<parameter value 2>" and so on. You must always pass in the name of the service definition you want to call (as the form will have to), but TM pre-generates this parameter for you. An example for the parameter string would be "sfmServiceName=My Groovy Dynamic Data&cityName=Boston&mycustomparameter=value".
Click "Test Service" to start a test. The dynamic data servlet for the selected space will be invoked and passed the specified parameters. If all goes well, you will see the return value provided by your Groovy script displayed in your browser just as they would be returned to the form.

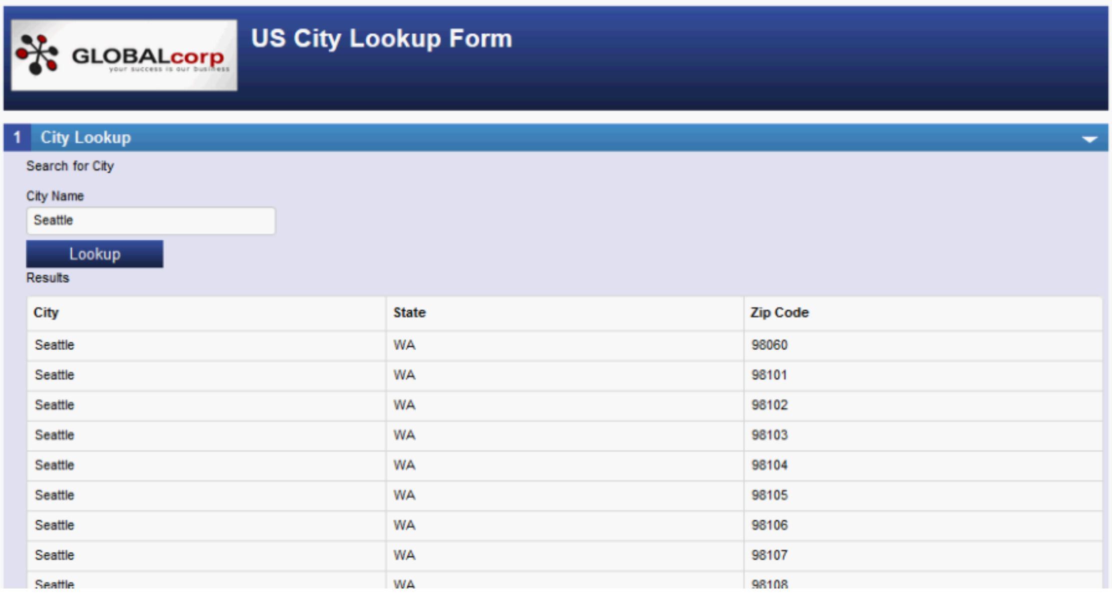
Adding a dynamic data call to a Composer form

This section shows how to add a dynamic data call to a Composer form, assuming a dynamic data service is already set up in Transaction Manager. To illustrate the steps, we use a simple, publicly available web service to provide the lookup data, which is exposed by a service definition in TM (see previous section for details of the example). We will use the Dynamic Data Assistant provided by Composer, which will generate a set of input fields, a button to trigger the lookup and a table to display the results.
Create a new form in Composer or open one of your existing forms. Find the "Dynamic Data Assistant" palette item and drag it onto a section in your form.



The dynamic data assistant

Enter the name of the dynamic data lookup service defined in Transaction Manager, in our case "US City Lookup".
Enter the name of the input block ("Search for City") and add a single input field, a text field named "cityName". Note: You can assign a label for display purposes, but the field name must match the input parameter expected by the TM service.
Enter the name of the result block, select "Table" as the result block type and add three result block text fields as shown.
Click "Finish". The assistant will create specialized sections performing the lookup. You can customize the styling and arrangement of the elements as well as the handling of results returned by TM; however, this is beyond the scope of this example.
Now publish the form to the TM instance where your dynamic data service is set up. Render the form, enter search data and click the lookup button. Your search results should be displayed in the table as in the example screenshot below.



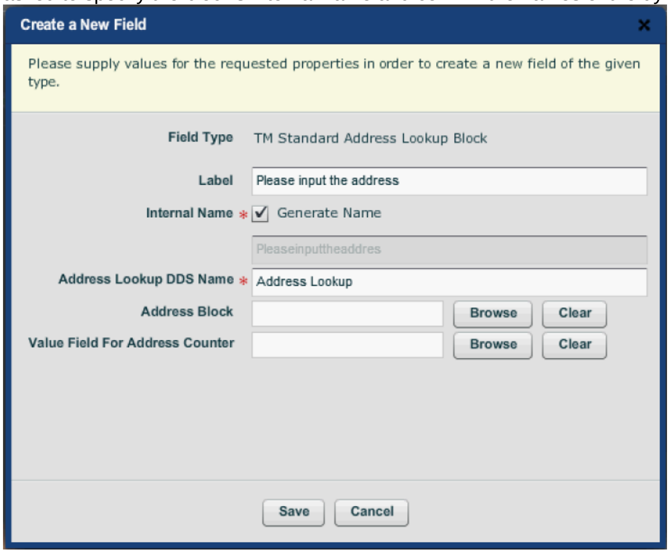
The results of a dynamic data call at runtime

Can my form look up and validate an address while the user is entering it?

Avoka Transact provides the capability to use live address lookup and validation using the Mastersoft Harmony Address Lookup service ({+}http://www.mastersoftgroup.com/hh-faq/+). You can add predefined address lookup widgets to your Composer forms, which will call a dynamic data service during address lookup to provide users with a list of address suggestions as they type.
This mechanism allows you to add dynamic address lookup to your forms in minutes, without having to do any integration work beyond having your server added to the whitelist for Veda service calls (your Avoka account manager will be happy to assist).
Note that transactions including Mastersoft address lookup calls are subject to additional transaction fees. Please contact your Avoka account manager for details.

Adding an address lookup widget to a form

Mastersoft address lookup is available in Composer from version 4.1 SP2 onwards.
To add a Mastersoft address lookup block to the form, find the "TM Standard Address Lookup Block" in the palette and drag it onto your form. You will be asked to specify the block's internal name and confirm the names of the dynamic data services on TM (see screenshot below).

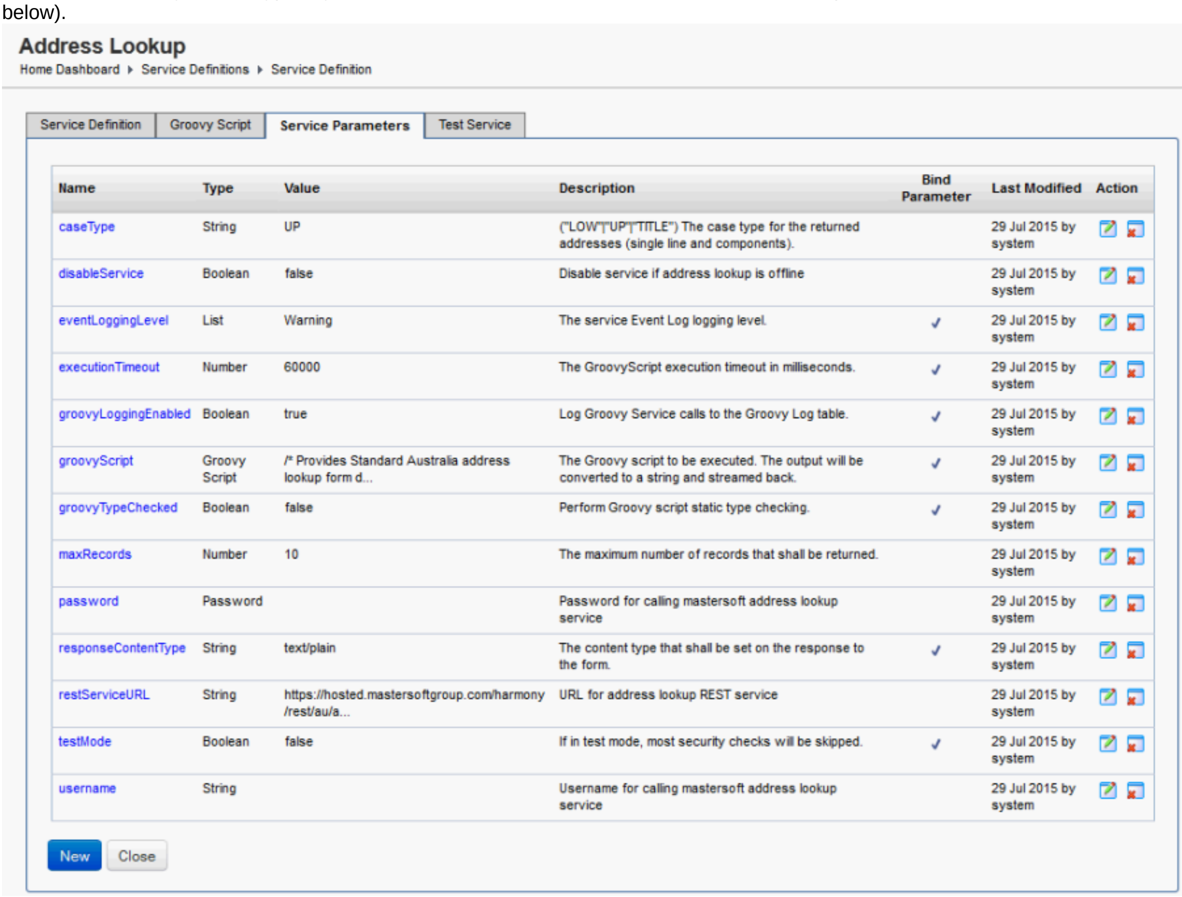


Configuring a standard address lookup widget

Enter a unique internal name for the block. Do not change the dynamic data service name fields - by default, TM (version 4.1.8 onwards) will contain the correct service. Once you click "Save", a new address lookup block will be added to your form.
You can configure some additional aspects of the address lookup block (e.g. how many characters the user needs to type before address lookup is initiated) by editing the properties of the block and switching to the "Data" tab.
Once you are satisfied, publish your form to Transaction Manager. The following section will tell you how to set up address lookup in TM.

The Mastersoft address lookup service in TM

To enable Mastersoft Harmony address lookup, TM exposes a dynamic data service named "Address Lookup". These services are fully set up but need to be configured with your Mastersoft credentials before you can use them. Please set up an account with Mastersoft and follow their directions.
Once you have set up an account with Mastersoft, configure the service by logging on to the TM management console and accessing "System > Service Definitions". Filter by service type "Dynamic Data" and edit the service named "Address Lookup". Switch to the "Service Parameters" tab (see screenshot below).



Name	Type	Value	Description	Bind Parameter	Last Modified	Action
caseType	String	UP	("LOW"|"UP"|"TITLE") The case type for the returned addresses (single line and components).		29 Jul 2015 by system	
disableService	Boolean	false	Disable service if address lookup is offline		29 Jul 2015 by system	
eventLoggingLevel	List	Warning	The service Event Log logging level.	✓	29 Jul 2015 by system	
executionTimeout	Number	60000	The GroovyScript execution timeout in milliseconds.	✓	29 Jul 2015 by system	
groovyLoggingEnabled	Boolean	true	Log Groovy Service calls to the Groovy Log table.	✓	29 Jul 2015 by system	
groovyScript	Groovy Script	/* Provides Standard Australia address lookup form d...	The Groovy script to be executed. The output will be converted to a string and streamed back.	✓	29 Jul 2015 by system	
groovyTypeChecked	Boolean	false	Perform Groovy script static type checking.	✓	29 Jul 2015 by system	
maxRecords	Number	10	The maximum number of records that shall be returned.		29 Jul 2015 by system	
password	Password		Password for calling mastersoft address lookup service		29 Jul 2015 by system	
responseContentType	String	text/plain	The content type that shall be set on the response to the form.	✓	29 Jul 2015 by system	
restServiceURL	String	https://hosted.mastersoftgroup.com/harmony /rest/au/a...	URL for address lookup REST service		29 Jul 2015 by system	
testMode	Boolean	false	If in test mode, most security checks will be skipped.	✓	29 Jul 2015 by system	
username	String		Username for calling mastersoft address lookup service		29 Jul 2015 by system	

The service parameters of the "Address Lookup" service

Edit the "username" and "password" parameters and enter your credentials. Save your changes.

You can also change other parameters, such as the maximum number of search results that will be returned to the form for each call.
Now your service is ready to be used by forms containing Mastersoft address lookup widgets. Please note that TM transactions including Mastersoft address lookup calls will incur additional fees; please contact your Avoka account manager for details.

How do I stop the Reader security warnings appearing when my PDF form makes a web service call?

Enter topic text here.

Form Validation (Manager v5.0)

XXX this is currently not a very meaty topic - worth fleshing out?

How can I verify submitted data on the server before it is delivered?

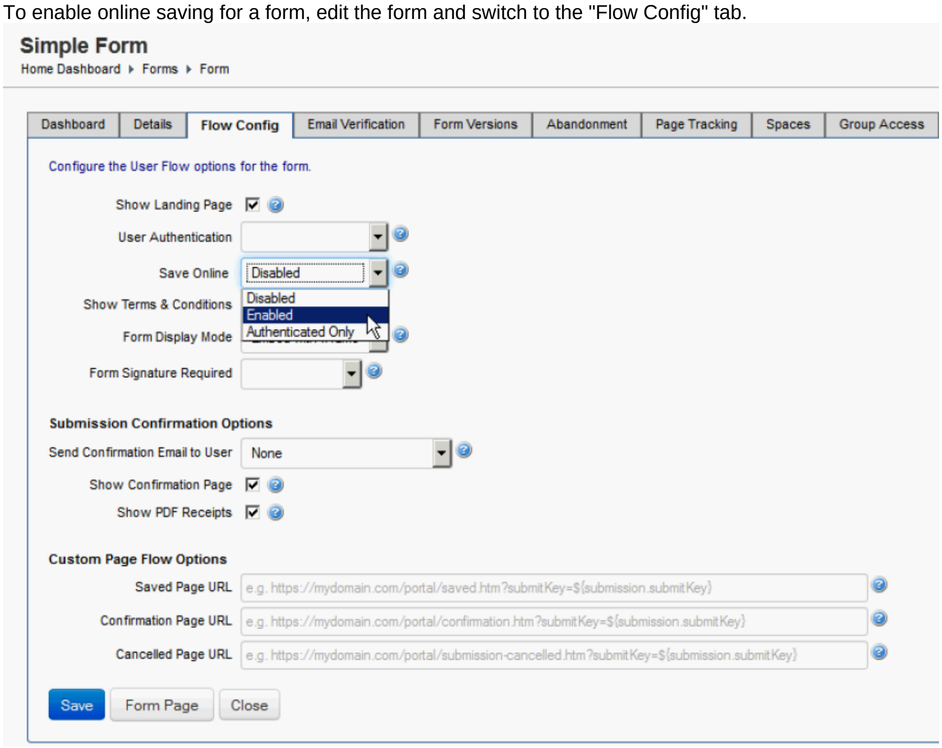
XXX is this still relevant? these days we'd probably use Groovy

Form Saving and Retrieval (Manager v5.0)

For you and your users' convenience, your forms can be configured to allow "online" saving, where a form can be saved as a draft to Transaction Manager, and later be securely retrieved by the original user. PDF forms can also be configured for "offline" saving, where the user saves a copy of the form to their computer.
This section discusses common use cases around form saving.

How do I enable a form to be saved online?

To enable online saving for a form, edit the form and switch to the "Flow Config" tab.



Enabling online save for a form
If you want to ensure that users can save only while logged in (using a TM user account), choose "Authenticated Only" from the "Save Online" dropdown. Transaction Manager will reject attempts to save the form online if the user was not authenticated when rendering the form. If an authenticated user saves a form online, they will be shown a page confirming that the form was saved, and can reaccess the form by logging on to TM and accessing their To Do list.
If your users are not logged in to TM when filling in the form or you want to allow all users to save the form, choose "Enabled" from the "Save Online" dropdown. In this case, Transaction Manager will still allow attempts to save the form online if the user is authenticated. This also allows submission sharing between anonymous and registered users. If an anonymous user saves the form online, they will need to pass a security challenge to return to their form - more on this in the next section.
Once you have configured the online save setting, save your changes.

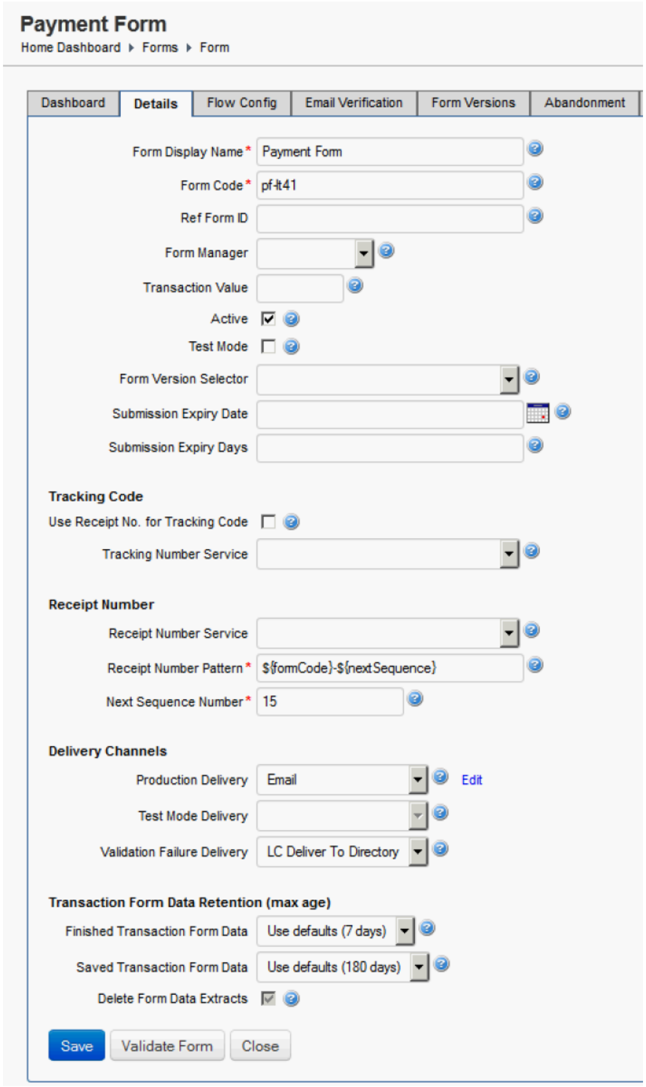
Can I allow anonymous users to save my form online?

Transaction Manager provides support for anonymous online saving and, just as importantly, a secure way for anonymous users to return to their saved form later while denying access to other users.
There are two ways you can secure saved form retrieval for anonymous users:

1. **Reference number only** When the user saves the form online, a page is displayed to the user containing customizable messaging as well as the reference number (which is the same as the tracking code or receipt number generated for the form). The user can send themselves an email with the reference number and a link back to the form. When returning to the form later, the user enters their reference number. If it is correct, the user can resume filling in the form; otherwise, access is denied.
2. **Reference number and form data challenge** In addition to the above mechanics, TM can also extract data from the saved form and ask the user to enter this data (in the form of a question such as "What is your place of birth?") in addition to the reference number before the user can access the saved form. Both the question text and the XPath in the submitted XML where the answer can be found are configured as form version properties in Transaction Manager. While this mode adds security, it is also important to consider carefully what question to ask the user. The data item to be asked about must be set before the form is saved (the form needs to ensure this - if no answer is defined when the form is saved, TM will treat this as an error case). Additionally, the data item must be both private (unlikely to be known by attackers) and unambiguous (the user must be able to easily remember the answer when prompted, and also must be able to enter the answer in the correct format). Dates can be problematic due to the multitude of date formats. Note that if a user forgets the answer to the form data challenge, there is no way for them to get back to the form. The answer itself is not stored in TM for security reasons (only a hash of the answer is), so it cannot be retrieved by support personnel. In such a case, the user will need to fill in the form again from scratch.

Generating a secure reference number for anonymously saved forms

When returning to an anonymously saved form, the user is prompted to enter the corresponding reference number. This is the same as the tracking code / receipt number generated for the form submission, and was displayed to the user after saving the form (if the user used the TM feature to have an email sent to them, this email generally also contains the reference number).
To make it easier for the user, the reference number should not be overly long. However, in the interests of security, the reference number should not be guessable (e.g. simply be an incrementing sequence). The reference number must also uniquely identify the submission. To achieve all these objectives, you will need to look into how tracking codes / receipt numbers are generated for your form.
To view or modify tracking code and receipt number settings, edit the form and ensure you are on the "Form" tab.



Changing the tracking code / receipt number settings

The "Tracking Code" section tells you whether the form uses tracking codes or receipt number as the reference number (receipt number is used if "Use Receipt No. for Tracking Code" is ticked).
When using tracking codes, you can choose a specific service to generate the tracking code ("Tracking Number Service" dropdown) or leave the dropdown blank to use the system default service.
Note that tracking codes are designed to be easily used as a reference number: They are short, alphanumeric strings that are randomly generated. You can define the length and set of characters that will be used to generate tracking codes by editing the corresponding service of type "Tracking Number".
When using receipt number ("Use Receipt No. for Tracking Code" is ticked), you need to ensure that the receipt number is appropriate for use as a reference number. Please review the settings in the "Receipt Number" section.
The "Receipt Number Service" dropdown determines the service that is used to calculate the receipt number (if blank, the default service is used - to find out which service that is, go to "System > Service Definitions" and search for services of type "Receipt Number"). There are currently two receipt number services that are shipped with Transaction Manager. These services are described in detail in the TM Administration Guide, chapter "Form Configuration", section "Receipt Number Generation".

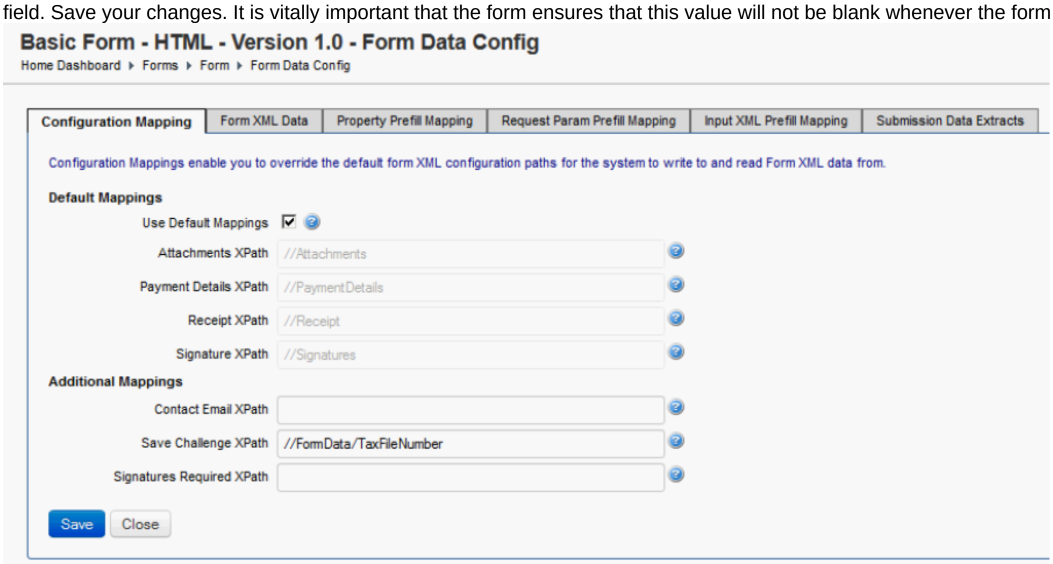
- Sequence Pattern Receipt Number Service Do not only use the variables "${submissionId}" and "${nextSequence}" as they are sequential and therefore easily guessable. A good starting point for a pattern is "${submissionId}-${submitKey.substring(0,4)}", which combines the submission OID (for uniqueness) with part of the submit key (obscurity) while still remaining reasonably short.
- Groovy Receipt Number Service Ensure your script returns suitable values that are not guessable.

