Saiku Documentation

Release 3.x

Tom Barber

Dec 14, 2018

Contents

1 **Background** **1**
- 1.1 OLAP Analysis . . . 1
- 1.2 Architecture . . . 2
- 1.3 Open Source . . . 2

2 **Administration Guide** **3**
- 2.1 Installation Guide . . . 3
- 2.2 Administrators Console . . . 5
- 2.3 Logging . . . 11
- 2.4 Schema . . . 11
- 2.5 Saiku Settings . . . 17
- 2.6 Embedding . . . 17
- 2.7 Backups and Restoration . . . 18
- 2.8 Migration From Saiku 2.x . . . 19
- 2.9 Webdav Access . . . 19
- 2.10 Safe Mode . . . 20
- 2.11 Localization . . . 20
- 2.12 Default Reports . . . 21
- 2.13 Migrating From Pentaho Analyzer . . . 22

3 **User Guide** **23**
- 3.1 Creating A Query . . . 23
- 3.2 Filtering . . . 23
- 3.3 Charts And Graphs . . . 24
- 3.4 MDX Mode . . . 25
- 3.5 Calculated Members . . . 25
- 3.6 Parameter Support . . . 25
- 3.7 Drilling . . . 26
- 3.8 Totals And Subtotals . . . 26
- 3.9 Dashboards . . . 26
- 3.10 Saiku Tips And Tricks . . . 29
- 3.11 How To Articles . . . 29

4 **Project Information** **31**
- 4.1 Saiku Project Information . . . 31

5 **Development** **33**

5.1 Development 33
5.2 Server Documentation 34

6 Getting Help 35
6.1 IRC 35
6.2 Commercial Support 35

7 Licensing 37

CHAPTER 1

Background

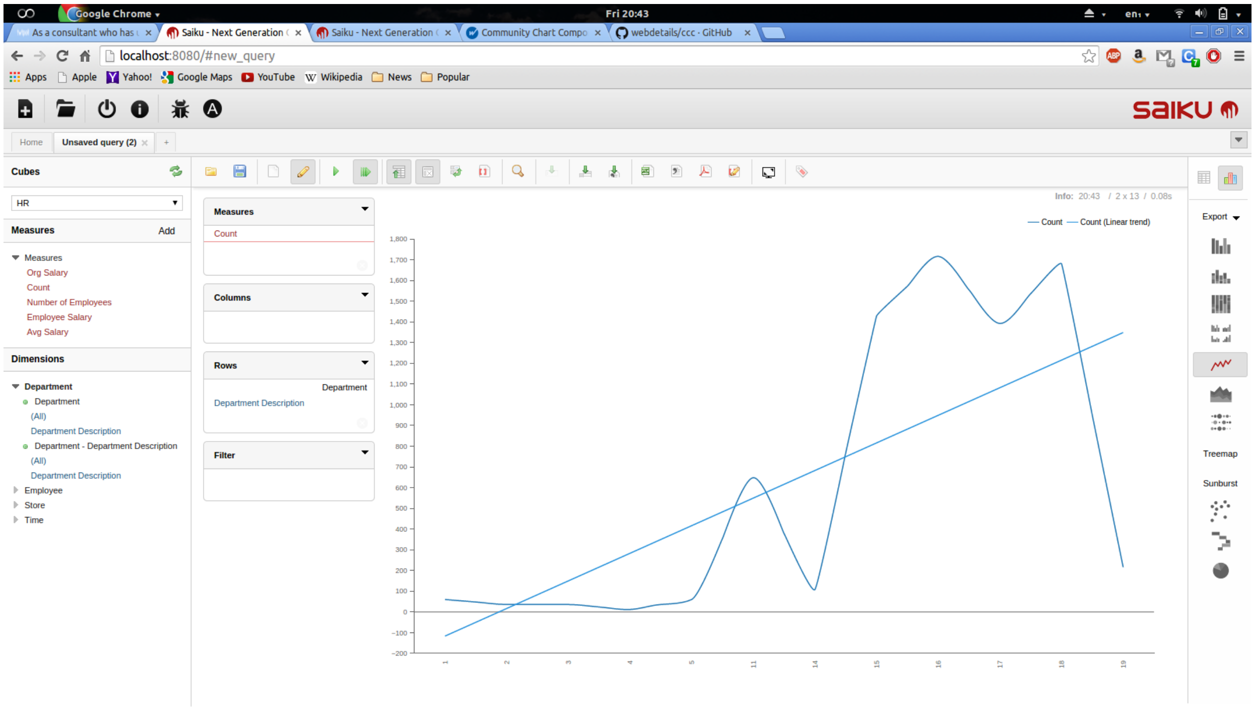
Saiku was founded in 2008 by Tom Barber and Paul Stoellberger. Originally called the Pentaho Analysis Tool, if started life as a basic GWT based wrapper around the OLAP4J library. Over the years it has evolved, and after a complete rewrite in 2010, it was reborn as Saiku.

Saiku offers a user friendly, web based analytics solution that lets users, quickly and easily analyse corporate data and create and share reports. The solution connects to a range of OLAP Servers including Mondrian, Microsoft Analysis Services, SAP BW and Oracle Hyperion and can be deployed rapidly and cost effectively to allow users to explore data in real time.

1.1 OLAP Analysis

On-Line Analytical Processing (OLAP) is a category of software technology that enables analysts, managers and executives to gain insight into data through fast, consistent, interactive access to a wide variety of possible views of information that has been transformed from raw data to reflect the real dimensionality of the enterprise as understood by the user. OLAP functionality is characterized by dynamic multi-dimensional analysis of consolidated enterprise data supporting end user analytical and navigational activities (The OLAP Council)

By harnessing the power of OLAP, Saiku allows users to choose the measures and dimensions they need to analyse and "slice and dice" the data and drill into the detail to uncover relationships, opportunities and issues. The intuitive user interface lets users drill down and up, filter, pivot, sort, and chart against OLAP and In-Memory engines. Utilizing the Olap4J library, Saiku is the first application on the market to offer support for Mondrian's Scenario feature allowing non destructive editing of query results, giving users the ability to adjust the figures and perform "what-if" analysis over their data. By harnessing the power of Mondrian, Saiku offers scalable in-memory analysis. Large amounts of data can be stored in memory in a distributed manner across the local network, offering greatly improved performance over large data warehouses as the aggregated data is retrieved from the network instead of reading from disk.



1.2 Architecture

Saiku is a modular analysis suite offering lightweight OLAP which remains easily embeddable, extendable and configurable. The Saiku RESTful server connects to existing OLAP systems, which powers user-friendly, intuitive analytics via our lightweight JQuery based frontend.

Look and feel can be completely customized as the user interface is based entirely on open standards. The supplied user interface is written in HTML, Javascript and CSS, making it easy to change or entirely replace the user interface. By using RESTful standards, the server can be easily integrated into different user interface technologies and 3rd party applications, the only requirements being that the 3rd party application can send and receive HTTP communication and understand JSON data instructions. The client application needs no understanding of MDX or related query languages.

1.3 Open Source

To make sure Saiku survives and sticks to its open source roots, Analytical Labs was created as a business to front Saiku and offer consultancy and development around the Saiku infrastructure. Analytical Labs is based in London.

Saiku is freely available to download from [www.saikuanalytics.com](http://www.analytical-labs.com). Organisations can use the solution internally without having to pay license fees. For customers who would like to use the solution in an Enterprise environment a number of options are available for support and to embed the solution into commercial applications.

CHAPTER 2

Administration Guide

2.1 Installation Guide

2.1.1 Deploy Saiku UI Under A Different Path

By default Saiku will provide the UI under localhost:8080 if you want to serve it under a different path you have to make a few changes. Step-by-step guide Install Saiku as normal Move tomcat/webapps/ROOT to your new folder, for example tomcat/webapps/saikuui Update the platformbean in tomcat/webapps/saiku/WEB-INF/saiku-beans.xml to point to your new path, ../webapps/ROOT/js/saiku/plugins/ —-> ../webapps/saikuui/js/saiku/plugins/ Refresh the browser or restart your server.

