Developing a Credit Dashboard

A Major Qualifying Project Report

Submitted to the faculty of
WORCESTER POLYTECHNIC INSTITUTE
In partial fulfillment of the requirements for the
Degree of Bachelor of Science

Submitted on 01/08/2013

Project Sponsor:

Bank of America – Merrill Lynch

Submitted by:

Nicolas Dupuis, Management Engineering

Alexandra Mutty, Electrical and Computer Engineering

Fangwen Yuan, Computer Science

Submitted to:

Onsite Liaison:

Peter Bartek, Vice President; Programmer Prof MKTS

Project Advisors:

Kevin Sweeney, School of Business, WPI

Micha Hofri, Department of Computer Science, WPI

Xinming Huang, Department of Electrical and Computer Engineering, WPI

Abstract

Quartz, the all-in-one platform developed in Bank of America's Global Markets Research Technology (GMRT) department, is heavily used for development teams, but lacks an easy method for monitoring system health. We spoke with GMRT Quartz users to identify gadgets that would be useful to them in their work, and developed seven gadgets to fill their needs. Our final deliverable was a Credit Dashboard containing seven gadgets that monitor the health of various components of Quartz.

Executive Summary

Bank of America, like most large-scale financial, corporate, and wealth management institutions, is heavily involved in credit trading. An entire division of the company is dedicated to the buying and selling of credit derivatives. The Global Markets Research Technology (GMRT) department supports trading activities by developing proprietary and nonproprietary software to make trading as easy and efficient as possible. One of the most important projects in GMRT is the development of Quartz, an all-in-one system for market data, risk measures, approval workflow, and a central location for all tools an employee will need. The final vision for Quartz is to have one platform containing front-to-back pricing, risk management, lifecycle, settlements, and analytics. This system would allow for quick turnaround time on new instruments and tools for users, and easier collaboration for developers by working on the same base code.

Quartz is a large system made up of many smaller components, so a large issue it faces is in monitoring system health. Specialized Quartz Dashboards are already incorporated to display interesting statistics on Quartz users and development progress in teams, but there are no global monitors on processes and applications. The Quartz system needs some way to allow GMRT users to monitor applications and processes that are pertinent to their work.

Our project was to develop a Credit Dashboard to contain gadgets that monitor various Quartz components' health. The gadgets developed monitor different applications and queues within Sandra, the object-orientated database where information is stored. The Credit Dashboard acts as a central location for the gadgets to be contained, and which all users have access to through Quartz.

We had two main objectives for the completion of this project: Identify desired gadgets, and develop those gadgets for the Credit Dashboard. We spoke with our sponsor liaison, and project manager, Peter Bartek about his ideas for needed gadgets, and interviewed with other GMRT Quartz users to find out what tools they would like to have. The gadgets were developed through a defined process: generate idea, define specifications, create concept art, create proof of concept, tailor gadget to need, and finally a finished gadget that could be put into production.

Each gadget was developed to fill a need or purpose for one user, and monitors a specific application or queue. In pertinent gadgets, we included the ability to drill down into more detail through the use of hyperlinks to intermediate pages or Sandra database items. The code for each gadget can be

easily edited and adapted for another user's purpose, if they would like to use the Credit Dashboard. The first gadgets on the Credit Dashboard were created by Bartek as proof of concept for the project, and contain information on the number of items in different queues. We added five unique gadgets (seven gadgets total) to the Dashboard and later eliminated the original gadgets from Bartek when their data was incorporated into one of the gadgets we designed.

Our first gadget, named Sandra Database Last Five Trades, pulls information on the last five trades entered into a Processed queue of the Sandra database and displays the fields in a table. The next gadget developed, Blotter Last Five Events, displays the last five events that a user has entered in their electronic blotter, called E-Dash (or CST Sales Tool). The Last Ten Alerts gadget provides a table of the last ten Alerts from a queue of alerts sent from an application, and generates popup notifications when a new alert has arrived. The Trades and Alerts gadget displays information on items in various queues, including the queues displayed in the original gadgets by Bartek. A table is used to display the number of items, a pie chart for each queue displays the ratio of different queues inside each larger queue, and a bar chart displays each queue, and their ratios, in a single chart. Blotter User Statistics is a gadget developed to display users launching their electronic blotter. A tree map shows the usage of the application in comparison with different department teams, and a graph displays the users and launches over the last ten days.

The final deliverable of this project is a Credit Dashboard with seven gadgets that monitor various aspects of system health within Quartz for users to view. The Credit Dashboard can be further improved through the generation of new gadget ideas by identifying users' needs, and the addition of more gadgets to the Dashboard. If more teams within GMRT begin using the Dashboard, further improvements can be made by creating individual customizable dashboards for users. This can be achieved by first editing current and future gadgets to allow for easy customization for displaying information from different applications or queue paths. Once gadgets are customizable to users, the Credit Dashboard could allow for the addition and removal of gadgets at a user's convenience, and preference storage so that upon launch, a user's customization is saved.

A personal, customizable Dashboard would greatly benefit users by allowing them to easily monitor system health, and any data relevant to their work, so that issues, errors, or discrepancies can be dealt with quickly before it becomes a larger issue.

Acknowledgements

We would like to thank the following individuals for their contributions to our project. Without their assistance, the project would not have been completed to its fullest possible extent.

Professor Kevin Sweeney

Professor Micha Hofri

Professor Xinming Huang

For providing guidance and encouragement throughout the project

Peter Bartek

Fred Hsu

Gokul Menon

Regina Roberson

Mike Yang

For their assistance and guidance in making the development and application processes of the project systematic and successful

Table of Contents

Table of Figures

Chapter 1: Introduction

Over the last one hundred years, technology has drastically changed the way in which humans conduct their lives. Social interactions, careers, travel, and businesses have all been affected by the wave of technological advances, and the advent of computers and access to internet further served to improve the quality of human life. Banks too have been modernized and improved through the use of new technologies to more easily and efficiently run their business.

We observed what seemed to be a strong commitment to quality at Bank of America, including the firm's technology department. In an effort to create a single system for all market data, trades, risk assessment and approval workflow, a core team of technology employees developed Quartz. Quartz permits all users access to the same base materials, which allows for easier collaboration, sharing, and access to data. The system offers many different applications to users, from browsers to development tools and soon, an electronic blotter.

With such a large system containing so many smaller components, keeping track of specific applications becomes a difficult chore. A user can look through DB Browser, a browser for the Sandra database that contains all items, to find a specific process, queue of events, or transaction, but it is not easy to see something as simple as the last recorded trade that failed. Going through the process of finding this information takes time and effort, and Quartz users have many more important things they could be doing instead.

Quartz would benefit from a central location for system information that is easy to view and keep track of any wanted occurrences. The Quartz Core Team has already developed Dashboards that display statistics on the adoption of Quartz and developer's work, displayed on TV's on the 31st Floor of the Bank of America Tower at One Bryant Park, and also available through Quartz. While statistics on Quartz users and lines of code committed is pertinent and interesting to look at, additional system information and monitoring of processes would be beneficial to users.

Our project aims to fill this need through the creation of a Credit Dashboard containing gadgets that monitor system information. The Dashboard acts as an easy to view central location for system information that all users have access to, and the gadgets are developed and designed to display relevant information in a style that is both easy to understand and aesthetically pleasing. Each gadget begins with a high-level view of an application, process, or queue, but has the ability to drill down into more information by providing links to either intermediate pages containing more information, or straight to the Sandra database to see the item directly. This way, users receive both an overview of relevant information, and easy access to more details if they need to know more.

Chapter 2: Background

This chapter reviews information pertinent to the understanding and development of the creation of the Credit Dashboard.

2.1 Credit Trading at Bank of America

Credit Trading involves the buying and selling of corporate bonds and credit derivatives. A corporate bond is a debt security issued by a corporation and sold to investors. The bond is then paid for by future operations of the company. The interest rate at which the company pays back its bond holders is determined by their creditworthiness and credit rating. A company with a higher credit rating will pay a lower coupon rate than a company with a low credit rating. This is because the company with the higher credit rating is deemed to be more likely to pay back their investors and is therefore less likely to default. The higher coupon rate is required on the lower credit rated company in order to entice investors to take on the extra risk associated with lending them money. In order to protect themselves from the interest rate risk and default risk of holding corporate bonds, Bank of America and other banks will also trade credit derivatives.

Bank of America uses credit derivatives to manage their exposure to credit risk. The most common credit derivatives are credit default swaps (CDS). A CDS acts like a form of insurance that protects against the failure to repay, or default, of the underlying asset. When Bank of America purchases a CDS, the firm receives the credit protection while the seller guarantees the protection in the case of default. The risk of default is then held by the seller of the CDS instead of by Bank of America. This is used to manage the risk of their portfolio. With the large number of mortgage loans, auto loans, and personal loans that are issued by banks it is important that they are able to protect themselves from the non-repayment of these loans in order to stay in business. Since banks trade with each other, they all trade the same instruments while relying on their proprietary software and strategies to give them an edge in the market.

2.2 Global Markets Research Technology

The Global Markets Research Technology (GMRT) Division within Bank of America develops proprietary and non-proprietary software in order to support the Global Markets businesses including Equity, Electronic Trading, Rates & Currencies, Credit and Structured Products, Commodities, Research, Sales, and Capital Markets. The Credit Trading Technology Department supports the sales teams and traders on the trading floor. They develop applications to help make the day-to-day operations of the sales teams and traders easier. Ideas for applications will either be generated themselves or they will meet directly with the teams to talk about any issues they are having and ideas they have for applications. If an application is feasible, it will be assigned to a team to begin development. Once an application is finished it will be released to a few sales team members or traders as a beta application to work out any bugs in the system. Here, the teams will make sure the application works well with the current systems being used and functions properly. After beta testing, the application will be released to the teams for use in their everyday operations. Continuous monitoring of the application as well as updates will be performed as long as it is in use.

Over the last few years, one of the major projects from the GMRT department is Quartz, which aims to become an all-in-one system for market data, risk analysis, trades, and approval workflow. The Quartz Documentation lists that the ultimate vision for Quartz is:

- One platform for trades, market data, analytics, and risk measures across asset groups
- Front-to-back pricing, risk management, lifecycle, settlements and approval workflow
- Quants/Tech/Front Office/Middle Office/Operations developers work on the same code base – tear down silos
- Quick turnaround for new instruments, risk and Front Office tools without sacrificing maintainability

Quartz is heavily used in GMRT, but many sectors of the company never open it. The Quartz Core Team and many others in the department work towards making applications that would interest users into using the system.

2.3 Quartz

Quartz is an object-orientated all-in-one platform for trades, market data, risk management, and development groups created and used exclusively by Bank of America. The concept behind this system is transparency; all users have access to view, contribute, and modify content in the system, with some restrictions. This allows for users to more easily access data, analytics, and other tools needed for work. Collaboration is much simpler for developers, as they all work on the same base code. The system also provides quick turnaround time for new instruments and tools through a unique approval workflow system.

The overall system is called Quartz, but is made up of many smaller components and applications. The base tool used is QZDesktop, which users can install on their computers to access Quartz similar to how you would a Windows folder system (figure 1). Users can find shortcuts to Quartz applications, add applications to their favorites, and easily add programs that they want to automatically start. The applications are organized by folder, applications for Credit located in that folder, and applications for Development in the Dev folder.

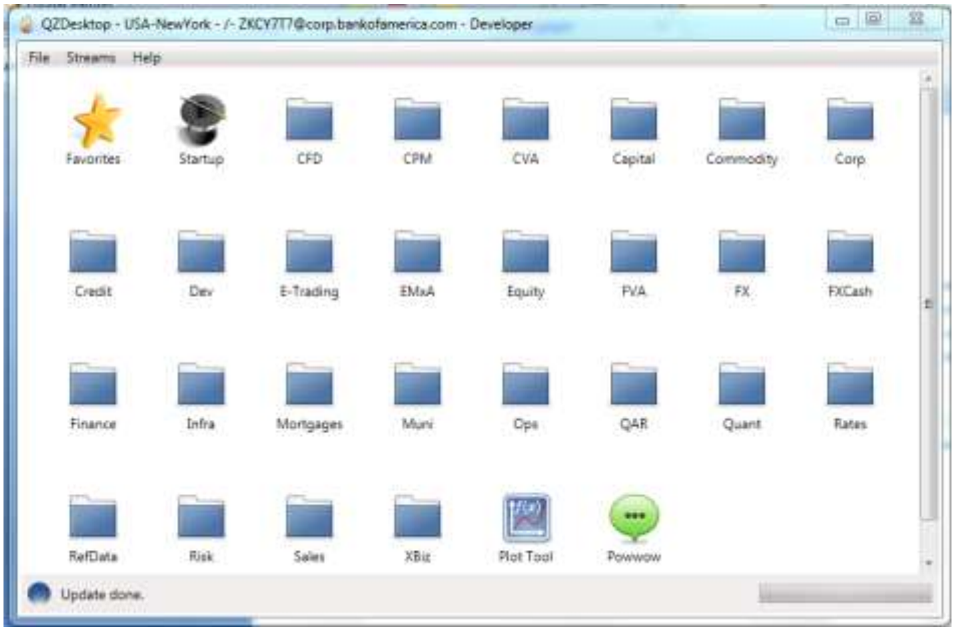
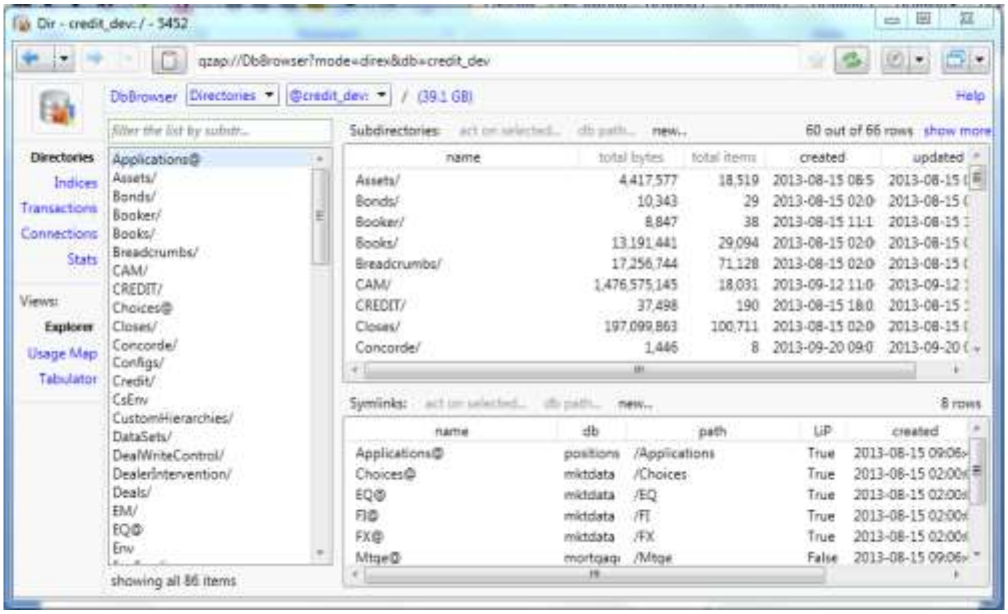


Figure 1. QZDesktop

QZDesktop offers many applications for different user areas. The applications general to all Quartz users and applications most pertinent to this project are described in the following sections.

2.3.1 Sandra: The object database

All object types and transactions that occur through Quartz are stored in the Sandra Database. The transactions can be viewed using the DB Browser, which allows a user to switch from several different databases, and filter through different subdirectories to find a specific transaction (figure 2). Based on the location of an object inside of the DB Browser, a program can retrieve information from the Sandra Database, or use the location to generate a URL to launch the object in the DB Browser.



Applications@
Assets/
Bonds/
Booker/
Books/
Breadcrumbs/
CAM/
CREDIT/
Choices@
Closes/
Concorde/
Configs/
Credit/
CsEnv
CustomHierarchies/
DataSets/
DealWriteControl/
DealerIntervention/
Deals/
EM/
EQ@
Env

name	total bytes	total items	created	updated
Assets/	4,417,577	18,519	2013-08-15 08:5	2013-08-15 (
Bonds/	10,343	29	2013-08-15 02:0	2013-08-15 (
Booker/	8,847	38	2013-08-15 11:1	2013-08-15 1
Books/	13,191,441	29,094	2013-08-15 02:0	2013-08-15 (
Breadcrumbs/	17,256,744	71,128	2013-08-15 02:0	2013-08-15 (
CAM/	1,476,575,145	18,031	2013-09-12 11:0	2013-09-12 1
CREDIT/	37,498	190	2013-08-15 18:0	2013-08-15 1
Closes/	197,099,863	100,711	2013-08-15 02:0	2013-08-15 (
Concorde/	1,446	8	2013-09-20 09:0	2013-09-20 (

name	db	path	UP	created
Applications@	positions	/Applications	True	2013-08-15 09:06
Choices@	mktdata	/Choices	True	2013-08-15 02:00
EQ@	mktdata	/EQ	True	2013-08-15 02:00
FI@	mktdata	/FI	True	2013-08-15 02:00
FX@	mktdata	/FX	True	2013-08-15 02:00
Mtge@	mortgage	/Mtge	False	2013-08-15 09:06

Figure 2. Sandra Database through DB Browser

DB Browser has three views, the Explorer, Usage map, and Tabulator. For most cases, the Explorer view (figure 2) is the easiest way to navigate, as it functions like a folder with subfolders. Clicking into an item drills down into that folder, and any folders under it. The Tabulator works in the same way, but in the view of a table blotter. The Usage Map view, while visually appealing, does not make it easy to dive into less well-used directories, as the directories with the largest amount of data appear larger than those with less data (figure 3).