Enabling a form data challenge in addition to the reference number

To enable the additional form data challenge, you need to configure the save challenge question to be presented to the user as well as the location of the challenge answer in the submitted form.
To configure the save challenge question, create a value for the form property "Save Challenge Question". To do so, edit the current form version, switch to the "Properties" tab and create the property as shown in the example below. Make sure to include any relevant instructions (such as required date format). Example: "What is your Tax File Number (enter without spaces)?".

The form property for the save challenge question

To configure the XPath in the form data where the challenge answer will be read from, edit the form data configuration and fill in the "Save Challenge XPath" field. Save your changes. It is vitally important that the form ensures that this value will not be blank whenever the form is saved.

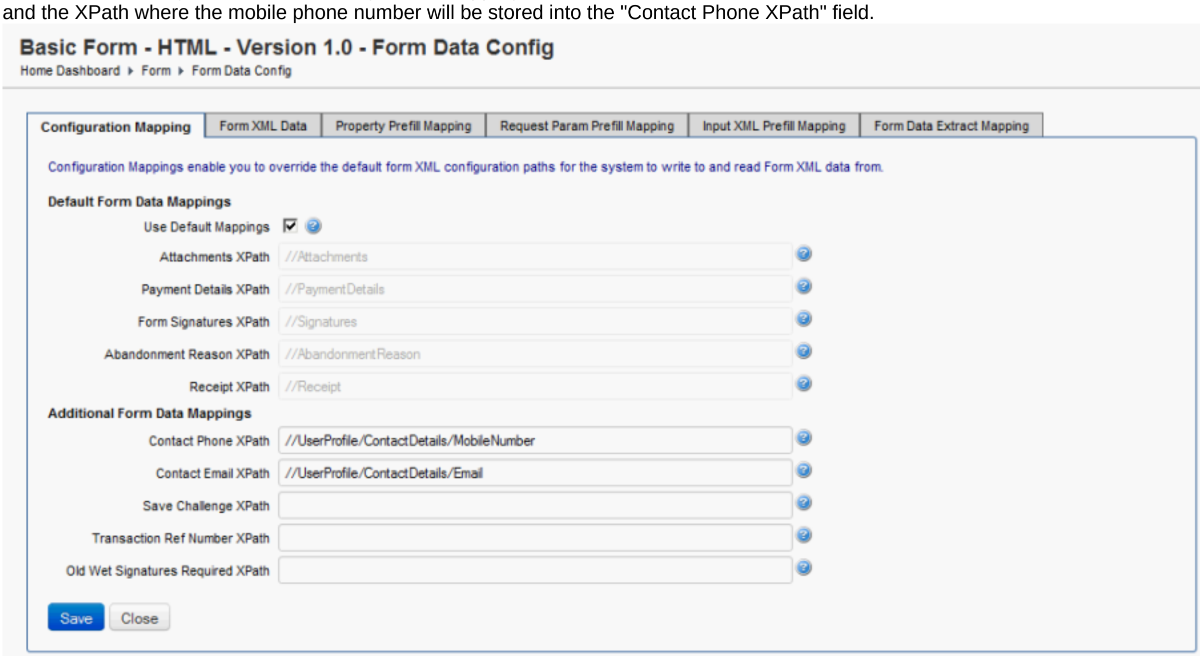


Configuring the challenge answer XPath

Extracting contact details entered by the user

Often it is important to know a user's email address or mobile phone number. While TM allows registered users to configure email addresses and phone numbers in their user profile, this does not help if users are submitting your form anonymously. To provide you with a more generic way of capturing user contact information, TM can be configured to extract a contact email address and/or mobile phone number from submitted form data, which is then stored against the submission and displayed in the TM management console.

To set up automated extraction of the user's email address and / or mobile phone number for a form, edit the form data configuration for the current form version and ensure you are on the "Configuration Mapping" tab. Enter the XPath where the email address will be stored into the "Contact Email XPath" field, and the XPath where the mobile phone number will be stored into the "Contact Phone XPath" field.

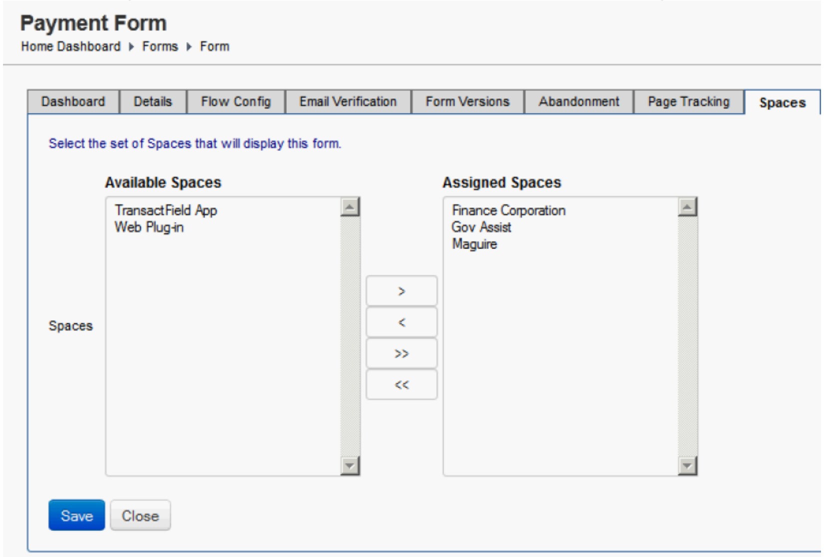


Configuring the XPath containing the contact email address and mobile phone number

Whenever a form is submitted or saved, the content of the specified element is read from the submitted data and stored against the submission. This email address is used by TM during submission email verification and can also be used by custom services (e.g. Groovy delivery services).

Can saved forms be shared across user spaces, including the TransactField App?

By default, saved forms are shared across all spaces on which the form is hosted. The TransactField App can synchronize saved forms with the TM server and is therefore able to display forms saved on other spaces (and *vice versa*).
Users can open and complete saved forms on any space (assuming the form can be rendered and submitted on all configured spaces).
To see what spaces a form is hosted on, edit the form and switch to the "Spaces" tab.

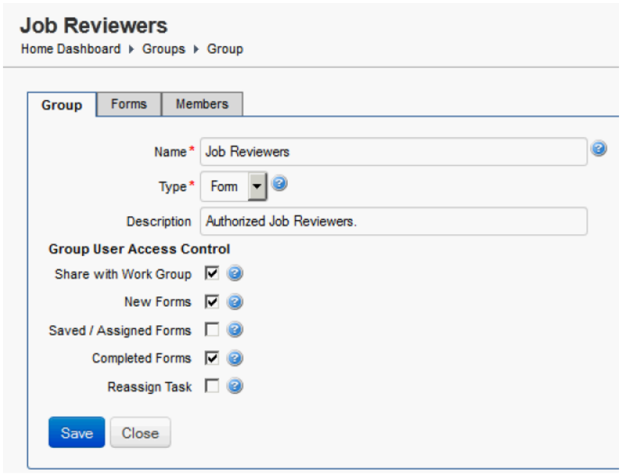


The list of spaces hosting a form
The set of spaces that are listed under "Spaces" can be used to host the form, and potentially share saved forms.

Can I assign a saved form to another person to complete?

Currently, Transaction Manager supports form sharing via form work groups. Form work groups are groups set up in TM that users can be assigned to (or, for externally managed users such as LDAP user accounts, group associations can be loaded at login time). Form work groups can then be associated with a number of forms and control access to these forms. All users in a form work group share access to all forms belonging to the group.
To further control group access, a form group can be configured to allow shared access to one or more of the following (see also the image below):

- opening new instances of the forms
- completing saved or otherwise incomplete submissions for these forms
- viewing submissions and receipts for these forms
- reassigning tasks based on these forms



Access control options for a form work group
This means that if you have a form work group where all access is shared, user A can open a form and save it, then user B can open the saved form and submit it, and both user A and user B will see it in their submission history (assuming both users are in the form work group in question). Group membership is evaluated dynamically, i.e. you can assign a new user to the group and they will instantly be able to see all relevant forms and submissions - or conversely, you can remove a user from a group to revoke access to these forms.
To avoid multiple users working on the same saved submission, a warning message is displayed on the space if another user has recently opened a saved submission but has not submitted it yet.
To summarise, if you want a group of users to be able to pick up each others' saved submissions, create a form work group with at least "Saved / Assigned Forms" permissions and assign users to it. All users in the group will then be able to see and complete each others' saved forms. Again, you can also assign users to the group as needed (e.g. if all current group members are unavailable but a submission must be completed, you can assign another user to the group temporarily and have them complete the submission).

Can I display custom messages after a user saves a form?

You can customize various messages displayed to the user after saving the form. This includes an additional message on the "Submission Saved" page as well as the subject and body template of the email anonymous users can send themselves with a link back to the form. All these messages can be customized separately for each form version.

Adding a custom message to the "Submission Saved" page

To add a custom message that will be displayed on the "Submission Saved" page, edit the form, switch to the "Form Versions" tab, edit the current version and switch to the "Properties" tab. Click "New", set the property type to "Saved Message" and enter your message (including HTML formatting) into the "Value" field. Save the property value.
In addition, the form has the opportunity to set or modify the saved message. When the form is rendered, the "Saved Message" form version property is populated into the "SavedMessage" node in the system profile. The form can modify this value as desired, and when the form is saved, TM reads out the saved message from the form data and stores it in the submission entity to be displayed on the "Submission Saved" page.
If a saved message is set, the "Submission Saved" page will display it in addition to its usual content. See the screenshot below for an example (the custom message is the text in the "Please Note" section).

The "Submission Saved" page, including a custom message

Customizing the "Submission Saved" email sent to users

On the "Submission Saved" page, users can elect to send themselves an email with a link to the saved form (as well as the reference number if the form was saved anonymously). You can customize the subject, body and sender email address used in the email for each user space and overwrite these settings on an organization and/or form version level.
The email subject and message are provided as Apache Velocity templates that can be customized with the following variables:

- form The form entity. You can reference attributes, e.g. "${form.formName}".
- formUrl The URL back to the saved form. This should always be included in the email body, especially for anonymously saved forms.
- submission The submission entity. You can reference attributes, e.g. "${submission.receiptNumber}", which should be included for anonymous users, since it is used as the reference number needed to return to the form.
- portal The space on which the form was saved.
- environmentName The name of the TM environment, as specified in the space property "Admin Environment Name" associated with the "Admin Console" space.

If the email message should be usable for both anonymous and authenticated online saves, it should include conditional text such as:
<p> You recently saved a ${form.formName} at ${portal.name}. </p> #if ($submission.user) <p> Please open your form at: </p> #else <p> Your save reference number is: ${submission.receiptNumber} </p> <p> Please provide this number when you open your form at: </p> #end <p> ${formUrl}. </p>
This will display the return URL for authenticated users, and the reference number as well as a return URL for anonymous users.
To configure the email sender, subject template and body template for a space, modify the space properties "Email Sender Address", "Email Saved Form Subject", "Email Saved Form Message". Note that the email sender address is not specific to the "Submission Saved" email.
To override the email sender, subject template and body template on the organization level, set the organization properties "Email Sender Address", "Email Saved Form Subject" and "Email Saved Form Message".
To override the email sender, subject template and body template on the form version level, set the form version properties "Email Sender Address", "Email Saved Form Subject" and "Email Saved Form Message".

Can I allow users to re-use a PDF form for multiple submissions?

If you enable offline submission for PDF forms, some users may save a form once and then try to re-use it for multiple submissions. By default, Transaction Manager rejects additional submissions to avoid ambiguity and improve traceability. However, if you would like to allow your users to create multiple submissions, you can configure this on the form version level.
To enable multiple submissions for a form version, edit the form, switch to the "Form Versions" tab and edit the current version. Switch to the "PDF Options" tab and tick "Allow Multiple Submissions", then save your change.

Enabling multiple submissions for a form version

Form Submission (Manager v5.0)

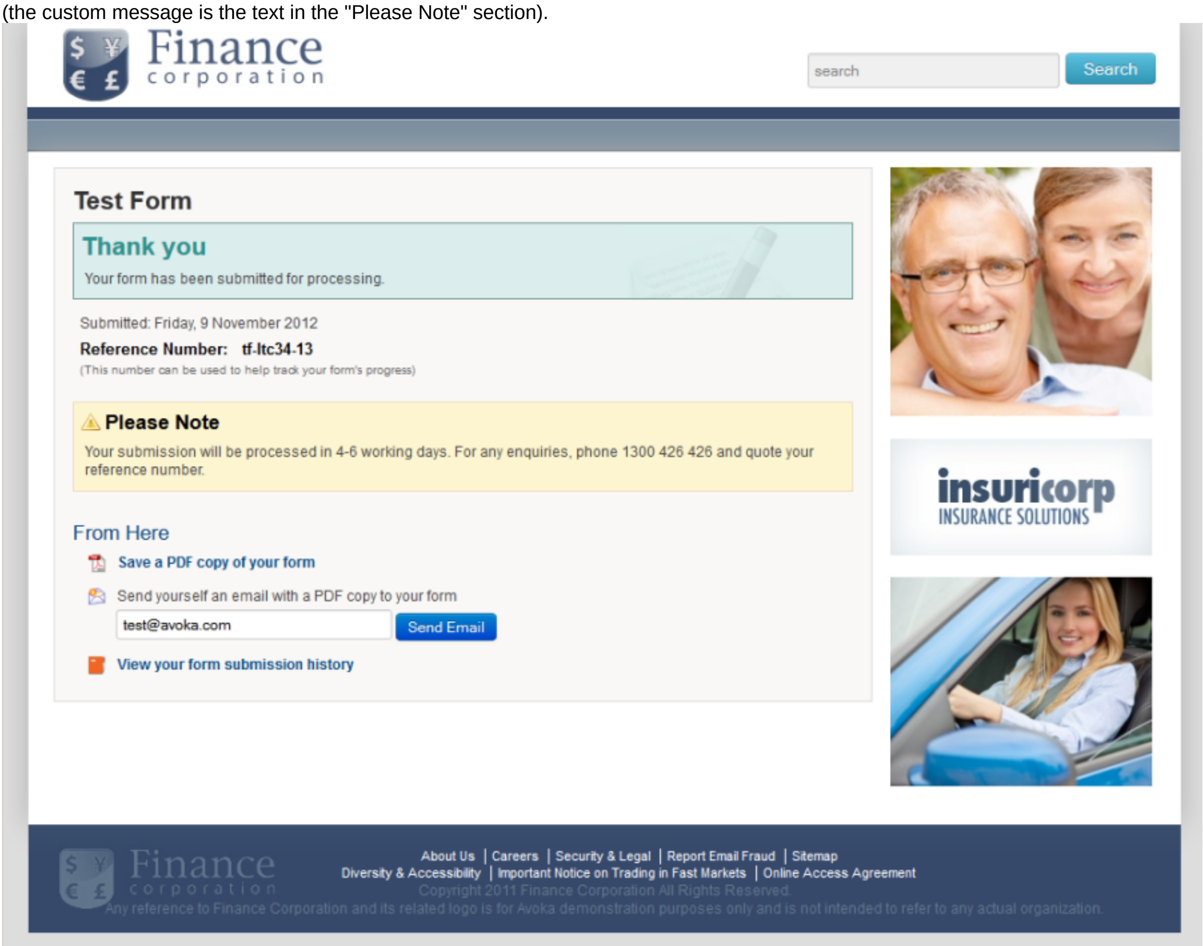
When a form is submitted by the user, the user's work may or may not be done. Complex forms may require additional steps such as attachments, payments and/or signatures. At the end of this process, however, the user is always presented with a confirmation page that offers a receipt and other important information.
This chapter discussed submission-related functionality as well as the confirmation page. Subsequent chapters will delve into optional functionality like attachments, payments and signatures.

Can I customize the content of the submission confirmation page?

There are two ways of customizing the submission confirmation page: You can either add a custom message to the submission confirmation page, or if you need to make more extensive changes, you can modify the entire HTML page template.

Adding a custom message to the submission confirmation page

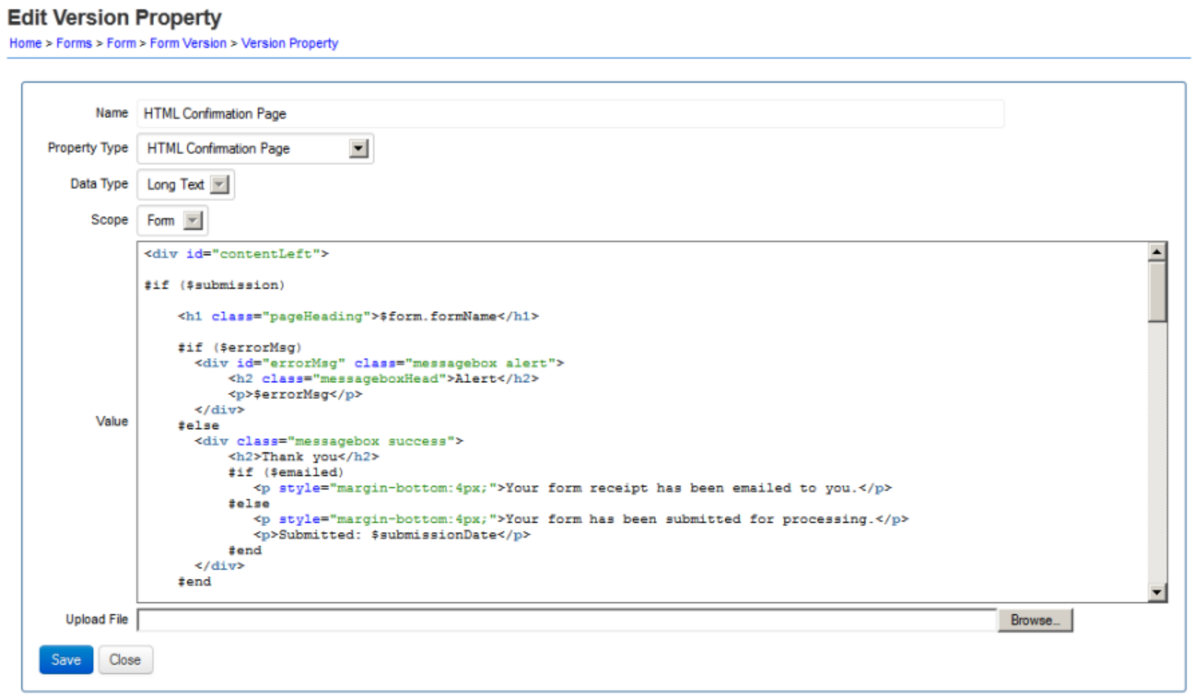
To add a custom message that will be displayed on the "Submission Confirmation" page, edit the form, switch to the "Form Versions" tab, edit the current version and switch to the "Properties" tab. Click "New", set the property type to "Submission Message" and enter your message (including HTML formatting) into the "Value" field. Save the property value.
In addition, the form has the opportunity to set or modify the submission message. When the form is rendered, the "Submission Message" form version property is populated into the "SubmissionMessage" node in the system profile. The form can modify this value as desired, and when the form is submitted, TM reads out the submission message from the form data and stores it in the submission entity to be displayed on the "Submission Confirmation" page.
If a submission message is set, the "Submission Confirmation" page will display it in addition to its usual content. See the screenshot below for an example (the custom message is the text in the "Please Note" section).



The "Submission Confirmation" page, including a custom submission message

Customizing the HTML template of the "Submission Confirmation" page (form version specific)

If you would like to make layout changes to the submission confirmation page or add new content (such as custom form version property values), you can adjust the content of the submission confirmation page on a form version level.
To do so, edit the form, switch to the "Form Versions" tab and edit the current form version. Switch to the "Properties" tab. The property you are looking for is named "HTML Confirmation Page". If it is set already for your form version, edit it; otherwise click "New" and select "HTML Confirmation Page" from the "Property Type" dropdown.



Customizing the HTML template of the submission confirmation page

When you create the property, TM will automatically load the current submission confirmation page HTML template as the initial value (as shown in the screenshot above). You can now adjust the layout and content.

To access form version properties, use the notation "${formProperties.PropertyName}", replacing "PropertyName" with the name of the property (omitting whitespace). For example, if you have added a form version property named "Contact Email Address", you could reference it in the page template using "${formProperties.ContactEmailAddress}".

Make sure to test your changes after saving the form version property. Other forms will not be affected by these changes.

Customizing the HTML template of the "Submission Confirmation" page (global)

If you would like to customize the HTML template of the "Submission Confirmation" page across the board (i.e. in a non-form-specific manner), you can edit the submission confirmation page template on the space level. This template applies to all forms that do not define the form version property "HTML Confirmation Page" when the confirmation page is viewed on that particular space.

To customize the submission confirmation page template for a space, edit the space and ensure it is editable ("Content Editable" checkbox). Switch to the "Pages" tab and edit the entry named "Submission Confirmation" (see image below for an example).

Editing the submission confirmation page for a portal

Adjust the page content and save your changes. The changed submission confirmation page will be used immediately. Ensure to test your changes carefully.

How can I customize the submission confirmation page based on the submitted data?

The standard submission confirmation page displays important information such as the receipt number to the user. You can customize the submission confirmation page as we have seen in the preceding sections, but what if you want to add more data specific to the submission (e.g. the type of application the user made, or additional instructions to the user that are specific to selections in the form)? TM addresses this by providing the set of submission data extract mappings (data extracted from the submitted XML data at submission time) on the confirmation page. You can display data extract values directly (e.g. the user's name) or use them to make decisions on what to display (e.g. additional instructions depending on application type).

If you would like to use this mechanism to customize the submission confirmation page, you will need to set up a set of data extract mappings for your form(s) first. Please refer to the TM Administration Guide, chapter "Form Version Configuration", section "Form Data Configuration Setup". Note that changes to the set of data extract mappings apply to subsequent submissions only.

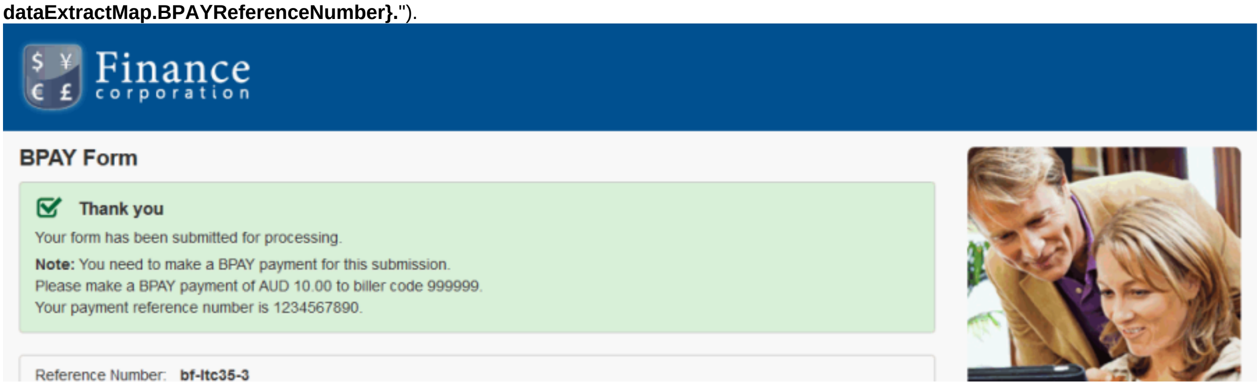
Once you have set up data extract mappings, you can customize the submission confirmation page to reference data extract values. If the data extract values are form specific, you will probably want to customize the confirmation page for your form version; on the other hand, if all forms share a set of data extract values, you may want to customize the global submission confirmation page template. Both options are described in the preceding sections.