2.1.2 Saiku Enterprise Manual Deployment

New in Saiku EE 3.7 we now have a build for manual deploymet of Saiku for those of you who don't want to use the supplied Tomcat distribution. To install this you can donwload the package from here. To install it follow these instructions: Unzip archive Deploy ROOT.war and saiku.war to your server and unzip/extract them. Copy data.zip and repository.zip to another location which your server can access and unzip into separate folders. Set the correct permissions on the folders so that your server can read and write to these files and folders. Update the configs you need to update web.xml, saiku-beans.xml and applicationContext-spring-security-jdbc by default, these are located in the saiku.war webapp in WEB-INF here is an example: sed -i.bak -e 's/${foodmart_url}/jdbc:h2:/tmp/saiku/data/foodmart;MODE=MySQL/g' /var/lib/tomcat7/webapps/saiku/WEB-INF/web.xml sed -i.bak -e 's/${earthquake_url}/jdbc:h2:/tmp/saiku/data/earthquakes;MODE=MySQL/g' /var/lib/tomcat7/webapps/saiku/WEB-INF/web.xml sed -i.bak -e 's/../../data//tmp/saiku/data//g' /var/lib/tomcat7/webapps/saiku/WEB-INF/web.xml sed -i.bak -e 's/../../repository///tmp/saiku/repository//g' /var/lib/tomcat7/webapps/saiku/WEB-INF/web.xml sed -i.bak -e 's/../../repository///tmp/saiku/repository//g' /var/lib/tomcat7/webapps/saiku/WEB-INF/saiku-beans.xml sed -i.bak -e 's/../../data//tmp/saiku/data//g' /var/lib/tomcat7/webapps/saiku/WEB-INF/saiku-beans.xml sed -i.bak -e 's/../webapps//var/lib/tomcat7/webapps//g' /var/lib/tomcat7/webapps/saiku/WEB-INF/saiku-beans.xml sed -i.bak -e 's/../../data//tmp/saiku/data/g' /var/lib/tomcat7/webapps/saiku/WEB-INF/applicationContext-spring-security-jdbc.xml Restart your server, and visit the home page: http://localhost:8080 you should be presented with a login screen and be able to login with the default admin/admin credentials.

2.1.3 Upgrade

iUpgrading Saiku is very simple. First stop the server by running the stop-saiku script so that the databases and repository are shut down cleanly. Next install the new Saiku server in a different location. Finally replace the data and repository directories in the new installation with the folders from your old installation, this will copy over all your existing users, reports, schema and datasources. Restart Saiku. If you customised any of the configuration files then these will need upgrading as well.

2.1.4 Saiku Enterprise Edition

Requirements Saiku has very few requirements, it will run in less than 1GB of RAM, and will run on any computer running Java. Currently the only prerequisite is JDK7. Downloading Saiku Enterprise Edition To install Saiku Enterprise Edition the first job is to download the installer and create a license from our licensing portal. You license is tied to your server hostname and as such you need to make sure you provide the correct hostname, this is not the url you connect to your server with but the dns hostname. To find your hostname on Linux/Mac open a shell and type hostname On windows you need to run ipconfig /all Running the installer The installer is an executable jar file, on most operating systems you should be able to double click and run the file, this will pop up the first page of the installer. If you are wanting to install Saiku Enterprise on a remote server, or a computer without a graphics card, you can run java -jar ./saikuinstaller-x.y.z.jar -console and you will be given a text based installer. If you are installing in console mode when you get to the screen asking for the license, pass it the path to the license file. For example if you put the license in /tmp the path would be /tmp/license.lic Once the installer is complete it will then start Saiku server and you should be able to find a login screen at http://servername:8080/. To login to the standard install you will be able to login with admin/admin. If for some reason you do not see a login screen, look in installation directory/server/tomcat/logs/catalina.out This will contain any errors that occurred during the server startup.

2.1.5 Saiku Community Edition

This guide should help you when installing Saiku 3.0 for the first time. Installation: Make sure you have a version of JDK 7 installed. Check that the JAVA_HOME environment variable is set. Download the zip or gzip file for Saiku Unzip the archive to your desired location. Running Saiku 3.0 for the first time: Open a command prompt or terminal. Navigate to where the installer copied the files Execute either start-saiku.sh or start-saiku.bat (depending or your operating system). Open a browser and navigate to the URL: http://<server-name-or-ip>:8080/ You should be greeted by the Saiku login page. To login to Saiku you will need a free license. You can get this http://licensing.meteorite.bi Things to check if you’re having problems: Are you using JDK7? Is the JAVA_HOME variable set in the OS? Is port 8080 blocked in the firewall?

2.1.6 Other Methods

Docker Image We supply Saiku Community Edition as a docker image for easy deployment. To deploy Saiku run: docker pull buggtb/saikuce for Saiku Community Edition and it will pull down the latest snapshot build. Or for enterprise run: docker pull buggtb/saikueeofficial and it will install the latest stable release. Saiku will run a full server on port 8080 on the running docker container. Juju Charm Meteorite BI have teamed up with Canonical to get Saiku Analytics into the Juju Charm Store, whilst we work on finishing up the acceptance you can still test the Saiku CE Juju charm by running: juju deploy tomcat juju deploy cs:~f-tom-n/trusty/saikuanalytics juju add-relation saikuanalytics tomcat juju expose tomcat Again, this will run Saiku on Tomcat, the default port is 8080.

2.1.7 Pentaho Plugin

To install Saiku EE plugin, do this via the Pentaho Marketplace. This one click install will install the EE plugin and set it up ready for use after a server restart. The Pentaho plugin has 2 items that need configuring. To install the license you can place it in ~/.pentaho or pentaho-solutions/system/saiku/ named license.lic. If you install it in ~/.pentaho of course it will survive re-installation which can make administration easier. The other item that needs configuring is the user file. For named user licenses you need to configure the users allowed to login with your license. So if you purchased a single user, you can enter a single username, if you purchased 100 users, you can enter 100 user names etc. This file can be located in the same place as the license and is called users.txt. In this file enter a username per row, so for the pentaho default users you'd have: joe suzy pat tiffany

Make sure the file ends with a blank line. Once these are configured, you can log out and back into the Pentaho BI server and the new license and users configuration will be picked up by the Saiku plugin.

2.2 Administrators Console

Saiku Administration Console was first released with Saiku 3.0.0 because we moved from flat files to a JCR based repository. Currently the Administration Console is used to administering Users, Data Sources, Schema and Licenses. Users with the administrator role(by default ROLE_ADMIN) have access to the Admin console by clicking the A icon in the main toolbar.

Once you have opened the Admin Console tab you will see a tree menu on the left side of the window. Contained in this menu is the user management section, data source management, maintenance area and license information.

2.2.1 User Management

User Management

Users are stored within the Saiku H2 database, more configuration options will be coming in Saiku 3.x, although you can change the Spring configuration files now to point to a different database an administer the users outside of the Saiku Administration console.

Adding a User

To add a user to Saiku login as an Administrator and click on the “A” logo on the toolbar. Under user management select Add User.

Now enter a username (mandatory), and optionally an email address. You can then enter a user password, and add some roles to the new user to allow them to see certain files and folders. Finally click Save User and the user will be added to the database.

Removing a User

To remove a user, select the existing user from the list on the left hand panel of the Administration Console. You can then click the Remove User button to delete the user.

Edit a User

To edit a user, change roles, password etc, you can select the user from the list on the left and then make adjustments to the users details in the main edit panel. Once you have made your desired changes you can then click the Save button to save the users details.

By default Saiku user management is performed using the user management screen within the Saiku Admin Console.

To add a user you click on the Add User item.

You will then be presented with the user creation area.

Username	The username that the user will login with
Email Address	The users email address. Currently this is unused, but we have report emailing and scheduled delivery on the roadmap that would make use of this feature.
Password	The users password
Roles	Roles assigned to that user for access permissions and data source security. These are free form and can be anything you like, but it is recommended you follow a naming policy. For example, our default roles are ROLE_ADMIN and ROLE_USER.

Once you have filled in the users profile you can then press save and the user will be added to Saiku.

To edit a user, click on that user in the user list and make changes to their profile

2.2.2 Data Source Management

Data Source Management

Schema Upload

All Saiku connections require a Mondrian schema definition to tell Saiku how the database maps to a cube. To find out more about schema design I would read some of the resources available on the internet, for example: http://type-exit.org/adventures-with-open-source-bi/2010/07/a-basic-mondrian-cube-introducing-the-star-schema/.

Once you have designed your cube you can then upload it to the Saiku server, to do this you need to login as an administrator and go to the Administration console. Then click on the Add Schema button on the left pane. In the main window click the browse button and select your schema file. Next enter a unique name for this schema and then press the upload button. Assuming all goes well you will see an okay message and the new schema listed in the left hand panel.

Remove Old Schemas

To remove an old schema select the schema from the list on the left hand side, and then in the main screen click the Remove button. Your schema will then be removed from the server.

Data Source Creation

Once you have a valid schema available and uploaded to the server, you can then create a data source.