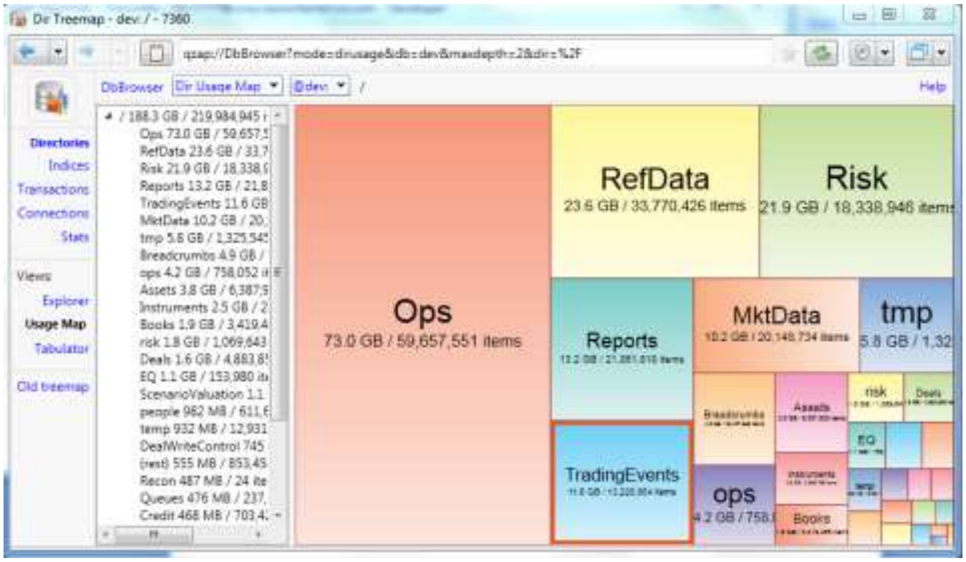


Figure 3. Sandra Database through DB Browser, Usage Map view

The Browser is difficult to navigate through – objects are usually stored with a generated name, so searching for a specific trade without the exact title would be impossible. To access an object, a user needs to first be inside the correct database inside Sandra, and then navigate correctly through the subdirectories before filtering for the exact object name.

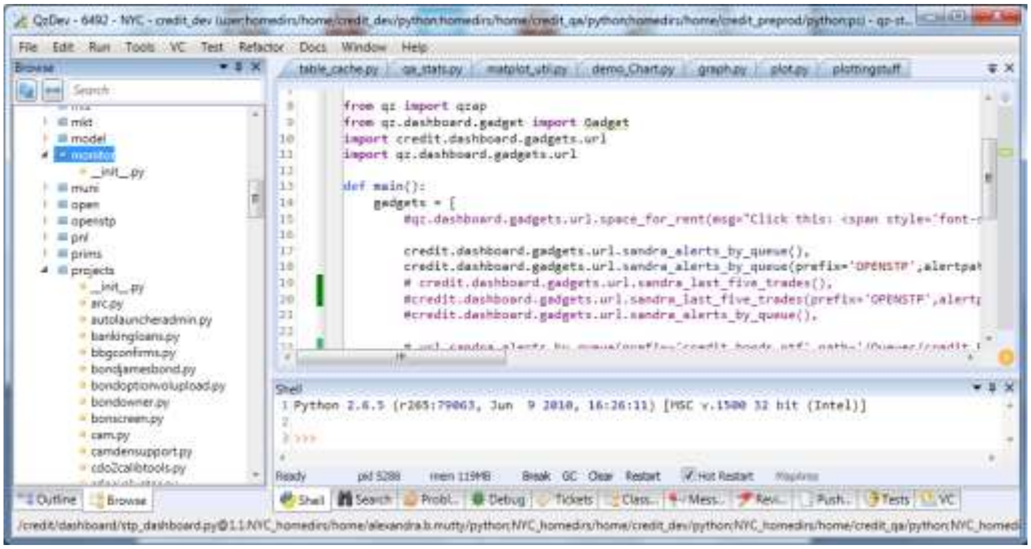
Searching by index can, in some cases, be more beneficial than going through the directory. Information in the Indices is stored in an index and can be accessed if the user knows what index they need to search in. The DB Browser also provides the ability to browse recent transactions, recent connections to applications, and message statistics.

2.3.2 Quack: Quartz Access Control

Quack controls access for groups of users to certain areas.

2.3.3 QZDev: Development application

QZDev is an application for the development of code projects (figure 4). Code can be written, reviewed, and put into production through this application. A user writes or changes code, "commits" it, and "pushes" it to another user or to be reviewed. This system designs means that collaboration and code sharing is very easy. The code is kept local to the user until it has been committed so that an error does not affect other systems. After review, it can be put into production. This system allows for quick turnaround, and changes can be pushed, reviewed, and put into production in half a day.



```
from qz import qzap
from qz.dashboard.gadget import Gadget
import credit.dashboard.gadgets.url
import qz.dashboard.gadgets.url

def main():
    gadgets = [
        #qz.dashboard.gadgets.url.space_for_rent(msg="Click this: <span style='font-

        credit.dashboard.gadgets.url.sandra_alerts_by_queue(),
        credit.dashboard.gadgets.url.sandra_alerts_by_queue(prefix='OPENSTP',alertpat
        # credit.dashboard.gadgets.url.sandra_last_five_trades(),
        #credit.dashboard.gadgets.url.sandra_last_five_trades(prefix='OPENSTP',alert
        #credit.dashboard.gadgets.url.sandra_alerts_by_queue(),
```

```
1 Python 2.6.5 (r265:79063, Jun  9 2010, 16:26:11) [MSC v.1500 32 bit (Intel)]
2
3 >>>
```

Figure 4. QZDev application

QZDev is the program we will be using for the development of the gadgets for this project. The program features searching, debugging, and testing capabilities that can be used in the development of coding projects. Some of the key features that were used during this project are on the left hand sidebar, which provides browsing through python files, and an outline of the current python document. This helps a developer to keep track of where they are in a large project. The searching capability was also heavily used in this project, as it can find the source code for any definitions that were imported into the current document. The most important tool of any development application is, of course, the debugger, which was heavily used over the course of this project to find errors.

2.3.4 DAG: Quartz Dependency Graph

DAG is a parse-time extension for python and contains a framework and programming model for developers. The model features subscriptions, layers, and filtering so that time dependent data can be updated without non-time dependent data having to be as well. Developers create DAG objects that are automatically serialized into the Sandra database.

2.3.5 AMPS: Real-time publish/subscribe

AMPS is a third party message system that allows users to subscribe or publish from real time data. For example, an application that looks at the last five trades completed would configure AMPS to notify the application when another trade had completed to keep the information updated.

2.3.6 Inform: Real-time Sandra notifications

Inform is a component of Quartz that uses the AMPS messaging system to notify of changes in the Sandra database based on the Sandra transactions. Inform allows users to subscribe to changes, and be notified of that change. In the case of an application, inform will tell an application that something had changed, and the application itself will call to update or make any changes. This application, used with the correct programming style, can allow programs to run quicker and more efficiently. Rather than updating every few seconds, the program can refresh only when a change has been made. This application can also be used to make a single change, rather than updating an entire batch of data, making it more efficient. For example, a program that displays the last ten events to occur in the database can add the latest event in and knock the oldest out instead of finding the last ten events and displaying them a second time.

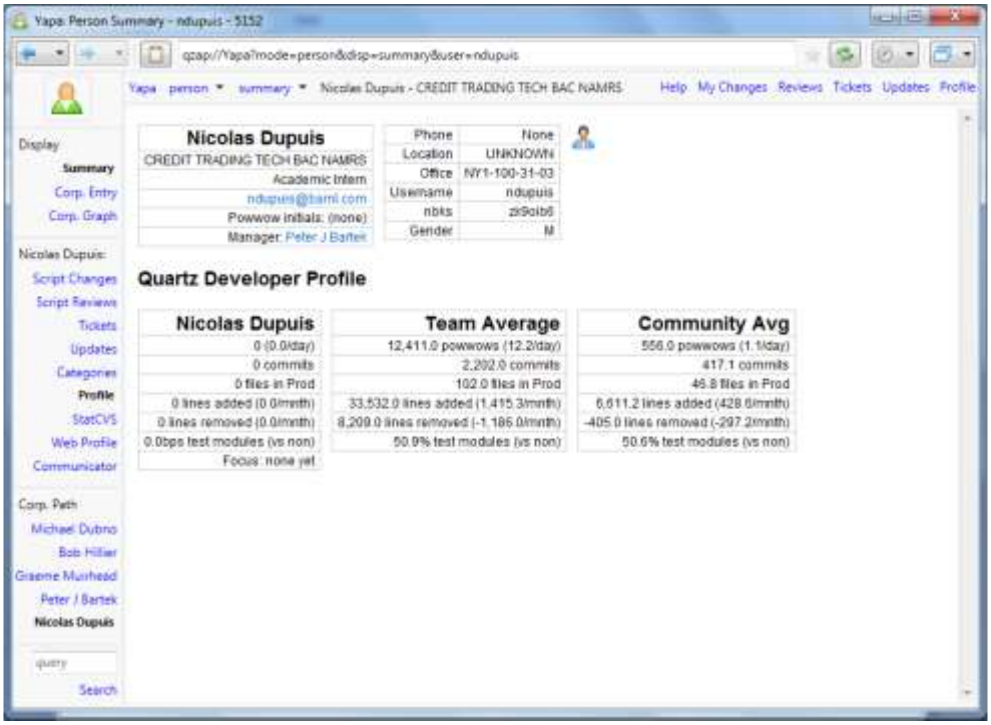
This feature was used frequently in the gadgets created in this project, as many of them featured real-time information. The ability to make an update when a change has been made meant that the information displayed in our gadgets were as accurate and up to date as possible.

2.3.7 Bob: Quartz Batch Scheduler

Bob is the scheduling system for Quartz, controlled through the Bob Monitor. Bob jobs can be specified by creating job files in QZDev.

2.3.8 Qzap: Quartz Addressable Protocol

Qzap allows for users to translate and access information as if they were working in a web browser. Applications can be "Qzapified" to contain a Qzap URL and be accessed without losing the state of the application. For example, employee information pages are accessed by browsing through Qzap, or typing in the Qzap URL (figure 5). For convenience, Qzap saves a user's top preferences, so that a suggestion menu of links drops down from the top search bar in the same way most web browsers do. To come soon in Qzap is the ability to see your search history and save favorite pages.



Nicolas Dupuis
CREDIT TRADING TECH BAC NAMRS
Academic Intern
ndupuis@baml.com
Powwow initials: (none)
Manager: Peter J Bartek

Phone	None
Location	UNKNOWN
Office	NY1-100-31-03
Username	ndupuis
nbks	zk9oib6
Gender	M

Quartz Developer Profile

Nicolas Dupuis	Team Average	Community Avg
0 (0.0/day)	12,411.0 powwows (12.2/day)	556.0 powwows (1.1/day)
0 commits	2,202.0 commits	417.1 commits
0 files in Prod	102.0 files in Prod	46.8 files in Prod
0 lines added (0.0/mnth)	33,532.0 lines added (1,415.3/mnth)	6,611.2 lines added (428.6/mnth)
0 lines removed (0.0/mnth)	8,209.0 lines removed (-1,186.0/mnth)	-405.0 lines removed (-297.2/mnth)
0.0bps test modules (vs non)	50.9% test modules (vs non)	50.6% test modules (vs non)
Focus: none yet		

Figure 5. Employee Profile through Qzap

Most Quartz applications are addressable through Qzap, including the DB Browser. The Qzap application is particularly helpful as it not only provides users with the ability to easily navigate to certain applications, but also provides hyper-linking from text to launching a Qzap URL. Part of the Qzap code includes NavURL, which can be used to generate Qzap URLs and to launch them from their applications or pages. In this project, we use Qzap and NavURL to generate Qzap URLs for displayed information, so that a user can drill down into more information by clicking on the item they want to view.

2.3.9 Powwow: The Quartz communication system

Powwow is Quartz’s built in communication system. In Powwow are different groups (similar to internet chat rooms) that users can join to speak with each other. Users can also create their own groups. They are used often for questions during any large assemblies or lectures that are offered, and as a quick and simple messaging system for users.

2.3.10 Quartz Deal Model: A format for all trades

To make the Quartz system as simple as possible, there needs to be consistency across the platform. For trades, the model to follow is the Quartz Deal Model, which formats trades into an object-orientated view (figure 6). In it, you have a Position (buy or sell). A Position is made up of some number of the same Instrument. A Book contains Positions, and a portfolio contains Books or other Portfolios. Altogether, this makes up a Deal. The content of a Deal is edited through the use of Events. For example, rather than editing the Deal so that we have 5000 Instruments instead of 10000, an event is created that does this. This keeps track of the history of a Deal, and prevents mistakes.

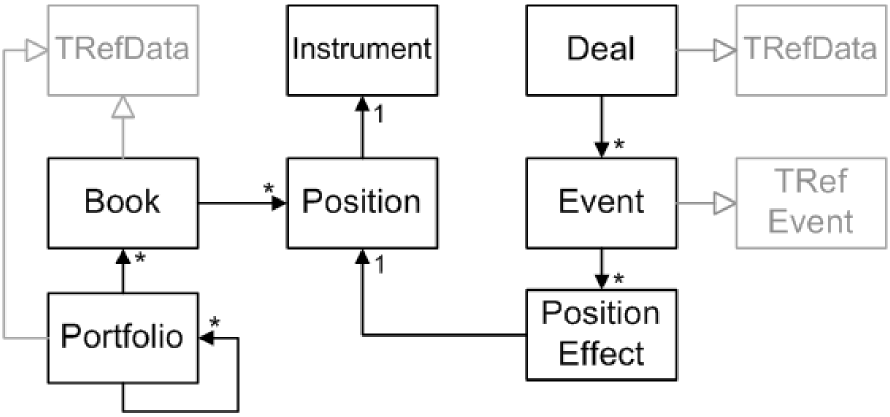
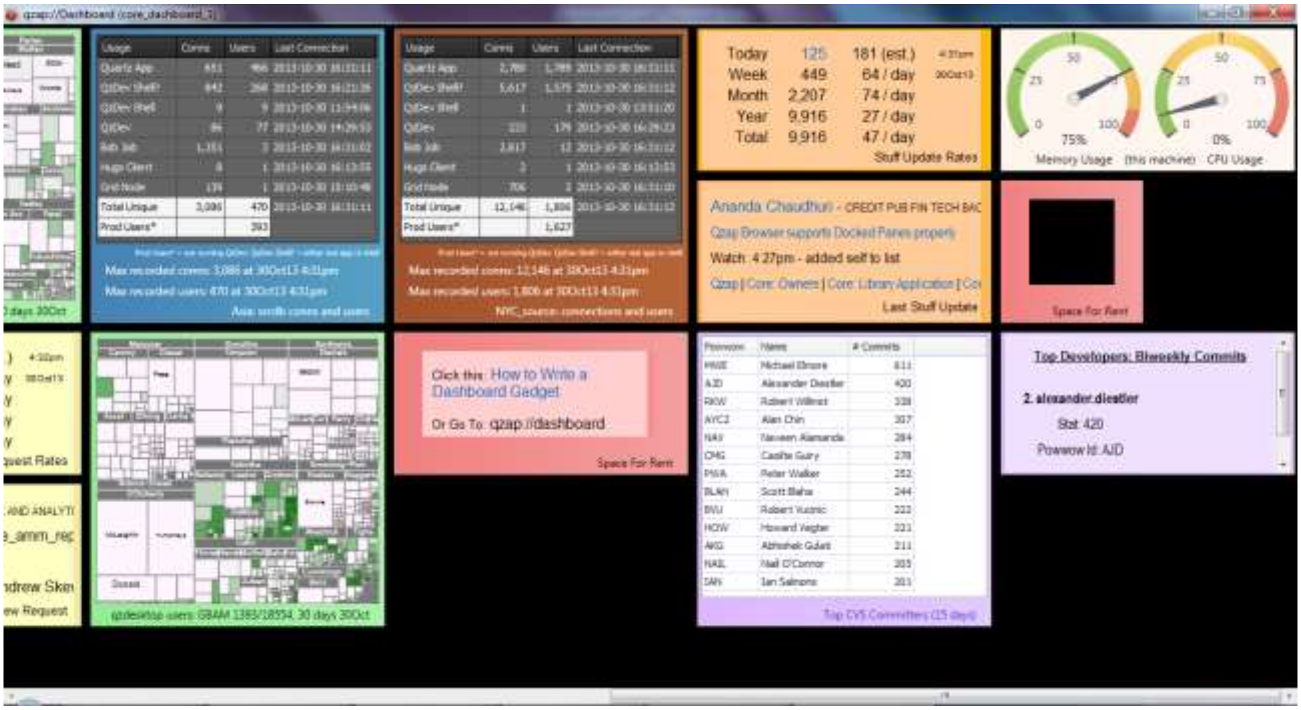