Here is an example for Velocity directives that add logic to a message box to display additional messaging if the user will need to make a BPAY payment

after form submission (the logic regarding the additional messaging is bolded below):
<div class="alert alert-success panelMargin-left"">
<div class="row">
<i class="icon-success"></i>Thank you

#if ($session.emailed)
<p>Your form receipt has been emailed to you.</p>
#else
<p>Your form has been submitted for processing.</p>
#if (${submission.dataExtractMap.PaymentType} == 'BPAY')
**<p>Note: You need to make a BPAY payment for this submission.
Please make a BPAY payment of AUD ${submission.dataExtractMap.PaymentTotal} to biller code ${submission.dataExtractMap.BPAYBillerCode}.
Your payment reference number is ${submission.dataExtractMap.BPAYReferenceNumber}.**
#end
#end

</div>
</div>
As you can see, a specific data extract value can be accessed via ${submission.dataExtractMap.<name> where <name> should be replaced with the name of the data extract element (e.g. "PaymentType"). The example shows both conditional logic based on data extract values ("**#if (${submission.dataExtractMap.PaymentType} == 'BPAY')**") as well as simple display of data extract values ("**Your payment reference number is ${submission.dataExtractMap.BPAYReferenceNumber}.**").



Confirmation page with data extracted from the submission

The above is a very basic example. You can add much more logic and elaborate styling to your confirmation page to suit your needs.

Can I send a confirmation email to users who have submitted a form?

If the form is configured to allow users to view their form receipts, the submission confirmation page will automatically allow the user to send themselves a confirmation email including the receipt. Transaction Manager does not actively send an email, instead allowing the user to customize the email address they would like it sent to and giving them the opportunity to react to email errors (e.g. user error when entering the email address). Alternatively, users can simply download the receipt.
The next section explains how you can customize the contents of the "Submission Confirmation" email.

How can I customize the confirmation email sent to the user?

You can customize the email subject and body text of the email for each user space and overwrite these settings on an organization and/or form version level. Generally, the email subject and message are provided as Apache Velocity templates that can be customized with the following variables:

- form The form entity. You can reference attributes, e.g. "${form.formName}".
- submission The submission entity. You can reference attributes, e.g. "${submission.receiptNumber}".
- submissionExtractData The data extract values for the submission. Reference individual values as attributes, e.g. "${submissionExtractData.dataExtractKey}" if the extract value name is "dataExtractKey".
- portal The space on which the form was submitted.
- environmentName The name of the TM environment, as specified in the space property "Admin Environment Name" associated with the "Admin Console" space.

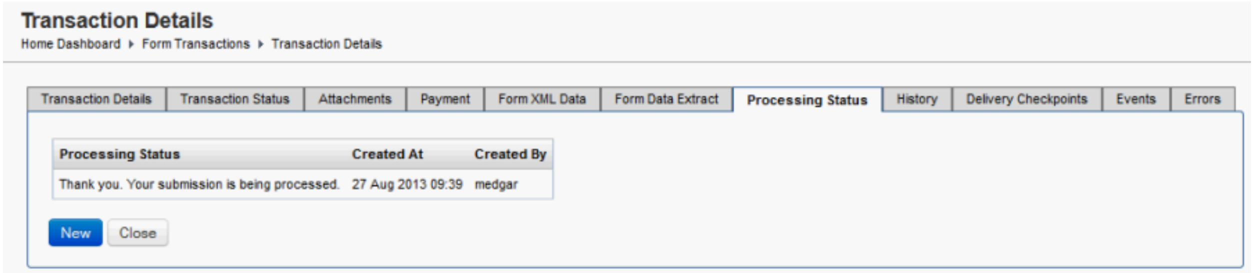
To configure the email sender, subject template and body template for a space, modify the space properties "Email Sender Address", "Email Form Receipt Subject", "Email Form Receipt Message". Note that the email sender address is not specific to the "Submission Confirmation" email.
To override the email sender, subject template and body template on the organization level, set the organization properties "Email Sender Address", "Email Form Receipt Subject" and "Email Form Receipt Message".
To override the email sender, subject template and body template on the form version level, set the form version properties "Email Sender Address", "Email Form Receipt Subject" and "Email Form Receipt Message".

Can I send the user status updates after submission?

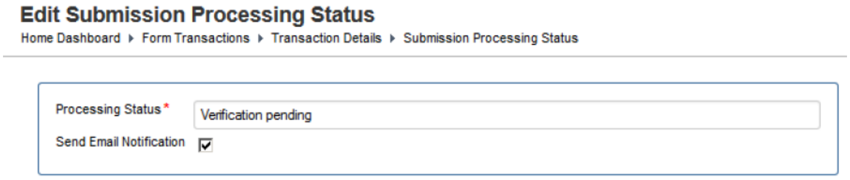
Often submission in TM is followed by further processing steps at the organization. If this is the case, you can notify users of back-office status updates as they occur. These processing status updates are displayed in the space and can also be sent to the user via email.

Adding a processing status update

There are two ways of creating processing status updates: directly via the TM management console or via a web service API. For more information about the web service method, please refer to the "Transact Web Service Guide", chapter "Forms Server Web Service".
To manually create a processing status, log on to the TM management console and access the submission log. View details for the submission you want to update and switch to the "Processing Status" tab. You will see the list of currently existing processing statuses (see the image below for an example) as well as the creation date and user. TM itself may also create some processing status values (e.g. during submission email verification).



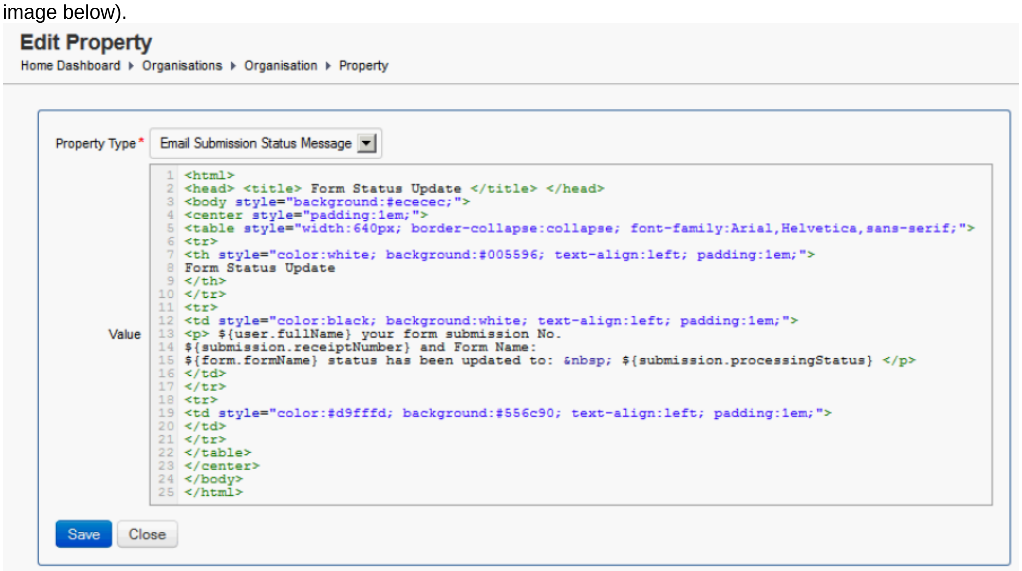
The list of processing status values for a submission
Click "New" to add a processing status (see image below).



Adding a processing status update
Enter the new processing status (e.g. "Approved"). This string will be displayed to the user.
Tick "Send Email Notification" to send an email to the user, telling them that the submission status has been updated.
Save the processing status. The new status will be displayed on the space (if the user has access to the submission list), and an email will be sent to the user if you chose that option.

Customizing processing status update emails

You can customize the processing status update email that is sent to the user on multiple levels: The global subject and message templates are defined as deployment properties (named "Email Submission Status Subject" and "Email Submission Status Message", respectively). If you adjust these values, they apply to processing status updates for all forms by default.
In addition, you can override the email subject and/or message on the organization and form level. This enables you to have a global template for most forms, but define a tailored email template for specific forms as needed.
To override the email subject / message on the organization level, edit the organization and add an organization property of type "Email Submission Status Subject" and/or "Email Submission Status Message". The default template value will automatically be populated as the property value for you to modify (see image below).



Customizing the processing status email for an organization
Edit the value and save.
Note: Editing organization properties does not immediately flow through into all form versions. In TM, form versions are isolated from changes to organization properties and other values - all changes to a form version need to be made explicitly. This helps preserve old form versions. To apply the new organization property value to a form version, edit the form version, switch to the "Properties" tab and click "Sync Org Properties".
To override the email subject / message on the form level, edit the form version and add a form version property of type "Email Submission Status Subject" and/or "Email Submission Status Message". The default template value will automatically be populated as the property value for you to modify.
Edit the value and save. All submissions using this form version will use the new template in the future, overriding values defined on the global and organization level.

Can I update a user's profile based on data they filled into my form?

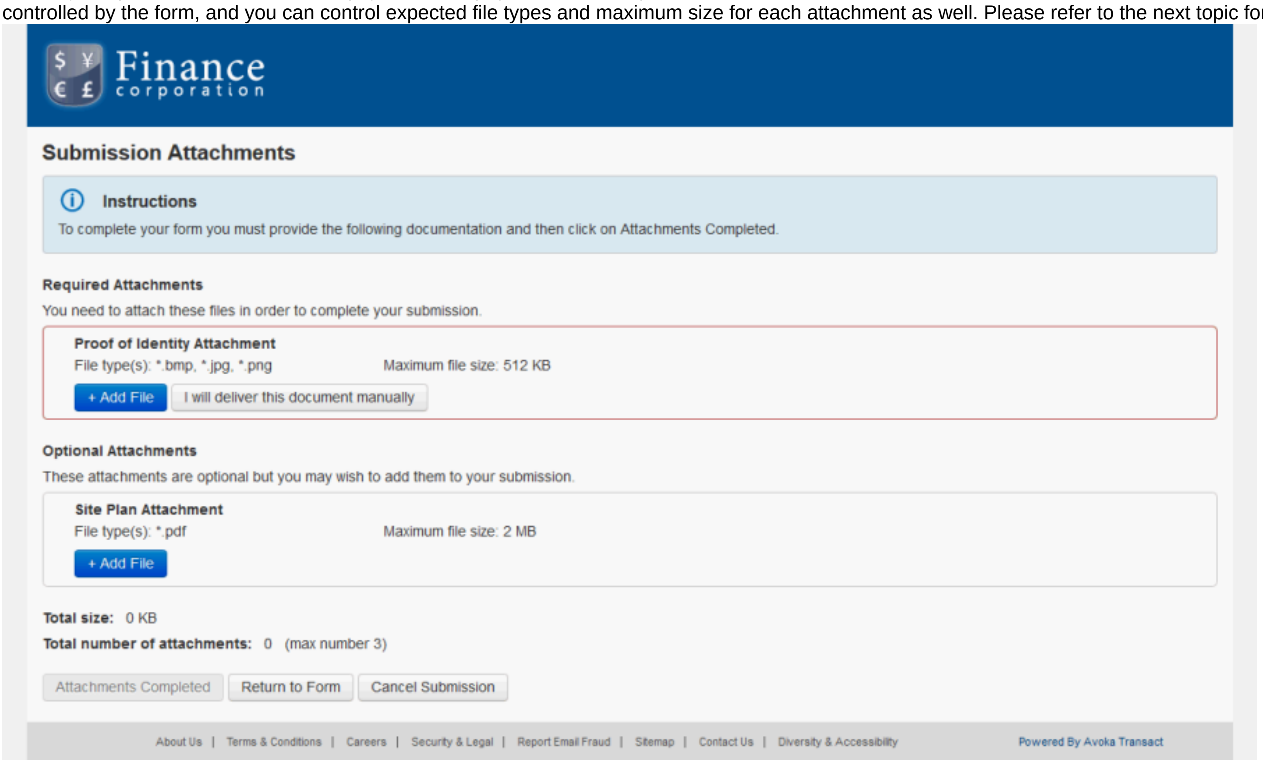
If you are using TM-managed user accounts in your user spaces, Transaction Manager not only allows your users to manually update their profiles, but also allows data that the user filled into a form to flow back into the user profile.
Whether this happens or not is the form's decision - TM checks the element "UpdateUserProfile" in the SystemProfile at submission time. If it is set to "true", TM will then look up the property prefill mappings with a scope of "User" for the form version. Each mapping that is found is applied in reverse; i.e. TM looks up the XPath in the submitted XML, and writes its value to the profile property specified in the mapping.
Therefore, this mechanism will update only those profile properties for which property prefill mappings have been set up. This mechanism also ensures that the form receives the current values for these properties at render time, via property prefill.

Attachments (Manager v5.0)

Apart from the fields filled in by the user, forms often require supporting documentation, such as proof of identity. These additional documents (here called attachments) may be provided electronically or physically (via mail, fax or personal drop-off). Transaction Manager supports all these use cases and allows you to make it easy for the user to directly upload attachments while constraining the attachment size and adding automated virus scanning of attachments using an external virus scanner.
Attachments can be dynamically requested depending on selections made in the form. Composer allows you to define attachments and the rules when each attachment will be requested, and in what format. Alternatively, you can maintain the set of attachments in Transaction Manager.

How can I allow the form to decide which attachments will be required?

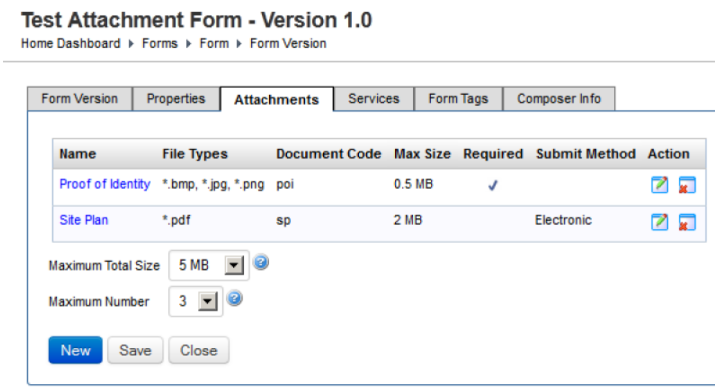
The form is always in control of the attachments that will be requested when the form is submitted. You can either define the logic around required attachments inside the form, or define a set of attachments in Transaction Manager, which is then passed to the form, which can modify the set of attachments as the user makes choices. At submission time, Transaction Manager will read the list of required and optional attachments from the submitted data, and displays an attachment page to the user.
To use the attachment functionality, your form's seed file needs to contain a special "Attachments" node where attachments will be located at submission time. By default, Transaction Manager looks for this node using the XPath "//Attachments", though the search path can be customized.
If you are using Composer forms, the "Attachments" node will be added and its contents managed for you when you add attachments to the form. Composer provides widgets that help you set up your attachments and rules to add attachments based on user input to the form. Please refer to the Composer User Guide, section "Transaction Manager Integration" for details.
Once a form that requires attachments is submitted to Transaction Manager, the user will be presented with a page allowing them to upload files or specify that they will provide the attachment in physical form (e.g. via mail or fax). Whether both these submit methods are available to the user can also be controlled by the form, and you can control expected file types and maximum size for each attachment as well. Please refer to the next topic for more details.



The attachment page presented to portal users

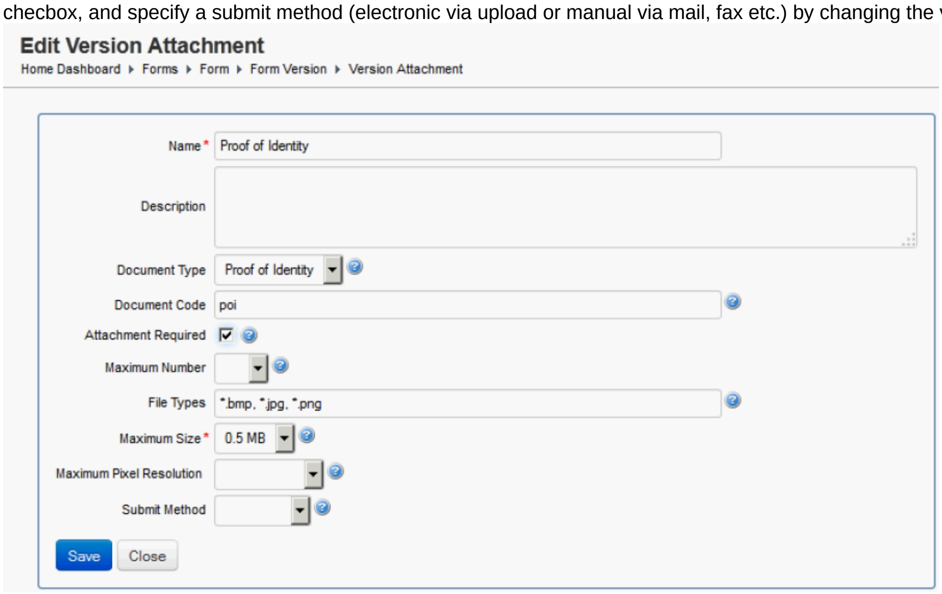
How can I prevent users from attaching too many/unsuitable files?

Each individual attachment requested by Transaction Manager can be restricted to a maximum file size and one or several file types (e.g. *.jpg). In addition, the entire set of attachments for a form submission can be restricted to a maximum size and number of attachments This can be necessary when a PDF form allows the user to attach files directly to the form, and also because users can in theory upload multiple files for each attachment defined in Transaction Manager (which allows for flexible handling of attachments where the user may have multiple files to satisfy a requirement for supporting documentation).
To restrict the total number and size of attachments that are possible for a form, edit the form in the TM management console, switch to the "Form Versions" tab and edit the current version. Switch to the "Attachments" tab.



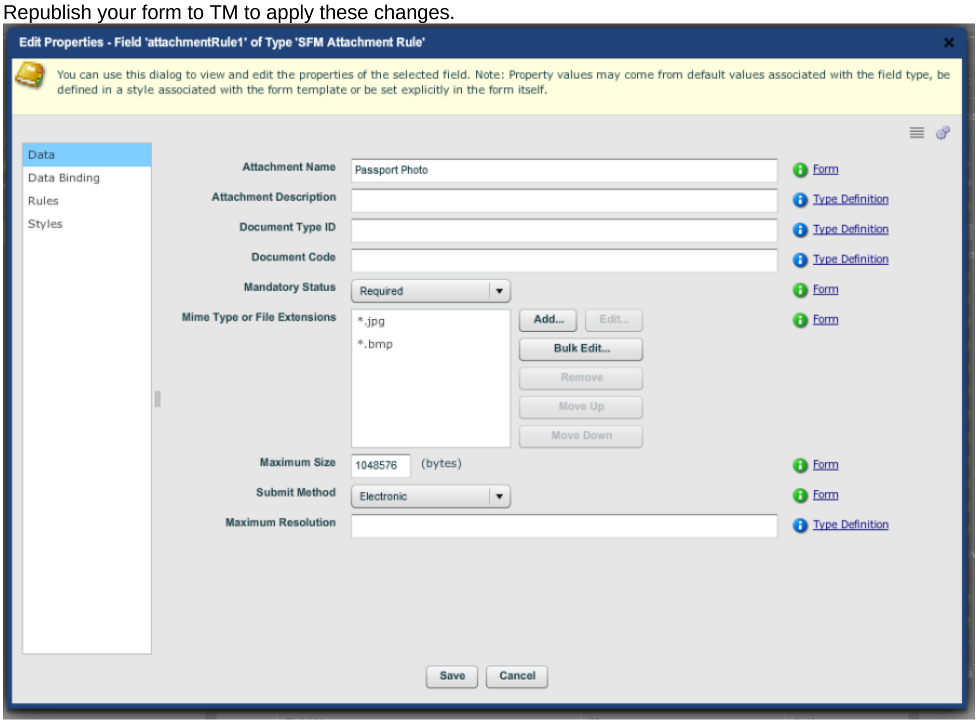
Configuring the total number and size of attachments for a form

You can restrict the total size and number of all attachments combined by entering values into the "Maximum Total Size" and "Maximum Number" fields. Save your changes. Now all future submissions for the form will be constrained to the configured maximum number and size of attachments. Constraints for individual file attachments still apply.
To control the maximum size and file types for individual attachments, you can specify a maximum size and a set of supported file types. If the attachments are set up in Transaction Manager, you can navigate to the "Attachments" tab as above and edit an individual attachment (see screenshot below). Edit the "File Types" field to restrict the file extensions users can upload, the "Maximum Size" field to restrict each attached file to a maximum size and the "Maximum Pixel Resolution" to restrict uploaded images where applicable. In addition, you can make the attachment mandatory by ticking the "Attachment Required" checbox, and specify a submit method (electronic via upload or manual via mail, fax etc.) by changing the value of the "Submit Method" field.



Editing the constraints on an individual attachment set up in SFM

If the attachments are set up in Composer, open your form in Composer and edit the attachment rule associated with the attachment. On the "Data" screen, you can enter a maximum size in bytes, add a set of file extensions to constrain the file type, make the attachment mandatory and determine the submit method.
Republish your form to TM to apply these changes.



Editing the constraints on an attachment set up in SFC

How can I guard against users uploading infected files?

Transaction Manager allows you to configure a virus scanning service. If configured, all attachments will be automatically virus scanned and processed only if no virus could be found.
Currently, Transaction Manager supports two virus scan engines out of the box: ClamAV ({+}http://www.clamav.net/+) and Symantec ({+}http://www.symantec.com/business/scan-engine+). To use either of these, you would purchase and install your engine of choice, and then configure a service definition and service connection in Transaction Manager.
Details on how to configure and activate a virus scanning service can be found in the Transaction Manager Administration Guide, chapter "Managing Form Attachments", section "Virus Scanning".

Payments (Manager v5.0)

Sometimes forms require that a payment be made after the form has been filled in, e.g. to pay for services requested. Transaction Manager supports a range of payment options, including credit card payment via a variety of providers, externally hosted payment (e.g. PayPal) and payment processed outside of TM (e.g. BPAY).
The Transaction Manager payment module does not itself process payments and does not store credit card numbers. The actual payment is processed by an external payment provider, but TM does monitor the payment provider, allows the user to reattempt payment and ensures payment is made before the submission can proceed. You can review the payment log and, if required, manually mark payments as completed via the management console.
Please refer to the TM Administration Guide for more on payments and the set of supported payment providers.

How do I add payments to a Composer form?

XXX according to David, we don't have the best solution atm

How can I enable payments for my form?