In the Administration Console you click on the Add Data Source button in the left panel. Once this is done you then give the data source a unique name, and if you are wanting to use the embedded Mondrian server you select the Mondrian connection type, if you are wanting to connect to a MSAS/Essbase or remote Mondrian server (or anything else that speaks XML/A) you can select XMLA from the connection type dropdown.

Next you enter the JDBC or XMLA url to the server, this depends on the database type so you will have to consult your documentation, here are a few examples:

jdbc:mysql://localhost/foodmart

jdbc:h2:../../data/foodmart

jdbc:postgresql://localhost/foodmart

you get the idea.

In the next box you need to select the schema that is valid for this database, and then your JDBC driver for example:

org.h2.Driver

com.mysql.jdbc.Driver

org.postgresql.Driver

You also need to make sure that the driver is available to the Saiku server by adding it to the tomcat classpath if it is not already available. To do this you can place it in tomcat/webapps/saiku/WEB-INF/lib and then restart the Saiku server.

Finally you need to add the username and password for the database you are connecting to and then click save. You should then see the data source listed in the list on the left hand panel.

If you are a Saiku 2.x user and you already have a number data sources available in a text file, or you want to add some details that require extra configuration on the commend line, you can click the Advanced button and enter the details into the text box instead of the input fields.

![](wiki-attachment:datasource.png)

Remove Data Source

If you no longer require a data source you can select it from the list on the left hand panel and then click Remove from the main window, this will remove the data source from the system.

Refresh Data Source Cache

Saiku will by default cache the data returned from queries to improve performance where possible. If you update the data available within the data base you can tell it to refresh the database cache by selecting the data source from the list and then clicking the Refresh Cache button.

Advanced Data Source Definitions

Saiku 3.x also accepts old style data source definitions with some minor changes because of the JCR repository. Once you have a schema uploaded you can then click the Add Data Source button and then Advanced. At this point you will be presented with a large text area field.

A Saiku 3 advanced data source looks like this:

type=OLAP name=dsname driver=mondrian.olap4j.MondrianOlap4jDriver location=jdbc:mondrian:Jdbc=jdbc:jtds:sqlserver://localhost:1433/database;Catalog=mondrian:///datasources/myschema.xml;JdbcDriver
username=sa password=password

As you can see, it looks very much like the Saiku 2.x data source except the catalog path has changed to a mondrian:// VFS pointer.

Securing data with roles

Saiku can restrict access to data at row level by applying role based security to your data source. To do this you need to define an Advanced datasource and apply the following:

security.enabled=true security.type=lookup security.mapping=ROLE_ADMIN=No HR Cube

Line 1 is self explanatory

Line 2 is the type of mapping, you can use lookup which will attempt to map Spring roles to Mondrian roles defined in the security.mapping row, or you can use one2one which will attempt to lookup the users roles defined in Saiku within the cube definition and apply those as security. If you are using XMLA you can also use passthrough, which will pass the security on to the XMLA provider.

Line 3 is if you use lookup type, this allows you to map Saiku roles to cube roles, so in this example the users who have the role ROLE_ADMIN will be mapped to the No HR Cube role within the schema defintion which looks like:

```
<Role name='No HR Cube'>
    <SchemaGrant access='all'>
        <CubeGrant cube='HR' access='none'/>
    </SchemaGrant>
</Role>
```

As you can see this role prevents access to the whole HR cube.

Additonal Connection Techniques

Connecting To Microsoft Analysis Server

Saiku provides an excellent interface to Microsoft Analysis Server and is very easy to setu. If you aren't using Analysis Services role based security you can select the XMLA menu option from the new data sources menu then enter the url where you have installed the MSMDPump DLL

If you want to use role based security you can still do this using an Advanced Data source, for example:

type=OLAP name=Internal driver=org.olap4j.driver.xmla.XmlaOlap4jDriver location=jdbc:xmla:Server=http://localhost/OLAP/msmdpump.dll username=admin password=admin security.enabled=true security.type=passthrough

By enabling security and adding the passthrough type Saiku will now pass the logged in users username and password to the XMLA endpoint so that Analysis Services can validate the user and assign the correct permissions to their queries.

Connect Saiku To Mssql With Instance Names

MSSQL has the concept of different instances on the same server which you address with instance names. Because of the way Saiku creates a connection string, this can be problematic, so here is one solution.

A big thanks to Amit at avatisolutions.com for the explanation!

Configure MSSQL to assign instances to dynamic ports (https://support.microsoft.com/en-us/kb/823938)

For main Saiku database connections you can now use the dynamic port to connect to a specific instance: jdbc:jtds:sqlserver://myserver:51218

For Saiku Schema Designer you can continue using the instance name as before: jdbc:jtds:sqlserver://myserver;instance=Amit

Saiku data sources come in two parts, the schema and the database connection.

The first job is to upload a valid Mondrian schema, in Saiku 3 we use Mondrian 4 which has an alternative schema design to Mondrian 3. Mondrian 4 is supposed to be able to read Mondrian 3 schema, but your mileage will vary, it is recommended you write your schema in Mondrian 4 syntax, or use the Saiku Schema Designer.

To upload a schema to Saiku, in the Admin Console, there is an Add Schema item in the left tree. Click this row and then choose your file to upload and give it a name.

When you press upload, the schema will be uploaded to the server and saved in its repository.

If you need to change it or remove it you can click on its name in the Schema subsection and press remove.

Next you need to add a data source, to do this click the Add Data Source item and you will then be presented with the data source configuration dialog.

In this section you can add a human readable name, select between Mondrian and XMLA connection types, and more.

Name	A human readable name for the data source.
Connection Type	Mondrian, XMLA or Mongo(EE only), change to XMLA if you are wanting to connect to an XMLA data source like Microsoft Analysis Services or Oracle Essbase.
URL	In Mondrian mode, this is the JDBC url of your database, for example, jdbc:mysql://localhost/foodmart. In XMLA mode the URL is the http:// url to your XMLA data source.
Schema	Not available in XMLA mode. This is the schema you have designed and uploaded to Saiku.
Jdbc Driver	The class of the JDBC driver, for example com.mysql.jdbc.Driver.
Username	Your data source username.
Password	Your data source password.
Advanced	This opens a freeform text area for more advanced connection types, for example if you need to use a digital schema processor.

2.2.3 Maintenance

The maintenance area allows you to backup and restore various aspects of the Saiku server.

To backup your repository press the Backup Now! button. This will download a zip file to your server containing the contents of your repository.

To restore the repository, select the backup file and press the Restore Repository button.

If you are migrating from Saiku 2.x you can also use the restore legacy reports button, to upload a zip file containing legacy reports which will be deployed to the server, due to changes in file structure and data sources, this is an experimental feature.

You might find it more flexible and easier to manage in a production environment to mount Saiku as a Webdav drive and backup/restore and maintain files using a file system instead of the administration console. Find out more here.

2.2.4 License Information

License Management

Enterprise Edition Only

Uploading a new license

If you trial has ended and you want to replace the license with a full license or if your full license need changing, then visit, http://servername:8080/upload.html from this screen you can add a new license and click the upload button.

Assuming everything goes okay, then you will be notified that the upload was successful. Log out and login again for the new license to take effect.

Installing your EE license in the Pentaho BI server

If you purchase a Saiku EE license and run Saiku within a Pentaho BI Server, here's how you install the license

Once your license has been issued, copy it to pentaho-solutions/system/saiku. If this server is remote you could use SCP via Linux or Putty on Windows or a Dropbox public folder to copy the file to the remote server. You should override an existing file called license.lic

Saiku normally uses named users, to maintain the list within the BI Server you need to edit the users.txt file located within the pentaho-solutions/system/saiku file. Make sure all valid logins are entered into this file 1 username per line, for example:

```
admin
joe
sue
```

You license will be limited to a certain number of users, as such the plugin will read the first x number of lines to find valid users. For example if you buy 5 users, the first 5 user names within that file will be valid and everyone else contained with the file will be excluded.

Maintaining the Named User List

If you have a user limit on your server license then you need to maintain the named user list. This list is a list of usernames allowed to login to Saiku, this list should be less than or equal to the number of licenses purchased. If you add more users than purchased seats then those users will not be able to login to Saiku. To add users navigate to the Admin Console and then Users List, by default this list is prepopulated with the Admin user, and you need to keep an Administrative user on the list otherwise you won't be able to access the admin panel. To add users simply add their name to the username box and click add user.