Figure 6. Quartz Deal Model

2.3.11 Dashboard: Dashboard of Quartz statistics

The Dashboards (Dashboard and Dashboard 2) contain gadgets that display information on Quartz statistics (figures 7 and 8). For example, one gadget has a tree map of Quartz users, to see what sections of the company are still not utilizing it, and what sections are using it frequently. There are also gadgets to display the machine’s CPU and RAM usage, Apps that look at conflicts in Sandra, Last Review Requests, and more. The Dashboards can be loaded from a user’s computer, but are also displayed on TVs on the 31st Floor of the Bank of America Tower, near the Quartz Core Team.



qzap://Dashboard (core_dashboard_1)

Usage	Conns	Users	Last Connection
Quartz App	651	466	2013-10-30 16:31:11
QzDev Shell†	842	266	2013-10-30 16:21:26
QzDev Shell	9	9	2013-10-30 11:54:06
QzDev	86	77	2013-10-30 14:29:53
Rob Job	1,351	3	2013-10-30 16:31:02
Hugs Client	8	1	2013-10-30 16:13:55
Grid Node	139	1	2013-10-30 15:10:48
Total Unique	3,086	470	2013-10-30 16:31:11
Prod Users*		393	

Max recorded conns: 3,086 at 30Oct13 4:31pm

Max recorded users: 470 at 30Oct13 4:31pm

Asia: smth conns and users

Usage	Conns	Users	Last Connection
Quartz App	2,789	1,789	2013-10-30 16:31:11
QzDev Shell†	5,617	1,575	2013-10-30 16:31:12
QzDev Shell	1	1	2013-10-30 13:31:20
QzDev	222	179	2013-10-30 16:29:23
Rob Job	2,817	12	2013-10-30 16:31:12
Hugs Client	2	1	2013-10-30 16:13:53
Grid Node	706	2	2013-10-30 16:31:10
Total Unique	12,146	1,806	2013-10-30 16:31:12
Prod Users*		1,627	

Max recorded conns: 12,146 at 30Oct13 4:31pm

Max recorded users: 1,806 at 30Oct13 4:31pm

NYC_source: connections and users

Today	125	181 (est.)	4:31pm
Week	449	64 / day	30Oct13
Month	2,207	74 / day	
Year	9,916	27 / day	
Total	9,916	47 / day	

Stuff Update Rates

Ananda Chaudhuri - CREDIT PUB FIN TECH BAC

Qzap Browser supports Docked Panes properly

Watch: 4:27pm - added self to list

Qzap | Core: Owners | Core: Library Application | Co

Last Stuff Update

Space For Rent

Click this: How to Write a Dashboard Gadget

Or Go To: qzap://dashboard

Space For Rent

Powwow	Name	# Commits
MWE	Michael Elmore	611
AJD	Alexander Diestler	420
RKW	Robert Willmot	338
AYC2	Alan Chin	307
NAV	Naveen Alamanda	284
CMG	Caolite Gury	278
PWA	Peter Walker	252
BLAN	Scott Blaha	244
BVU	Robert Vucinic	222
HOW	Howard Vegter	221
AKG	Abhishek Gulati	211
NAIL	Niall O'Connor	203
IAN	Ian Salmons	203

Top CVS Committers (15 days)

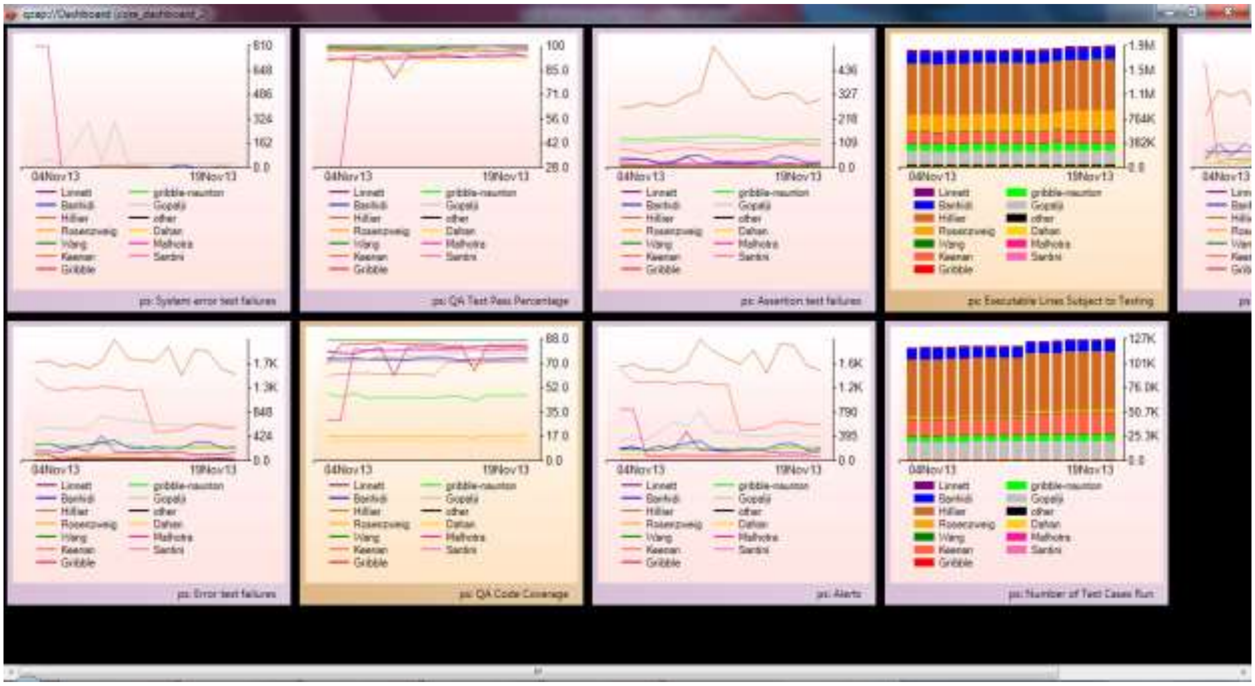
Top Developers: Biweekly Commits

2. alexander.diestler

Stat: 420

Powwow Id: AJD

Figure 7. Quartz Dashboard



qzap://Dashboard (core_dashboard_2)

Figure 8. Quartz Dashboard 2

The Dashboards are desktop applications that, on startup, check a global gadget directory for what gadgets to display. A Developer could create their own Dashboard and gadgets by importing the base dashboard definitions and adding their own gadgets. All of the gadgets to be displayed need to be referenced in the Core team’s qzap_gadgets.py file, which acts as a sort of directory for a dashboard to see what gadgets are available, and then called from the dashboard main(). As long as the user does not commit or push any files, the dashboard and gadgets will remain local to them. An example of a dashboard main() calling one gadget named sandralastfivetrades that utilizes clickable Qzap URL links:

```
def main():
    gadgets = [
        credit.dashboard.gadgets.url.sandra_last_five_trades(),
    ]
    for navUrl in gadgets:
        level, elem = qzap.launchAppPageFromURL(str(navUrl))
    return elem
```

2.4 Python

The main programming language used for the project is Python. Comparing to other mainstream programming languages, Java and C++ for instance, Python requires less detailed syntax and provides better readability, and thus is easier to learn. It supports multiple programming paradigms, including both object-oriented and functional programming, making it a flexible and dynamic system. Python code can be packaged into standalone executable programs and runs on Windows, Linux/Unix, Mac OSX, and has been ported to the Java and .NET virtual machines. As such a versatile programming language, it is used often for the building of credit trading technology platforms and applications.

2.5 Application Development

Basic software development utilizes the waterfall model method when programming (figure 9). The waterfall method encompasses the following order of steps: project requirement, business analysis, software design, implementation, testing, and maintenance. Developers will typically follow this order of steps; however they can go back to prior steps when needed.

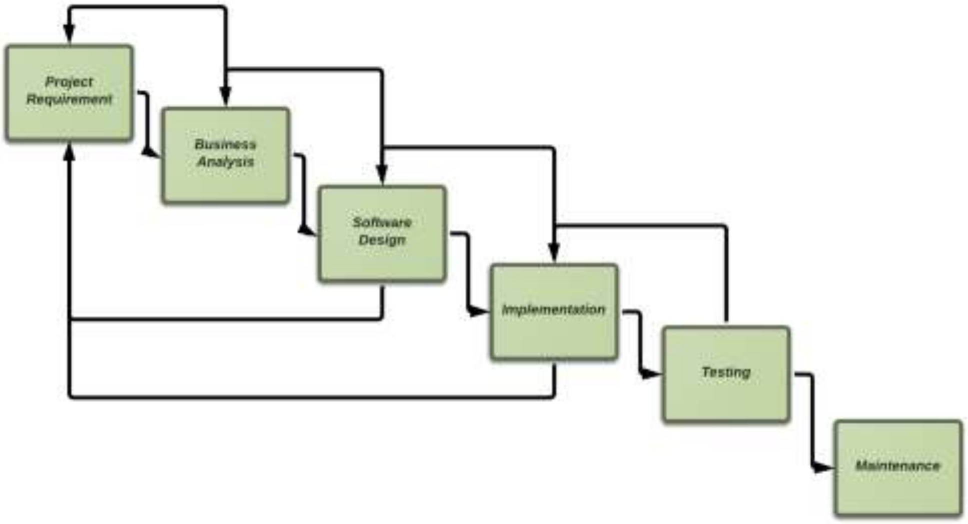


Figure 9. Waterfall Model

During the project requirement and business analysis phase, information is gathered about the required outcomes of the project. This is where the project scope, requirements, and timeline are initially decided upon. The software design phase is where all new features are discussed and a general design of the software will be created. During implementation is where the major software development and research takes place. Testing takes place concurrently with the implementation process. Any errors found are corrected and the performance of the product is tested across multiple platforms. Maintenance is carried out throughout the life of the product to ensure sustainability.

Chapter 3: Methodology

The goal of this project was to create a Credit Dashboard containing gadgets to display the overall health of the credit system of Quartz, and easily drill down into details when necessary. This tool is beneficial for users to be able to quickly identify and access any issues that may be occurring.

The following chapter explains the methods and processes followed to achieve the project goal of creating a Credit Dashboard. Project tasks were completed in a linear fashion, concurrent to the writing of the final report (figure 10). The final week of the project was dedicated to final revisions, and preparation for the final presentation on December 17th, 2013.

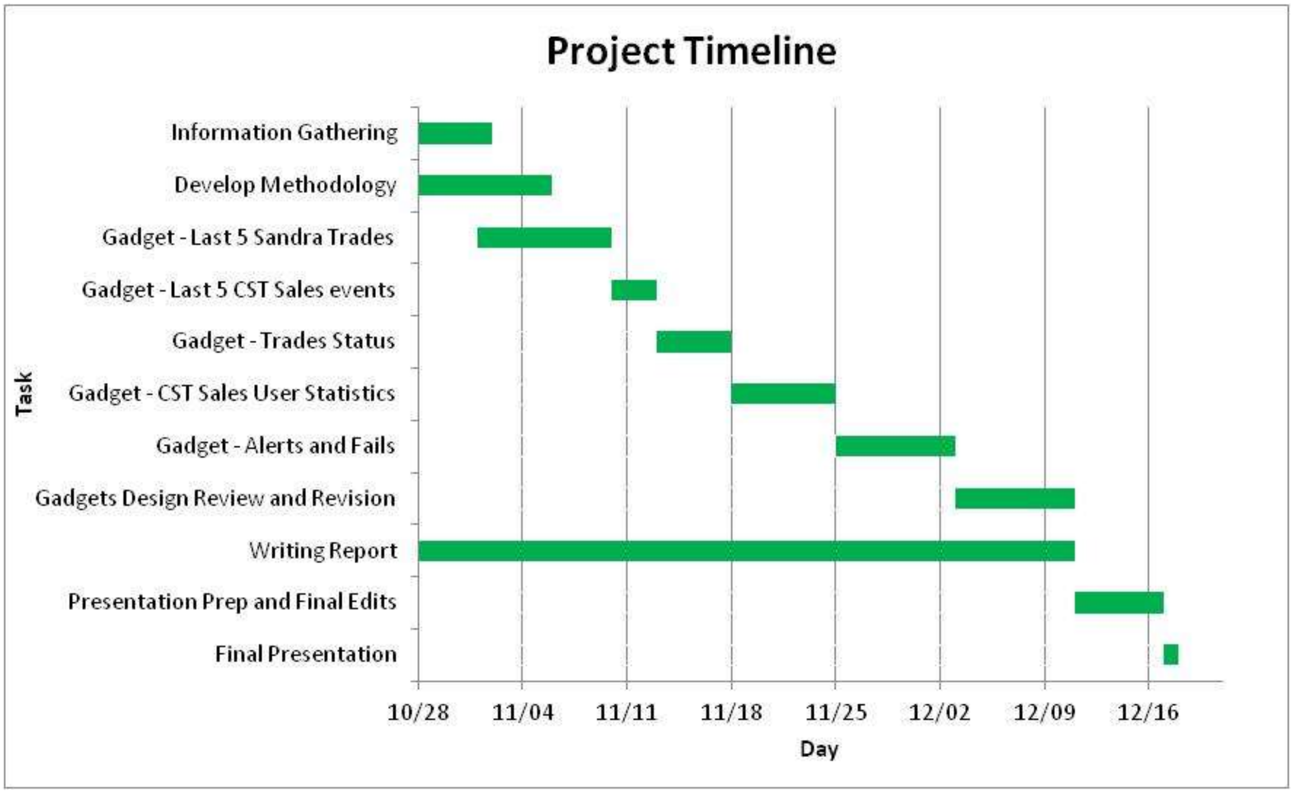
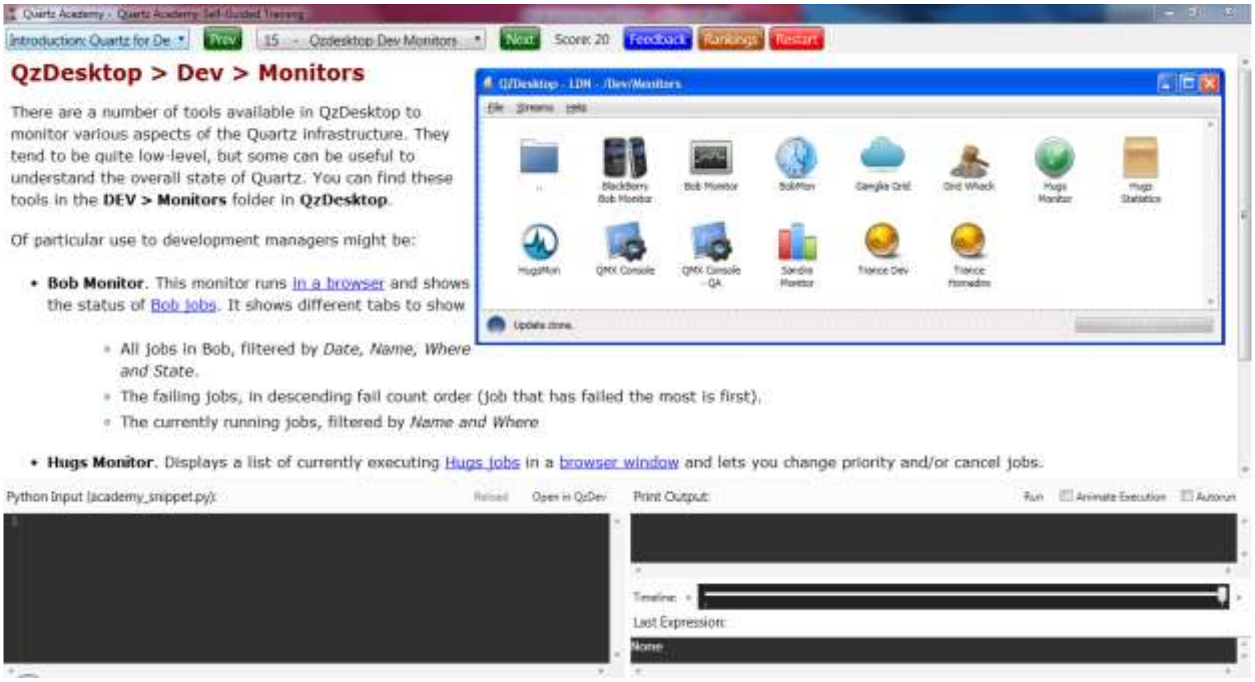


Figure 10. Project Timeline

3.1 Research

This project required a solid understanding of Quartz, Bank of America's all-in-one platform for developers, market data, trades, risk management, and approval workflows. As Quartz is an internal system, created and used exclusively by Bank of America, all research into the system was through the available tutorials, provided by Bank of America for new hires to learn about the system. The tutorials are recorded presentations on different Quartz topics, by different members of the Quartz Core team. We began with the Introduction tutorial, and then jumped into relevant topics as necessary. Peter Bartek, our sponsor liaison for the project, gave us a brief overview and demonstration of the Quartz system in our first meeting (Meeting Logs, Appendix A).