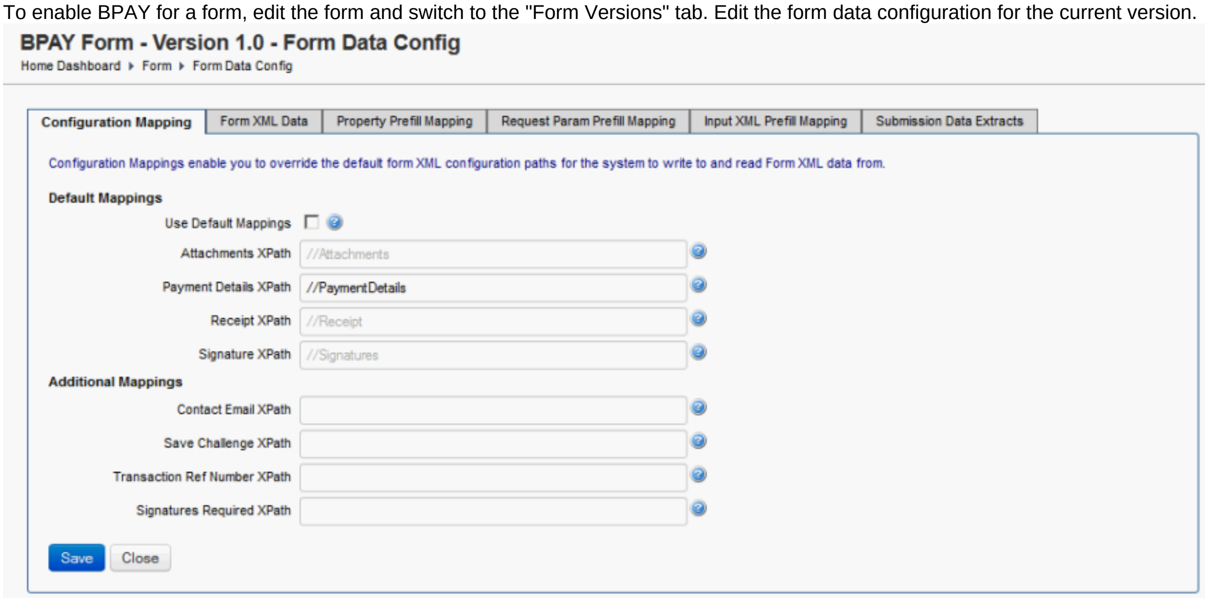
Transaction Manager can request payment for your forms from the user. The payment method and amount payable are controlled by the form (which can vary the amount depending on choices made by the user). At submission time, Transaction Manager reads the payment information from a special "PaymentDetails" node in the form.
By default, Transaction Manager looks for this node using the XPath "//PaymentDetails", though the search path can be customized.
If you are using Composer forms, the "PaymentDetails" node will be added and its contents managed for you.
There are three payment methods supported by Transaction Manager (further information about payment types and the various payment services can be found in the TM Administration Guide, section "Online Payment"):

- Credit/debit card The user pays via credit/debit card on a payment page provided by the TM user space. No credit card information is stored in TM; payment is made against a payment gateway, which then transfers the amount to the organization's bank account.
- Hosted The user is directed to an external payment provider (e.g. PayPal). The payment is made via the provider's website, and the user is then redirected back to TM along with the payment status.
- BPAY BPAY ({+}http://www.bpay.com.au/+) is an Australian system to make payments to registered organizations identified by a BPAY biller code. For each payment, the user is also given a reference number to quote on their payment. The actual payment is made outside of TM, so TM will not prompt the user for payment. The payment information should be shown to the user in the form itself (and on the form receipt), and the BPAY information can also be added to the submission confirmation page displayed in Transaction Manager.

To enable payments in a Composer form, you can add an "TM Payment Total" widget to your form. You can update this field with the amount payable as the user makes choices. If you need to declare a GST (General Sales Tax) amount, you can use the "TM GST Total" widget which contains the amount of GST that is included in the total payment.
You can allow the user to choose the payment type by adding a "TM Payment Type Selector" palette item to your form. If you would like to hard code a specific payment type, you can drop on a widget (e.g. "Data Field") and bind it to "$record.SFMData.PaymentDetails.PaymentType". You can then set its value to one of "Credit/Debit Card", "Hosted" or "BPAY" to hard code the desired payment method. There are other alternatives, such as pre-populating the available payment methods into the form.
Important note: If you set a payment total amount in your form, you MUST also specify a valid payment type (one of the values mentioned above), or Transaction Manager will reject the form submission.

How can I use BPAY with my form?

BPAY ({+}http://www.bpay.com.au+) is a way to make online or telephone payments using two identifiers: the biller code identifying the business receiving the payment, and a reference number identifying the payment itself.
Because BPAY involves the user making a bank transaction, BPAY payments cannot be monitored by Transaction Manager and there will be no payment step during submission. The organization hosting the form will need to ensure that the user makes the payment (TM does record the amount the form requested to make this easier).
To enable BPAY for a form, edit the form and switch to the "Form Versions" tab. Edit the form data configuration for the current version.



Adding a configuration mapping for the payment details node
Add a mapping for the "Payment Details XPath" to the payment element in your form. Save your changes.
Your form needs to ensure that the chosen payment details element contains the following at submission time:

- PaymentType = "BPAY"
- BPAYBillerCode = biller code of the business receiving the payment
- BPAYReferenceNumber = reference number for this payment

Here is an example of a valid payment details element that will cause TM to register a BPAY payment for $10 with a biller code of 999999 and a reference number of 1234567890:

<PaymentDetails> <Payment> <SubTotal/> <GST/> <ServiceFee/> <Total>10.00000000</Total> </Payment> <BPAYBillerCode>999999</BPAYBillerCode> <BPAYReferenceNumber>1234567890</BPAYReferenceNumber> <BSBNumber/> <EFTAccountNumber/> <ClientReferenceNumber/> <PaymentType>BPAY</PaymentType> <AccountNumber/> <ItemList/> </PaymentDetails>

TM will make a note of the payment amount and the BPAY details. However, as payment for BPAY is made via bank transfer, TM cannot monitor the payment. Therefore, payment status is set to "Completed".

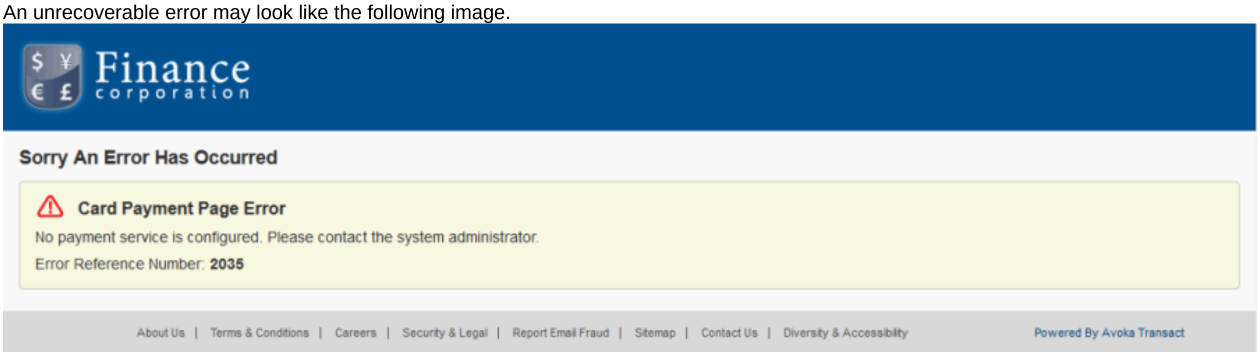
It is important to let users know that payment has to be completed outside of TM. This can be done in the form and/or by customizing the submission confirmation page for the form.

How can I diagnose payment errors?

When using Transaction Manager to process payments (e.g. credit card payments), payment errors may occur, especially during testing. These range from user errors (e.g. invalid card number) to configuration errors (e.g. organization not correctly configured for payment in TM) to connection errors or errors returned by the payment provider (e.g. insufficient funds). Where this is feasible, the user will be offered the opportunity to reattempt payment, but in some cases (e.g. configuration errors) TM prevents the user from further attempts as the error is known to be unrecoverable.

Recoverable errors generally just lead to an error message being displayed on the payment page. Unrecoverable errors mostly lead to an error being logged to the TM error log, and an error reference number being displayed to the user.

An unrecoverable error may look like the following image.

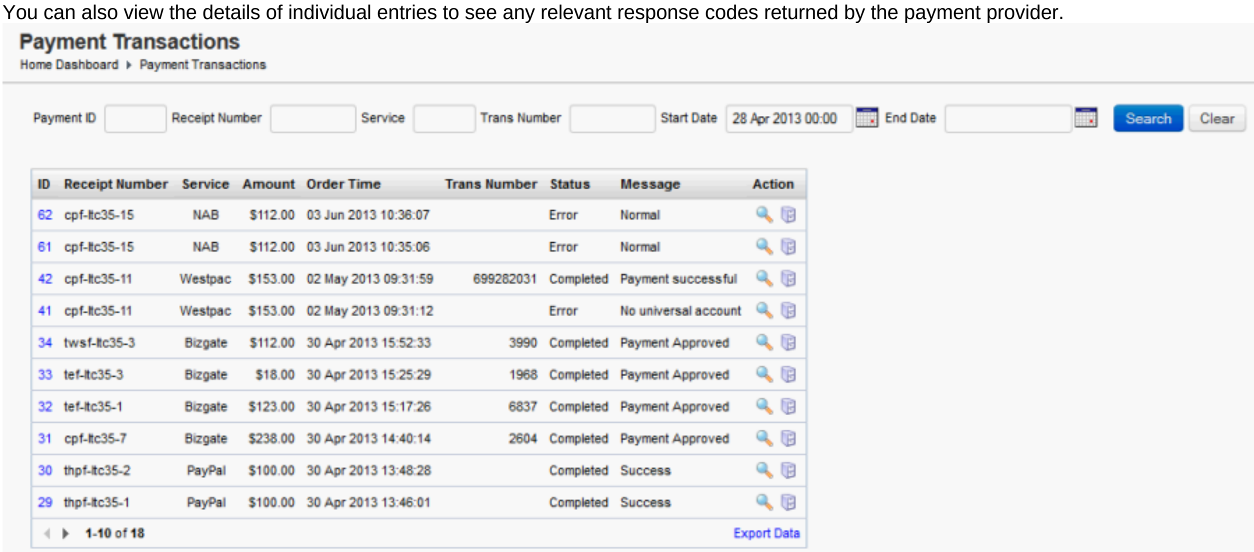


An unrecoverable payment error

The message in this case indicates that no payment service is configured for the organization in TM. To see further details about the error, log on to the TM management console and access "System > Error Log". Find the error log entry whose ID matches the reference number displayed on the payment error page. The error log entry will have additional information that can help debug the problem.

In addition, failed payment attempts will also be logged to the payment log (assuming the organization is configured sufficiently and the payment service can actually be invoked). In the management console, go to "Operations > Payment Transactions". You can find all payments for a submission by filtering by receipt number. The "Status" column indicates whether payment was successful.

You can also view the details of individual entries to see any relevant response codes returned by the payment provider.



ID	Receipt Number	Service	Amount	Order Time	Trans Number	Status	Message	Action
62	cpf-ltc35-15	NAB	$112.00	03 Jun 2013 10:36:07		Error	Normal	
61	cpf-ltc35-15	NAB	$112.00	03 Jun 2013 10:35:06		Error	Normal	
42	cpf-ltc35-11	Westpac	$153.00	02 May 2013 09:31:59	699282031	Completed	Payment successful	
41	cpf-ltc35-11	Westpac	$153.00	02 May 2013 09:31:12		Error	No universal account	
34	twsf-ltc35-3	Bizgate	$112.00	30 Apr 2013 15:52:33	3990	Completed	Payment Approved	
33	tef-ltc35-3	Bizgate	$18.00	30 Apr 2013 15:25:29	1968	Completed	Payment Approved	
32	tef-ltc35-1	Bizgate	$123.00	30 Apr 2013 15:17:26	6837	Completed	Payment Approved	
31	cpf-ltc35-7	Bizgate	$238.00	30 Apr 2013 14:40:14	2604	Completed	Payment Approved	
30	thpf-ltc35-2	PayPal	$100.00	30 Apr 2013 13:48:28		Completed	Success	
29	thpf-ltc35-1	PayPal	$100.00	30 Apr 2013 13:46:01		Completed	Success	

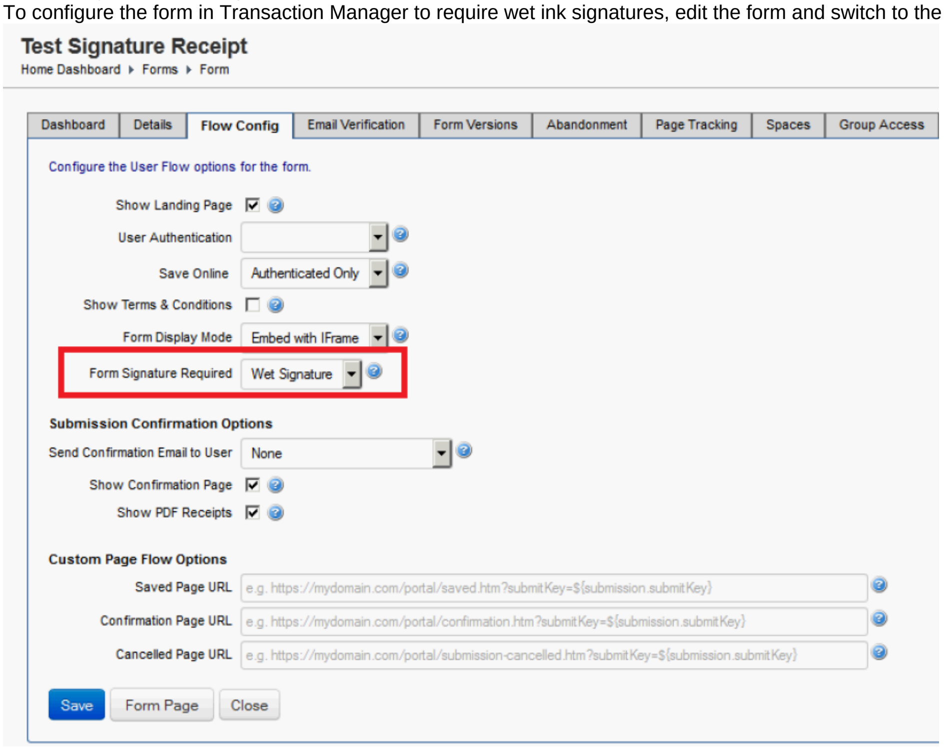
The list of payment attempts

Signatures Options (Manager v5.0)

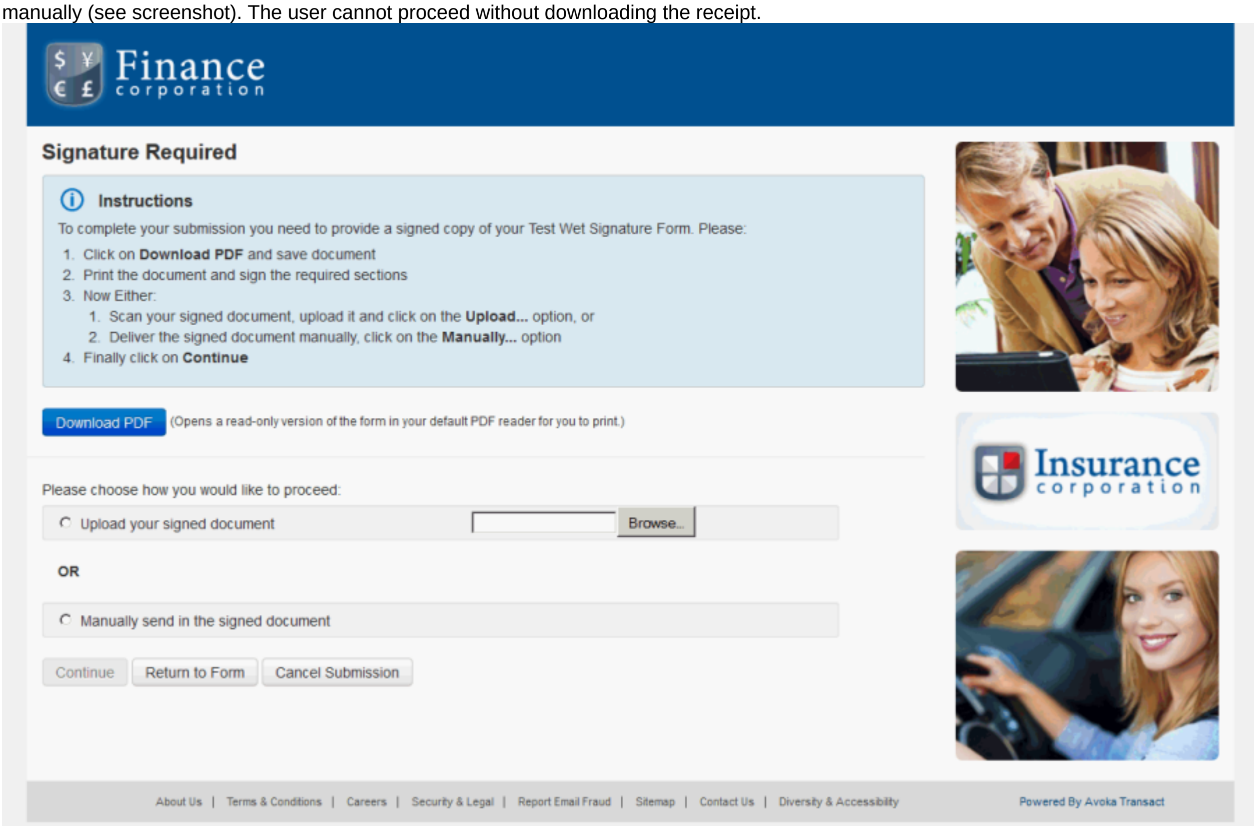
Like paper forms, SmartForms often need to request the user's signature to confirm their submission. Transaction Manager can provide multiple ways to enable user signatures.
Signature methods are discussed in the Transaction Manager Administration Guide, chapter "Signatures".

How can I ask for wet ink signatures after form submission?

Transaction Manager provides a specialized wet ink signature page as an optional post-submission step. To configure wet ink signatures for a form, you can change a form setting in the TM management console, or modify your form to set an element in the "Signatures" section. The latter allows the form to decide whether signatures are required.
To configure the form in Transaction Manager to require wet ink signatures, edit the form and switch to the "Flow Config" panel.



Enabling wet ink signatures for a form
Select "Wet Signature" from the "Form Signature Required" dropdown and save.
Now, whenever a user submits the form, they will be asked to download and sign the receipt, and then either directly attach the scanned receipt or send it in manually (see screenshot). The user cannot proceed without downloading the receipt.



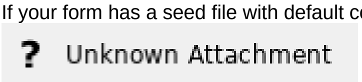
The wet signatures page on the user portal
Please refer to the next section to learn how you can let the form decide whether signatures are required.

Can the form decide whether signatures will be required?

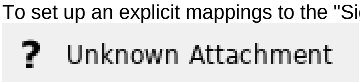
In addition to configuring signature options in Transaction Manager, the form itself can also decide if signatures will be required, and if so, what signature method is to be used. This gives you greater flexibility in the form (e.g. signature required only in some cases, user can choose how to sign the form).
The form can communicate the information regarding signature type via a specialized section named "Signatures" in TM. The section has the following structure:
<Signatures>
<SignatureTypeSelected />
</Signatures>
The form can set three different values for the "SignatureTypeSelected" element to indicate that signatures are required:

- Wet The user must provide a signed copy of the form receipt. The example spaces in TM provide a wet signature page where the user can download the receipt and choose how they will provide the signed receipt (see previous section for a screenshot).
- ESignature The user is presented with an electronically signed receipt and can choose whether to accept the receipt or cancel the submission. The example user spaces provide an eSignature page to accomplish this.
- Digital Digital signatures using a certificate can be done in the form. If the "SignatureTypeSelected" is "Digital", Transaction Manager assumes that signatures have been completed in the form and sets the submission status accordingly. No further request for signatures will be made of the user.

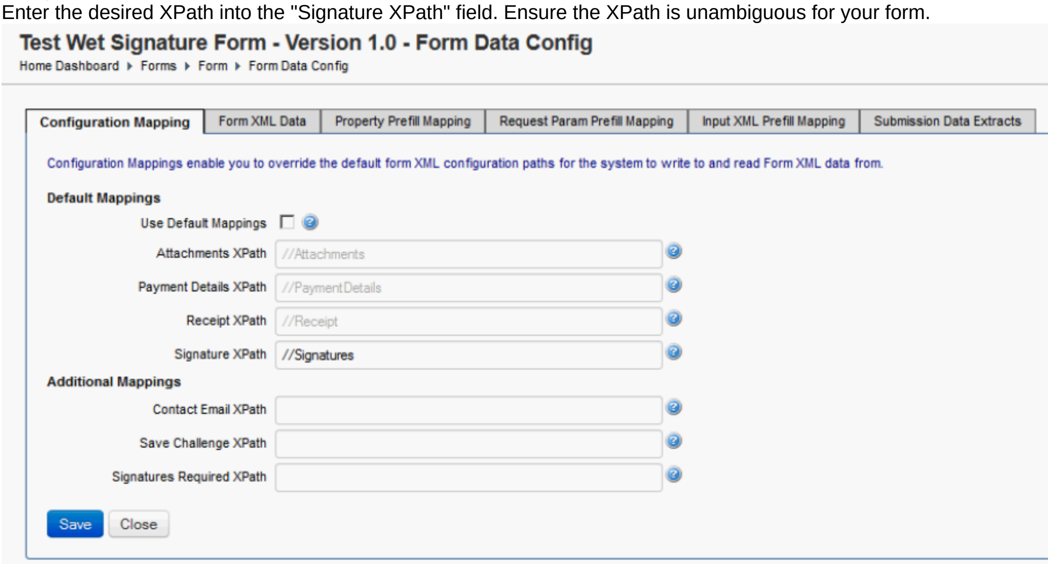
If any other value (including no value) is used by the form, the value will be ignored, and if a signature method has been set up for the form in the TM management console, it will be used.
If your form has a seed file with default configuration mappings (edit the form, switch to "Form Versions" tab, click "Form Data Configuration" (



) next to the current version and check if "Use Default Mappings" is ticked), then the "Signatures" element is looked up using the XPath "//Signatures". This means that your element can be located anywhere in the seed file and must be named "Signatures". If this is not the case, or if you have multiple "Signatures" elements in your seed file, you will need to specify the mapping explicitly.
To set up an explicit mappings to the "Signatures" element, edit the form, switch to "Form Versions" tab and click "Form Data Configuration" (



) next to the current version.
Untick "Use Default Mappings" so you can edit the mapping. **Note:** If your form uses attachments, payment or a receipt section populated by TM, ensure you enter mappings for the corresponding XPaths as well, since default mappings no longer apply. You can re-enter the default mappings, which are displayed as a placeholder in the fields.
Enter the desired XPath into the "Signature XPath" field. Ensure the XPath is unambiguous for your form.



Mapping the "Signatures" element
Save the mappings.
In Composer, you can use a wet signature widget to add a signature block, which will allow users to sign the receipt, and you can also set the selected signature type to a value of your choosing.

Receipts (Manager v5.0)

Receipts are often an important part of the form submission process, providing the user with an uneditable copy of the data they submitted. Transaction Manager generates PDF receipts for all form submissions, along with a unique reference number (the receipt number) that identifies the submission. You can configure whether or not to allow the user to access the receipt, and include it in submission delivery.