If your administrative user is not Admin you also need to change the main username which will allow you access to the server if you don't have the user in the named user list. To adjust this you need to look in tomcat/webapps/saiku/WEB-INF/saiku-beans.xml and find the licenseUtilsBean. To override the default user change the adminuser property to match your username.

The list of named users is independent of any users you have in the user management section as different authentication mechanisms may not use the User Management section to create users.

The list will allow you to add more users than the license permits, but they are evaluated in list order, so for example, if you have a 10 user license and add 15 users, if the person trying to authenticate comes in the last 5, they will not be allowed to login to Saiku.

Saiku User Quota

Included in the release of Saiku 3.6 was the new User Quota system. This allows Saiku Analytics customers with full licenses to allow unlicensed users to login up to 30 times. This allows users without a license to get a taste of Saiku and at that time you can then make a decision about buying the user a license. Once the user has used has used up all of their available logins they will be locked out and you are then required to buy a license for continued use. Users are informed of their quota usage in the query workspace, and administrators can find out about the quota usage in the About box which can be accessed from the main toolbar.

Saiku uses licenses to manage users and authentication. In the license area you can see the license type you have and its expiry date.

If you are an EE user who doesn't have a RAM based or unlimited user license, you also need to maintain the users list. This list is the valid users allowed to login to your platform. For example, if you purchase a 20 user Saiku license, the first 20 users in this list will be allowed to login to Saiku as stipulated in your license details.

2.3 Logging

2.3.1 Log Files

By default the log files are stored in *saiku-server/tomcat/logs*.

In this directory you will find, *catalina.out* and *saiku.log* both of which contain useful information when debugging or monitoring the Saiku server.

Administrators can also access these log files by browsing: *http://localhost:8080/saiku/rest/saiku/admin/log/{logfilename}*

So for example to view the current saiku.log open a new tab and enter: *http://localhost:8080/saiku/rest/saiku/admin/log/saiku.log*

2.4 Schema

A schema is the document used to define how your dimensions and measures map to your data source. In its raw form it is an XML document which describes the physical table structure and how those tables link together, and it describes your cubes, and the dimensions and measures they each contain.

2.4.1 Advanced Date Filtering

Enterprise Edition Only Saiku now supports advanced date filters, allowing you to do dynamic date ranges like Last x Periods, Yesterday, Today etc. To take advantage of these features you need to make some adjustments to your schema so Saiku knows how to apply the filters.

Firstly to use the Last Periods, and quick date selectors, you need to implement the Pentaho defined AnalyzerDateFormat. This format allows us to see how your date dimension is structured on any given level. So for example:

```
<Hierarchy name="Time" hasAll="false">
   <Level attribute="Year">
      <Annotations>
        <Annotation name="AnalyzerDateFormat">[yyyy]</Annotation>
      </Annotations>
   </Level>
   <Level attribute="Quarter">
      <Annotations>
        <Annotation name="AnalyzerDateFormat">[yyyy].[Qq]</Annotation>
      </Annotations>
   </Level>
   <Level attribute="Month">
     <Annotations>
       <Annotation name="AnalyzerDateFormat">[yyyy].[Qq].[mm]</Annotation>
     </Annotations>
   </Level>
 </Hierarchy>
 <Hierarchy name="Weekly" hasAll="true">
   <Level attribute="Year">
```

(continues on next page)

(continued from previous page)

```
     <Annotations>
       <Annotation name="AnalyzerDateFormat">[yyyy]</Annotation>
     </Annotations>
   </Level>
   <Level attribute="Week">
     <Annotations>
       <Annotation name="AnalyzerDateFormat">[yyyy].[Qq].[ww]</Annotation>
     </Annotations>
   </Level>
   <Level attribute="Day">
     <Annotations>
      <Annotation name="AnalyzerDateFormat">[yyyy].[Qq].[ww].[dd]</Annotation>
     </Annotations>
   </Level>
  </Hierarchy>
```

As you can see from the above snippet, on each level we declare how the MDX would look if you were to hand write the query but instead of putting dates in we just put the format of the levels in. The query then replaces these annotations with actual dates when the query is run. This means that when you reopen the query a new date is entered and as such, they are dynamic.

Secondly we have the Operator date selector which allows us to select from different date ranges which are not dynamic but frequently used. For example between, after, before and does not equal. This part of the dialog is controlled by the SaikuDayFormatString annotation. As you can see in the snippet, we apply this only to a day level. SaikuDateProperty refers to a member property you have defined in your schema which is a date string. yyyy/mm/dd, dd/mm/yyyy etc and the SaikuDayFormatString annotation is the format the dates are in.

```
<Level attribute="Day">
   <Annotations>
     <Annotation name="AnalyzerDateFormat">[yyyy].[Qq].[ww].[dd]</Annotation>
     <Annotation name="SaikuDayFormatString">yyyy/mm/dd</Annotation>
   </Annotations>
</Level>
```

2.4.2 Conditional Formatting

Saiku supports conditional formatting defined in the schema, and in an MDX query. Currently you can't apply run time formatting to drag and drop queries but this is in the roadmap. Schema Based Formatting To apply schema conditional formatting, you can open your schema and create a calculated member similar to this:

```
<CalculatedMembers>
 <CalculatedMember name='Profit' dimension='Measures'>
    <Formula>[Measures].[Store Sales] - [Measures].[Store Cost]</Formula>
    <CalculatedMemberProperty name="FORMAT_STRING" expression="Iif(([Measures].[Store␣
↪Sales]) &lt; 10000, '|(#,##0.00 &#8364;)|style=red', '|#,##0.00 &#8364;|style=green
↪')"/>
 </CalculatedMember>
</CalculatedMembers>
```

This snippet says that if the Store Sales value is less than 10,000 then its style is the colour red and apply a format mask to the number as well which ends with the Euro symbol. Similarly if it is more than 10,000 then make the cell green and apply a format mask. MDX Query Formatting You can also add conditional formatting to an MDX query:

```
WITH
MEMBER Measures.ProfitFormatted as Measures.Profit, FORMAT_STRING = IIf(Measures.
↪Profit > 2000, "|### €|style='lightgreen'|link='http://www\.analytical-labs\.com?
↪profit\=" || Cast(Product.CurrentMember.Name as STRING) || "'|","|(### €)|style='red
↪'|")
SET [~ROWS] AS
     {[Promotion].[Media Type].[Media Type].Members}
SELECT
NON EMPTY {[Measures].[ProfitFormatted]} ON COLUMNS,
NON EMPTY [~ROWS] ON ROWS
FROM [Sales]
```

In this example you can see it applies a format which includes a link to an external url and passes the current member as a parameter.

2.4.3 Creating A Schema Using Saiku Schema Designer

What Is A Schema?

A schema in its raw form is an XML document that defines how the data is laid out in your database. Within Saiku you might have multiple schema, each containing multiple cubes, within a cube are collections of Dimensions and Measures. The schema allows Saiku to display the UI elements that lets users discover answers to their data in a drag and drop format.

Understanding How A Schema Is Laid Out

A Saiku schema contains a number of distinct elements, the first section is called the Physical Schema, this is a description of how the tables you want to make available to you users are related to one another. For example you might have a fact table related to sales, this probably wont have a key. You then have a dimension for customers, this has a key on a CUSTOMER_TK field. The customer dimension joins to the sales fact table via the CUSTOMER_TK fields in both tables, so you then link these tables together using those keys. This means when the SQL is executed, Saiku understands how the tables join together. Once you have decided which tables are available you can then build the cubes, dimensions and measures that make up the main bulk of a schema. A cube is a collection of dimensions and measures relating to a specific set of information, for example, you might have a cube for sales data, another for marketing stats and so on. Dimensions reside both inside and outside of cubes, dimensions outside of cubes can be shared across cubes, unsurprisingly these are known as Shared Dimensions. Dimensions are usually textual data that can consist of hierarchies and levels. Returning to our customer dimension in the earlier description, a Dimension could be called Customer, within a Dimension you could have 0, 1 or many hierarchies, the choice is often driven by reporting requirements. In our customer example a hierarchy may simply be called Customer, within that hierarchy you might have a number of levels, for example Country, County, Town. You might then decide you want a second hierarchy called Category, where you have sorted your customers into distinct groups, this might contain a level called Groups. In doing this you could then easily create a report that shows sales by customer location and cross-tabulated with group information. Lastly we have Measures. Measures are numerical stats that allow us to, measure, the data we are interrogating. So, sales totals, sales counts, cost, tax and so on are likely measures in a sales type cube. Measures in Saiku are always numerical.