Concurrently with watching the tutorial videos, we used Quartz Academy to learn about the system (figure 11). Quartz Academy is an interactive approach to learning the system, where important topics are broken down into modules and can be easily stepped through by the user. Topics include an introduction to Quartz, extensive Python tutorials, and how to use the development environment within Quartz.



QzDesktop > Dev > Monitors

There are a number of tools available in QzDesktop to monitor various aspects of the Quartz infrastructure. They tend to be quite low-level, but some can be useful to understand the overall state of Quartz. You can find these tools in the **DEV > Monitors** folder in **QzDesktop**.

Of particular use to development managers might be:

- **Bob Monitor**. This monitor runs in a browser and shows the status of Bob jobs. It shows different tabs to show
  - All jobs in Bob, filtered by *Date, Name, Where and State*.
  - The failing jobs, in descending fail count order (job that has failed the most is first).
  - The currently running jobs, filtered by *Name and Where*
- **Hugs Monitor**. Displays a list of currently executing Hugs jobs in a browser window and lets you change priority and/or cancel jobs.

Figure 11. Quartz Academy Example

However, the best method for learning Quartz was actually using it. We spent time clicking through the QZDesktop application and opening up applications to see what they did. For the applications we used frequently, such as QZDev, AMPS, and Sandra, we would go back to the tutorial videos and watch the videos relevant to the application for a more solid understanding of how to use it.

These three methods combined served to help us learn to easily navigate and use pertinent Quartz applications for the purpose of completing our project objectives (see Chapter 2.3).

3.2 Project Objectives

We identified two main objectives for the completion of our project goal to create a Dashboard to display the system health for Quartz:

- Identify gadget specifications that teams would find useful and relevant in their work
- Create a dashboard with gadgets to display relevant information

In this section we summarize the methods used in order to complete each of the objectives.

3.2.1 Identify Desired Gadgets

The Credit Dashboard project began as an idea from the mind of Peter Bartek, who thought that a dashboard displaying a high level view of Quartz with the ability to drill down into details would be beneficial for users to see what is happening from a system view. Coming into this project, Peter Bartek already had a few ideas for possible gadgets; last five trades, list of Queues, technical errors, and business alerts.

We met with different development teams' members to see what they would be interested in seeing on the Credit Dashboard (Meetings Log, Appendix A). Each interview started with the same outline of questions (Appendix B), but became more discussion-based as we learned about what role each member played and what they would like to see in a credit dashboard. We used the ideas and concepts talked about in these interviews to develop new gadgets for the Dashboard. Meetings occurred on a week-by-week basis as gadgets were developed, so that each meeting could start with a view of the most recent Dashboard and gadgets created.

3.2.2 Create Dashboard and Gadgets

For each desired gadget from the previous section, we met the need of the user by developing the gadget for the Dashboard. The creation of each gadget followed the same process from idea to finished product (figure 12).

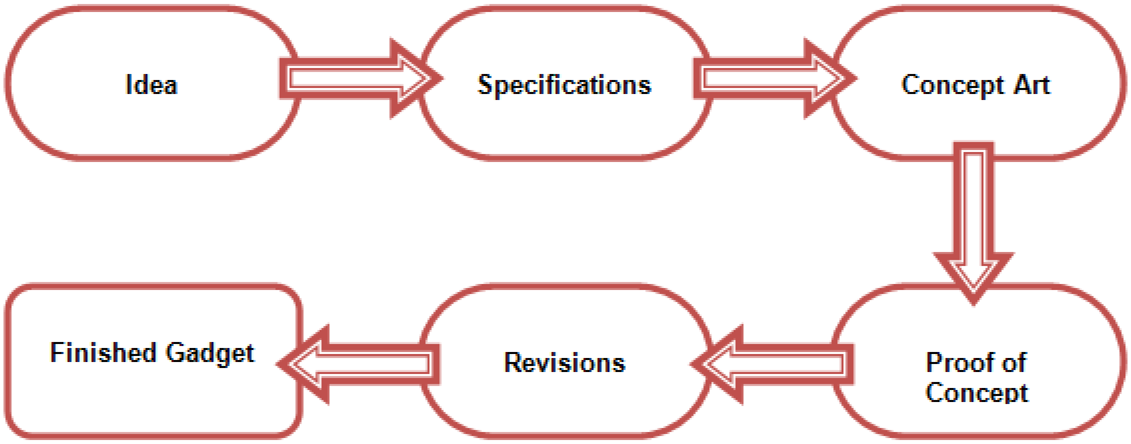


Figure 12. Gadget Development Process

Idea

Each gadget starts as an idea – a vague solution to either a current problem or to fill the need of a user. For the majority of the gadgets developed, the idea came from one of the team members, to fill one of their needs. Other times, the idea was generated from our project group to fill some need of a user, which in most cases was the ability to see a high level view of Quartz with the ability to drill-down when needed.

Specifications

After we have an idea, we flesh out the specifications for the gadget. We aim to answer the following questions:

- What does the Gadget do?
- What teams/who will be using the gadget?
- What information do we need to gather for the gadget?
- What information will be displayed?
- How will the information be displayed?

Concept Art

Once there is an understanding of what need the gadget fulfills and what it should do, concept art is created for how we would like the gadget to look. The information needs to be displayed in a way that is easy to follow, uncluttered, and aesthetically pleasing. By creating a picture of our idea of the final product, the process of writing the code for the gadget is made simpler by having an understanding of the final picture. The concept art was usually created in Excel to allow for easy manipulation of tables and graphs as ideas changed throughout the process.

Proof of Concept

From the concept art, we create a gadget that fulfills the gadget specifications. This requires using QZDev to write, test, and revise the code through the Application Development Process (see Chapter 2.5) until we have a working product.

Revisions

Once we have a working prototype of the gadget, we bring the gadget back to the original user that desired it for a design review and gather feedback. If necessary, we continue revising until we have a product the user finds satisfactory.

Finished Gadget

The finished gadget is added to the Credit Dashboard and pushed into production.

3.3 Gadget Production

When a gadget is finished, it goes through a series of steps to be put into production. The code is first pushed to, and reviewed by, our group's immediate manager (and sponsor liaison), Peter Bartek. Bartek reviews the code before pushing it up to be into production. Later edits to production code can be made, pushed, and reviewed in the same manner.

When a gadget is in Production, it means that the code is globally available to Quartz users – everyone can view it, and edit it in their own user area, but cannot make global changes. The gadgets are globally available, such that when each user launches the Credit Dashboard Icon, it references the global gadgets available. Because users can see the base code, they have the ability to adapt it to their own purposes, or make revisions and optimizations that they think would make the Dashboard or gadget run more efficiently. Their edits can be pushed through to the core team to production.

Chapter 4: Gadget Specifications

This chapter explains the structure of the created Credit Dashboard, how the dashboard and gadgets interact with one another, and the details of the purpose, reasoning, and development of each gadget on the Credit Dashboard.

4.1 The Credit Dashboard

The Credit Dashboard displays gadgets containing a high-level view of the credit system's functions, and is the culmination of this project's goals. Each gadget contains separate information on different relevant systems inside the credit division at Bank of America, and the Credit Dashboard acts as a central location for the information (figure 13).

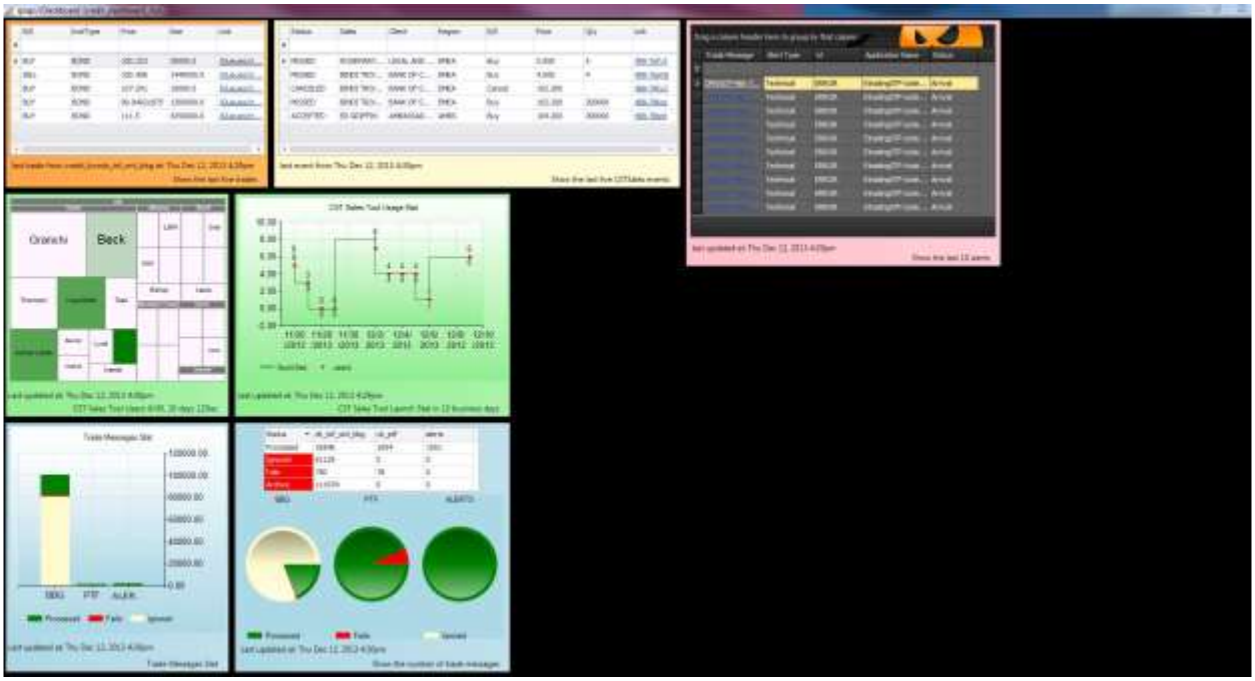


Figure 13. The Credit Dashboard

The Credit Dashboard is a desktop application, which enables a user to click the icon and have the program immediately launch. The icon to launch the Credit Dashboard is located inside QZDesktop, under the Credit/Beta path directory (figure 14).

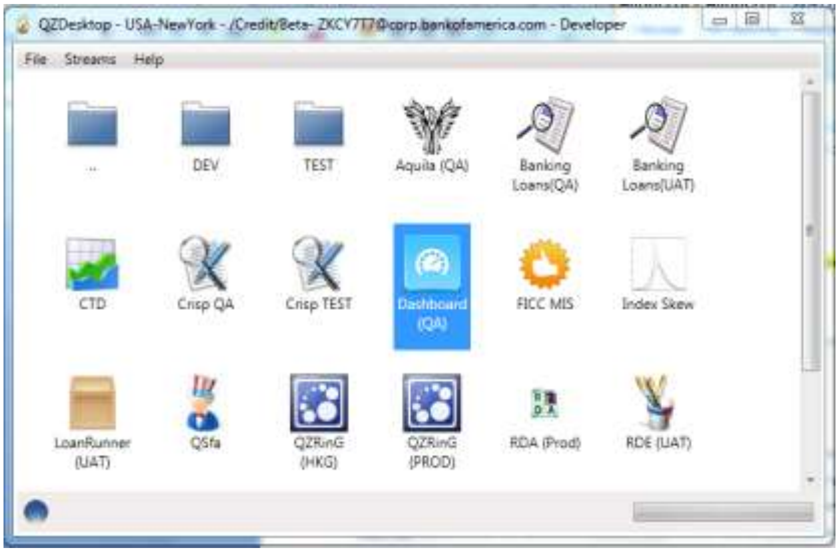


Figure 14. QZDesktop with link to Credit Dashboard (Dashboard (QA))

Chapter 2.3.11 explains the two dashboards the Quartz Core Team created for displaying Quartz statistics, and how they work. The Credit Dashboard works in exactly the same way. When the Credit Dashboard icon is clicked, it launches the user's version of the Credit Dashboard (which is the global version that every user has access to, unless they have made edits). Inside the dashboard code, all libraries and definitions (including the definitions for each gadget) are imported from the base code directory that each user has access to. This allows the dashboard to inherit that code and use its functions and definitions. The dashboard then calls for what gadgets to be displayed using the definitions it inherited (figure 15).

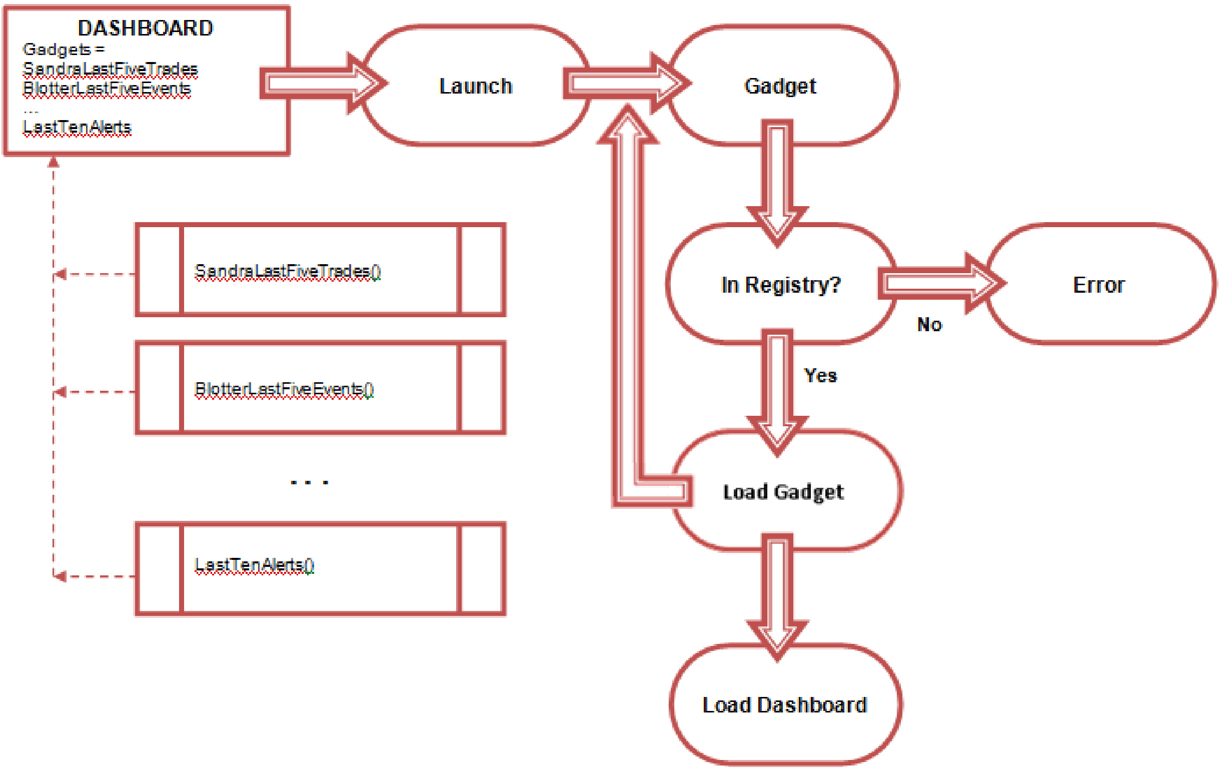


Figure 15. Process of Launching the Credit Dashboard

When the dashboard is launched, each gadget is taken in turn and checked that it is inside the global gadget registry. If the gadget is inside, the gadget is loaded. If the gadget is not in the registry, the program will throw an error. Once all the gadgets have been loaded, the dashboard is launched. A Bob Job running in the background gathers the actual information for each gadget, and upon launch, the gadget pulls from the stored Bob Job so that it does not have to do this work itself.

Currently, there are seven gadgets (five unique gadgets) on the Credit Dashboard. Two more gadgets that were created by Peter Bartek as proof of concept for this project were removed when their information was incorporated into gadgets our team created. Gadgets can be edited, and more gadgets can be added at later dates, if necessary. The Credit Dashboard and corresponding gadgets' code is viewable to all users. A user can suggest revisions, or add in a gadget they have created, and push the idea to the Quartz Core team if they are interested in having it added to the global Credit Dashboard.

4.2 Gadgets

In this section, each gadget on the Credit Dashboard is detailed, from its initial conception as an idea, the need for it, and the process of development.

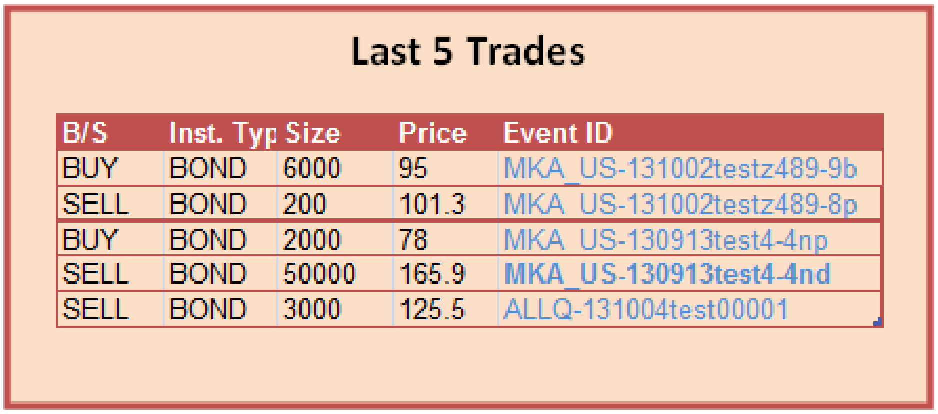
4.2.1 Sandra Database Last 5 Trades

Each time a trade occurs, it is stored in the Sandra database, viewable through the Sandra DB Browser. A trade first goes into a Queue, where it is processed or put into a Fails subdirectory if it cannot be processed. When finding a specific trade, a user has to open DB Browser, navigate to the correct database (as the DB Browser default is dev), locate Queues, and then search for their specific trade inside the Processed subdirectory.