How do I include a custom barcode in the PDF receipt?

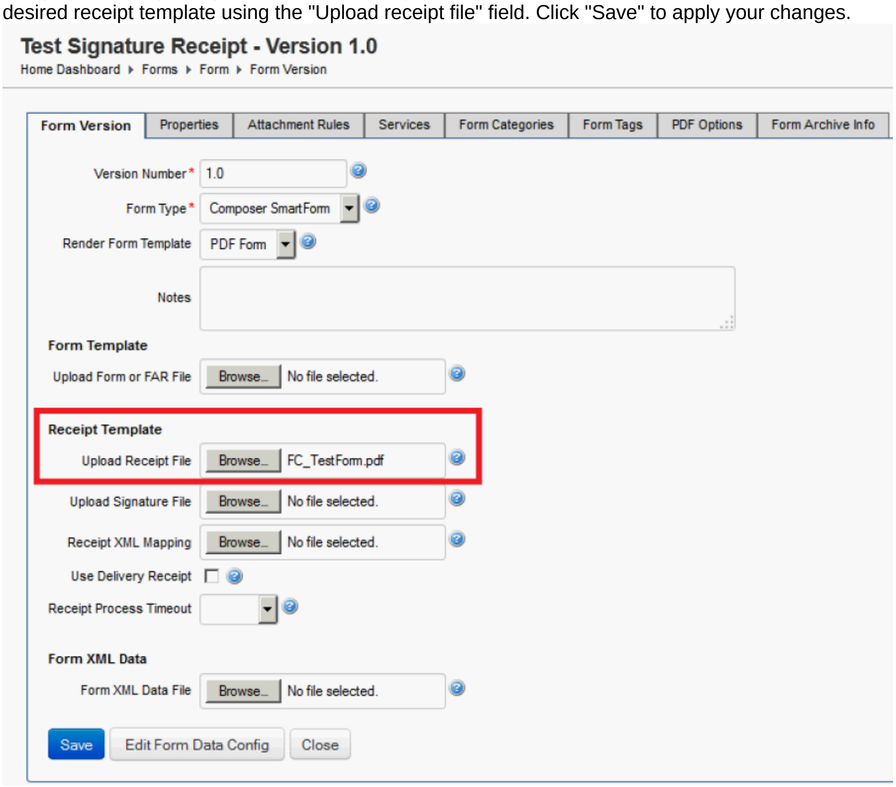
XXX get more info from Composer team

Can I control how receipt numbers are generated?

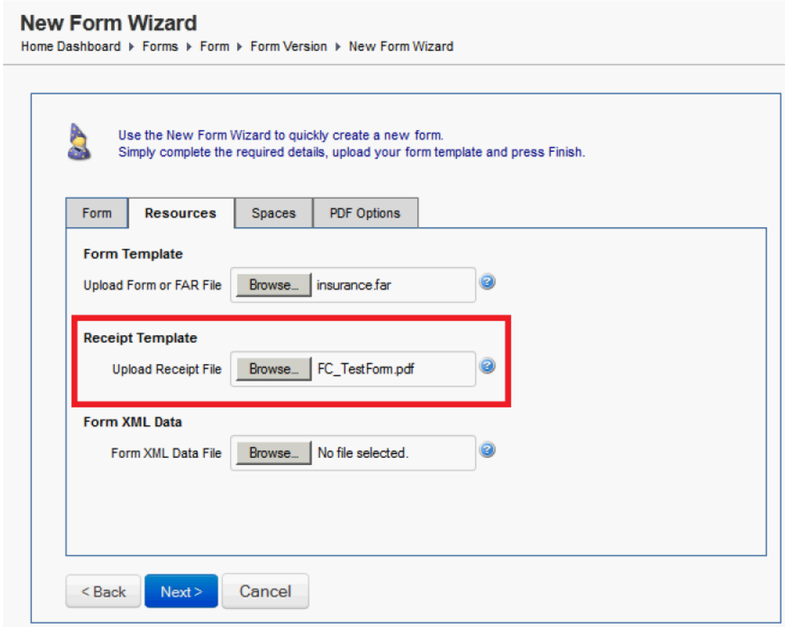
Transaction Manager offers you multiple ways to generate receipt numbers for a form. You can specify a parameterized pattern that TM will use to create the receipt number, invoke a web service you provide, or implement your own receipt number algorithm using a Groovy script. Groovy is a very simple but quite powerful scripting language that will be familiar to Java or C# programmers.
Receipt number generation is configured on a form level, and uses service definitions defined globally in Transaction Manager (allowing re-use of receipt number services between forms).
The Transaction Manager Administration Guide discusses receipt number generation in detail and shows you how to configure your form accordingly. Please refer to chapter "Form Configuration", section "Receipt Number Generation".

Can I use a separate PDF template for the receipt?

You can use a separate PDF file as a receipt template for all form types. For Composer SmartForms, a PDF receipt template can be automatically generated by Composer (depending on the template used by your Composer forms).
Note: Ensure your receipt template uses the same form data structure as the form template. Otherwise the receipt may not render correctly.
To upload a receipt template in the TM management console, edit the form and switch to the "Form Versions" tab. Edit the current version and upload the desired receipt template using the "Upload receipt file" field. Click "Save" to apply your changes.



Specifying a receipt template for an existing form version
When creating a new form using the "New Form" wizard, you can upload a receipt template on the "Resources" tab (if a suitable receipt template is not already contained in the FAR file for Composer SmartForms).



Specifying a receipt template on the "New Form" wizard

The PDF receipt won't render, what can I do?

XXX test harness, logs

Can I use HTML receipts?

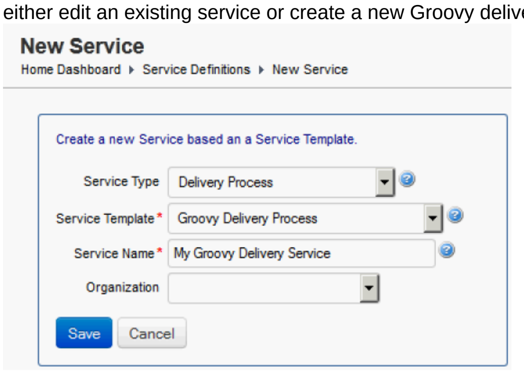
XXX what is our policy here? how does mixed mode work (is there a system defaultable service that can handle both? or do you have to decide on one and configure ALL other forms explicitly?)
XXX PhantomJS, how to set up service, Composer side: enable this in the template (is enabled in Maguire template only), totally transparent to form developer, at publish time Composer includes a .dat file (encrypted HTML) as the receipt template

Delivery (Manager v5.0)

Once the user submits a form (and also completes all subsequent steps such as payment, attachments and signatures), the submission is delivered to a backend system. There are multiple ways to deliver submissions built into Transaction Manager, and you can configure delivery on a form level.
Transaction Manager can be configured to automatically purge submission data from the TM database after it has been delivered.

How can I deliver submitted form data to my own database?

There are multiple delivery methods supported by Transaction Manager. These range from simple email delivery to web service delivery as well as custom delivery via a Groovy script delivery service.
The Transaction Manager Administration Guide, chapter "Delivery Configuration" discusses all available delivery methods as well as how to configure delivery for a form. This section will explain custom delivery using a Groovy script in further detail. Groovy is a very simple but quite powerful scripting language that will be familiar to Java or C# programmers. It is used in Transaction Manager to allow administrators to define and modify behavior at runtime.
Custom delivery allows you to adjust the delivery logic by modifying a Groovy script. This allows you to deliver submission data to a wide range of backend systems and customize exactly what data will be delivered. Because you can provide and modify the delivery logic, this is an extremely flexible delivery method and can be used to address a multitude of special requirements. If the logic needs to change, the Groovy script can be modified at runtime.
To set up a new Groovy delivery service in Transaction Manager, go to "System > Service Definitions" and filter by service type "Delivery Process". You can either edit an existing service or create a new Groovy delivery service based on a template. In the latter case, click "New".



Creating a Groovy delivery service based on a template
Choose "Delivery Process" from the "Service Type" dropdown. Choose a suitable service template (e.g. "Groovy Delivert Process"). Enter the name of the new service into the "Service Name" field and click "Save".
Please refer to the TM Groovy services guide ("System > Transact Services Guide") for details and examples.

Can I deliver submission data to SalesForce CRM?

You can publish to SalesForce CRM as well as a multitude of other systems using the Groovy script based delivery mechanism in Transaction Manager.
The following Groovy script shows how submission data can be delivered to SalesForce. Please refer to the previous section to learn how to set up a Groovy delivery service.

```
import groovyx.net.http.*
import groovy.json.JsonSlurper;
import static groovyx.net.http.ContentType.JSON
import static groovyx.net.http.Method.POST
// Get contact details
def AvokaSmartForm = new XmlParser().parseText(submissionXml);
def model = [:]
model.email = AvokaSmartForm.YourDetails.Name.Email.text()
model.firstname = AvokaSmartForm.YourDetails.Name.Firstname.text()
model.surname = AvokaSmartForm.YourDetails.Name.Surname.text()
model.organisationName = AvokaSmartForm.YourDetails.Name.OrganisationName.text()
model.country = AvokaSmartForm.YourDetails.Name.Location.Country.text()
model.contactPhoneNo = AvokaSmartForm.YourDetails.Name.ContactPhoneNo.text()
model.description = AvokaSmartForm.Enquiry.QuestionsComments.Questions.text()
model.state = AvokaSmartForm.YourDetails.Name.Location.State.text()
def eventLogService = new com.avoka.fc.core.service.EventLogService()
// Create Lead in SalesForce
// initialze a new builder and give a default URL
def http = new HTTPBuilder('https://www.salesforce.com/servlet/')
def postBody = [
oid: 'XXXXX', // replace with correct value
first_name: model.firstname,
last_name: model.surname,
email: model.email,
phone: model.contactPhoneNo,
company: model.organisationName,
state: model.state,
country: model.country,
lead_source: 'Test Enquiry',
Campaign_ID: '108A00000001XYZ',
description: model.description,
]
http.post( path : 'servlet.WebToLead?encoding=UTF-8',
body : postBody,
requestContentType: ContentType.URLENC ) { resp ->
eventLogService.logInfoEvent("SalesForce Response: " + resp.statusLine, submission)
}
```

The script needs to be adjusted to suit your circumstances. The example implementation publishes data extracted from an enquiry form into SalesForce and logs an information event against the submission when this has happened.

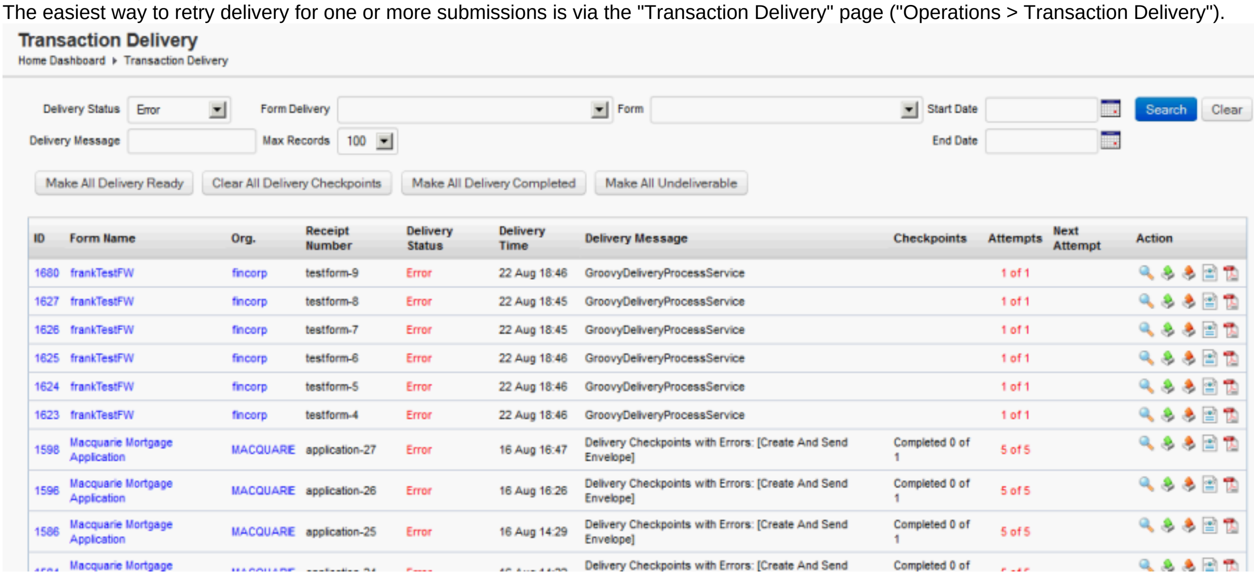
Can I call a web service to deliver the submitted form data?

You have three options to use a web service for submission delivery:

1. **Use web service delivery if you can provide a web service implementing the Forms Server web service.** Refer to the Transaction Manager Web Service Guide for more information on these APIs. The Transact Integration Agent is a Transact component that is installed on a client site and provides an out-of-the-box implementation along error handling, push or pull delivery, and configurable delivery of submission data on to your backend.
2. **Use Groovy delivery to invoke arbitrary web services.** This option requires you to write a Groovy script that calls the web service and handles delivery. This method allows you to integrate with a web service of your choice, but the implementation needs to be considered carefully (e.g. error handling) and will require thorough testing. To achieve this, create a new Groovy delivery service as described in the previous section and customize the Groovy script.
3. **Use REST delivery to pull down submissions on demand** This option allows you to call a REST endpoint to retrieve submission data. Please refer to the Transact services guide in the TM management console ("System > Transact Services Guide") for details and examples.

How do I retry delivery if there is an error?

While submission delivery is done automatically by TM according to the delivery channel configured, there is a multitude of delivery errors that may occur (e.g. network outages, constraint violations in backend systems, errors in Groovy delivery scripts). Once the source of the error has been found and addressed, submissions that encountered delivery errors need to be redelivered. Depending on submission volume and length of the outage, many submissions may be affected.
TM offers a range of tools to manage submission delivery retry, ranging from automated retries to manual delivery retry (for individual submissions or in bulk). Groovy services can make use of delivery checkpoints to record which delivery steps have been successfully completed, and which ones have to be executed during a retry.
The easiest way to retry delivery for one or more submissions is via the "Transaction Delivery" page ("Operations > Transaction Delivery").



ID	Form Name	Org.	Receipt Number	Delivery Status	Delivery Time	Delivery Message	Checkpoints	Attempts	Next Attempt	Action
1680	frankTestFW	fincorp	testform-9	Error	22 Aug 18:46	GroovyDeliveryProcessService		1 of 1		
1627	frankTestFW	fincorp	testform-8	Error	22 Aug 18:45	GroovyDeliveryProcessService		1 of 1		
1626	frankTestFW	fincorp	testform-7	Error	22 Aug 18:45	GroovyDeliveryProcessService		1 of 1		
1625	frankTestFW	fincorp	testform-6	Error	22 Aug 18:46	GroovyDeliveryProcessService		1 of 1		
1624	frankTestFW	fincorp	testform-5	Error	22 Aug 18:46	GroovyDeliveryProcessService		1 of 1		
1623	frankTestFW	fincorp	testform-4	Error	22 Aug 18:46	GroovyDeliveryProcessService		1 of 1		
1598	Macquarie Mortgage Application	MACQUARE	application-27	Error	16 Aug 16:47	Delivery Checkpoints with Errors: [Create And Send Envelope]	Completed 0 of 1	5 of 5		
1596	Macquarie Mortgage Application	MACQUARE	application-26	Error	16 Aug 16:26	Delivery Checkpoints with Errors: [Create And Send Envelope]	Completed 0 of 1	5 of 5		
1586	Macquarie Mortgage Application	MACQUARE	application-25	Error	16 Aug 14:29	Delivery Checkpoints with Errors: [Create And Send Envelope]	Completed 0 of 1	5 of 5		
[illegible]	Macquarie Mortgage	[illegible]	[illegible]	[illegible]	[illegible]	Delivery Checkpoints with Errors: [Create And Send	Completed 0 of	[illegible]		

The "Transaction Delivery" page
You can filter by various search criteria, chief among them delivery status. If you search for delivery status "Error", you will see the set of submissions for which delivery has failed (note that not all records may be shown - you can increase the value of the "Max Records" dropdown). As shown in the screenshot above, the delivery time and error message are shown. Information about the number of completed delivery checkpoints and the number of automated retries will be shown where applicable.

You can then either retry delivery for individual submissions (to retry delivery as is and to clear out all delivery checkpoints and then retry delivery) or change the delivery status for all search results via the buttons above the table (e.g. "Make All Delivery Ready" to retry delivery for all submissions shown). Monitor the results to ensure delivery is completing normally.

How can I deliver form data submitted to a TM cloud instance to an Enterprise Information System in another datacentre?

Enter topic text here.

How can I transform the form XML data into an enterprise XML data schema?

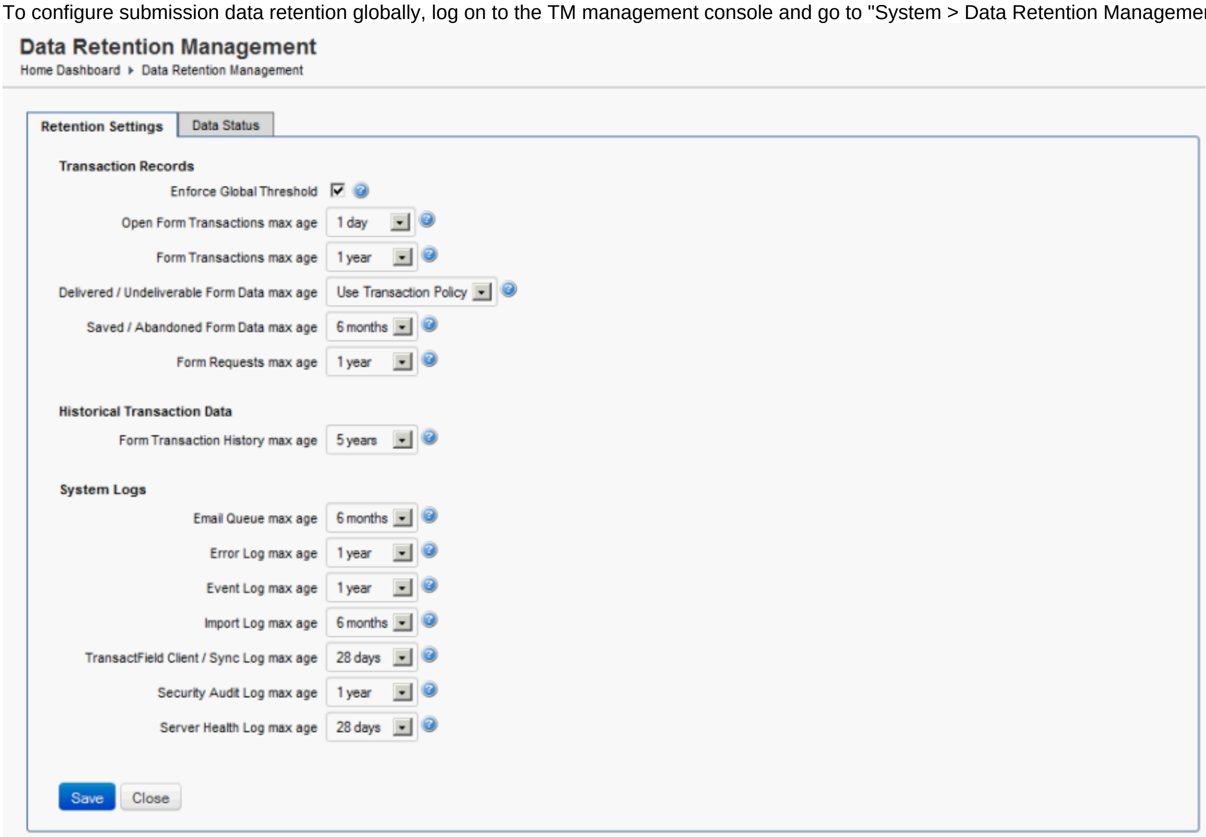
Enter topic text here.

How can I purge form submission data from TM once it has been delivered into our system?

TM gives you a wide range of options around data retention. You can specify these settings globally, and override some important submission data retention settings for specific organizations if needed. Please refer to the TM Administration Guide, chapter "Operations", section "Data Retention".

Configure global retention settings

XXX This is covered in the admin guide and doesn't add anything here
To configure submission data retention globally, log on to the TM management console and go to "System > Data Retention Management".

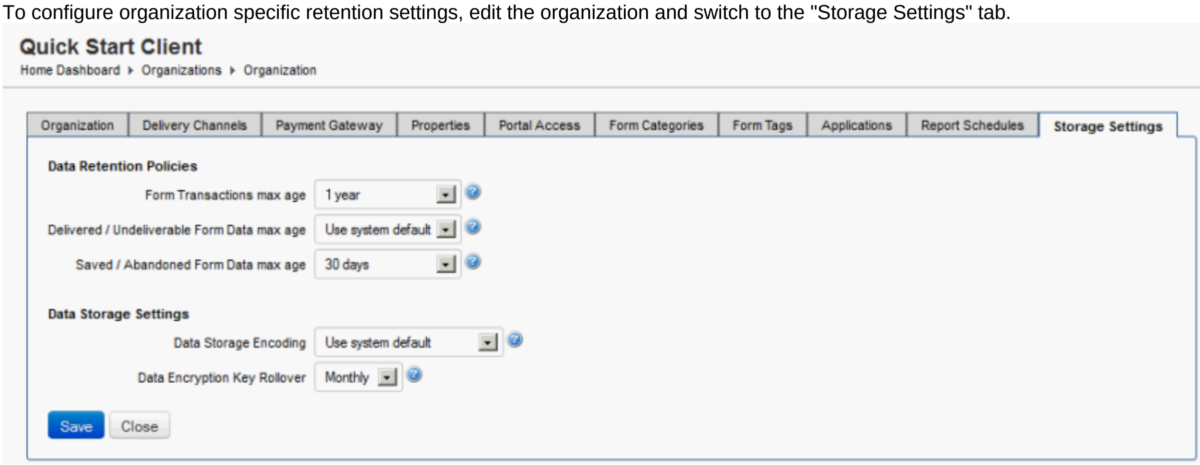


Global data retention settings

The first section, "Transaction Records", concerns submission and request log data. You can specify retention settings separately for saved (including abandoned) submissions, delivered (including undeliverable) submissions and all form submission (which includes saved and delivered submissions as well). In addition, you can specify how long request log entries shall be retained.
Most of the above settings can be overridden on the organization level (see the next section). If you would like to enforce the global data retention thresholds as the maximum values that can be chosen on the organization level, you can tick the "Enforce Global Threshold" checkbox.
Records exceeding the maximum configured age will be deleted from Transaction Manager. Ensure you have suitable delivery settings in place for all forms to avoid data loss.
In addition to maintaining submissions and form requests with associated data, Transaction Manager also maintains summary tables for statistical purposes. These tables are much more compact than the submission and request log, and are used by various reports and the dashboard in Transaction Manager. You can configure the retention policy for these summary tables in the "Historical Transaction Data" section.
Save your changes.

Configure organization specific retention settings

To configure organization specific retention settings, edit the organization and switch to the "Storage Settings" tab.