Connecting

Opening Saiku Schema Designer

To create a Saiku schema the first task is to open the schema designer. Administrators can click on the cube icon in the main toolbar.

This will then ask if you want to create a new schema or open an existing one. If you click on New you are then prompted for a schema name, which is a human readable name for the collection of cubes, it also needs to be unique as this is what the saved schema is called. When you click on create schema you are then asked to connect to a database.

Connecting to a database

To create a schema you need a working data source that Saiku can connect to. This can be either a standard JDBC compliant database, or a Saiku Mongo connection. Select the connection type you want and fill in the data source information.

Driver: this is the jdbc driver class, some common examples are:

MySQL: *com.mysql.jdbc.Driver*

PostgreSQL: *org.postgresql.Driver*

Oracle: *oracle.jdbc.OracleDriver*

MSSQL: *net.sourceforge.jtds.jdbc.Driver*

URL: This is the JDBC url for your database connection, some common examples are:

MySQL: *jdbc:mysql://database*

PostgreSQL: *jdbc:postgresql://host/database*

Oracle: *jdbc:oracle:<drivertype>:@<database>*

MSSQL: *jdbc:jtds:sqlserver://server/*

Username: Database Username

Password: Database Password

Once this is done, you will be given the ability to select a database and schema from that database. You should then be given a list of tables available to use within your Saiku schema.

Importing Tables

Defining Which Tables Are Available Within The Schema

A Saiku schema doesn't need to have access to all tables within your schema, this would just confuse matters, so first we decide which tables we want to include. At the top of the page you can see a list of tables and a summary of their first fields, if you need a table in your schema then drag the table from the top row and drop it on the canvas below.

If your table has a key field, like an ID or TK you can select the key in the next field, the key can also be made up of a number of fields.

You dimension tables should all have an ID, but your fact table may not have an ID field so you can just press cancel. Once you have selected your tables you can then join the tables that link together. Drag a table join circle from one table to the target, usually you would drag from source to target, so for example if you join a dimension to the fact table, you would drag from the dimension to the fact table and select the foreign key, this will create the join.

If you join the wrong way, don't worry you can just click on the join and change the order. Once you are happy with your tables, joins and keys you can then create your cubes.

Creating Cubes

Once you have your tables defined it is time to create your cubes, dimensions and measures.

Cubes contain collections of dimensions and measures, so for example you might have a cube that deals with Sales, and another cube that deals with Purchases and so on.

When you open the schema designer you will see we have pre populated the designer with a blank cube.

By default this cube is called Cube 1 which isn't very useful so the first thing you should do is give it a proper name.

Dimensions

Now you have done this we can create some dimensions. To do this you drag the tables you allocated in the screen before across from the left column to the centre column, when you drop it the schema designer will ask you if you want to "Auto generate all the attributes from every column of <tablename>?" If you have a table full of 1 level attributes then this can be a quick and useful way to import all your attributes quickly, but often you want to be able to define the keys and names of just a few columns or customise it entirely, in which case you can click Yes and edit it after the fact or No and insert the attributes manually.

Attributes

Attributes are mappings of table columns to the schema, so for example you might have a customer table and within that a customer name, address, phone number, country etc. These are attributes in your dimension, so create attributes to include them in your dimension.

If you have a hierarchy you might need to create a joined key that explains to Saiku how the hierarchical join would work, for example if you wanted to create a hierarchy that allowed users to select Country then drill down to State you would likely have to create a compound key in the schema of Country and State and select a name column that assigns State to the name.

If you don't want to define Hierarchies and Levels and instead just want a selection of 1 level attributes available to your users, you can check Has Hierarchy and then you don't need to create a hierarchy for it to be displayed in the Saiku UI, if you plan to create Hierarchies and Levels, leave this blank.

Hierarchies

Hierarchies are collections of levels, levels are attributes ordered as such that the drill and display order makes sense.

So using our customer dimension as an example, you might have a hierarchy also called Customers and an all member name called All Customers, within that you might have a few levels, Country, State, City, Customer Name. This would be a logical hierarchy for customers where you want to be able to slice and dice by geography. Similarly if you were creating time dimensions, Year, Month, Day would be a logical level order.

You can add as many hierarchies as you require per dimension, most will only have 1 or 2, but there is no restriction on the amount of hierarchies you have.

Cubes

So as mentioned earlier a cube is a collection of dimensions and measures. The dimensions we have created are shared globally so they need to be imported into the cube we are creating and along with that one or more measures need to be defined so we can "measure" our data.

First we have to understand the concept of measure groups, for those of you who have used Saiku 2.x or Mondrian 3 then there used to be a concept called virtual cubes where you would define multiple cubes and if you needed to compare measures in different cubes you would create a virtual cube, in Saiku 3 we use Measure Groups, these are tied to tables and you can have 1 or more of them. You don't have to have different fact tables for measure groups, you might just want to bucket your measures into different sections(currently the Saiku UI doesn't expose this but it is in development), or as discussed you might want to include more than one fact table in your cube.

Once you have created your measure groups you can select one for editing, first you need to select the fact table you will be creating the measures from, and give it a name, Default is fine, you might want to call it something different.

Next you add measures to your cube. Measures are always numeric but can be Sum/Average/Min/Max etc.

Give the measure a name, select the column that represents the measure and the aggregation you want to use. If you want to apply a format string you can add it in the box below, although this isn't mandatory.

Add as many measures as you require, then press the next button.

Lastly we need to link the dimensions we created in the previous section to the cube. So in the Select Dimension box select the dimension you want to import into your cube. If the dimension is based on the same table as the fact table used in the current measure group then the Select Foreign Key box will disabled itself as you don't require a key, but if your dimension is linking to an external table then you need to select the column in the fact table that links to this external table.

Once you have added all your dimension links, make sure you press the Update button to save the changes to that measure group.

Saving/Exporting

Saving

To save the schema to the Saiku repository, exit the Cubes and Dimensions dialog by pressing the Save button, and then click on the Create Schema button in the main dashboard designer toolbar:

This will publish your schema to the repository saving it in a file with the same name as the schema. You can then go across to the Administration panel to create or update a data source to use your new schema.

Note Currently you cannot open a saved schema for futher editing, so it is recommended that you keep the schema designer open until you are happy with the schema design. To make further modifications you can export your schema (see below) and open the file in a text editor.

Exporting

If you want to make further changes to your schema, backup or use it elsewhere you can click the Download button in the Administration panel under the schema you want to export and it will download the schema to an XML file on your local computer. You can then make changes to the file locally and upload it back to the server for further use.

2.4.4 More Information

Advanced Date Filtering - Learn how to use the Advanced Date Filter, for period on period filtering etc.

Conditional Formatting - How to make cells behave differently when different measure values are returned

Using Saiku Schema Designer - Create schema without a text editor using Saiku Schema Designer

Mondrian in Action - The Definitive Mondrian 4 book

Foodmart test schema - The Foodmart schema which powers the test suite and demo data, covers most use cases

Schema auto generated documentation - Find out what the various schema attributes do in Mondrian 4

2.5 Saiku Settings

Saiku has a number of UI settings that can be adjusted by editing *saiku-server/tomcat/webapps/ROOT/js/saiku/Settings.js*

Here are some of the most common:

Property	Default	Comments
VERSION		Current version for various dialogs. If you want to reskin Saiku you might want to change this.
BASE_URL	window.location.origin	The base url of the Saiku server, usually detected automatically but can be overridden.
TOMCAT_WEBAPP	/saiku	If you want to point to an alternative Saiku Server location, you can adjust this url.
REST_MOUNT_POINT	/rest/saiku/	Rest mount point, only needs changing if you make changes to the webapp.
DIMENSION_PREFETCH	true	
DIMENSION_SHOW_ALL	true	
DIMENSION_SHOW_REDUCED	false	
TABLE_LAZY	true	Enable or Disable table lazy loading, improves performance on large tables.
TABLE_LAZY_SIZE	1000	Number of rows to be rendered initially.
TABLE_LAZY_LOAD_ITEMS	20	Batch size to load new items.
TABLE_LAZY_LOAD_TIME	20	Load throttling for lazy loading.
MEMBERS_FROM_RESULT	true	Limit the members displayed in the filter dialog to those returned by the previous query. (Can be changed in the query)
MEMBERS_LIMIT	3000	Max members to import into the filter dialog.
MEMBERS_LIMIT	75	Search result limit.
ALLOW_PARAMETERS	true	Allow query parametrization.
DEFAULT_VIEW_STATE	view	Default to View or Edit mode.
EVALUATION_PANEL_LOGIN	true	Show or hide the evaluation credentials notice on the login page.

