# Overview of ANet/Tableau integration project

Broadly speaking, the various aims of this project are:

* Provide analytics and data visualization capabilities in ANet with a best-of-breed commercial solution, which would be broadly used across Active’s application suite.
* Provide a modern user experience for reporting in general.
* Reduce OLTP database load by moving reporting to a separate reporting database.
* Improve reporting performance by running reports from denormalized, aggregated data sources.
* Ultimately to phase out the existing reporting solutions.

As of this writing, there is a significant amount of discussion (controversy) about how Tableau should fit into the overall ANet reporting solution:

* Tableau’s primary strength is as an ad hoc data visualization tool, so initial integration will focus on that aspect.
* Tableau does have the ability to generate simple tabular views, but lacks the formatting and internal programmability of ANet’s own reporting engine, which is based on the .Net ActiveReports DLL. Most of this complex report logic would have to be implemented in the data tier to have report-ready data for specific purposes.
* Tableau does have the ability to generate CSV downloads, but only of the data actually shown in a visualization. One of product’s asks, based on requests from the Y’s in particular, is to be able to download the detailed data behind a visualization. This can be done by providing a tabular visualization which runs against less aggregated data, but there are performance trade-offs in this approach.
* Tableau’s ability to schedule report executions is limited.
* The thinking is that good data visualization will replace some of the need for report execution and CSV downloads, but the degree to which this is true is unknown.
* A tactical approach to the OLTP problem would be to run some of the existing reports against the reporting database.

# ANet/Tableau integration model

The discussion below makes the following assumptions:

* Tableau will be part of the ANet solution, even though there is ongoing discussion about exactly which functionality should be implemented in Tableau.
* There will be an initial implementation of Tableau (“R1”) rolled out to a population of beta users (YMCAs) with a small collection of reports (<15, primarily membership, customer, financial).
* There was discussion of having R1 customers log directly into Tableau, and run reports directly from the Tableau UI. However, we anticipated problems then transitioning those customers into the integrated experience, which would remove some functionality.
* As a result, R1 will contain the minimal AN modifications to provide integration usable by the R1 orgs.

## R1 integration

* Tableau will be administered manually by Active staff – no SCM automation required for limited number of beta orgs. Two users will be created for each org:
  + A “reader” user with the ability to view reports.
  + A “contributor’ user with the additional ability to customize reports.
* There will be two new user profile permissions in ANet:
  + Ability to view Tableau reports
  + Ability to customize and save Tableau reports
  + No report-level security as we have for existing reports; this will be in a later release.
* Launching reports from AN menu system
  + We will use existing menu system and add the generic reports into it. Product has a mockup for a prettier UI, but this would be in a later release.
  + Unlike current ANet reports, no filter page will be displayed. No one likes the current UI, so we’re bringing them directly into Tableau for filtering.
* Report display:
  + The report will loaded into either the current frame or another browser tab/window, like other AN reports, based on current AN settings, using Tableau’s “embed URL” format.
  + Using Tableau’s “trusted authentication” system, the report will be executed as either the reader or contributor user, based on the ANet user’s permissions.
  + Data will be automatically restricted to the current org.
  + Single-site report permissions will not be implemented in R1.
* Creating customized (adhoc) workbooks:
  + Users with customize permission will be able to create customized workbooks and save them.
  + All customized reports for an org will automatically be saved in an org-specific “library” project.
  + Tableau’s REST API will be used to enumerate the available reports for that org, to update the Tableau reports table driving the menu. (This is the probably most complex task in R1)
* New configuration settings:
  + New license setting for Tableau, to control which orgs see the new functionality
  + A few data center-wide settings in the ActiveNetSites database, and edited in the portal, to configure Tableau server URL and access credentials.

## R2+ functionality

The following list is everything which has been identified as being part of the final integration, based on our experience with JReports. The details of that functionality, and when it will be developed, are currently unknown.

* A new Tableau launch UI is being designed by product.
* Report level security to control which user profiles can execute and customize which reports.
* Site-level security, to allow ANet’s single-site/multi-site security permissions to be honored by the Tableau implementation. This will probably mainly be a function of workbook or datamart functionality, but may require some limited enabling in ActiveNet.
* Automate creation of Tableau setup of projects, groups and users for orgs: This will probably be done by the SCM team, using Tableau’s command-line interface, tabcmd.
* It is possible we will need to create create one Tableau user for each ANet user, rather than just two role-specific users. Unfortunately, this is hard because the REST API doesn’t provide all necessary capability; instead, their command line tool would have to be automated and executed as a result of data changes in the servlet. However, the only known functionality which requires this is the ability to save personal dashboard filters. Current thinking is that this isn’t very important, so we may not need to do this. To prevent this from becoming a problem in R1, we will disable the filter saving capability through the embed URL.