Peter Bartek expressed that he would find it helpful to be able to see the last few trades from the Bloomberg Queue in the Sandra database without having to navigate through this system. In this case, a gadget that could display the most important information about the last 5 trades, and allow for the ability to drill down to view more information about each trade would be beneficial.

We looked at past trades to determine what information from a trade was most important to display to the user: the instrument type, size, position, price, and unique event ID. When looking at the gadget, users should have a broad overview of the trades, and are able to drill down to find what they want. Clicking on the Event ID should link them to the Sandra database and to the open trade they clicked on for easy access to information.

We first created a design for what we envisioned the final gadget to look like (figure 16).

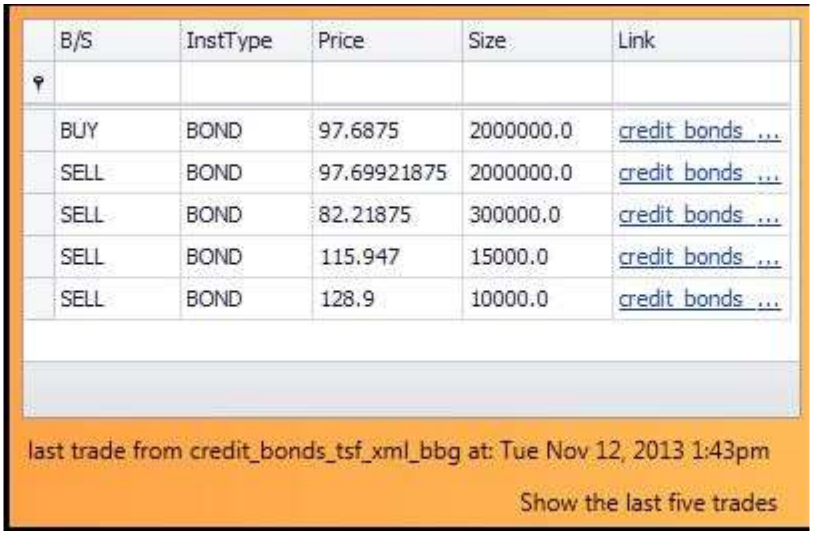


Last 5 Trades

B/S	Inst. Typ	Size	Price	Event ID
BUY	BOND	6000	95	MKA_US-131002testz489-9b
SELL	BOND	200	101.3	MKA_US-131002testz489-8p
BUY	BOND	2000	78	MKA_US-130913test4-4np
SELL	BOND	50000	165.9	MKA_US-130913test4-4nd
SELL	BOND	3000	125.5	ALLQ-131004test00001

Figure 16. Concept Art Last Five Trades

In the development of the gadget, we stayed true to the concept art with only a few minor changes (figure 17). We began by creating the table the information would go into, and then solving how to fill the information with data, and where to take that data from. All of the information comes from Sandra Database, and inside each transaction object for a trade is where we found the specific information about the trade. Inside the Sandra browser, all the trades we wanted to display are located in the credit_dev database, inside Queues/Processed. Once we had filled the table with the last five trades (with the correct information), we used Inform (See Chapter 2.3.6) to be notified whenever a new transaction is added into the database. When a new transaction is entered, the gadget again looks for the last five trades and fills the table with those. The last piece to be worked on was creating hyperlinks in the table, as Quartz uses Qzap URLS as opposed to those you would enter into a web browser. For each trade in the table, a URL is first generated and then used to hyperlink the text to the transaction in the Sandra Database.



B/S	InstType	Price	Size	Link
BUY	BOND	97.6875	2000000.0	credit_bonds_...
SELL	BOND	97.69921875	2000000.0	credit_bonds_...
SELL	BOND	82.21875	300000.0	credit_bonds_...
SELL	BOND	115.947	15000.0	credit_bonds_...
SELL	BOND	128.9	10000.0	credit_bonds_...

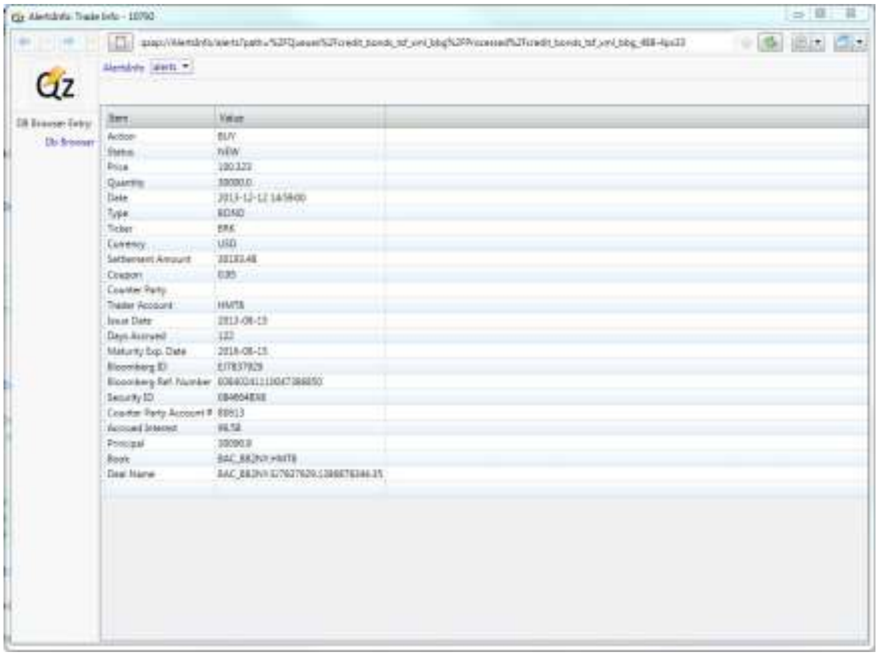
last trade from credit_bonds_tsf_xml_bbg at: Tue Nov 12, 2013 1:43pm

Show the last five trades

Figure 17. Sandra Last 5 Trades Gadget

The size of the columns in the table can be adjusted by dragging the column divider, but will reset to default size after the Credit Dashboard is closed and reopened. The trades themselves are not in any particular order, so clicking on the column headers can allow you to sort the table by that header. The information is updated when the Credit Dashboard is opened, and any time a new transaction is added in the Sandra database.

After a design review, we determined that it would be helpful to have an intermediate page between the link in the table, and the actual item in the Sandra database (figure 18). The page between would contain more information about the trade than the table, but less than the item in the database. The page also provides a link to the item in the database, if the user needs to find something specific that wasn't deemed necessary for the intermediate page.



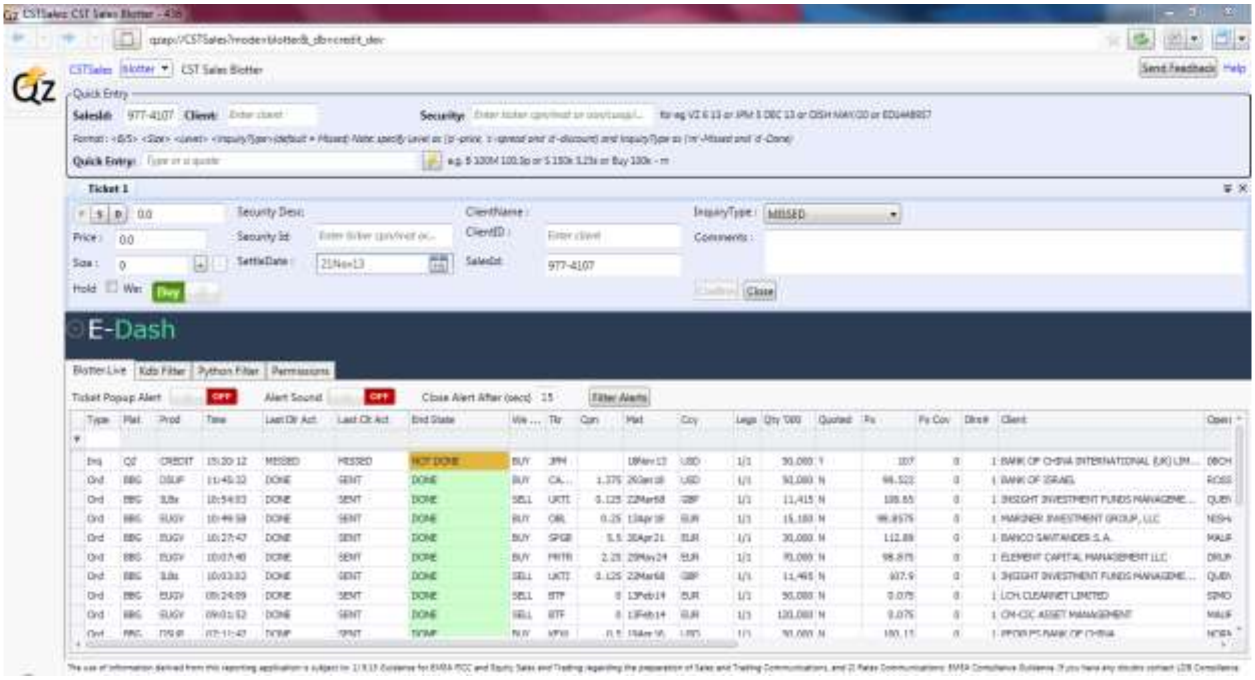
Item	Value
Action	BUY
Status	NEW
Price	100.323
Quantity	30000.0
Date	2013-12-12 14:58:00
Type	BOND
Ticker	BRK
Currency	USD
Settlement Amount	30183.48
Coupon	0.85
Counter Party	
Trader Account	HMTB
Issue Date	2013-08-13
Days Accrued	122
Maturity Exp. Date	2018-08-15
Bloomberg ID	EJ7837829
Bloomberg Ref. Number	0084002411100473888850
Security ID	084664BX8
Counter Party Account #	89813
Accrued Interest	86.58
Principal	30090.9
Book	BAC_BB2NY-HMTB
Deal Name	BAC_BB2NY:EJ7837829:1386876346.35

Figure 18. Intermediate Trade Page

The trade page takes an object path, and fills a QZtable with the fields we have chosen for that object. The table is then "Qzapified" so that it pops up in the Qzap browser and has an addressable URL.

4.2.2 Blotter Last Five Events

Gokul Menon and Peter Bartek are continuing to develop an electronic blotter, named CST Sales Tool, for the sales and trading teams that would suit their needs (figure 19). Many of the intended users still use paper blotters, and do not have any inclination to change their personal recording system to incorporate an electronic blotter unless it has qualities they find desirable. One of the popular qualities that has been incorporated is the ability to see past trades – whether they were completed or not. Current systems allow you to see past completed trades, but traders find it helpful to keep track of which clients they were not able to close with for any future interactions they may have with them. The more information they have about a client, the better they can tailor their sales to them. From this, Menon determined that it would be helpful to have the last five trades readily available to view.



CSTSales: CST Sales Blotter - 436

qzap://CSTSales?mode=blotter&_db=credit_dev

CSTSales blotter CST Sales Blotter

Quick Entry

SalesId: 977-4107 Client: Enter client Security: Enter ticker cpn/mat or cusip/isin... for eg VZ 6 13 or JPM 5 DEC 13 or DISH MAY 20 or ED1448937

Format : <B/S> <Size> <Level> <InquiryType>(default = Missed) Note: specify Level as (p -price, s -spread and d -discount) and InquiryType as (m -Missed and d -Done)

Quick Entry: Type or a quote e.g. $ 100M 100.5p or S 150k 3.25s or Buy 100k - m

Ticket 1

Price : 0.0 Security Desc: ClientName : InquiryType : MISSED

Size : 0 Security Id: Enter ticker cpn/mat or... ClientID : Enter client Comments :

Hold ☐ Win: Buy SettleDate : 21Nov13 SalesId: 977-4107

E-Dash

Ticket Popup Alert OFF Alert Sound OFF Close Alert After (secs) 15

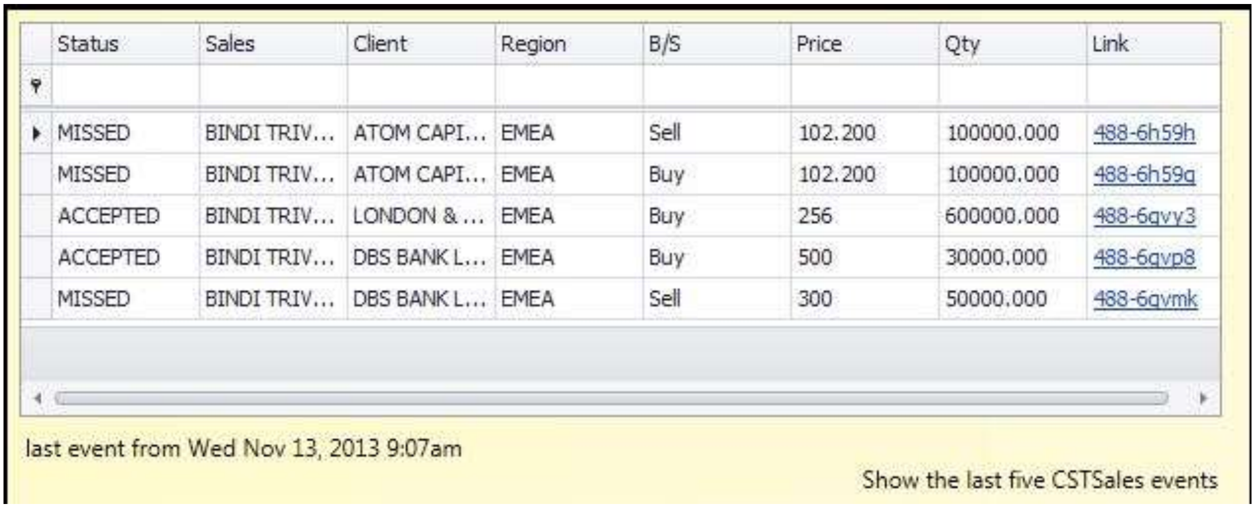
Type	Plat	Prod	Time	Last Clr Act	Last Clt Act	End State	We...	Tkr	Cpn	Mat	Ccy	Legs	Qty '000	Quoted	Px	Px Cov	Dlrs#	Client	Oper
Inq	QZ	CREDIT	15:20:12	MISSED	MISSED	NOT DONE	BUY	JPM		18Nov13	USD	1/1	50,000	Y	107	0	1	BANK OF CHINA INTERNATIONAL (UK) LIM...	DBCH
Ord	BBG	DSUF	11:40:32	DONE	SENT	DONE	BUY	CA...	1.375	29Jan18	USD	1/1	50,000	N	98.522	0	1	BANK OF ISRAEL	RCSS
Ord	BBG	SUBx	10:54:03	DONE	SENT	DONE	SELL	UKTI	0.125	22Mar68	GBP	1/1	11,415	N	108.65	0	1	INSIGHT INVESTMENT FUNDS MANAGEME...	QUBY
Ord	BBG	EUGV	10:49:58	DONE	SENT	DONE	BUY	OBL	0.25	13Apr18	EUR	1/1	15,100	N	99.8575	0	1	MARINER INVESTMENT GROUP, LLC	NISH
Ord	BBG	EUGV	10:27:47	DONE	SENT	DONE	BUY	SPGB	5.5	30Apr21	EUR	1/1	30,000	N	112.88	0	1	BANCO SANTANDER S.A.	MALF
Ord	BBG	EUGV	10:07:40	DONE	SENT	DONE	BUY	FRTR	2.25	25May24	EUR	1/1	70,000	N	98.875	0	1	ELEMENT CAPITAL MANAGEMENT LLC	DRUP
Ord	BBG	SUBx	10:03:03	DONE	SENT	DONE	SELL	UKTI	0.125	22Mar68	GBP	1/1	11,465	N	107.9	0	1	INSIGHT INVESTMENT FUNDS MANAGEME...	QUBY
Ord	BBG	EUGV	09:24:09	DONE	SENT	DONE	SELL	BTF	0	13Feb14	EUR	1/1	50,000	N	0.075	0	1	LCH.CLEARNET LIMITED	SIMO
Ord	BBG	EUGV	09:01:52	DONE	SENT	DONE	SELL	BTF	0	13Feb14	EUR	1/1	120,000	N	0.075	0	1	CM-CIC ASSET MANAGEMENT	MALF
Ord	BBG	DSUF	07:11:47	DONE	SENT	DONE	BUY	KFW	0.5	19Apr16	USD	1/1	50,000	N	100.13	0	1	PEOPLES BANK OF CHINA	NORA

The use of information derived from this reporting application is subject to [illegible] Guidance for EMEA FICC and Equity Sales and Trading regarding the preparation of Sales and Trading Communications, and [illegible] Communications EMEA Compliance Guidance. If you have any doubts contact [illegible] Compliance

Figure 19. CST Sales Tool Electronic Blotter

When an event is created in the blotter, it is saved as either an event of a new Deal, or an event of an existing Deal (see Chapter 2.3.10). Every CST Sales Tool Event begins with the prefix CSTSales, so when retrieving the data, we filtered the events inside Deal Events by those beginning with the same prefix, to look at only the events pertaining to the electronic blotter. This way, the events displayed are the last five trade events submitted.