2.6 Embedding

Saiku can be embedded in a number of ways. Because of its restful nature, the server can be completely decoupled and used as a standalone service. If you just need Analytics in your webapp, then you have two choices.

2.6.1 IFrame Embedding

You can embed the Saiku UI within an IFrame and default to either a Table or Chart view. This is good when simple embedding within a wiki or webpage is required and no interaction between application and report is required. You could also change the CSS on the saiku server to mimic the target webpage so they look similar.

2.6.2 Saiku Embed Framework

Saiku Embed Framework is ideal for use cases when you require deeper embedding and interaction between source and target. For example you might want to create an application in which is passes parameters to Saiku reports to render the output, to do this Saiku Embed Framework is ideal.

Saiku Embed Framework

Initialize The Client

```
var myClient = new SaikuClient({
            server: "/saiku",
            path: "/rest/saiku/embed",
            user: "admin",
            password: "admin"
    });
```

Execute A Query

```
myClient.execute({
                  file: "/homes/home:admin/yardsyoy2.saiku",
                  htmlObject: "#saiku",
                  render: "chart",
                  mode: "line",
                  chartDefinition: {
                       width: 560,
                       colors: colors,
                       extensionPoints: {
                          legend: true,
                          legendShape: 'circle',
                          legendSize: {width: '100%'},
                          legendLabel_textStyle: "#990000",
                          legendFont: 'normal 11px "Open Sans"'
                        }
                   },
                   zoom: true,
                   params: {
                       teamname: rfiSchooldropdown.str
                   }
                  });
```

2.7 Backups and Restoration

It is recommended you backup the Jackrabbit repository from time to time in case of corruption.

To do this you need to navigate to the Backup/Restore menu in the Administration console. You can then click the backup button to get a zipped up export of the entire JCR repository.

To restore the repository you can unzip the latest backup you made and then upload the XML file to the server which will restore the repository. The restoration attempts to sync with whatever is already there so if you want to start afresh, you are advised to remove what it already in the repository using the Webdav access feature.

2.8 Migration From Saiku 2.x

Saiku 3 has a lot of new features in the backend our biggest change is adding a JCR repository. This change allows us to add extra functionality to the server and removes the need for having a bunch of files residing directly within the webapp, but it does add an extra layer of complexity when it comes to migrating and managing the new repository.

2.8.1 Importing Legacy Schema and Data Sources

As a one off import you can import legacy data sources and schema by placing them in the .... directory. This will cause the Saiku Server to attempt to import them into the JCR repository when the server starts and make them available to report off. Due to the complexity of these things, some changes will have to be made to the datasource definitions once they are imported because the catalogue uri is now Catalog=mondrian:///datasources/myschemapath.xml , the importer will try and change this setting during the import but due the various ways of configuring the Saiku server your mileage may vary.

2.8.2 Importing Legacy Reports

We also realise that people who have been using the Saiku Server 2.x will have a number of legacy reports that they may like to make available in the new server. To manage this we have created a legacy reports import facility. If you navigate to the Backup/Restore menu on the Administration console you will see the Restore Legacy Reports option. To use this you need to zip up your old reports in a zip file, don't organise them into different sub directories as the server doesn't know how to manage them, just put them all in the root directory. Then upload the zip file to the server. The server will then extract these reports to /etc/legacyreports where you can then move them to various other directories.

Because there are various differences with both Saiku 3 and Mondrian 4 in the way they process queries although we try our best to translate queries from the Saiku 2.x format into Saiku 3 queries, you may find that some do not work and will need manually recreating. Luckily the repository and file formats for Saiku 3 won't be changing so this is a one time upgrade.

2.9 Webdav Access

As Saiku 3 Standalone has migrated to a Jackrabbit JCR repository, we now allow access to this repository via Webdav.

Webdav drives look like network drives to operating systems and so you can "mount" Saiku's repository in My Computer on Windows or as a drive in Finder on Mac OS.

The default path to the Webdav server is dav://<hostname>:<port>/saiku/repository/default/homes

and once you have mounted it, you can navigate the files and directories using My Computer you can also access the repository via a browser on the same url.

Webdav Clients include Cadaver(Linux), Cyberduck(Windows/Mac) and many others.

The default credentials are admin/sa!kuanalyt!cs.

To change the password, you need to follow the instructions in tomcat/webapps/saiku/WEB-INF/saiku-beans.xml

2.10 Safe Mode

Saiku tries to start its Mondrian connections when the application is booted. This usually works fine, but occasionally due to misconfiguration or other error the Mondrian engine gets stuck trying to connect and the platform doesn't start properly.

To remedy this issue Saiku supports a "safe mode" which will start the platform but prevent any datasources from connecting allowing you to remove the troublesome connection and restore your Saiku server back to a working state.

To start Saiku in safe mode, add *-Dsaiku.safemode=true* to your *CATALINA_OPTS* or *JAVA_OPTS* and restart your Saiku server.

2.11 Localization

It is possible to attain a localized version of a Saiku view in the Saiku standalone version.

Follow these steps:

In the admin pane, add following to the datasource: *"DynamicSchemaProcessor=mondrian.i18n.LocalizingDynamicSchemaProcessor;locale=en_US;"*

Go to *WEB-INF/classes/mondrian.properties* and enable the *LocalizingDynamicSchemaProcessor* by adding following line: *"mondrian.rolap.localePropFile=locale.locale"*. This line tells that you will add a folder 'locale' that contains files prefixed with 'locale' containing the translations.

Create the folder 'locale' in *WEB-inf/classes/* and add some files: *locale_en_US.properties locale_fr_FR.properties*

Provide translations in your mondrian schema by changing variable names with the ${} notation.

Provide these translations in the properties files of step 3. Test your set-up From this point, you can manually check whether your translations work by changing the locale in the datasource definition.

Enable the ChangeLocale plugin by going to the Saiku Settings.js file and adding 'ChangeLocale' to the PLUGINS array:

```
PLUGINS: [
  "Chart",
  "ChangeLocale"
],
```

Make sure the locales are mapped in the settings.js file:

```
Settings.MONDRIAN_LOCALES = {
  "English": "en_US",
  "Dutch": "nl_BE",
  "French": "fr_FR"
};
```

A new button will appear which will change the locale in the data source definition and reload the connection with the translations.

The headers will not be translated in cause you use a Mondrian version prior to 4.3.0.1-SPARK.

2.11.1 Reports with non-Latin characters

Saiku reporting engine uses *Apache FOP <https://xmlgraphics.apache.org/fop/>* to generate PDF files. This framework comes out of the box with a set of Adobe's *Base-14 Fonts <https://xmlgraphics.apache.org/fop/0.95/fonts.html#Base-14-Fonts>*, those does not include non-latin characters. So, in order to export PDF reports for languages containing those characters (e.g. Chinese, Greeek, Russian or Arabic), the user need to tell FOP where those fonts are stored and which font to use.

In the Saiku installation directory, locate the file: *"saiku/saiku-core/saiku-web/src/main/resources/org/saiku/web/export/fop_config.xml"* This file, is well documented, containing examples on how to specify a directory containing True Type fonts, add one specific font or let FOP automatically load the operational system's fonts.

To specify a directory containing True Type fonts (bulk load):

```
<!-- "bulk" font mapping -->
<directory recursive="true">FULL_PATH_TO_YOUR_FONTS_DIRECTORY</directory>
```

To use one specific font:

```
<!-- your specific font mapping -->
<font kerning="yes" embed-url="FULL_PATH_TO_YOUR_FONT_FILE.ttf" embedding-mode="subset
↪">
    <font-triplet name="YOUR_FONT_NAME" style="normal" weight="normal"/>
</font>
```

To let FOP automatically load the operational system's fonts (very slow if you have many fonts installed):

```
<!-- automatically detect operating system installed fonts -->
<auto-detect/>
```

In order to change the PDF report font, you have to change the file: *"saiku/saiku-core/saiku-web/src/main/resources/org/saiku/web/export/xhtml2fo.xsl"*, the report font size and font family are specified on lines 9 and 10, respectively:

```
<xsl:param name="font-size" select="'YOUR FONT SIZE'"/>
<xsl:param name="font.symbol" select="'YOUR FONT FAMILY NAME'"/>
```

Be careful to keep parameters values inside sigle quotes.