Organization-specific data retention settings

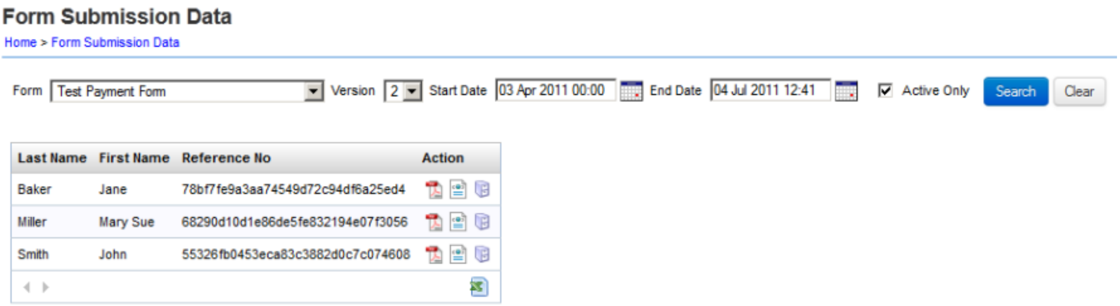
The first section, "Data Retention Policies", allows you to adjust the submission data retention settings for saved, delivered and all submissions. The "Use system default" option means that global data retention settings will be used; alternatively, choose a different value to override the global policies for this organization.
Save your changes.

Reporting (Manager v5.0)

Transaction Manager includes extensive reporting capabilities including a dashboard and submission data summary reports built into the management console, as well as flexible reports based on the BIRT framework that can be viewed on-screen or emailed out on a schedule.
The TM Administration Guide talks about the BIRT reports (chapter "Reports") and charts (chapter "Operations", section "Charts") in detail.

Can I create an summary Excel report based on submitted data?

Transaction Manager includes support for extracting values from the submitted XML at submission time and storing them to the TM database, where they can be queried and reported on more easily. In addition, TM provides a summary page in the management console where a tabular representation of the data extracted for a specific form can be viewed and exported to Excel.
To learn how to set up data extract mappings for a form, please refer to the Transaction Manager Administration Guide, chapter "Form Version Configuration", section "Setting Up Data Extract Mappings". Once set up, TM will store data extracted from the form at submission time.
To view the extracted data in a tabular summary in the TM management console, go to "Operations > Form Submission Data". Select the form and version from the respective dropdowns, and optionally filter by submission date.



Viewing data extracted from submissions for a form version

To obtain an Excel report based on the search results, click the icon at the bottom right of the table.

Task Creation (Manager v5.0)

TM allows you to assign tasks to users and user groups. These tasks can reference a form that is to be filled in, and optional prefill data, or they could reference an existing submission that is to be reviewed. Tasks can be created via the management console or a web service interface. For details about the Task / User web service please refer to the TM Web Service Guide.
Note: This chapter references version 2 of the Task / User web service specifically. Version 1 is still exposed by TM, but this is for backward compatibility reasons only, and version 2 should be used for new projects.

Can I create a task to review an existing submission?

Transaction Manager allows you to create review tasks, which are based on a previous submission, as well as form tasks.
Using the Transaction Manager Task / User web service API version 2, you can use the CreateFormReviewTask operation to create a review task. The operation is described in detail in the TM Web Service Guide.
You can also create tasks using the TM management console. To create a review task, go to "Operations > Assigned Tasks" and click "New". Select "Review" from the "Task Type" dropdown and enter the OID of the existing submission into the "Previous Submission ID" field. If you do not provide form XML data, the XML associated with the existing submission will be passed into the task form by default. Fill in the other fields as usual.

How can I create a task using the web service API?

XXX

Can I assign tasks when using the Web Plugin?

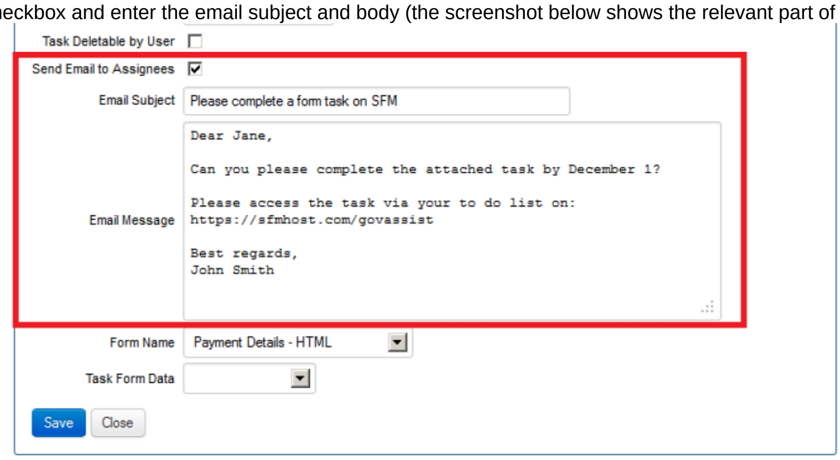
Enter topic text here.

Can I add file attachments when creating a task?

Enter topic text here.

How can I notify assignees of a new task?

Assignees will see new tasks appear in their To Do list on the user space. In addition, you can have an email sent to the assignee(s) when the task is created.
When using the Task / User web service version 2, the two "Create*Task" operations contain parameters that allow you to control whether an email will be sent ("sendEmailFlag") as well as the email subject and message ("emailSubject", "emailMessage". For more details on these operations, please refer to the Transaction Manager Web Service Guide.
When creating a task using the TM management console ("Operations > Assigned Tasks", click "New"), you can tick the "Send Email to Assignees" checkbox and enter the email subject and body (the screenshot below shows the relevant part of the task creation page).



Email settings for a new task

Old topics (Manager v5.0)

Enter topic text here.

Portal Integration Options

Transaction Manager offers a variety of options to expose forms to your target audience. You can use the Self Service portal, which provides a complete solution for end-to-end form fill-in. If you already have a web portal into which you would like to integrate form fill-in, you can easily add calls to the Web Plugin module to render forms. If you wish to provide your users with sophisticated form lookup as well as offering existing resources, you can use the Finder module to provide custom form and resource finding.

Using the SmartForm Web Plugin

This chapter provides information about the SmartForm Web Plugin, including ways to integrate the web plugin with your own web environment.
XXX review and rework entire chapter

Introduction

The SmartForm Web Plugin is a JavaScript library that users can incorporate into their own web environment. It provides functionality to link to or embed specific forms hosted in SmartForm Manager.
The SmartForm Web Plugin is generally used if you already have your own branded web portal and want to embed forms inside it without styling your own SmartForm Manager Self Service Portal. Note that the Web Plugin supports only unauthenticated (or externally authenticated) transactions and does not provide a personalized user area.

Integrating with the Web Plugin

The SmartForm Web Plugin is used by referencing the "smartform-webplugin.js" script from the web-plugin portal, then referencing either the "Avoka. SmartForm.WebPlugin.showEmbeddedForm()", or the Avoka.SmartForm.WebPlugin.showPopupForm() function from the script in your web pages, depending on what you are trying to achieve. Both functions require at least 3 common parameters; formCode, userAccountId, and redirectTarget. Further parameters can be supplied to include prefill data for the form, or specify window options for popup forms, as described in the following sections.

Using Popup Forms

Links in a web page to a popup form can be implemented with the following code:

Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'

Using Embedded Forms

Forms can be embedded into a web page using the following code:

Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'

Parameters

Both popup and embedded forms require 3 common parameters:

- formCode The unique form code of the target form.
- userAccountId The user's login name on the portal (XXX this is presumably not required - or does it need to be the empty string for anonymous requests?)
- redirectTarget The URL of your submission confirmation page.

Additionally, data can be prefilled into the form using a "prefillData" parameter. This parameter should contain a list of name/value pairs. In order for these parameters to be mapped into the form at runtime, you will need to set up request parameter prefill mappings with matching names for the form's data configuration. Please refer to the SmartForm Manager Administration Guide, section "Setting up Request Parameter Prefill" for information on how to set up these mappings.
The following example shows data prefilled into a popup form.

Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'

Using SmartForm Finder

XXX

Introduction

XXX this is out of the admin guide - rework?
Users can search for forms on the Turnkey Self Service Portal by using conventional search fields and filtering. If there is a large number of forms or if external resources should be referenced as well, however, more sophisticated form finding methods may be needed.
SmartForm Manager provides a way for users to search for forms or external resources by answering a series of questions about their needs. Each answer is used to filter the set of forms and resources.
A set of questions that are used to filter forms is called a finder category. It contains a form data file describing the forms and resources that shall be exposed as well as a user interface definition file. This separation is useful when a form data file is exposed multiple times via different user interface definitions.
The SmartForm Finder module is an example of how a finder category might be rendered, providing a user interface to progress through the set of questions and display matching forms. In addition, users can also search the entire set of forms and resources by keyword.

Setting Up Finder Categories

XXX refer to admin guide or duplicate the section?

Accessing SmartForm Finder

XXX refer to admin guide or duplicate the section as done below?
Please note: The Transact Finder module contains an example page that will display the finder category with code "grants", if one exists. Generally, agencies would provide their own finder page using a Transaction Manager finder category. For more information around this, please refer to the Transact Finder Examples module ("avoka-sf-finder-examples.war"), which is created by the TM installer.
To be able to access the Transact Finder module, it must be deployed on your Transaction Manager instance. The relevant deployment module is named "avoka-sf-finder.war".
To access Transact Finder, create a finder category with code "grants" and then access
<TM server root>/finder e.g. http://transactionmanager.com/finder
Figure 121 shows the Transact Finder after the user has answered two questions. Note also the global search field and the quick navigation bar at the top that allows the user to jump to one of the previous questions.

Old SSO

Enter topic text here.

Anonymous form access_2

Transaction Manager can be used to present forms to anonymous users. You can configure whether authentication is required at the form level. However, there are several reasons for wanting Transaction Manager to know the identity of the user, such as:

- Prefill (or pre-populate) the form with information you already know about the user, such as their first and last name. Prefilling a form makes it easier for the user to complete their form.
- Prefill the form with information you can look up about this user in external system, such as their account number or their existing services.
- Restrict or grant access to different forms based on the groups the user belongs to.

Prefill for anonymous users_2

It is possible to achieve prefill of forms without Single Sign On implementation. For example, using the TM Web Plugin, the hosting web application can embed the form, passing a number of URL parameters to Transaction Manager. Transaction Manager, in turn, can inject these parameters into the form's XML, or use these URL parameters to look up information in external systems (please see chapter "Prefill" for details and other prefill options). In many cases, this is actually the simplest way to implement prefill.

Single Sign On implementations_2

There are many different vendors who provide SSO solutions, as well as open source solutions. These solutions tend to be quite different from each other in terms of the way that they are implemented. There are a variety of techniques in use, including browser plugins, http headers, cookies, Java applets, smart cards, one time password tokens, integrated windows authentication, and Kerberos. There are also some other techniques to implement single sign-on such as OpenId, used by Google, Yahoo, and MySpace, and many others.
Transaction Manager does not support any particular SSO implementation out of the box. However, Transaction Manager does provide an SSO based authentication provider whose functionality can be specified at runtime by modifying Groovy scripts. This allows integration with many SSO solutions without requiring a custom build of TM.
To configure SSO authentication, log in to the TM administration console and go to "Security > Security Managers". Edit the security manager named "SSO Security Manager" (or create a new security manager based on the "SSO Security Manager" template. You can provide three Groovy scripts to customize the SSO authentication behavior: the "Get SSO Auth Token Script" and "Auth OK Response Script" scripts on the "SSO Auth Filter" tab and the authentication provider's Groovy script parameter, which handles authentication itself. Please refer to the TM Groovy services guide ("System > Groovy Services Guide") for more details about these scripts and example implementations.

SAML_2

A common Single Sign On standard is known as SAML, or Security Assertion Markup Language. SAML is an XML-based open standard data format for exchanging authentication and authorization data between parties. Importantly, SAML assumes the existence of a security provider, which is not specified by the standard.
There are many different ways and patterns in which SAML can be implemented, and many different security provider implementations. SAML is really more suited to loosely coupled federated SSO across multiple domains than it is to SSO across multiple web applications in a single enterprise - however it may be used in this context.
Transaction Manager has been successfully integrated with SAML-based implementations. The complexity and duration of the integration depend greatly on the specific implementation.

Additional ideas (Manager v5.0)

Garry:
add something around best practice for defining print-friendly forms
form validation: For 3.4 specifically, maybe add something here on in-field validation, also touch on specific ways to validate ie per section, per form etc
attachments: Provide something about best practice configuration for attachments within the TransactField App
payment: How can we connect through to other payment providers
delivery: touch on Groovy script delivery Also, maybe touch on different stages of submission here to include why we would use certain endpoints, such as form submission pre-processor and submission completed processor
reporting: I also would like to use the submission data extract option to view a cut of the data as it comes in
Lena:
describe the Composer publish workaround to keep publishing to a specific form in TM after the FAR form key changes
jobs, form bundles
Keyword index
No index entries found.

Data Retention Management (Manager v5.0)

Transaction Manager incorporates a sophisticated data retention management capability which is used to control the growth of database tables, to maintain high system performance, and also to purge transaction form data which contains Personally Identifiable Information (PII) entered by end users.

Many entities have a maximum age, after which TM will delete them automatically using scheduled jobs. These maximum ages can be adjusted globally, and some transaction related settings can be defined on an organization and/or form level.

Data Retention Policy Mode

There are two data retention policy modes in TM: **strict** and **relaxed**.

The mode affects how long (as a maximum) transaction related data can be kept in TM. For either mode, you can still configure the exact values for individual data retention settings.

The following describes the differences between the two modes. Note that the data retention settings mentioned will be discussed in section Transaction Record Settings below.

	Strict Mode	**Relaxed Mode**
Saved Transactions	max **180 days**, default 30 days	max 2 years, default 30 days
Finished Transaction PII Data	max **30 days**, default 7 days	max 2 years, default 7 days
Transaction Records	max **2 years**, default 90 days	max 5 years, default 180 days
Finished Job Records	max **2 years**, default 90 days	max 2 years, default 180 days
Submission Data Extracts	**deleted with Finished Txn PII Data**	optionally deleted with Finished Txn PII Data
Submission Properties	**deleted with Finished Txn PII Data**	deleted with Transaction Records

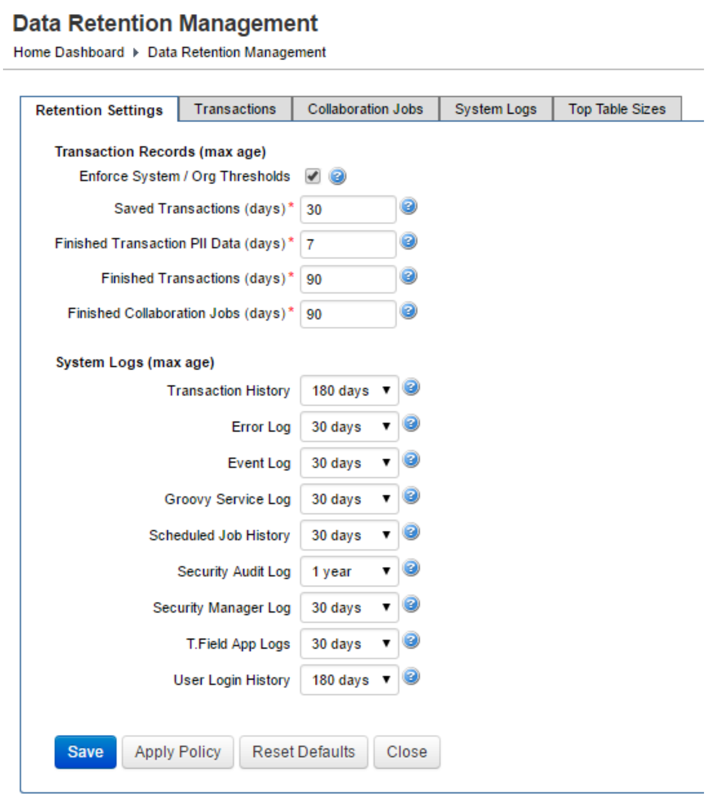
The data retention policy mode is selected in the TM installer (both during a new installation and an upgrade). You can see what data retention mode your server is using by accessing "System > System Info" in the "Build Properties" section.

Generally, we recommend that you use strict mode whenever possible, as it restricts the maximum transaction data age, avoiding problems caused by long data retention (database growth, performance decrease, long storage of sensitive personally identifiable information).

Data Retention Settings

Configuration of the data retention policies is performed using the "System > Data Retention Management" page. Access to this page is controlled via the Transaction Manager "Service Edit" permission.

Changes made to the retention settings will be used the next time data retention jobs run. If you want to start a data purge run immediately, click "Apply Policy" after saving your changes.



Read on for details of the individual settings and their meaning.

Transaction Record Settings

Transactions are a central entity in TM. Each transaction corresponds to a form submitted by a user and is associated with a multitude of other entities such as submission history entries, submission properties, data extracts and attachments.

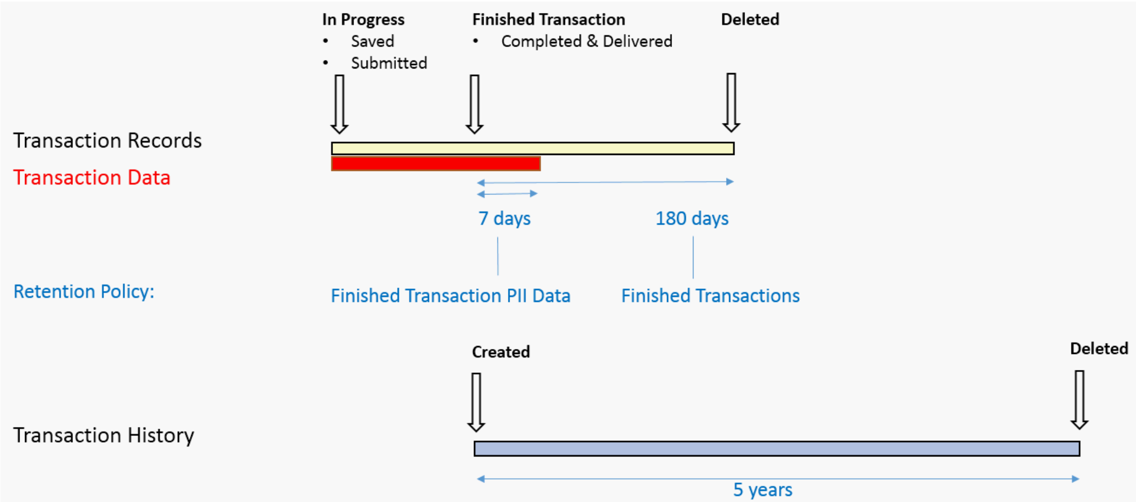
Transactions are crucial to data retention for two reasons:

1. Transactions are continually generated and contain a large amount of data.
   This can lead to large database sizes and degraded performance if data is not purged regularly.
2. Transactions contain user-entered personally identifiable information.
   This can pose a security risk if attackers gain access to data stored in TM. While TM comes with a number of data security measures, it is still advisable to ensure that personally identifiable information is purged once transactions have finished.

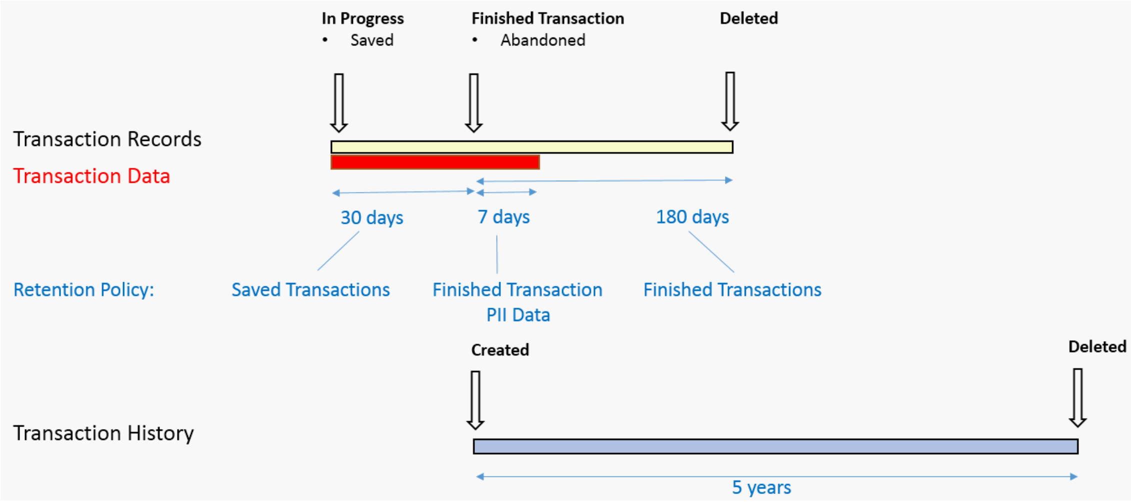
So generally, you want to keep your purge settings strict, so that data is purged regularly, keeping the system performant and secure. TM supports this approach by ensuring that data is deleted only when the submission is in a finished state (either completed successfully and delivered to the back end, or abandoned).

To understand the settings that are available, we need to look at the transaction life cycle. There are three different categories a transaction can fall into for purposes of data retention:

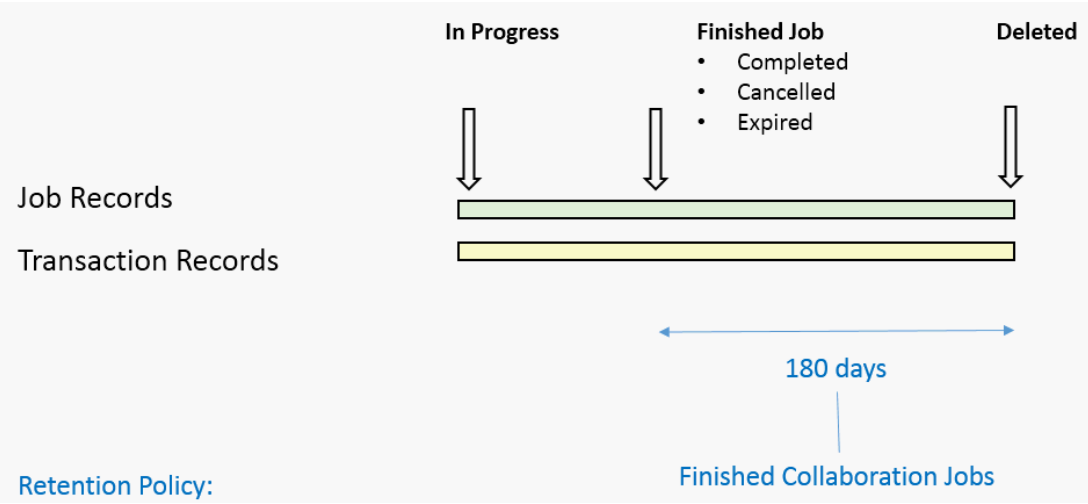
1. **Completed Transactions**
   The submission is not part of a collaboration job and has been successfully completed by the user, and delivered to the backend system (if configured in TM). No further user interaction is anticipated, and as all the data has been delivered, the countdown to submission deletion starts at completion time. The maximum age of completed submission records is controlled by the "Finished Transactions" setting.
   Note that submission data (including submitted XML and attachment data) makes up the lion's share of a submission's data footprint. To alleviate this, submission data for finished submissions can be set to be deleted well before the actual submission record is deleted. This is controlled with the "Finished Transaction PII Data" setting. When PII data has been deleted, the submission is still visible in the TM submission log (but the XML data cannot be viewed, and attachments cannot be downloaded any more).