# Details of R1 Tableau integration

## Integration techniques

The following type of Tableau functionality will be used to integrate ANet with Tableau:

* Automated configuration of objects (projects, groups and users) and loading of reports in the Tableau server, using tabcmd command line tool. If done at all in R1, this would probably be done by the SCM team.
* Trusted authentication to implement user security in reports.
* REST API to query configuration information from the server.
* Embed URL to display the report.
* Alternatively, Tableau’s Javascript API can be used to display the report. This allows building a customized experience around the Tableau “Viz” object’s display area, but is a significant development task and doesn’t seem to be necessary for this project, at least in R1.

## Tableau object model

By object model, we need the configuration in the Tableau server of sites, projects, users, user groups, and workbooks. The basic requirements which that object model should address are:

* ANet configuration should be cleanly separate from that for other applications integrated with Tableau.
* We should be able to prevent any modifications to the generic reports, so that new versions can be deployed without clobbering customizations.
* To streamline deployment, there should be a single deployed instance of each generic report.
* We should be able to control which users can customize a report, and save that customization.
* We must ensure that customized reports are only visible to users of that org.
* It is desired to be able to organize reports by “domain” (i.e., membership, customer, financial, similar to the current ANet report menu top-level headings). This *can* be implemented in ANet, but in the model below, is implemented in the Tableau object model

To meet this requirements, the objects will be configured as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Site** | | | | There will be a single site for all of ANet, to separate ANet administration from that of other applications. |
|  | **Projects** | | | Domain\_<domain>: There will be one project for each domain, containing the workbooks for the generic reports. |
|  |  | | | Library\_<org>: There will be one project for each org, containing the reports customized by that org. User/project security will limit the users for an org to only being able to save into that org’s library, and not the domain projects. |
|  |  | **Workbooks** | | Deployment will publish workbooks into the appropriate domain projects  Customized workbooks will be saved into the org’s library project |
|  |  |  | **Views** | The view(s) in a workbook are what is actually launched by ANet as “reports” |
|  | **Users** | | | ActiveNet: Site administrator |
|  |  | | | <org>\_reader: Org-specific user |
|  | **Groups** | | | At this time, there does not seem to be a need for groups. If we do end up creating one Tableau user per ANet user, then reader and contributor groups will be created for each org, and permissions will be granted to groups. |

## Tableau user permissions by project

Here’s how the Tableau user permissions should be configured as a function of project:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Reader permissions for domain projects and org’s library project** | **Contributor permissions for domain projects** | **Contributor permissions for org’s library project** |
| **View** | True | True | True |
| **Web edit** | False | **True** | **True** |
| **Write / Web save** | False | False | False |
| **Download/save as** | False | False | **True** |
| **Delete** | False | False | **True** |
| **Filter** | True | True | True |
| **Add comment** | False | False | False |
| **View comment** | False | False | False |
| **View summary data** | True | True | True |
| **View underlying data** | False | False | False |
| **Export image** | True | True | True |
| **Share customized** | False | **True** | **True** |
| **Move** | False | False | False |
| **Set permissions** | False | False | False |
| **Connect** | False | True (???) | True (???) |

## Object creation lifecycle

* One time creation of site and site administrator user.
* One time creation of fixed projects for report “domains” (e.g., membership, facilities, revenue, etc.)
* Per release publishing of the generic workbooks created by Active into the appropriate domain project.
* Per org as it it provisioned, the following will be created, with a naming convention:
  + A project for the org’s custom reports (called the “library”).
  + A “reader” user who can permissions to execute reports and play with filters, but not customize reports
  + A “contributor” user who can customize reports and save as (ad hoc permission)
  + Permissions of those users to access the domain projects and the org’s library project, as above.

## Trusted authentication

Tableau’s trusted authentication will be used to:

* Ensure that a report displayed in ANet using the embed URL is running in the context of the desired Tableau user for that org and role.
* Ensure that report URLs can’t be distributed to give access to Tableau reports outside of ANet and its security system, since the trusted authentication tickets are one-time use and only good for 3 minutes.