2.12 Default Reports

If your users have regular reports you would like them to see when logging into Saiku you can now set Default Reports. The report definition can be set by user, role or globally. Inside tomcat/webapps/ROOT/js/saiku/Settings.js you will find this:

```
DEFAULT_REPORT_SHOW: true, // true/false
DEFAULT_REPORTS: {
'admin': [
    {
      path: '/homes/home:admin/test.saiku', // example: /homes/home:admin/chart.saiku
      visible: true // true/false
    }
   ],
'_': [
```

(continues on next page)

(continued from previous page)

```
       {
         path: 'ADD_PATH2',
         visible: false
        }
     ],
'ROLE_ADMIN': [
      {
         path: 'ADD_PATH3',
         visible: false
       }
   ]
 },
```

To edit the default reports, you just need to customise this to your own requirements. The first entry is for the user admin and they have 1 report visible when they login. The second entry is _ this is a global indicator. When anyone logs in they will be shown these reports. Finally, you'll see an entry for ROLE_ADMIN, this is role based report loading.

To find the path to the report you would like your users to see, open the repository browser and copy the path metadata like this:

2.13 Migrating From Pentaho Analyzer

Users of Pentaho Analzyer who would like to use Saiku with their existing queries can use this proof of concept utility to convert .xanalyzer files to .saiku Step-by-step guide Download the jar from https://dl.dropboxusercontent.com/u/8503756/saiku-xanalyzerconvert-3.7.5-jar-with-dependencies.jar Export your .xanalyzer query from your BI Server and extract it from the zip file. Execute the converter against the file: java -jar ./saiku-xanalyzerconvert-3.7.5-jar-with-dependencies.jar <path_to_file> <mondrian_4_true_false>.

For example: java -jar ./saiku-xanalyzerconvert-3.7.5-jar-with-dependencies.jar /tmp/test.xanalyzer true

If you want to run Pentaho Analyzer files on Saiku Standalone, make sure you set the Mondrian flag to true so it exports a slightly different format. If your connection details differ on your Saiku server you might need to configure the connection details at the bottom of the new Saiku file, you can open this file in any text editor. There are a lot of variables and differences between our tests and the real world, if your saved file isn't supported or doesn't work, please file a bug on JIRA with the sample xanalyzer file attached for us to debug.

CHAPTER 3

User Guide

3.1 Creating A Query

At the core of Saiku are the queries you will create to analyse data. You drag and drop dimensions and measures to build up the query you want. Under the hood Saiku uses the MDX query language to write the query but as an end user you don't need to see any MDX or understand what it means as we do all this for you. Open a new query window by clicking on the New Query icon . This will open a new query window and a blank canvas to create your query in. From the cubes list on the left hand panel select the cube you want to analyse from the drop down box.

When you select a cube this will populate the canvas with the available dimensions and measures.

To start analysing data all you need to do now is include at least one dimension in the rows drop area and either one measure in the measures drop area or another dimension in the columns drop area. Once you have fulfilled this critera then the query will execute and the results displayed on screen.

3.2 Filtering

3.2.1 Overview

Filtering Overview

There are a number of ways of refining the data you see from your query these include filtering the dimension members that appear in the resultset or using the Filter axis. A Saiku query has 4 drop zones to help you define your query, Measures, Columns, Rows and Filter. The filter axis allows you to filter what is visible in the result but using dimensions that aren't displayed on the table, for example, you may want to show store sales by region on a table, but only for 2014, and as such you would add the time dimension to the Filter axis. But, for example, if you wanted to show Q1 sales year on year, you would put the Year and Quarter levels on to the rows or columns axis, this means they would then show up in the table.

To select particular rows of data(members) then you need to click on the label in the dropbox to get the Filter popup dialog. Once you see this you can then select the members you want to see and move them to the selected side on the right and click the Ok button. When you press Ok the dialog will close and the query will be executed with the new filters in place.

3.2.2 Advanced

Include / Exclude Saiku now supports include & exclude within the filter dialog, which makes the filtering far more flexible on moving data sets. For example, you might just want to exclude 2 members, rather than include a fixed list which might need updating when your dimensions are updated. To exclude just add the members you want to exclude to the filter area and then instead of selecting the Include radio button, select the exclude button instead.

Searching and Pre-Filtering on Server If you have large dimensions, Saiku pre filters them on the server to improve user interface performance. This list on the left may be truncated because it would be too long and cause your browser to hang. To filter you can use the Search box at the top of the filter dialog.

You know if your filter dialog is being filtered when the Items: count equals the Display Limit count. The limit is adjustable by the administrator by looking in the Settings file

3.2.3 Defining Default Filters

Saiku supports the definition of default filters, which is done passing certain URL parameters. This feature is very useful when Saiku in embedded into a 3rd party platform.

Users can define two kinds of default filters:

- Hierarchy/Level value filters;
- MDX expression filter.

To define a **Hierarchy/Level value filter** users have to add URL parameters with the following syntax:

http://saiku_host:saiku_port?default_filter_[Hierarchy][Level]=value.

For instance:

http://saiku_host:saiku_port?default_filter_[Time].[Weekly].[Year]=1998.

You may add more than one parameter, e.g.:

http://saiku_host:saiku_port?default_filter_[Time].[Weekly].[Year]=1998&default_filter_[Time].[Weekly].[Day]=15.

Finally, in order to define a **MDX expression filter** users have to add URL parameters with the following syntax:

http://saiku_host:saiku_port?default_mdx_filter_AXIS=<filter expression>.

For instance:

http://saiku_host:saiku_port?default_mdx_filter_ROWS=[Customer].[Customers].[All Customers].

3.3 Charts And Graphs

3.3.1 Overview

Once you have designed your query charts are very straightforward in Saiku. Press the chart button in the top right hand corner of the workspace. This will show a bar chart.

If you want to show a different chart type you can select one from the list on the right hand side. Don't forget though, if you pivot the result set the chart data can be completely different so play around with your query if you don't see the layout you desire. For example:

The only differences between the two images above is the orientation of the result set, in the first image the time dimension is on the columns axis and the store state on the rows, in the latter its the other way around. This is why they call it data discovery!

3.3.2 Chart Properties

Saiku EE provides additional chart properties and features. You can click on the Properties button and you will be displayed with a Chart Settings dialog. Depending on the chart type you will be given a number of different options.

As you can see you can give your chart a title, and axis labels. You can also override the default colour set with your own colours, and disable the legend if you don't want it. Finally, if you are using a Line chart, you can add trend lines to the chart to show the trend over your dataset. There are 3 types Linear, Weighted Moving Average and Moving Average.

3.4 MDX Mode

If you want greater control over your query you can use Saiku's MDX mode. This allows users to edit the query itself rather than use the drag and drop tools available to you. You can either create a blank query or create something close to what you want and then edit the query itself. MDX is a multi dimensional query language and there are plenty of tutorials on the internet that explain how to write and understand MDX queries. http://mondrian.pentaho.com/documentation/mdx.php http://www.databasejournal.com/features/mssql/article.php/1495511/MDX-at-First-Glance-Introduction-to-SQL-Server-MDX-Essentials.htm

It is also a Microsoft standard so there is plenty of documentation available on the MSDN Network http://technet.microsoft.com/en-us/library/aa216767%28v=sql.80%29.aspx

3.5 Calculated Members

Starting in Saiku 3.0 we have a new Calculated Member feature that allows users to create dynamic measures within the user interface.

This pops up the calculated member dialog:

In this dialog you can edit the calculated measure name, the formula that defines the calculated measure and the output format. Edit the formula to match your requirements and press the Ok button. The calculated measure is then displayed in the measures list and you can then include it in the current query.

3.6 Parameter Support

For more advanced use cases, Saiku now supports parameters to inject variables into the query. If you embed a Saiku report into a 3rd party platform or want to provide an alternative filter mechanism, parameters allow you to define where users can make changes to filters on the fly without having to open the filter dialog box. Assign A Parameter Name Firstly you need to assign a parameter name. To do this put your level on an axis and click on it to popup the filter selection dialog. In here you will see the Parameter Name text box to add your parameter name, give it a name and press ok.

Populate the parameter value To use the parameter via the Saiku UI you can then enter the parameter value in the textbox at the top of the query:

Enter the value you want and press the query run button and your query will then be executed with the parameter in place:

```
 SET [~ROWS] AS
     {[Store].[Stores].[Mexico]}
 SELECT
 NON EMPTY {[Measures].[Unit Sales], [Measures].[Measure Name]} ON COLUMNS,
 NON EMPTY [~ROWS] ON ROWS
FROM [Sales]
```

3.7 Drilling

Overview

To access different subsections of data Saiku has two “Drilling” Methods Drill Through Drill through allows you to pick a cell and see the raw data that makes up the value of the measure you are looking at. For example:

First I select the drill through function by clicking on its icon.