Like with the Sandra Database Last 5 Trades (see Chapter 4.2.2), the position, instrument type, price, and quantity were the main pieces of information to display. Because the two gadgets are so similar in purpose, the initial design remained the same (figure 16). Links to each specific event is also included, like the previous gadget. Through the development process, we later added in new fields that were thought to be pertinent to a clear overview of an event, such as the Client, Region, and Status of the trade. Our final design and gadget looks very similar to that of the Sandra Database Last Five Trades (figure 20).



	Status	Sales	Client	Region	B/S	Price	Qty	Link
▸	MISSED	BINDI TRIV...	ATOM CAPI...	EMEA	Sell	102.200	100000.000	488-6h59h
	MISSED	BINDI TRIV...	ATOM CAPI...	EMEA	Buy	102.200	100000.000	488-6h59g
	ACCEPTED	BINDI TRIV...	LONDON & ...	EMEA	Buy	256	600000.000	488-6gvy3
	ACCEPTED	BINDI TRIV...	DBS BANK L...	EMEA	Buy	500	30000.000	488-6gvp8
	MISSED	BINDI TRIV...	DBS BANK L...	EMEA	Sell	300	50000.000	488-6gvmk

last event from Wed Nov 13, 2013 9:07am

Show the last five CSTSales events

Figure 20. Last Five Events from CST Sales Tool

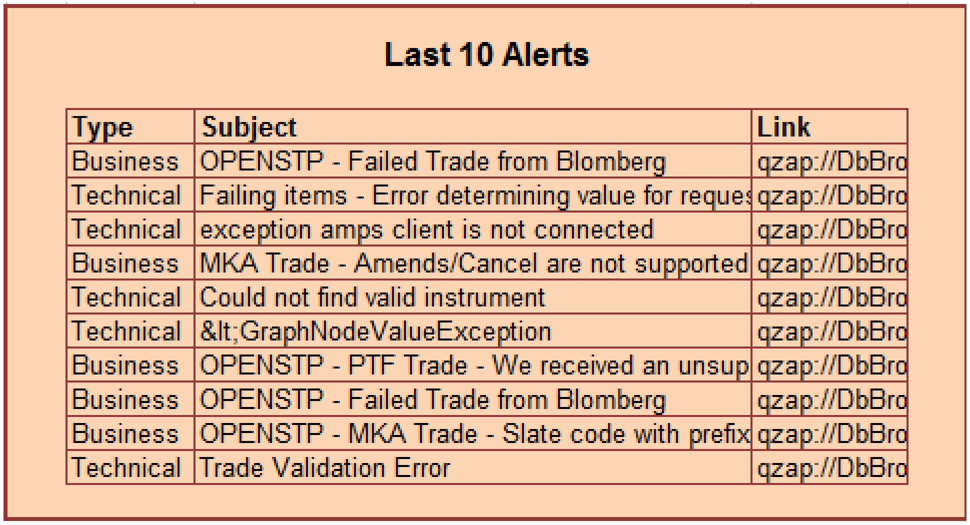
4.2.3 Last Ten Alerts

As part of an initiative to increase the number of users relying on Quartz, Bartek believes it would be helpful to have a gadget that would display recent alerts. Alerts are stored inside the Sandra database under Support and Alerts. From there, the object contains the subject of the alert (usually an error message), what application it occurred in, and several other details that are helpful for resolving issues. The alert first goes into a queue, and is later moved to Processed or Ignored queues depending on how the alert was recalculated. If an application has an error, there is code available to configure the application to send an alert into the Sandra database's Alerts Queue to be read.

In some of the support teams for Bank of America's applications, errors that occur have to be found and the error log sent to another user (usually via email) if they want to share it. Bartek believes this system could be improved by configuring an application to send an alert to Sandra, and then having a gadget that would display the last ten alerts for a user to easily see. This way, a user has real-time alerts updates, and links to the Sandra database alert. Ideally, Bartek hopes that soon users will go directly to Sandra for Alert information rather than going through error logs.

We met with Mike Yang, a Technology Programming Associate at Bank of America, and asked if he would be willing to send alerts from some of his applications into Sandra, to work with our gadget. He was interested in the idea of being able to easily see the last ten alerts and have links to the alert information readily at hand, rather than finding out later when an error log was emailed to him.

When starting the development of the gadget, we first designed something basic that would reveal the details of an alert, what type of alert it was, and a link to more information (figure 21).

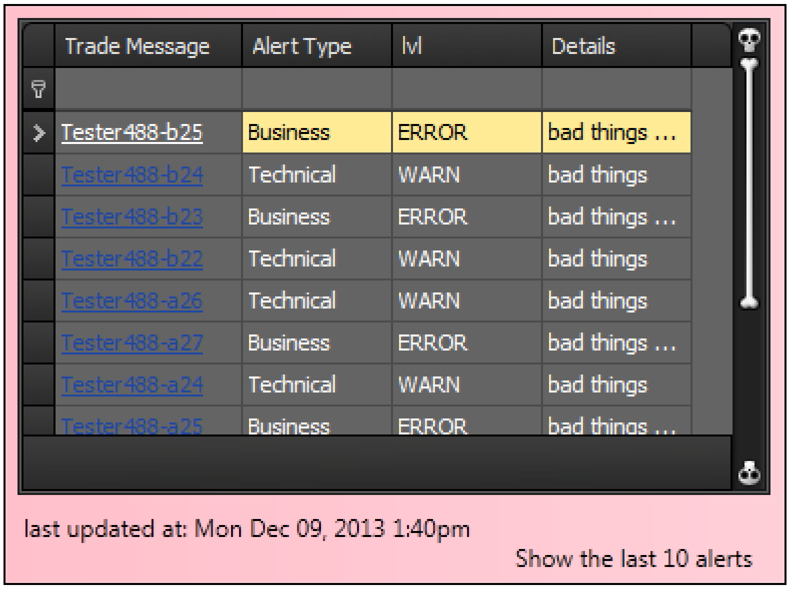


Last 10 Alerts

Type	Subject	Link
Business	OPENSTP - Failed Trade from Blomberg	qzap://DbBro
Technical	Failing items - Error determining value for reques	qzap://DbBro
Technical	exception amps client is not connected	qzap://DbBro
Business	MKA Trade - Amends/Cancel are not supported	qzap://DbBro
Technical	Could not find valid instrument	qzap://DbBro
Technical	<GraphNodeValueException	qzap://DbBro
Business	OPENSTP - PTF Trade - We received an unsup	qzap://DbBro
Business	OPENSTP - Failed Trade from Blomberg	qzap://DbBro
Business	OPENSTP - MKA Trade - Slate code with prefix	qzap://DbBro
Technical	Trade Validation Error	qzap://DbBro

Figure 21. Concept Art for Last Ten Alerts

Code for sending errors into Sandra alerts has already been developed, and only needed to be added into the production of Yang's applications. While waiting for him to work on this, we created a gadget that would take Alerts from one of the already available directories, using the same methods as used for the Sandra Last Five Trades gadget. The gadget displays alerts from all paths inside the Alerts Queue, meaning items from Processed, Failed and Signed Off queues inside Alerts are displayed in the table. We added in more fields we found relevant during the development, the most important being the level of the alert. The final prototype of the gadget contained real-time information about the last ten alerts, including the subject, alert type, level of alert, details, and a link to the alert inside the Sandra Database (figure 22).



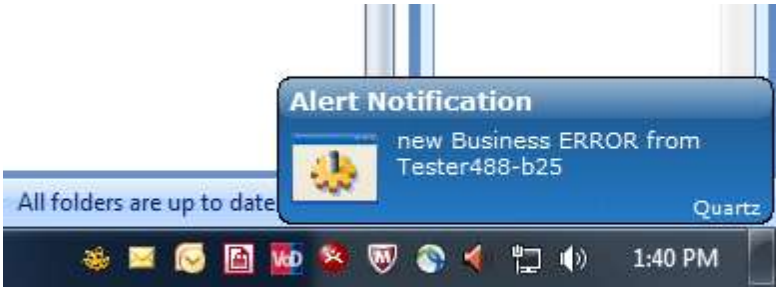
Trade Message	Alert Type	lvl	Details
Tester488-b25	Business	ERROR	bad things ...
Tester488-b24	Technical	WARN	bad things
Tester488-b23	Business	ERROR	bad things ...
Tester488-b22	Technical	WARN	bad things
Tester488-a26	Technical	WARN	bad things
Tester488-a27	Business	ERROR	bad things ...
Tester488-a24	Technical	WARN	bad things
Tester488-a25	Business	ERROR	bad things ...

last updated at: Mon Dec 09, 2013 1:40pm

Show the last 10 alerts

Figure 22. Gadget for Last 10 Alerts

After a Design Review, it was decided that the Alerts Gadget could be improved by incorporating a notification that would pop up in the bottom right hand corner of the screen when a new Alert was added to the Queue (figure 23). This way, if users wanted to deal with alerts right away, rather than when they happened to look at the gadget, they could be quickly informed.



Alert Notification

new Business ERROR from Tester488-b25

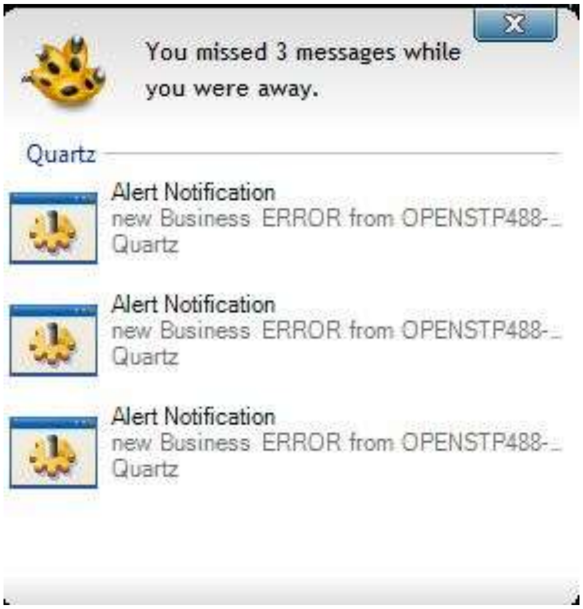
Quartz

All folders are up to date

1:40 PM

Figure 23. Notification of new Alert in Last 10 Alerts Gadget

To create the popup notifications with new Alerts, we used Growl, a third party notification system used by Bank of America developers. If multiple Alerts occur while the Dashboard is not running, all Alerts will display when the Dashboard is next launched (figure 24).



You missed 3 messages while you were away.

Quartz

Alert Notification
new Business ERROR from OPENSTP488-...
Quartz

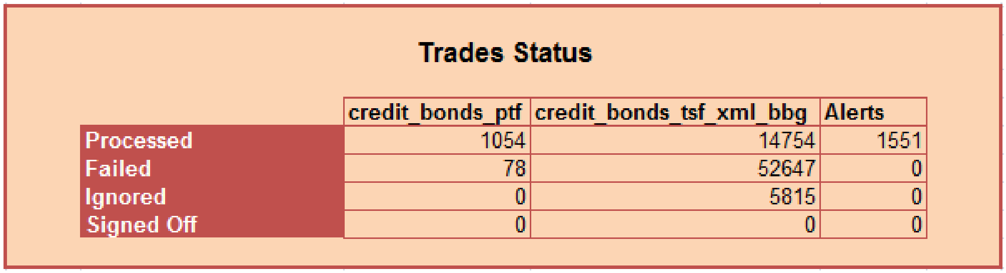
Alert Notification
new Business ERROR from OPENSTP488-...
Quartz

Alert Notification
new Business ERROR from OPENSTP488-...
Quartz

Figure 24. Alerts notifications when Launching Dashboard after new Alerts occur

4.2.4 Trades/Alerts Status

Viewing the last entered trades, or latest alerts gives a brief look into the details of the system, but Bartek explained that most people do not need to see details, but a broad spectrum of information to show that the system is working well. To create a high-level view of the system, we needed something that looked at all the information in a more expansive view, and simplified it in such a way to be visually appeasing. In this case, we decided a table of the Processed, Failed, Ignored, and Signed Off trades and Alerts would allow users to see just how much is being done, without going too far into details (figure 25). As an added visual aid, we decided to include a bar graph of the same information (figure 26), but as a separate gadget. Both gadgets display the same information, in two different mediums.



Trades Status

	credit_bonds_ptf	credit_bonds_tsf_xml_bbg	Alerts
Processed	1054	14754	1551
Failed	78	52647	0
Ignored	0	5815	0
Signed Off	0	0	0

Figure 25. Concept Art for Trades Status table

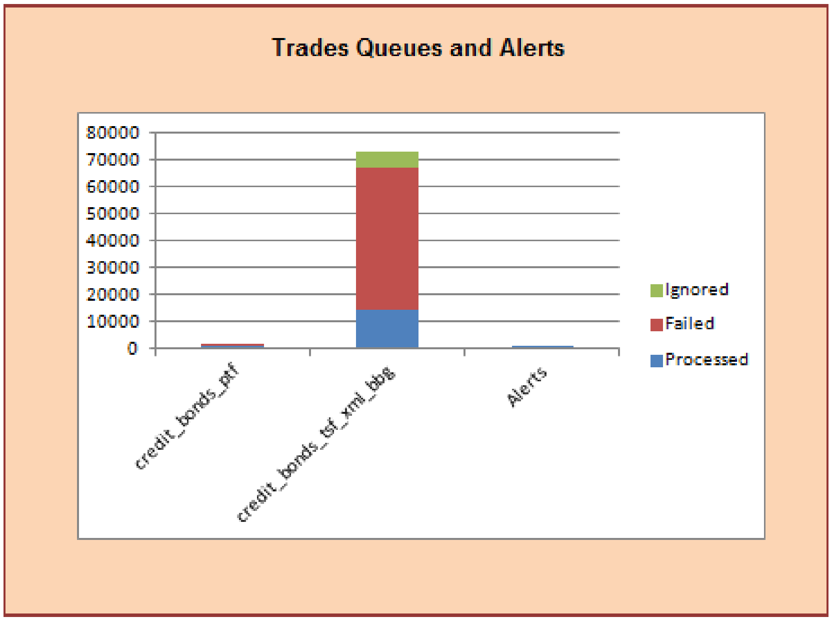
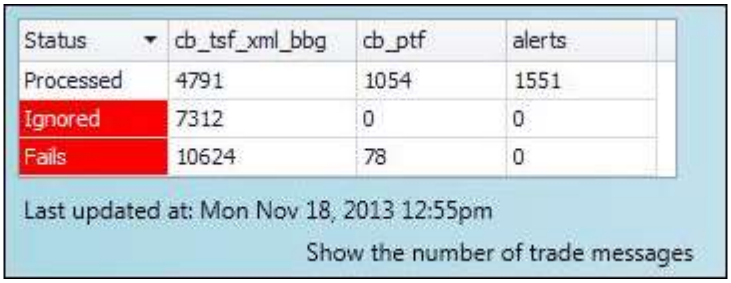


Figure 26. Concept Art for Trades Status Graph

The information displayed in the graph and table is all information we have accessed for gadgets in the previous sections. The Queues of trades and Alerts are all located inside the Sandra Database, under Queues and Support. The main obstacle of this gadget was to bring all this information, from a few different subdirectories, into one location for easy viewing. Once the information was in a table, the graph was more easily created. Examples of created graphs on gadgets were available through demo UIs, and by finding the source code for Quartz Dashboard gadgets with graphing capabilities. The initial gadgets looked very similar to our concept art (figures 27 and 28).