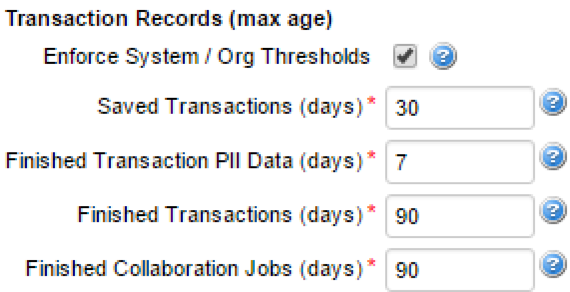
2. **Abandoned Transactions**
   The submission is not part of a collaboration job and has been abandoned (by the user, an administrator or automatically after a period of inactivity). Data for abandoned submissions may have been delivered if configured in TM. Generally, abandoned submissions will be quite common, especially caused by users opening a form (perhaps just for a first look) and then closing it without submitting it. While this is normal, the amount of data generated by abandoned submissions can build up, so it is important to delete these submission records.
   The maximum age for abandoned submissions is controlled by the "Finished Transactions" setting, with one exception: Submissions that were opened by users but never saved or submitted (as mentioned above) are deleted after two days.
   Note that submission data (including submitted XML and attachment data) makes up the lion's share of a submission's data footprint. To alleviate this, submission data for finished submissions can be set to be deleted well before the actual submission record is deleted. This is controlled with the "Finished Transaction PII Data" setting. When PII data has been deleted, the submission is still visible in the TM submission log (but the XML data cannot be viewed, and attachments cannot be downloaded any more).



3. **Collaboration Jobs**
   If a submission is part of a collaboration job, it is not purged as part of normal submission purging. Instead, TM uses the "Finished Collaboration Jobs" policy to purge the collaboration job and all its submissions once the entire job has finished.



Now let us look at the set of transaction related data retention settings available via "System > Data Retention Management", tab "Retention Settings".



The "Enforce System / Org Thresholds" relate to settings that can be overwritten on the organization and/or form level (more on that below). If ticked, then the policies defined at a lower level cannot be more lenient than the corresponding setting at a higher level.
For example, if this checkbox is ticked and the global "Finished Transaction PII Data" policy is set to 14 days, you cannot enter values greater than 14 days for the "Finished Transaction PII Data" policy on the organization or form level. If an organization's value of the "Finished Transaction PII Data" policy is 10 days, you cannot configure any of its forms to have a "Finished Transaction PII Data" policy value greater than 10 days.

Thresholds are only relevant to the "Saved Transactions" and "Finished Transaction PII Data" settings.

Note: Even if "Enforce System / Org Thresholds" is ticked, changes to global or organization policies DO NOT affect the values set on lower levels. For example, if an organization has a "Finished Transaction PII Data" policy value of 14 days and you change the global "Finished Transaction PII Data" policy value to 7 days, the organization value will remain the same until the organization settings are edited. This avoids problems around accidental changes to global policies trickling through the entire system.

The "Saved Transactions" setting allows you to specify the number of days after which a submission that was explicitly saved by a user will be abandoned (unless of course the user returns and submits the form). Note that the submission is not deleted at this point, but it counts as a finished transaction and the countdown for PII data deletion and submission record deletion start ticking at abandonment time.
This setting can be customized on both the organization and form levels.

The "Finished Transaction PII Data" setting determines how many days personally identifiable submission data (XML data, attachment data, submission extract data and submission properties) will be kept once a transaction has finished. This data takes up a large amount of space and poses some security

risk due to containing user entered data, so we recommend a strict value for this policy. The submission record itself will stay around after data deletion (see the next setting below).
This setting can be customized on both the organization and form levels.

The "Finished Transactions" setting determines how many days submission records and associated entities will be kept once a transaction has finished. If PII data has been deleted for a submission, the header records remain until this policy kicks in, allowing your administrators to find the submission in the TM logs.

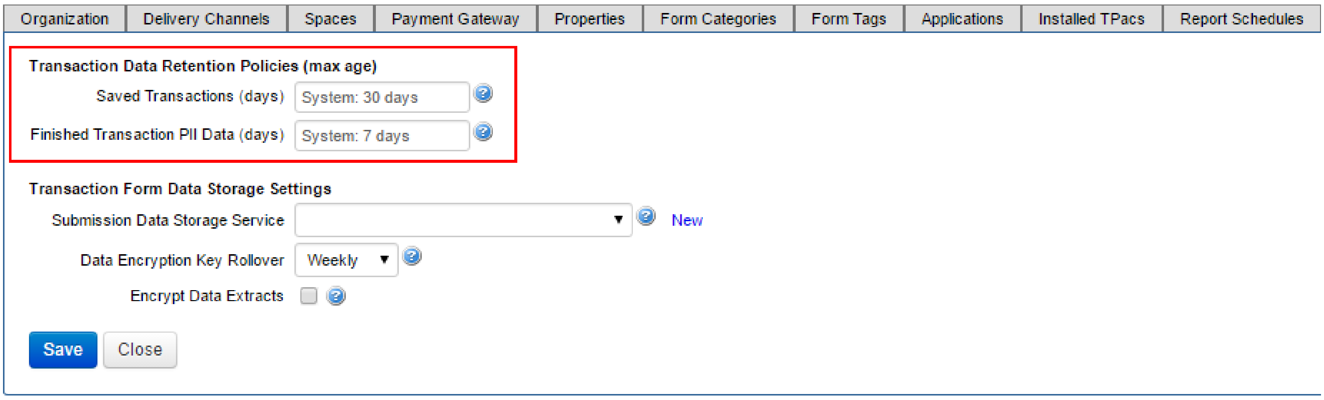
The "Finished Collaboration Jobs" setting determines how many days collaboration jobs and all their submissions will be kept once the entire job has finished.

Always save your changes after editing any field. If you would like to start a purge run straight away, save and then click "Apply Policy".

Organization Level Settings

Organizations by default use the global data retention settings; however, you can customize the "Saved Transactions" and "Finished Transaction PII Data" on the organization level if desired. These settings will then apply to all submissions associated with forms belonging to the organization.

To edit the organization data retention settings, edit the organization and switch to the "Data Management" tab.



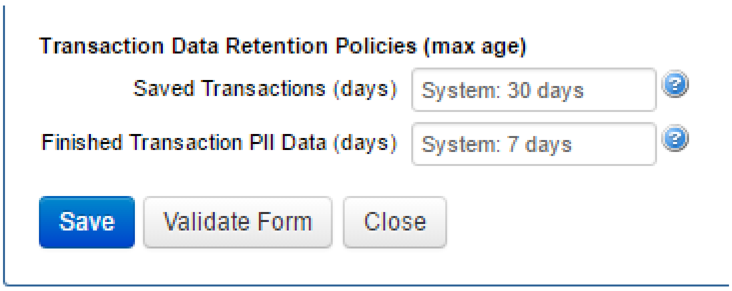
Enter custom values for "Saved Transactions" and "Finished Transaction PII Data" or leave them blank to use the global setting. If "Enforce System / Org Thresholds" is turned on, you will not be able to enter values larger than the current global policy value.

When you are done, save your changes. TM will apply the new policies to all submissions for this organization.

Form Level Settings

Forms by default use the organization (if defined) or global data retention settings. However, you can customize the "Saved Transactions" and "Finished Transaction PII Data" on the form level. These settings will apply to all submissions for this form.

To edit the form data retention settings, edit the form and switch to the "Details" tab. The relevant setting are at the bottom of the tab.



Enter custom values for "Saved Transactions" and "Finished Transaction PII Data" or leave them blank to use the global setting. If "Enforce System / Org Thresholds" is turned on, you will not be able to enter values larger than the current organization (if defined) or global policy value.

When you are done, save your changes. TM will apply the new policies to all submissions for this form.

Immediate Data Deletion

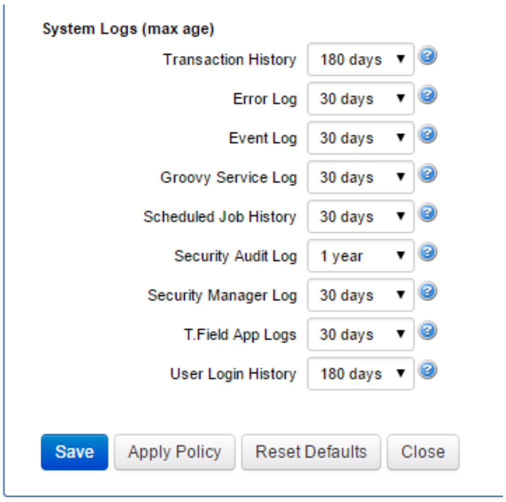
Please note for forms where the transaction data should be deleted immediately once it has been delivered, organizations will need to use the REST Delivery API which provides an option for immediate data deletion once positive delivery confirmation has been acknowledged.

Otherwise, the minimum possible time until data deletion is one day after the submission has been delivered.

System Log Settings

Apart from transaction data, TM stores a number of other entities that may need to be purged (such as various system logs and sent emails). Each of these settings generally refers to a single entity and do not concern personally identifiable information.

To change these settings, go to "System > Data Retention Management" and ensure you are on the "Retention Settings" tab. We will discuss the settings available in the "System Logs" section (the others were discussed in the previous section).



Whenever a submission is finished, a record containing information about the submission (such as final status, form name, time to submit and browser information) is written to a separate transaction history table. The transaction history table does not contain personally identifiable information, though the contact email address (if defined) is written to the transaction history table for security auditing purposes. The "Transaction History" setting determines how long transaction history records are kept, counting from the time the submission was originally started.

The "Error Log" setting determines how long error log entries and data that are not associated with a submission will be kept, counting from the time the error occurred.

The "Event Log" setting determines how long event log entries that are not associated with a submission will be kept, counting from the time the event was logged.

The "Groovy Service Log" setting determines how long log entries for Groovy service invocations will be kept, counting from invocation time.

The "Security Audit Log" setting determines how long audit log entries are kept, counting from the time the log was created. The audit log can be crucial to tracking down changes made by administrators, so on non-test servers, we recommend that you not purge it too soon.

TM creates a log entry every time a scheduled job runs. The "Scheduled Job History" setting determines how long TM will keep these history entries.

The "T.Field App Logs" setting determines how long TM will keep T.Field device logs and T.Field sync logs.

TM logs various events relating to user session (login, logout, session expiry, account lock etc.). The "User Login History" setting determines how long these logs will be kept.

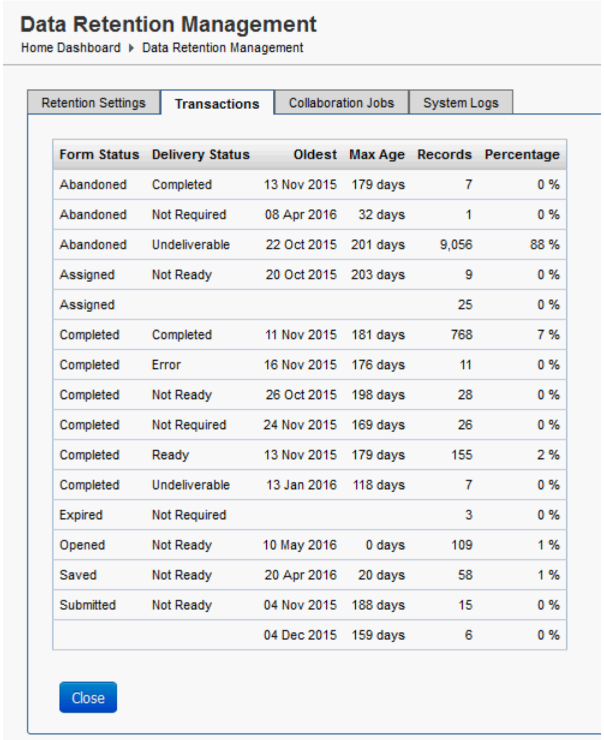
Always save your changes after editing any field. If you would like to start a purge run straight away, save and then click "Apply Policy".

Data Retention Status

To help decide on data retention settings or see how they are working out, it is useful to see how much data is currently stored on your TM server as well as other metrics such as the age of the oldest record. TM provides summaries on the data retention management page for your convenience.

To see a breakdown of transactions (excluding submissions associated with collaboration jobs), go to "System > Data Retention Management" and switch to the "Transactions" tab.

Transactions are grouped by form status and delivery status. Note that only certain combinations of these statuses can be eligible for purging (form status "Completed" or "Abandoned" and delivery status "Completed", "Undeliverable" or "Not Required").



Data Retention Management
Home Dashboard ▸ Data Retention Management

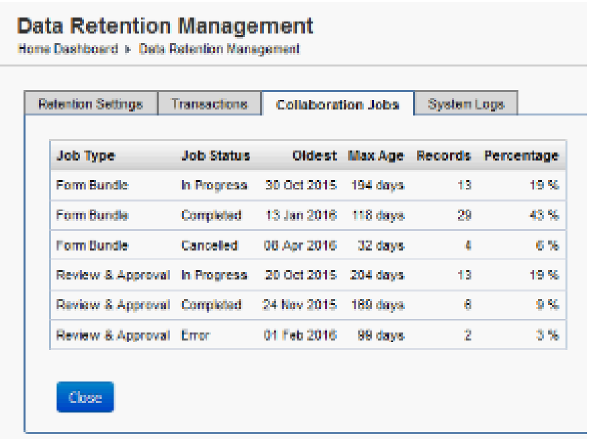
Retention Settings | Transactions | Collaboration Jobs | System Logs

Form Status	Delivery Status	Oldest	Max Age	Records	Percentage
Abandoned	Completed	13 Nov 2015	179 days	7	0 %
Abandoned	Not Required	08 Apr 2016	32 days	1	0 %
Abandoned	Undeliverable	22 Oct 2015	201 days	9,056	88 %
Assigned	Not Ready	20 Oct 2015	203 days	9	0 %
Assigned				25	0 %
Completed	Completed	11 Nov 2015	181 days	768	7 %
Completed	Error	16 Nov 2015	176 days	11	0 %
Completed	Not Ready	26 Oct 2015	198 days	28	0 %
Completed	Not Required	24 Nov 2015	169 days	26	0 %
Completed	Ready	13 Nov 2015	179 days	155	2 %
Completed	Undeliverable	13 Jan 2016	118 days	7	0 %
Expired	Not Required			3	0 %
Opened	Not Ready	10 May 2016	0 days	109	1 %
Saved	Not Ready	20 Apr 2016	20 days	58	1 %
Submitted	Not Ready	04 Nov 2015	188 days	15	0 %
		04 Dec 2015	159 days	6	0 %

Close

To see a breakdown of collaboration jobs, go to "System > Data Retention Management" and switch to the "Collaboration Jobs" tab.

Again, not all jobs are eligible for purging: Only collaboration jobs with a status of "Completed", "Canceled" and "Expired" are finished and subject to data retention times.



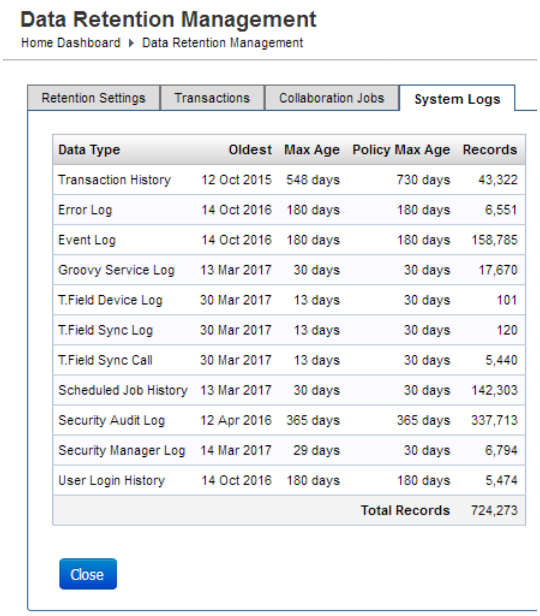
Data Retention Management
Home Dashboard ▸ Data Retention Management

Retention Settings | Transactions | Collaboration Jobs | System Logs

Job Type	Job Status	Oldest	Max Age	Records	Percentage
Form Bundle	In Progress	30 Oct 2015	194 days	13	19 %
Form Bundle	Completed	13 Jan 2016	118 days	29	43 %
Form Bundle	Cancelled	08 Apr 2016	32 days	4	6 %
Review & Approval	In Progress	20 Oct 2015	204 days	13	19 %
Review & Approval	Completed	24 Nov 2015	169 days	6	9 %
Review & Approval	Error	01 Feb 2016	99 days	2	3 %

Close

To see a breakdown of other system entities, go to "System > Data Retention Management" and switch to the "System Logs" tab.



Data Retention Management
Home Dashboard ▸ Data Retention Management

Retention Settings | Transactions | Collaboration Jobs | System Logs

Data Type	Oldest	Max Age	Policy Max Age	Records
Transaction History	12 Oct 2015	548 days	730 days	43,322
Error Log	14 Oct 2016	180 days	180 days	6,551
Event Log	14 Oct 2016	180 days	180 days	158,785
Groovy Service Log	13 Mar 2017	30 days	30 days	17,670
T.Field Device Log	30 Mar 2017	13 days	30 days	101
T.Field Sync Log	30 Mar 2017	13 days	30 days	120
T.Field Sync Call	30 Mar 2017	13 days	30 days	5,440
Scheduled Job History	13 Mar 2017	30 days	30 days	142,303
Security Audit Log	12 Apr 2016	365 days	365 days	337,713
Security Manager Log	14 Mar 2017	29 days	30 days	6,794
User Login History	14 Oct 2016	180 days	180 days	5,474
			Total Records	724,273

Close

TM Table Usage (MySQL only)

If your TM instance uses MySQL, you can see a breakdown of the largest database tables to help you assess how your system uses resources.

Go to "System > Data Retention Management" and switch to the "Top Table Sizes" tab,

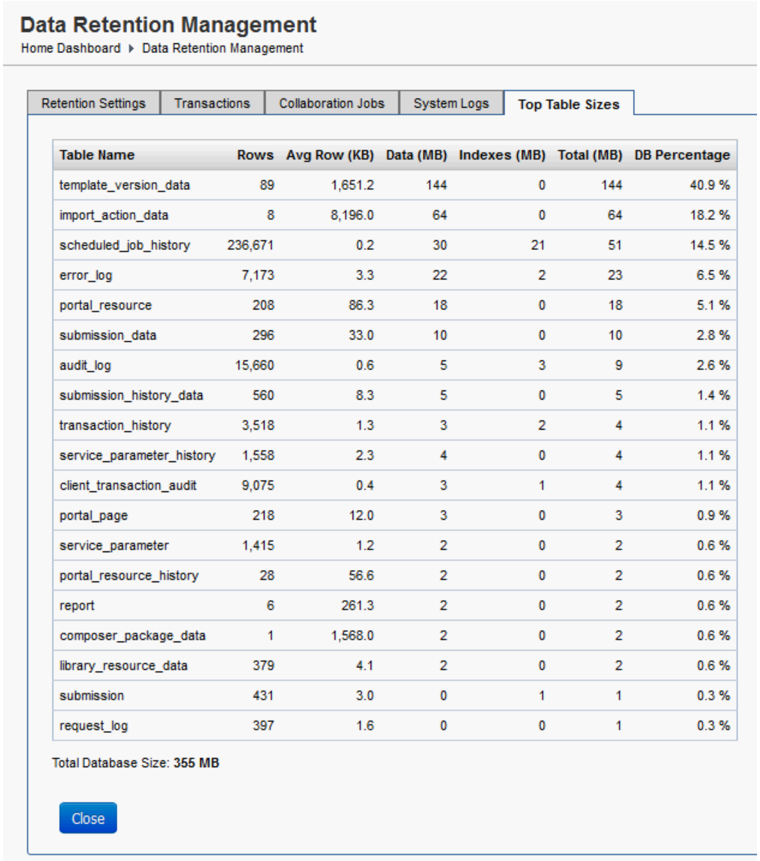


Table Name	Rows	Avg Row (KB)	Data (MB)	Indexes (MB)	Total (MB)	DB Percentage
template_version_data	89	1,651.2	144	0	144	40.9 %
import_action_data	8	8,196.0	64	0	64	18.2 %
scheduled_job_history	236,671	0.2	30	21	51	14.5 %
error_log	7,173	3.3	22	2	23	6.5 %
portal_resource	208	86.3	18	0	18	5.1 %
submission_data	296	33.0	10	0	10	2.8 %
audit_log	15,660	0.6	5	3	9	2.6 %
submission_history_data	560	8.3	5	0	5	1.4 %
transaction_history	3,518	1.3	3	2	4	1.1 %
service_parameter_history	1,558	2.3	4	0	4	1.1 %
client_transaction_audit	9,075	0.4	3	1	4	1.1 %
portal_page	218	12.0	3	0	3	0.9 %
service_parameter	1,415	1.2	2	0	2	0.6 %
portal_resource_history	28	56.6	2	0	2	0.6 %
report	6	261.3	2	0	2	0.6 %
composer_package_data	1	1,568.0	2	0	2	0.6 %
library_resource_data	379	4.1	2	0	2	0.6 %
submission	431	3.0	0	1	1	0.3 %
request_log	397	1.6	0	0	1	0.3 %

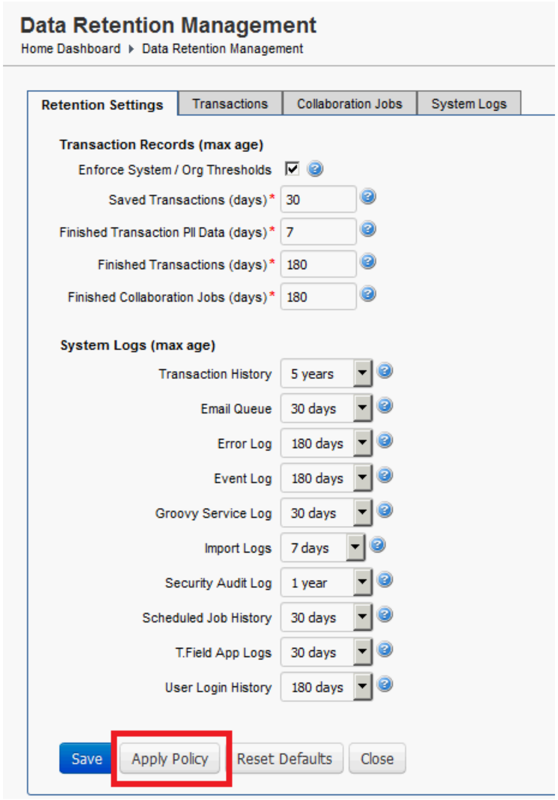
Total Database Size: **355 MB**

Data Retention Schedule

Data retention is performed continuously as part of the Transaction Processor service, which is run at short intervals throughout the day. This helps keep data from accumulating and spreads out the load of data purging. By default, the Transaction Processor service runs every 5 minutes and will purge up to 500 records for each purgable entity (this number is exposed as a service parameter and can be changed if needed - please consult Avoka Support if unsure).

