



DSA Project Checklist

DSA Project Checklist

The purpose of this document is to describe key activities that should be done as part of every DSA Project.

References

- DSA_Content_Breakdown.xls: <http://bit.ly/dsaxlsx>
- Digital Sales Aid Chatter Group: <http://bit.ly/dsachat>
- Base DSA Rally Project: <https://rally1.rallydev.com/#/9309886323d/userstories>
- DSA Architecture: <http://bit.ly/dsaarchitecture>
- DSA Wiki: <https://wiki.modelmetricssoftware.com/wiki/index.php/DSA>
- DSA Image Asset Specification: <http://bit.ly/dsaimages>

DSA Client Homework

There are several things the client needs to have available at the start of the project for us to be able to engage and deploy quickly. These are:

General

- Apple Enterprise Account (for Internal apps) or Developer Account (for App Store apps)
- Which SFDC Profiles will be using DSA.
- Consumer Key and Redirect URI from Connected Apps Setup.
- SFDC Login credentials for a System Administrator and test accounts with Profiles of users that will be using the app (Sales User Profile, etc.).

Application

- Name for the application
- Icon for the application

Labels

- Please complete *DSA_Content_Breakdown.xls* with:
 - What are the labels of each category?
 - What are the labels of each subcategory?
 - What articles of content belong to each subcategory?

Images

- Please complete the *DSA_Content_Breakdown.xls* with the image name for each of:
 - Main Screen Background
 - Category Buttons
 - Category Backgrounds
 - Subcategory Thumbnails
 - Top and Bottom menu logos
- Assemble and deliver images specified in *DSA_Content_Breakdown.xls*

Project Team Responsibilities

You're developing a custom application. The DSA library is a starting point, but this is still a custom development project, and requirements should be tracked and QA'd just like any other custom development project.

General Questions to Ask

- Which User Profiles will be using the app?
- How many Contact Records will be downloaded for each User type? Base DSA is tested to work with 250k Contacts. Larger volumes are possible, but will require additional QA. Where possible, it's best to try to limit the number of Contacts downloaded to improve the initial synchronization time and search times within the app.
- How much Content will be synchronized for each User type? Are there any extremely large videos?
- Which versions of iPad and iOS will be used?
- Will the app need to support Single Sign-On (SSO)? What type?
- What is considered to be an acceptable initial sync time?
- How will the app be distributed to end users? Apple App Store? Enterprise Distribution? Over the Air?
- Are there any MDM solutions, like Mobile Iron, that will be used for distribution? Which ones?

Project Manager Responsibilities

- Set up project in Rally, make sure team has access.

SBA Responsibilities

- List any User Stories that extend the functionality of Base DSA. You can find the Base DSA User Stories under the Special Projects Workspace in Rally.

Development Team Responsibilities

- Create a branch of the Master branch on the Base DSA project in Git. Use this as a starting point for any customizations required by the client. Further setup instructions can be found in the Readme.md file.
- Replace the default Consumer Key and Redirect URI in the app with values from the Client's Production Org.
- Follow Source Control Guidelines documented here: <http://bit.ly/m30scm>
- Use Phabricator for Code Reviews following the guidelines here: <http://bit.ly/m30codereview>
- Defects should be filed in Rally using these Defect Classifications: <http://bit.ly/declasses>

Methodology

Mobile Projects follow the Delivery [Methodology 3.0](#).

[WAITING ON IMAGE]

Some key aspects of this methodology for mobile projects include our:

- [Rally Setup for Project Management](#)
- [Environment Setup - iOS](#)
- [Develop Architecture](#)
- [Architecture Reviews](#)
- [Source Control](#)
- [Code Standards - General](#)
- [Code Standards - iOS](#)

Helpful Hints

- During development and QA, it is important to always remove the any old versions of the app before installing a new one. This ensures that any issues present in the old version haven't caused problems in the new one. This is necessary during development and QA only.
- If an updated build will be going out to end users, and they will not be able to delete the old version of the app, it is important that we QA the upgrade process. The app will need to be QA'd on a fresh install and also on an upgrade install from the previous released version.
- Salesforce has a max of 5 active sessions per username. If a 6th session becomes active, the oldest will be invalidated. Sharing logins during testing can cause Salesforce to invalidate older sessions, forcing the app to logout. It's best to avoid sharing logins while testing the app.

- Use test user with Profiles of real users. Do not login as a System Administrator with full data visibility, especially in orgs with a large number of records (> 250,000 Contacts). If a System Administrator or another Role with visibility to large record counts is a valid user of the app, discuss with the customer ways in which they can limit the number of Contacts synchronized to the device for these users. Some options are to impose a hard limit, to only download owned Contacts, or to only download Contacts with a specific value in a field.
- An Update of the DSA Managed Package does not affect existing mobile configurations.
- It is best to leave the DSA app in the foreground while syncing, especially during the initial sync. It will continue syncing in the background for as long as iOS lets it, but sync can be stopped by the OS if the app is in the background.

Troubleshooting

- Profile
 - Do they have the right access to all key objects related to the DSA and all custom config
 - Is the mobile config enabled for the right profiles?
 - Unless coded for all users of the DSA need to have "View Setup and Configuration = TRUE on their profile
- Content/Library
 - Does the user have the right access to at least View the content item(s) in question

FAQ

Do we deliver source code for DSA to the client?

Yes. Any request for source code should follow the process described in the [Code Delivery Checklist](#).

I found a defect in Base DSA, what do I do?

The project team is responsible for fixing defects that arise from custom functionality in the client app. If the team suspects that an issue is present in the underlying Base DSA, confirm it first with the latest Master build of Base DSA from TestFlight. If it is present in that build, file it as a Defect on the DSA Version 3 (RESTframework) project in Rally.

The project team developers may be best equipped to resolve the defect quickly. If this is the case, they should create a branch of Base DSA master, fix the issue, and submit it for Code Review using Phabricator.

My team's developers spent time fixing an issue in Base DSA. Does the client pay for this?

Time spent fixing Base DSA issues can be billed to the Base DSA project in PSE. Make sure Tom and Mihir are aware of the issues, and you're using Rally, Git, and Phabricator to document the issue and the fix.

What do you mean by “*DSA is an accelerator, not a product*”?

DSA is a library of code that we use to quickly create a custom app based off of our client's requirements as a part of a services engagement. This enables us to customize DSA for each of our clients needs, and allow them to be able to update it further with us, without us, or with someone else. However, once a project has ended, our clients will not receive updates or bug fixes.

My client wants to know the roadmap for some features they want implemented. Can R&D develop these features for a future release?

Feature requests are a great opportunity to bring your client's AE back into the picture. We are happy to scope out and implement custom feature functionality as a part of a paid services engagement. DSA does not have a public roadmap, and our clients shouldn't be expecting to get free updates with new functionality.

Is it possible to implement *[Feature X]*?

Probably. DSA is written as a native iOS app in Objective-c, so as long as something is possible to do in iOS, it should be possible to do in DSA with some effort.

Can DSA sync new records in the background?

Not really. iOS puts severe limits on background activity of 3rd party apps. This helps make iOS a very performant operating system, but it does mean that we can't maintain a synchronization process for more than 10 minutes after the app enters the background. It's not possible to push new records to the device while the app is in the background.