Status ▾	cb_tsf_xml_bbg	cb_ptf	alerts
Processed	4791	1054	1551
Ignored	7312	0	0
Fails	10624	78	0

Last updated at: Mon Nov 18, 2013 12:55pm

Show the number of trade messages

Figure 27. Initial Trades Status Table Gadget

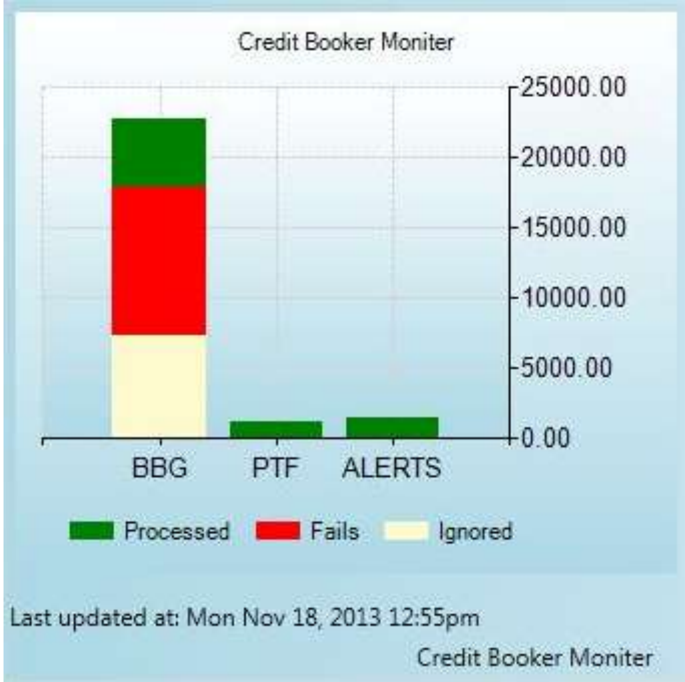
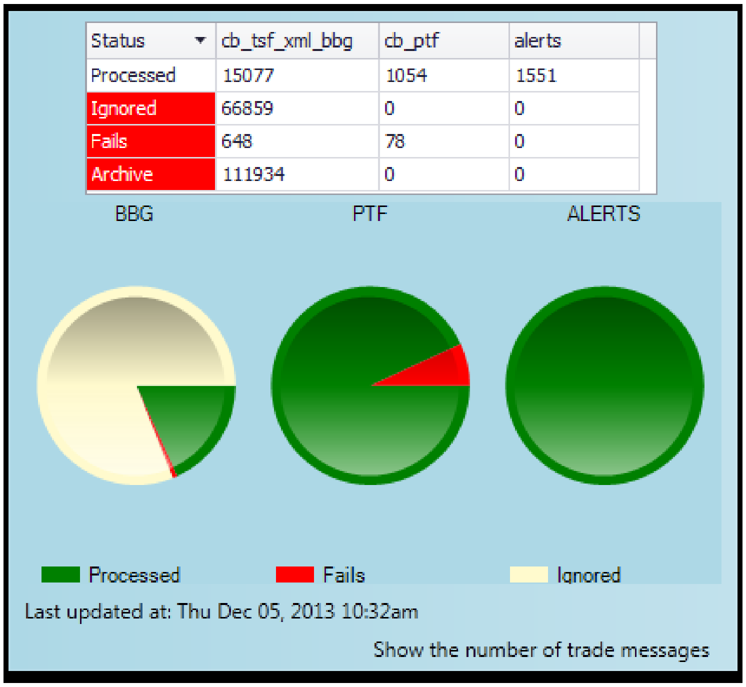


Figure 28. Initial Trades Status Graph Gadget

Because the cb_tsf_xml_bbg (BBG) has so much more data inside of it, the other Queues are dwarfed by comparison. We decided that adding charts under each Queue for their information would provide more readability, without detracting from the fact that there are quite a bit more transactions in the BBG Queue. Bartek also created an Archived Section in the Queues to throw old data, so that it would not get stuck inside the directories displayed in the graphs. Overall, the final gadgets contained a table of the Queues and Alerts items, pie charts displaying each Queue, and a bar chart containing the information from all three – the two Queues and Alerts (figures 29 and 30).

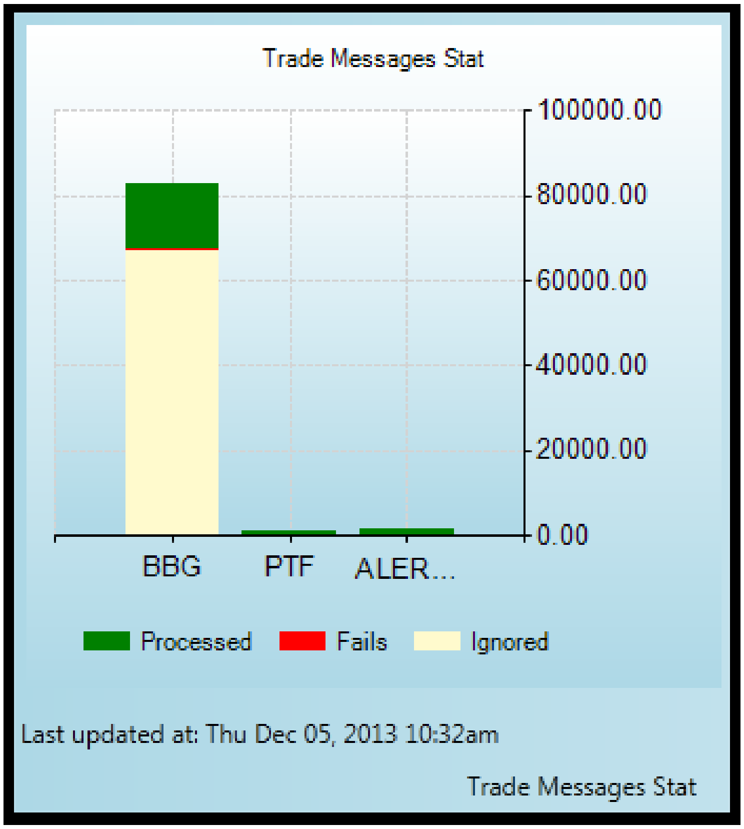


Status	cb_tsf_xml_bbg	cb_ptf	alerts
Processed	15077	1054	1551
Ignored	66859	0	0
Fails	648	78	0
Archive	111934	0	0

Last updated at: Thu Dec 05, 2013 10:32am

Show the number of trade messages

Figure 29. Trades Status Table and Chart Gadget



Last updated at: Thu Dec 05, 2013 10:32am

Trade Messages Stat

Figure 30. Trades Status Bar Chart Gadget.

4.2.5 Blotter User Statistics

With the recent development of the CST Sales Tool described in Chapter 4.2.2, it is important to keep track of how much the application is being used. The blotter is currently still in beta, but as it develops, Menon said it would be helpful to see how many users are using this application. If very few users are using it, the application may need to be changed or scrapped, and if a lot of users are using it, it may need maintenance or monitoring.

Quartz already keeps track of the number of unique users and launches of different applications, but both Menon and Bartek said it would be more helpful to keep track of the number of users entering information in the blotter, as opposed to just launching it. Additionally, users do not necessarily have to launch the application from its icon, as it has a unique URL that can be navigated to through any open Quartz browser. We met with Fred Hsu, a member of the Core Team that developed Quartz and Qzap, to see how we could get statistics on accessing a specific URL (see Meetings Log, Appendix A). Hsu explained that users' most recent one hundred browsing preferences from Qzap is saved to allow for easily accessing of recent and most visited pages. When a URL has not been visited enough, it drops out of the bottom of the one hundred saved to make room for one that has been. If this information is stored, it can be used to create user statistics for connections to the CST Sales Tool URL.

This gadget has a much farther reaching scope than just looking at the number of users accessing the electronic blotter. Once you have a method of determining the users navigating to CST Sales Tool, you could branch to pages inside that one application to get an idea of what features are being used. Additionally, this same approach can be brought to any other Qzap application to allow for more accurate application usage data.

To display the usage of the CST Sales Tool, we decided we should use a tree map (figure 31). This would allow for users to see the use of the blotter first among regions, and then different desks of trading, to the users that are utilizing the program. Each section of the tree map is based on how many users are in that region or desk; larger sections have more traders than smaller sections. The color indicates the usage of the application; darker colored sections use the application more than their lighter counterparts. The tree maps used by the current Dashboard (see Chapter 2.2.11) displays the users using Quartz from the previous day, and when the next day's statistics come through, the old information is discarded. Demo UI for creating a tree map was already available for us to use.

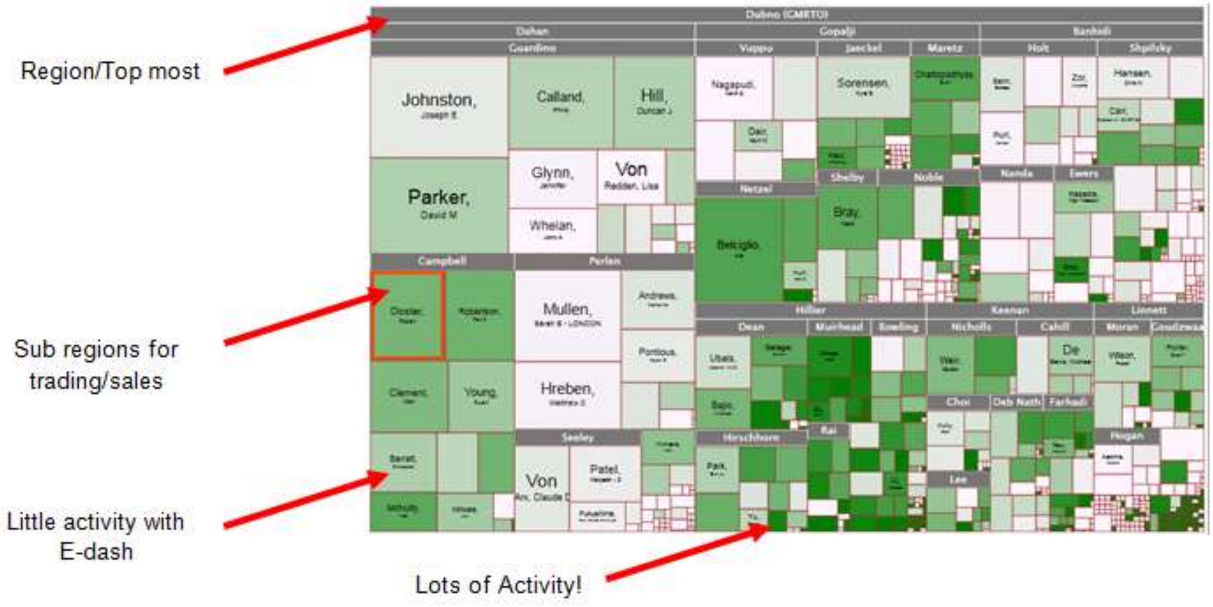


Figure 31. Concept Art for User Statistics Gadget

In order to display the activity in the application over time we first needed to create a Bob Job (see Chapter 2.3.7) that will capture the statistics (users connecting to the CST Sales Tool URL) for the day and store them for our use. Once we have this information stored, we can add all users' connections, unique users, and any other statistics we find important. The most recent day's information could be used for the tree map, and the backlog of user statistics could be used to create graphs over time for more visuals on user statistics (figure 32).

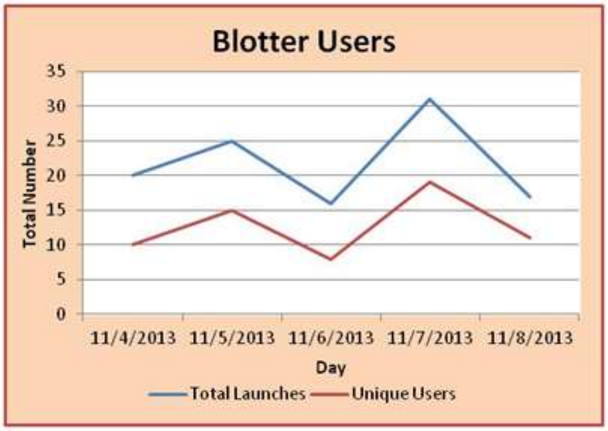
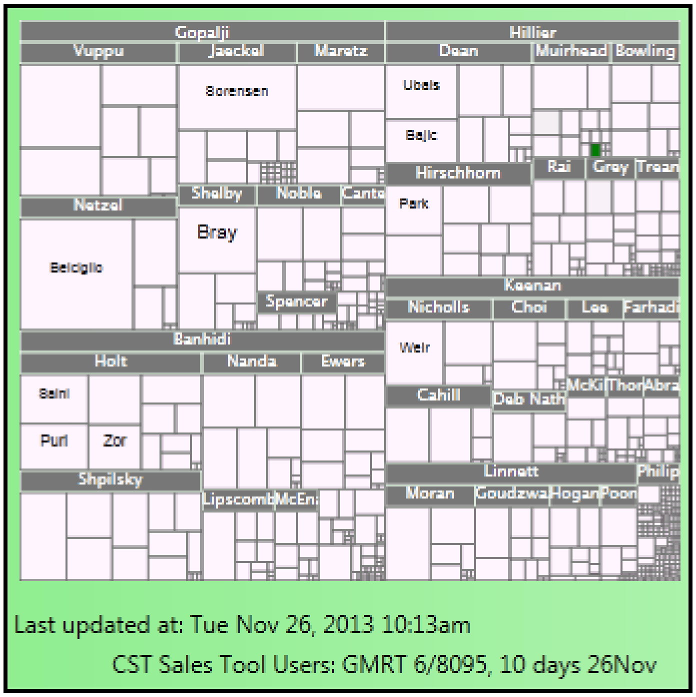


Figure 32. Concept Art for User Statistics Graph over time

What we have found with the User Statistics gadgets, is that there are not a lot of people currently using the blotter. That is to be expected, as the blotter is only in production in London, and is still being developed for other offices. Because of this, the tree map displayed for this gadget is considerably blank (figure 33). The only section colored is Bartek's department, where he, Menon, and our project group have been using the application.

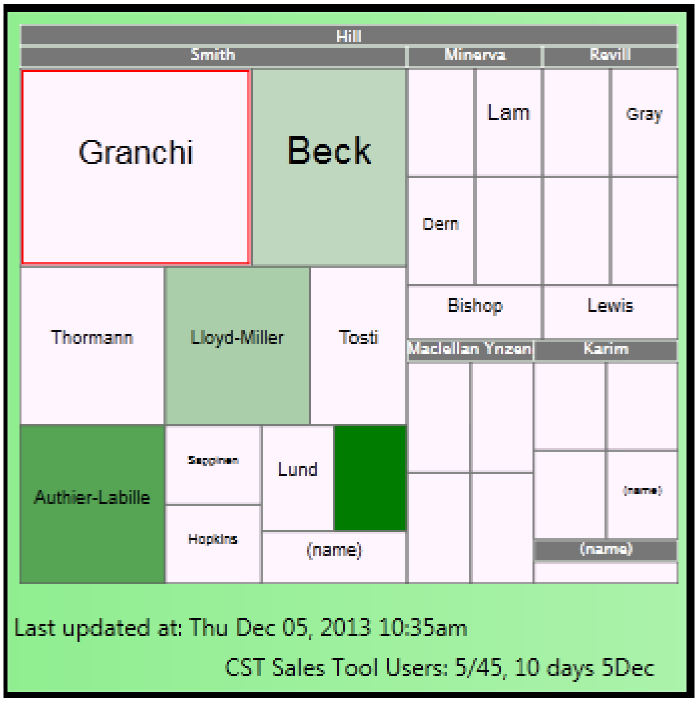


Last updated at: Tue Nov 26, 2013 10:13am

CST Sales Tool Users: GMRT 6/8095, 10 days 26Nov

Figure 33. Initial User Statistics Gadget

After a design review, it was decided that we did not need to display all of the departments, but only the ones relevant to business as we only really cared about how many people are utilizing the electronic blotter in London offices, where it is currently in production. Narrowing down the departments displayed allows for a better overall picture of the adaptation of the electronic blotter. Rather than use the default hierarchy from the demo UI, we created our own starting at the top with Graham Hill, the Head of the Security Products Sales, and the topmost of the business hierarchy inside Bank of America. In Quartz's Qzap browser we can see all the people he directs, and each person that they direct under him. This information was used for confirmation, and the actual data pulled for the hierarchy came from the Sandra employee profiles. When we had a working tree map using this hierarchy, we checked it against the Qzap profiles to be sure of accuracy. The finished tree map is a lot clearer to see, and displays only the business departments that we want to monitor for use of the electronic blotter (figure 34).

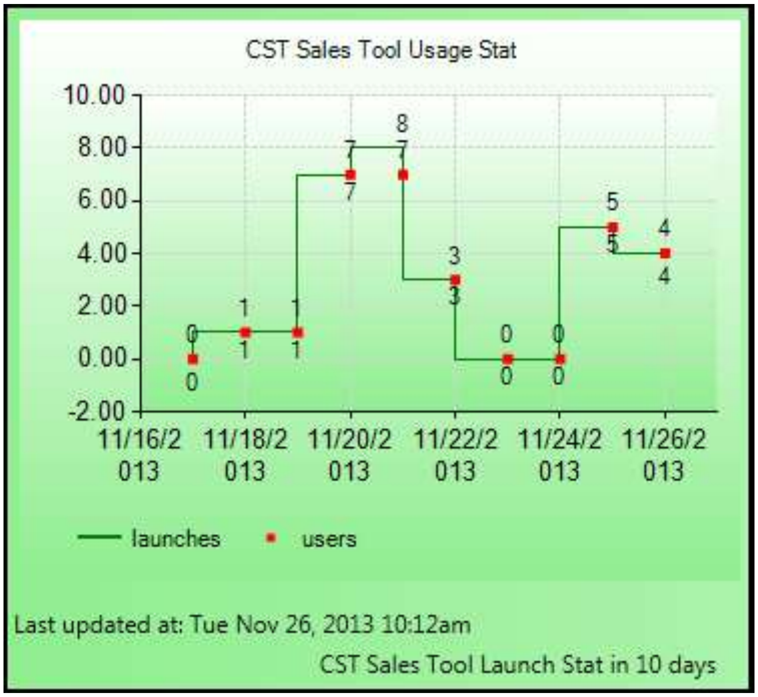


Last updated at: Thu Dec 05, 2013 10:35am

CST Sales Tool Users: 5/45, 10 days 5Dec

Figure 34. Users Statistics Gadget

The gadget to display the User Statistics over time also has very low numbers, but is much more interesting to look at (figure 35). Right now, the number of users and launches are low (less than ten per day), but when usage of the application picks up, this graph will be more helpful to understanding how many users are launching the application. The green line on the graph indicates the number of launches per day, and the red dot indicates users. When the electronic blotter application eventually goes into production in New York offices, this will be helpful to see how many people are using it, and how often they launch it.