Triggering Data Retention

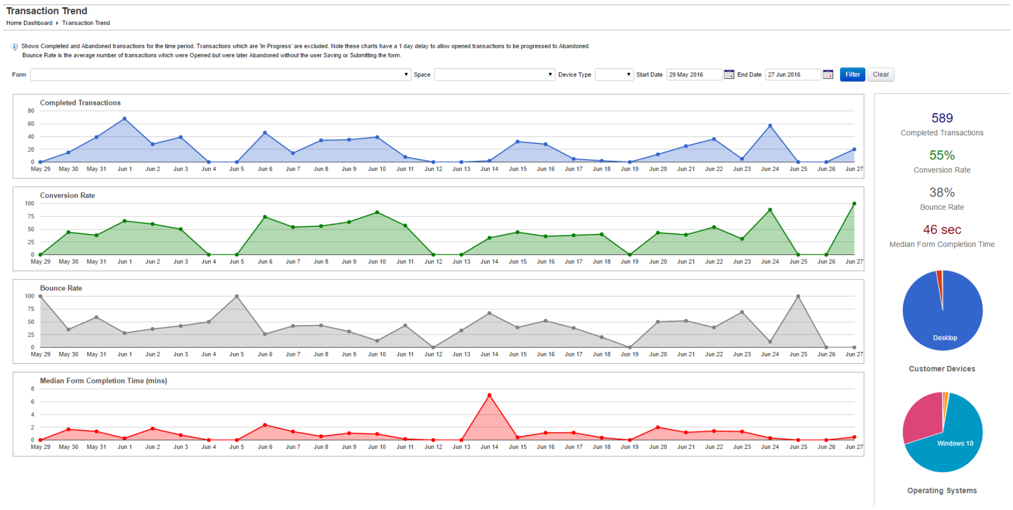
While TM continuously applies data retention policies as described above, you can also trigger a data purge manually. To do so, go to "System > Data Retention Management", ensure you are on the "Retention Settings" tab and click "Apply Policy" (if you made changes to the policies, click "Save" first, and then "Apply Policy").



TM will perform a data purge run similar to the pass done by the Transaction Processor service and will return a message as to how many entities were deleted.

Analytics Transaction Trend (Manager v5.0)

This topic discusses the Transact Manager Analytics view "Transaction Trend". This view is designed to provide key top line metrics and time series graphs about how transactions are performing from a business perspective.



Background

Form transactions supported in Avoka Transact can be more complex than you may initially think. Transactions will often run over may days, they will often involve multiple sessions, with saves and resumes, and can be across multiple devices.

Another special feature of transaction is that they can be abandoned in a variety of different ways which are important to understand from a business perspective. The types of abandonment include:

- **Bounces** - these are transactions which are opened when they view the form, but then the navigate away or close with window before starting work on the transaction.
- **Form Starts** - these are transactions where the user opens the form, performs some action like navigation or form data entry but then closes the window without performing an explicit user save
- **Background Saves** - these are transactions where the form has performed an automatic background save when the user navigates to the next section, but the user then closes with window without performing an explicit save
- **User Saves** - these are transactions where the user has performed an explicit save operation and then closes the window, however they never return to resume the transaction again
- **User Cancels** - these are transactions where the user has decided to explicitly cancel the transaction by clicking on the Cancel button.
- **Administrator Cancels** - these are transactions which may be manually abandoned by an administrator, these are very rare

The table below details when these abandonment status are determined and gives an indication of their relative frequency

Abandonment Type	Done Automatically	Done When	Frequency	Comments
Form Bounce	Yes	24 hours	Very High	
Form Starts	Yes	24 hours	High	Supported in TM 4.3.4
Background Saves	Yes	24 hours	High	
Use Saves	Yes	30 days default	Low	Period configurable at a Form, Org. or System level
User Cancels	No	done by user	Low	
Administrator Cancels	No	done by staff	Very Low	

The reason this is important is that we don't know whether an open transaction has been abandoned until 24 hours later. This effectively means we wont have daily conversion rate and bounce rate information available until 48 hours later.

The Transaction Trend view is based on information in the Transaction History table. This avoids very heavy analytics queries being performed against the transaction tables which would impact operational transaction processing. Transaction History records are created when a transaction has been finished, which means it has either been Completed by the user, or it has been Abandoned. Completed transaction history records are created in real-time, when a transaction is completed an corresponding history records is created at the same time. Abandoned transaction history records however are only created with a Transaction progresses to an Abandoned state, which could occur 24 hours later or some 30 days later.

The effect of this is you should not be looking at yesterdays daily conversion rate and bounce rate metrics as not all the abandoned records will have been processed.

Filter Criteria

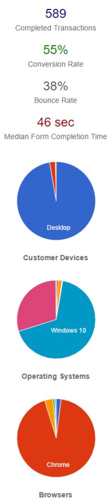
The filter criteria supported by the Transaction Trend view include:

- Organization - this filter on the top right will restrict the transaction trend analytics data to forms belonging to that organization, and will also filter down the list of Forms and Spaces available for filtering. Staff which belong to only one Organization will have this filter value automatically set.
- Form - this filter will restrict transaction trend analytics data to that belonging to the Form
- Space - this filter will restriction transaction trend analytics data to that associated with this Form Space
- Device Type - this filter will restrict transaction trend analytics data to that completed by the specified by the Device Type, which includes Desktop, Phone or Tablet devices.
- Start Date - this filter includes transactions which are completed on or after the Start Date

- End Date - this filter includes transactions which are completed on or before the End Date

Key Metrics

The Key Metrics pod on the right hand side show some key top line metrics that business owners should be focusing on.



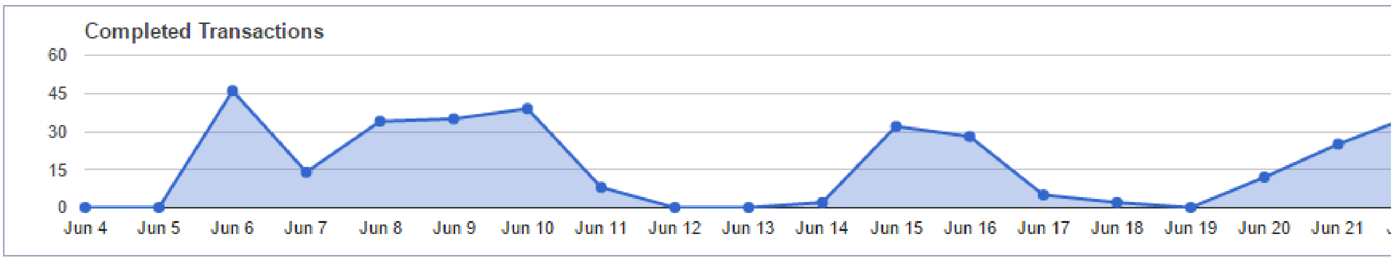
These metrics include

- **Completed Transactions** - the number of transactions completed by users for the given period and filter criteria
- **Completion Rate** - the ratio of Completed transactions / total transactions (including abandoned transactions) for the given period and filter criteria
- **Bounce Rate** - the ratio of Form Bounce transactions / total transaction for the given period and filter criteria
- **Median Form Completion Time** - the median time time users took to complete the form transaction, this time is cumulative time taken in each form transaction session
- **Customer Devices** - the percentage distribution of device types used to perform transactions
- **Operation Systems** - the percentage distribution of operating systems used to perform transactions
- **Browsers** - the percentage distribution of browsers used to perform transactions

Charts

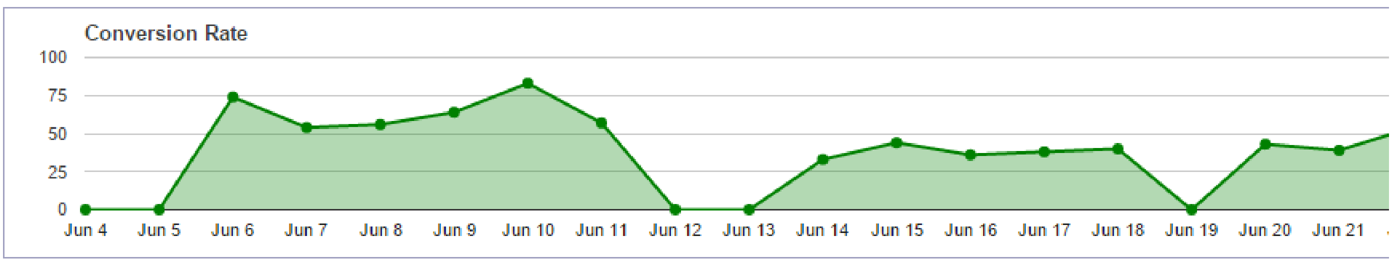
Completed Transactions

This chart provides a timeline display of total transaction completed which were started on the specified day. If a transaction is completed a few days later, it will still be attributed to the date it was started in the completed transaction counts for that day.



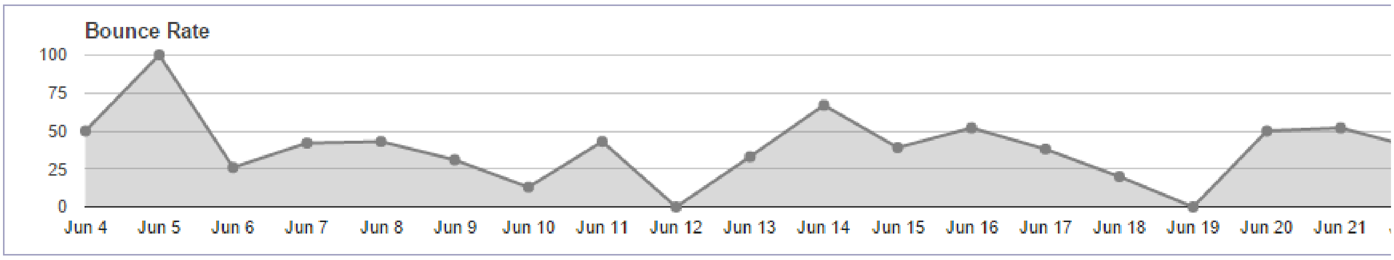
Conversion Rate

This chart provides a timeline display of total transaction completed which were started on the specified day. If a transaction is completed a few days later, it will still be attributed to the date it was started in the completed transaction counts for that day.



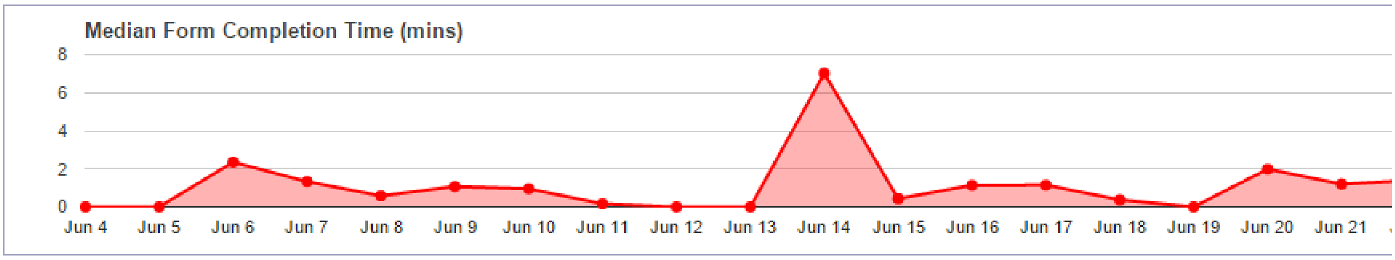
Bounce Rate

This chart provides a timeline display of total form bound transactions started on the specified day. A form bounce is a transaction opened by the user but which they never start, this is the equivalent of a web analytics page bounce when a user views an page but immediately navigates away or closes the window. In Transact a transaction is classified as bounce if they have not performed a save, either explicit user save or a form automatic background save in 24 hours. In real terms the daily bounce rate can be accurately calculated 2 days later. We cant provide the daily bounce rate for transactions started today or yesterday.



Median Form Completion Time (mins)

This chat provides a timeline display of the median time it took user to complete a form which was started on the specified day. The time taken to complete a form transaction is the sum of the transaction form sessions time. So if you spent 2 minutes in the first session and then resumed the transaction the next day and spent another 13 minutes completing the transaction the total time would be 15 minutes.



Frequently Asked Questions

Why does the bounce rate increase and the conversion rate change since yesterday ?

The daily conversion rate and bounce rate cannot be accurately determined for the first 48 hours. After 48 hours the bounce rate will be accurate, and the conversion rate will be fairly accurate but its values may change as other In Progress transactions started on that day are either Completed or Abandoned.

Why isn't the Conversion Rate the inverse of the Bounce Rate?

The Bounce Rate is only one type of abandonment, there are other types including abandoned Form Starts, User Saves, User Cancels and Administrator Cancels. These all contribute to the overall abandonment rate which is the inverse of the Conversion Rate.

Are Assigned Tasks or Collaboration Jobs included in the Bounce Rate?

Not assigned tasks and Collaboration job transactions are explicitly excluded for automatically abandoned form bounces and form starts.

Transact Licensing Reports (Manager v5.0)

Avoka Transact licensing is based on the concept of pay-per-transaction. To support licensing compliance Avoka Transact includes an automated license audit reporting system. This is covered in the standard Avoka Master Services Agreement (MSA) under the topic "Audit".

By using an automated licensing system Transact relieves customers from the burden audit reporting to ensure licensing compliance. Performing manual license reporting and auditing doesn't contribute to your bottom line.

Transaction license reporting is performed on a daily basis. These license reports contain meta-data information about the completed and abandoned form transactions, they contain no end-user form data or personal identifiable information (PII).

This license reporting system also provides summary system health information, to enable proactive operation monitoring of our customers systems and support capacity planning.

License Keys and TM Installations

Each Transact Manager instance (Prod, Staging, Dev. etc.) is configured with a unique customer licensing key which is used to support license reporting. With the Avoka Transact Cloud Service licensing configuration is performed by Avoka.

For on premise installations you will need to configure a Transact License Key for each TM instance before performing an installation or upgrade. Please do this ahead of your planned install or upgrade, so your IT staff will have the keys ready. To obtain License Keys from Avoka please contact: licensing@avoka.com.

If you have staff running TM instances locally on their computers for development purposes, you can obtain a per-user license key, by providing the named user's email address to Avoka support. Please note local TM licenses should not be used for any production workflows and are provided for local development purposes only.

Please see the Avoka Transaction Manager Installation Guide for information on how to install and upgrade Transact Manager and configure license keys.

Transaction Licensing Reports

Transaction license reports are provided on a daily basis to support transaction licensing accrual. Licensing reports include summary meta data on:

- Client Organizations - active during the reporting day
- Form Definitions - active during the reporting day
- Form Transactions - which were completed or abandoned during the reporting day
- Collaboration Jobs - created during the reporting day
- Services - billable 3rd party service invocations during the reporting day

These reports **DO NOT** include any user information, user entered information or personally identifiable information (PII).

System Health Reports

System Health reports provide hourly system health aggregate information to support pro-active operational monitoring and capacity growth planning. Hourly system health reports include aggregate counts on:

- transaction lifecycle event counts (form renders, saves, submits, deliveries, abandonments, deliveries)
- system error counts and event log counts
- groovy service and scheduled job invocation counts

Data Security

The Transaction Licensing system has a very secure data security architecture and strict access control policies. All data is encrypted at rest using AES-256 encryption with encryption keys managed in the AWS Key Management System (KMS). All data in transit is encrypted using TLS 1.1. Access is to licensing data controlled via AWS IAM named user accounts, within restricted networks.

Access to licensing reports is on a strictly need to know basis for Avoka licensing compliance staff and Avoka customer account management staff, and is a literally handful of people.

Access to system health reports is restricted to Avoka operational support staff and Avoka customer account management staff.

Network Rules

To support Transaction License reporting TM servers deployed on premise must have firewall rules which support access to the following AWS endpoint:

- https://s3-us-west-2.amazonaws.com

All traffic is REST service calls over HTTPS. Network traffic is daily for the transaction licensing reports running after midnight, and hourly for system health reports running after the hour. These reports are typically very small measuring around a KB for the daily transaction reports and around 90 bytes for system health reports.

Transact Licensing Reports Data (Manager v5.0)

The Transact Licensing Reports data files use GZIP compressed tabular CSV format. These report files include:

Instance

The licensed TM instance.

Name	Type	Description
tm_license_key	char(32) Not Null	primary key
billing_client_code	Varchar(20) Not Null	billing system client code
license_name	Varchar(50) Not Null	license name
hosting_type	Varchar(20) Not Null	specifies hosting type: [Cloud | On Premise]
env_type	Varchar(20) Not Null	TM environment type [PROD | UAT | TEST | DEV]
env_node_count	Integer Not Null	number of server nodes
env_version	Varchar(20) Not Null	TM Version Number
os_type	Varchar(10) Not Null	Operating System Type: linux, windows
db_type	Varchar(10) Not Null	Database Version: mysql, oracle, sqlserver
retention_mode	Varchar(10) Not Null	Data retention policy mode: strict, relaxed
manager_url	Varchar(100) Not Null	TM Manager Console URL
portal_count	Integer	number for form spaces
time_zone	Varchar(20) Not Null	time zone
update_timestamp	Timestamp Not Null	the date/time the instance was last included in an upload

Client

The TM client organization entity

Name	Type	Description
tm_license_key	char(32) Not Null	primary key (compound) & foreign key
client_oid	Integer Not Null	primary key (compound)
client_name	Varchar(100) Not Null	TM Client Name
client_code	Varchar(20) Not Null	TM Client Code
update_timestamp	Timestamp Not Null	the date/time the client was last included in an upload

Form

The transaction form metadata.

Name	Type	Description
tm_license_key	char(32) Not Null	primary key (compound) & foreign key
form_oid	Integer Not Null	primary key (compound)
client_oid	Integer Not Null	foreign key
form_name	Varchar(120) Not Null	TM Form Name
form_code	Varchar(20) Not Null	TM Form Code
update_timestamp	timestamp not null	the date/time the form was last included in an upload

Transaction

The finished transaction.

Name	Type	Description
tm_license_key	char(32) Not Null	primary key (compound) & foreign key
submission_oid	Integer Not Null	primary key (compound)

form_oid	Integer Not Null	foreign key
job_oid	Integer	foreign key
form_status	Varchar(20) Not Null	TM Form Status: Competed, Abandoned
form_abandonment_type	Varchar(20)	type of abandonment: Bounced, Started, Saved, Submitted, Cancelled
time_started	Timestamp	
time_finished	Timestamp	TM time: completed or abandoned
user_agent_device_type	Varchar(20)	users device type: Desktop, Tablet, Mobile
user_agent_browser_type	Varchar(20)	users browser type: Chrome, Internet Explorer, Edge, Firefox
user_agent_os_type	Varchar(20)	users operation system type: Windows, iOS, Android
form_session_count	Integer	number of form sessions
time_to_submit_sec	Integer	TM Time To Submit Seconds
time_elapsed_sec	Integer	TM Time in seconds between inital form render and submission
internal_txn_flag	Integer	specifies whether the form transaction is internal or external
space_name	Varchar(60)	specified the name of the Form Space (portal)
anon_participant_count	Integer	number of anonymous participants in transaction
auth_user_count	Integer	number of authenticated participants in transaction
attachment_count	Integer	file attachments count
attachment_total_size	Integer	file attachments total size in bytes
template_type	Varchar(20)	form template version type: maestro, composer, pdf
far_data_version	Varchar(20)	form template FAR data version: 4.3.0sp1
fw_flag	Integer	TransactField App transaction
error_count	Integer	number of error_log records associated with transaction

Job

The transaction job.

Name	Type	Description
tm_license_key	char(32) Not Null	primary key (compound) & foreign key
job_oid	Integer Not Null	primary key (compound)
client_oid	Integer Not Null	foreight key (client.client_id)
time_created	Timestamp Not Null	TM time job created

Service

The 3rd party licensed service invocation.

This information is used to support billing of 3rd party service licenses.

Name	Type	Description
tm_license_key	char(32) Not Null	primary key (compound) & foreign key
submission_service_log_oid	Integer Not Null	primary key (compound)
submission_oid	Integer Not Null	foreign key (transaction.submission_oid)
service_name	Varchar(60) Not Null	TM Service Name
service_url	Varchar(200)	endpoint URL of the external service called
service_info	Varchar(200)	additional information recorded with service invocation
time_created	Timestamp Not Null	TM time service invoked

System

The system health reports on aggregates for an hour period. Note TM will not upload historical error log data, it will only report the previous hours error count data.

This information used for system support to identify trending system issues and ensure systems are correctly provisioned with adequate capacity.

Name	Type	Description
tm_license_key	char(32) Not Null	primary key (compound) & foreign key
system_health_datetime_utc	Timestamp Not Null	
system_health_date	Date Not Null	primary key (compound)
system_health_hour	integer Not Null	primary key (compound)
txn_form_render_count	Integer Not Null	number of form request_log records
txn_receipt_render_count	Integer Not Null	number of receipt request_log records
txn_submit_count	Integer Not Null	number of form submissions (excludes saves)
txn_save_count	Integer Not Null	number of user form saves (excludes background saves)
txn_abandon_count	Integer Not Null	number of abandoned transactions
txn_delivery_count	Integer Not Null	number of submissions delivered
txn_delivery_error_count	Integer Not Null	total number of submissions in delivery error status
job_created_count	integer Not Null	number of jobs created
groovy_service_count	Integer Not Null	number of groovy service log
event_log_count	Integer Not Null	number of event log records
scheduled_job_count	Integer Not Null	number of quartz scheduled jobs run
fw_sync_count	Integer Not Null	number of TM field worker sync call log records
error_total_count	Integer Not Null	total number of error_log records
error_critical_count	Integer Not Null	number of TM critical errors for the period
error_groovy_count	Integer Not Null	number of TM system error_log records
error_collab_job_count	Integer Not Null	number of TM 'Collaboration Job' type error_log records
error_email_count	Integer Not Null	number of TM 'Email' type error_log records
error_form_delivery_count	Integer Not Null	number of TM 'Form Delivery' type error_log records
error_form_dynamic_data_count	Integer Not Null	number of TM 'Form Dynamic Data' type error_log records
error_form_render_count	Integer Not Null	number of TM 'Form Render' type error_log records
error_form_submission_count	Integer Not Null	number of TM 'Form Submission' type error_log records
error_livecycle_count	Integer Not Null	number of TM 'LiveCycle' type error_log records
error_management_count	Integer Not Null	number of TM 'Management' type error_log records
error_payment_gateway_count	Integer Not Null	number of TM 'Payment Gateway' type error_log records
error_receipt_render_count	Integer Not Null	number of TM 'Receipt Render' type error_log records
error_rest_soap_api_count	Integer Not Null	number of TM 'REST SOAP API' type error_log records
error_scheduled_job_count	Integer Not Null	number of TM 'Scheduled Job' type error_log records
error_security_manager_count	Integer Not Null	number of TM 'Security Manager' type error_log records
error_fw_sync_count	Integer Not Null	number of TM 'T.Field Sync' type error_log records
error_unclassified_count	Integer Not Null	number of TM 'Unclassified' type error_log records

Upload Log

The data upload verification log file. This information is used to verify the licensing data was correctly uploaded.

Name	Type	Description
tm_license_key	char(32) Not Null	primary key (compound) & foreign key
time_created	Timestamp Not Null	primary key (compound)
client_count	Integer Not Null	number of client records uploaded
form_count	Integer Not Null	number of form records uploaded

transaction_count	Integer Not Null	number of transaction records uploaded
job_count	Integer Not Null	number of transaction records uploaded
submission_service_count	Integer Not Null	number of service records uploaded

Transact Insights Analytics Data (Manager v5.0)

This page has been moved to Transact Insights Analytics Data.