The implementation will work as follows:

* The ActiveNet servlet servers will be configured in Tableau server as trusted IP addresses.
* Before each report execution, the servlet will request a ticket from the tableau server by posting to the Tableau server as follows and getting the 24 character ticket returned.

http://<Tableau server>/trusted  
?username=<Tableau user>&target\_site=<site id of Activenet site>

* When the report is executed, the ticket will be included in the embed URL:

http://<Tableau server>/trusted/<ticket>/t/ActiveNet/views/<workbook>/<view>

See this [Tableau documentation](http://onlinehelp.tableausoftware.com/v0.0/server/en-us/help.htm#trusted_auth.htm%3FTocPath%3DAdministrator%2520Guide%7CTrusted%2520Authentication%7C_____0) for an overview of trusted authentication.

## Report display (embed URL)

When a report is executed, it will be loaded into a window using the embed URL described below, which includes the ticket number, report name, org filter value, and a few UI parameters causes the initial display of the view into the target window.

http://<Tableau server>/trusted/<ticket>/t/ActiveNet/views /<workbook>/<view>?

Parameters (some of these need to be re-evaluated):

|  |  |
| --- | --- |
| Org Name=Auburn& | Specify the current org, to filter data |
| :embed=yes& | Hide the heading (this needs to be re-evaluated |
| :customViews=no& | Hide the “Remember my changes” drop-down, so no one can save filters. |
| :tabs=no | Don’t display workbook tabs; so only current view can be used |
| :toolbar=top& | Display the toolbar on the top |
| &:format=& | HTML format (not CSV/PDF/PNG export) |

See this [Tableau documentation](http://onlinehelp.tableausoftware.com/v0.0/server/en-us/help.htm#embed.htm%3FTocPath%3DAdministrator%2520Guide%7CEmbed%2520Views%7C_____0) for a description of the embedded view implementation.

## Enumerate reports through REST API

Based on the REST API documentation, the following calls should be necessary; note this has not been tested yet.

|  |  |
| --- | --- |
| POST /api/2.0/auth/signin | Sign in as site administrator (requires username and password) Get security token to use in header in all later calls. |
| GET /api/2.0/sites/site-name ?key=ActiveNet | Get site ID for ActiveNet site; save for later calls.  (Only necessary at first call after reload) |
| GET /api/2.0/sites/site-id /users/ | Get all users for site; find the correct ones for this org by name; Save IDs for later use.  (Only necessary at first call after reload) |
| GET /api/2.0/sites/site-id /users/user-id/workbooks | Get all workbooks for a user. Use the project name to determine where to put in menu Save workbook ID |
| GET /api/2.0/sites/site-id /workbooks/workbook-id/views | Get all views for workbook Save view name and ID |
| POST /api/2.0/auth/signout | Sign out |

The view information should be used to update a table of Tableau reports, used by the menu and security system, to match what is currently in Tableau.

* Store views into Tableau report table, keyed by view ID
* If ID does not exist, create record, use view name, and assign to “domain” = project name
* If ID does exist in table, unhide it
* If a record in the table was not enumerated, hide it

See this [Tableau documentation](http://onlinehelp.tableausoftware.com/v0.0/server/en-us/help.htm#rest_api.htm%3FTocPath%3DREST%2520API%7C_____0) for details on the REST API.

## Org-level data security

It is assumed in the above that there is a common org parameter to all the reports, which can be passed in the embed URL (e.g., Org Name=Auburn&) as there was in the sample report evaluated.

# Requirements at risk

The following are key product requirements which may not be able to be implemented:

* Taxonomy support (e.g., change the label and filter name for “Activity” to whatever it’s configured as in Anet).

# Technical background

## Dataflow to Tableau reports

## Requirements from product

## Tableau references

|  |  |
| --- | --- |
| [Tableau technical documentation](http://onlinehelp.tableausoftware.com/v0.0/server/en-us/help.htm#welcome.htm%3FTocPath%3D_____1) | Server administration REST API Javascript API  Embed URL |
| http://www.tableausoftware.com/products#video | Marketing video |
| [https://tableausoftware.webex.com/tableausoftware/lsr.php?RCID=55cdb3ddd2fb484bbb5cab13a016a631](https://owawest.activenetwork.com/owa/redir.aspx?C=x2aGG8h4H0CeKVo--JXM_JmqLtVcu9EIJkrqYL440UoyN65GMqlzqRYfgvyT04E187D33lkWIV4.&URL=https%3a%2f%2ftableausoftware.webex.com%2ftableausoftware%2flsr.php%3fRCID%3d55cdb3ddd2fb484bbb5cab13a016a631) | Webex of demo of Tableau for Active |