Last updated at: Tue Nov 26, 2013 10:12am

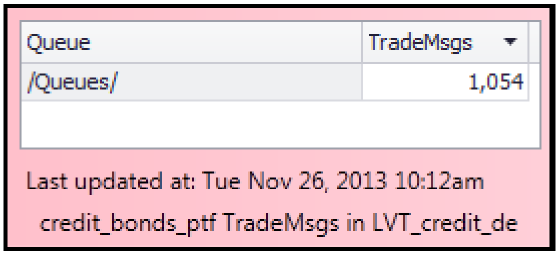
CST Sales Tool Launch Stat in 10 days

Figure 35. User Statistics Graph over Time Gadget

4.2.6 Queues

Before the start of our work on this project, Bartek created two gadgets for proof of concept that a Credit Dashboard could be made. He created two corresponding gadgets to display the Queues of Alerts, so that users could see the number of Alerts at any given point, which we improved upon with the Trades/Alerts Status Gadgets in Chapter 2.2.5.

In the first gadget, there is a table for the number of Processed transactions (or Trade Messages) in the LVT_credit_dev database Queue (figure 36). Clicking on the tile of the gadget launches Qzap and brings the user to the Queues directory in Sandra.



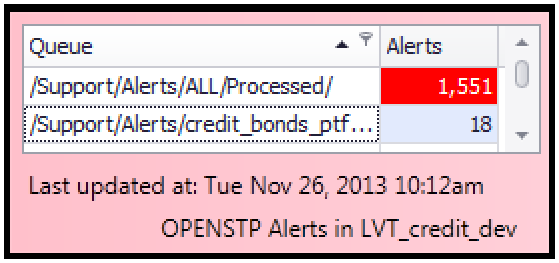
Queue	TradeMsgs
/Queues/	1,054

Last updated at: Tue Nov 26, 2013 10:12am

credit_bonds_ptf TradeMsgs in LVT_credit_de

Figure 36. Queues Gadget of Processed Trade Messages

The second gadget displays the number of Processed Alerts for three different Queues directories (figure 37). Clicking the tile of the gadget will launch Qzap and bring the user into the Alerts directory.



Queue	Alerts
/Support/Alerts/ALL/Processed/	1,551
/Support/Alerts/credit_bonds_ptf…	18

Last updated at: Tue Nov 26, 2013 10:12am

OPENSTP Alerts in LVT_credit_dev

Figure 37. Queues Gadget for Processed Alerts

The information displayed in these gadgets was later incorporated into the Trades/Alerts Status gadgets in Chapter 4.2.4, and removed from the Credit Dashboard.

Chapter 5: Conclusions and Recommendations

Over the seven weeks this project took place, we have created a Credit Dashboard that acts as a central location for seven gadgets designed to aid in monitoring system health. Though Bank of America has a recently developed yet sophisticated system for market data, risk measures, and approval workflow, named Quartz, it lacks a thorough monitoring of that system. The idea behind this project is system transparency; users need to have the ability to see what is happening within different aspects of Quartz in a manner that is both easy and convenient.

The Credit Dashboard fulfills this need through the display of the gadgets. Each gadget monitors a different application or queue, aimed to support different teams in the Global Markets Research Technology (GMRT) division within Bank of America. The benefits of the Credit Dashboard are twofold; the gadgets provide easily-read system information to help teams monitor their applications, and the Dashboards created by the Quartz Core Team are exposed as a viable method for monitoring system processes.

We recommend that Peter Bartek, our sponsor liaison and manager of this project, continue to contact different GMRT teams to talk about the Credit Dashboard and ask if they would have any interest in utilizing the Credit Dashboard. We have seven gadgets that can be demonstrated to potential users, and even adapted to their needs if they were interested in using it. The Alerts gadget specifically may interest potential users, as it can be easily customized for a user to receive alerts from an application they have developed.

When talking with different teams to gather interest in the dashboard, we recommend that ideas for gadgets might need to be gathered. Appendix B contains an interview outline that can be used to start a discussion for possible future gadgets. If at all possible, gadgets should be developed for teams, or instructions for teams to create their own if they are interested in using the Credit Dashboard.

For future gadgets, easy customization should be kept in mind during development. Current gadgets can be edited to a user’s specific application or queue through revision of the gadget code, but a UI component to do this on the gadget would make it much easier for users to adopt the Credit Dashboard into their work.

If GMRT teams do take interest in using the Credit Dashboard and having more customizable gadgets created, we would recommend that the Dashboard itself also become customizable for users. Our group's vision of the final Credit Dashboard would be a blank Dashboard with an option to add gadgets (figure 38). When adding gadgets to the Dashboard, users could customize them to display the applications and queues that they want to see, and not just the ones we have developed for different teams. Users should be able to organize and group the gadgets however is best for them, and if they close the Dashboard it should retain these preferences for its next launch. Gadgets could be kept in a directory to be added, or development of a "Gadget Store" could occur that would allow for browsing and adding gadgets in the same easy manner as App Stores.

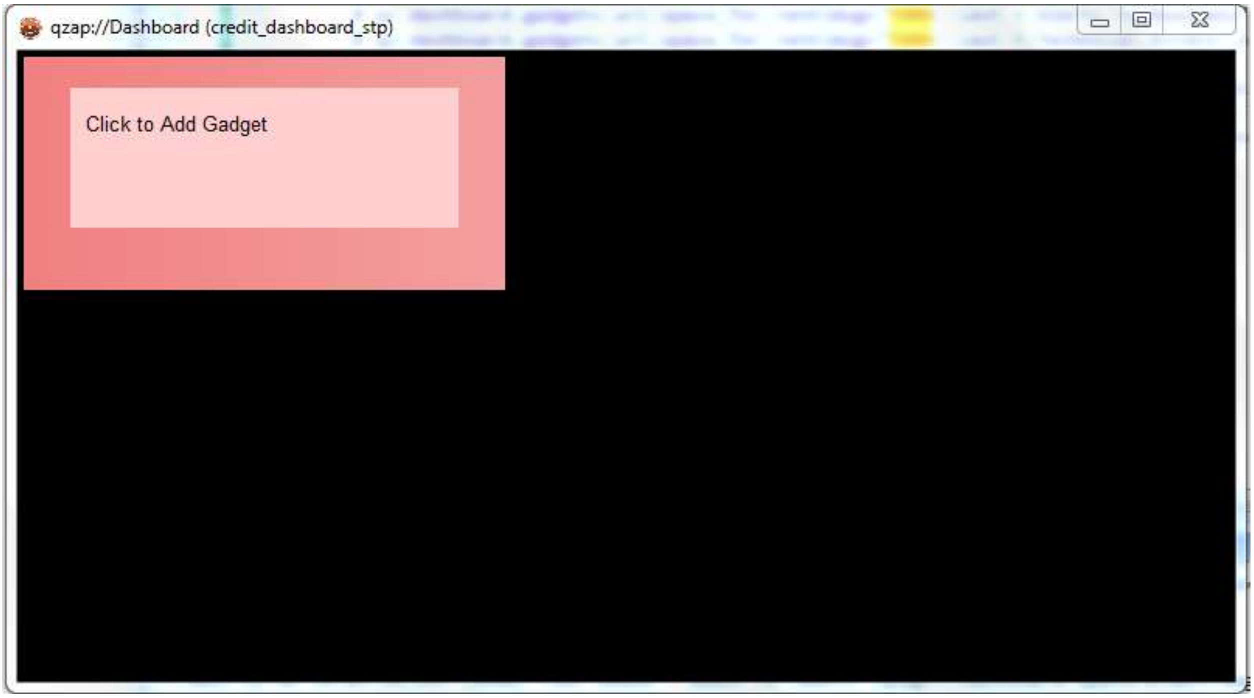


Figure 38. Concept for Blank Customizable Dashboard

A customizable Dashboard would help users monitor information pertinent to their interests. A large selection of gadgets and intuitive easy-to-use interface will help "sell" the Dashboard to potential users. As more users work with the Dashboard, the more gadgets can be created and resources devoted to improving the Dashboard. The final product would be a customizable monitoring system for users to keep track of applications, queues, alerts, and anything else that a gadget can be designed for.

References

America, B. o. Our History and Heritage.

Bank of America: A History. (August 24th, 2011). *The Telegraph*.

Growl Team, Growl (2004-2011),, Growl.info

GT&O Divisions. Retrieved 11/1/2013, 2013. Bank of America Flagscape

James, B. R., & James, M. (1954). The Story of Bank of America: Biography of a Bank. New York, New York: Bank of America N.T. and S.A.

Johnston, M. (1990). Roller Coaster: The Bank of America and the Future of American Banking. New York, New York: Ticknor & Fields.

Merton, R. C. (1974). On the pricing of corporate debt: The risk structure of interest rates*. *The Journal of Finance, 29*(2), 449-470.

PBS. A.P. Giannini. *Who Made America?*. PBS.org

Quartz Core Team, Bank of America. AMPS - Real-time Publish/Subscribe. *Quartz Project Documentation*.

Quartz Core Team, Bank of America. Bob - The Quartz Batch Scheduler. *Quartz Project Documentation*.

Quartz Core Team, Bank of America. Dag - The Quartz Dependency Graph. *Quartz Project Documentation*.

Quartz Core Team, Bank of America. Hugs - Quartz Grid Scheduler. *Quartz Project Documentation*.

Quartz Core Team, Bank of America. Inform: Real-Time Sandra Notifications. *Quartz Project Documentation*.

Quartz Core Team, Bank of America. Quack - Quartz Access Control. *Quartz Project Documentation*.

Quartz Core Team, Bank of America. Qzap - Quartz Addressable Protocol. *Quartz Project Documentation*.

Quartz Core Team, Bank of America. QzDesktop - Launches Quartz Applications on Windows. *Quartz Project Documentation*.

Quartz Core Team, Bank of America. QzDev - User Manual. *Quartz Project Documentation*.

Quartz Core Team, Bank of America. Sandra - The Sandra Object Database. *Quartz Project Documentation*.

Quartz Core Team, Bank of America. The Quartz Deal Model. *Quartz Project Documentation*.

Quartz Core Team, Bank of America. UI - The Quartz UI Framework. *Quartz Project Documentation*.

Stulz, R. M. (2009). *Credit default swaps and the credit crisis*: National Bureau of Economic Research.

Telegraph, T. (August 24th, 2011). Bank of America: a history.

Weistroffer, C., Speyer, B., & Walter, N. (2009). Credit default swaps. *Deutsche bank research*.

Appendix A: Meetings Log

Project Meetings Log

Date	Attendees	Formal/Informal	Main Topics
10/28/2013	Peter Bartek Nicolas Dupuis Alexandra Mutty Fangwen Yuan	Informal	- Introductions / Tour of BofA floors - Project Discussion - Explanation of Quartz
10/30/2013	Peter Bartek Nicolas Dupuis Alexandra Mutty Fangwen Yuan	Informal	- Create a presentation on Credit Dashboard (Pros/Cons) to "sell" to teams - Begin development of gadget to view last 5 trades
10/30/2013	Peter Bartek Nicolas Dupuis Gokul Menon Sales/Trade staff	Formal	- Peter and Gokul ask Sales/Trading staff to Demo Quartz electronic blotter, take feedback - Program ran really slowly, would need to be faster
10/31/2013	Peter Bartek Gokul Menon Alexandra Mutty Sales/Trade staff	Formal	- Peter and Gokul ask Sales/Trading staff to Demo Quartz electronic blotter, take feedback - Traders want to see trades that haven't been completed (client backed out, or they did
11/04/2013	Peter Bartek Nicolas Dupuis Alexandra Mutty Fangwen Yuan	Informal	- Gadget Development update - Set up meeting with Gokul Menon to discuss possible gadget ideas - Make a Code project (tool to keep track of project progress)
11/04/2013	Nicolas Dupuis Gokul Menon Alexandra Mutty Fangwen Yuan	Formal	- Presentation of Credit Dashboard idea - Gokul's Role – Peter Bartek gets project, breaks down into parts, Gokul completes part(s) - Gadget ideas: Last 5 trades through Blotter, User stats
11/07/2103	Peter Bartek Nicolas Dupuis Alexandra Mutty Fangwen Yuan	Informal	- Gadget ideas/specifications - Decide on method for drilling down in gadget
11/08/2013	Peter Bartek Nicolas Dupuis Alexandra Mutty Fangwen Yuan	Informal	- Begin development for top CST Sales Users - Possible idea for information: use tree map - Meet with Fred Hsu to find out about storing user URL info
11/08/2013	Nicolas Dupuis Gokul Menon Alexandra Mutty Fangwen Yuan	Informal	- Where CST Sales Tool trades are stored - How to access app usage information - User statistics: possibly looking at how many users "touch" a trade

11/08/2013	Peter Bartek Nicolas Dupuis Fred Hsu Alexandra Mutty Fangwen Yuan	Informal	- Demonstrated Sandra Last 5 Trades gadget - Explained Credit Dashboard concept - User statistics idea – need their URL history - Where users URL history is stored
11/13/2013	Peter Bartek Nicolas Dupuis Alexandra Mutty Fangwen Yuan	Informal	- Concept for User Stats gadget - Where to find user preferences for User Stats. Gadget - Will need to create a Bob Job to store stats
11/15/2013	Peter Bartek Nicolas Dupuis Alexandra Mutty Fangwen Yuan	Informal	- Demo with QzDev - Worked through Demo bar chart code
11/19/2013	Nicolas Dupuis Alexandra Mutty Mike Yang Fangwen Yuan	Formal	- Specs for alerts and fails gadget - Possible email option? - What information to display
11/21/2013	Peter Bartek Nicolas Dupuis Alexandra Mutty Mike Yang Fangwen Yuan	Informal	- Clarification on Alerts and Fails gadget - Where to find information - What the gadget should look like
12/02/2013	Peter Bartek Nicolas Dupuis Alexandra Mutty Fangwen Yuan	Informal	- Design Review - Intermediate pages for links - New hierarchy for treemap (limit to business)
12/02/2013	Peter Bartek Sam Beck Nicolas Dupuis Alexandra Mutty Fangwen Yuan	Informal	- Into to Credit Dashboard - Interest in Alerts - FWD tutorial on setting up alerts in applications

Appendix B: Interview Outline

We first explain our project and the purpose of the dashboard, followed by a quick demonstration of the Dashboard with the gadgets that were completed by the time of the meeting.

- What is your role in the company?
- What work do you do on a daily basis?
- What projects are you currently working on?
- What Quartz Applications do you use frequently?
- What information do you look at daily within the Quartz tools?
- What information would you like to have readily available?
- Would you use a dashboard?
- What tools would you like to see?

Appendix C: A Brief History of Bank of America

From its earliest beginnings, Bank of America has been interested in a financial system that works for the consumers. In 1904, Amadeo Giovanni formed the Bank of Italy in San Francisco, California as a people's bank. At the time, most American banks discriminated heavily against the poor and new immigrants to the country, and the Bank of Italy filled the banking needs of the lower class.

Throughout the next 80 years, Bank of Italy, later renamed Bank of America and Italy, and finally Bank America, steadily grew to become a trusted and widespread Californian bank. Following the relaxation of federal banking regulations that allowed banks to expand across State borders, Bank America acquired Seafirst Corporation of Seattle, Washington, and began its expansion across the United States. Acquisitions of Security Pacific National Bank in 1992 and Continental Illinois National Bank and Trust Co. in 1994 served to help spread Bank America across the continent. Plagued by financial troubles in the late 1990's, Bank America was later merged with Nations Bank to become the Bank of America Corporation.

In the 1950's, Bank America had led the revolution behind credit and banking cards with BankAmericard (later named VISA), and in 2005 Bank of America became a credit card giant following the acquisition of MBNA, the largest independent credit card issuer. Shortly after, Bank of America acquired the United States Trust Company and Countrywide Financial in 2006 and 2008, respectively. Just days away from bankruptcy, Merrill Lynch merged with Bank of America in 2009 under pressure from federal government. Though the merge initially meant that Bank of America inherited the severe losses suffered by Merrill Lynch leading up to their bankruptcy, it also served to make Bank of America the largest wealth management company in the world, and later went on to be most profitable in the investment banking market.

Bank of America began in consumer banking, and even today consumer banking remains the largest division of the corporation. With over 5,800 retail branches and 18,000 ATMs located across the United States, Bank of America is one of the largest banks in the World. As of 2013, Relbanks statistics places Bank of America as the 11th largest financial institution, with a combined total of $2.21 trillion dollars in assets. In the United States, it is the second largest bank, behind J.P Morgan Chase. Although globalization means that most companies, including Bank of America, have expanded across the World, 90% of Bank of America's revenue is generated in the United States domestic markets, and this remains their core strategy as a banking institution.