Then I select the cell I am interested in on the result. In this example I have selected the Deluxe Supermarket row. I’ve then decided I want all the available information returned to me.

When I click okay, the output Saiku then shows me a table with the row level data, in this case a breakdown of the rows which made up the Deluxe Supermarket cell. Drill Across Drill across is a useful feature that allows users to find out more about a specific area of a query for example:

I click on the drill across icon.

In this sample query I have the total yards gained by USC and I want to know more about how they got them. I could rearrange the query by moving elements around or I could click on drill across. In this example I want to know the away team code and the Yards gained and Touchdowns scored, so I check those boxes and press ok.

When the query is executed, you can see that my Team Name level that was on Rows is now in the filter. Away Team Name has been moved to Rows and Touchdowns added to the measures. This then gives me the breakdown of the yardage and touchdowns scored by away team against USC.

3.8 Totals And Subtotals

Saiku has for sometime, supported Totals and recently introduced subtotal support. Totals

To add grandtotals to an axis, click on the drop down arrow in the Axis header to get the context menu. From there navigate to Grand Totals and select the total method you want to use. Subtotals Subtotals are used when you have multiple levels on an axis. To add subtotals create your query then click on the level you are interested in. At the bottom of the popup dialog you will see a subtotals box. Select the totals type you require and press ok.

The result will then be calculated and presented in the query:

3.9 Dashboards

Dashboards are reports that show a collection of sub reports on a single screen. These reports usually have something in common and allow users to see the bigger picture.

3.9.1 Dashboard Design

You can create dashboards within Saiku using the Dashboard Designer that was released in Saiku Enterprise 3.2. To create a dashboard, the first thing that is required are some saved queries created using the regular Saiku query designer. Once you have your required queries you can then create dashboards by clicking on the Create Dashboard button on the splash screen.

You will be greeted with a prompt asking if you want to create a new dashboard or open an existing one. Click on New and another dialog will open asking how many widgets you would like to have by default in the new dashboard. This isn't a fixed amount you can add or remove widgets later. Once you have entered the number of widgets and click create the blank dashboard will be created.

As you can see in the above screenshot we have a basic dashboard with a 2x2 grid. To add new widgets you can click on the Create Widget icon in the toolbar, to remove widget click on the widgets cog icon and click Remove Widget.

3.9.2 Filtering Reports

Of course just viewing the report might not provide the best insight into your data, and as such a filter might be needed to help you display different areas of your business. To add a filter, first you must add a parameter to your Saiku report. You can do this in the selections dialog and when you save this parameter it is then picked up and used by the dashboard designer.

Once you have added this report to your dashboard you can then see the Filter button enabled within the cog context menu. If you select that you can then choose which filter(if you have more than one) you would like to configure.

From the filter edit dialog you can give the filter a label so people understand the context. You also need to select the type, currently are Radio Button, Text Box, Dropdown and Dropdown Multiselect. If you have lots of values you might want Dropdown or Dropdown Multiselect if you just have a few the Radio Button dialog might be better, or if you want a user to type in a value just select Text Box. In the Content section you can then define if you want a user generated list of values or lookup the values from the data warehouse. If you click create list you are shown the following dialog

This lets you simply add elements to a list of values that the user can the choose from. If you select lookup you are shown the following

From here you can select Current Level which will lookup the values available to you from the current level in the query, this is the normal selection, if you would like to override it you can also select the override radio button. From here you can then select the schema, cube, dimension, hierarchy and level that the selection list is create from. Once you have the filter created press the save button and you are returned to the dashboard edit screen.

3.9.3 Viewing

To view a dashboard, in edit mode you can press the Show Dashboards button. This will open the dashboard in View mode and allow you to click around with it. If you aren't in Edit mode opening a new dashboard is simple. Click on the create dashboard icon on the splash screen

Then in the next dialog press the open button and find the saved dashboard.

When you select a dashboard you can see 4 buttons at the bottom of the open dialog. Edit puts you back into edit mode. View (New Tab) will open the dashboard within the Saiku UI like any other Saiku Report. View (New Window) will open the dashboard in a new browser tab or window which maximises screen space and gets rid of other UI elements. To filter a dashboard click on the Filter icon in the top left and the filter bar will pop out allowing you to refine the data you are viewing.

3.9.4 Adding Content

Once you have a grid organised you can then add content to your dashboard. To do this find the report you want to add in the tree on the left hand navigation window.

Then drag the report from the left hand panel to a widget in the main dashboard and drop it. This will then give you a table view of the report you have just added to the dashboard.

As you can see this widget isn't big enough to display the whole table, so we can adjust the widget size by dragging the resize handle in the bottom right corner of the widget.

You can remove reports from widgets by clicking on the cog icon and clicking the Remove Report menu item. Of course, we are dealing with graphical dashboards, so a table may not be the best way of displaying the information in the query. In this case we can switch from table mode to chart mode. To do this click on the cog icon and in the Render Type menu select Chart and then if you want to override the default chart type select the chart type you would like to use. You should now see your report rendered as a chart.

3.9.5 Adding Embed Code

To add a JavaScript code, add a report to the widget and open the context menu and click "Embed Code". This screen should appear for you:

In the text field "External Resources", you must add the URI of the external JavaScript libraries, as in this example:

In the field "JavaScript", you'll add your custom JS code, see example:

Then click the "Run" button and your code will run.

To retrieve the Saiku data, you must add the syntax correctly in your JavaScript code. See more information below. Adding Embed Code (styling table) It is also possible add some CSS styles in your table.

(info) SEE OTHER EXAMPLES HERE. Example of data returned from Saiku This is an example of Saiku data, you must understand how it is structured, for a better understanding with the syntax that will be used in the JavaScript code. ROWS (11/21) [0] { "name": "Pearl Chardonnay Wine", "data": [210, 54.28, 136.5] }, [1] { "name": "Pearl Imported Beer", "data": [175, 63.12, 159.25] }, [2] { "name": "Walrus Light Beer", "data": [187, 191.12, 484.33] }, [3] { "name": "Good White Zinfandel Wine", "data": [159, 247.85, 621.69] }, [4] { "name": "Good Light Beer", "data": [115, 100.04, 250.7] }, [5] { "name": "Pearl Light Wine", "data": [198, 46.07, 112.86] }, [6] { "name": "Pearl Chardonnay", "data": [189, 131.7, 328.86] }, [7] { "name": "Portsmouth Light Beer", "data": [175, 259.92, 663.25] }, [8] { "name": "Portsmouth Imported Beer", "data": [187, 152.32, 403.92] }, [9] { "name": "Good Imported Beer", "data": [154, 98.17, 249.48] }, [10] { "name": "Pearl Chablis Wine", "data": [209, 231.12, 560.12] } … COLS (4) [0] "Product Name", [1] "Unit Sales", [2] "Store Cost", [3] "Store Sales" Syntax (to add in code JS) Syntax Description {IDPANEL} Returns the id of the respective widget, for example: $('#{IDPANEL}').highcharts(…); {ROWS_name_VALUES} or {ROWS_data_VALUES} Returns an array of values of the object ROWS, for example: {ROWS_name_VALUES} >> ["Pearl Chardonnay Wine", "Pearl Imported Beer", "Walrus Light Beer", "Good White Zinfandel Wine"] {ROWS_data_VALUES} >> [[210, 54.28, 136.5], [175, 63.12, 159.25], [187, 191.12, 484.33], [159, 247.85, 621.69]] {COLS_name_VALUES} Returns an array of values of the object COLS, for example: {COLS_name_VALUES} >> ["Product Name", "Unit Sales", "Store Cost", "Store Sales"] {ROWS_name_data_EMPTY_name_data} Returns an object with names and data (or just names or data), for example: {ROWS_name_data_EMPTY_name_data} >> { "name": "Pearl Chardonnay Wine", "data": [210, 54.28, 136.5] }, { "name": "Pearl Imported Beer", "data": [175, 63.12, 159.25] }, { "name": "Walrus Light Beer", "data": [187, 191.12, 484.33] }, { "name": "Good White Zinfandel Wine", "data": [159, 247.85, 621.69] }

{ROWS_name_EMPTY_name} >> { "name": "Pearl Chardonnay Wine" }, { "name": "Pearl Imported Beer" }, { "name": "Walrus Light Beer" }, { "name": "Good White Zinfandel Wine" }

{ROWS_data_EMPTY_data} >> { "data": [210, 54.28, 136.5] }, { "data": [175, 63.12, 159.25] }, { "data": [187, 191.12, 484.33] }, { "data": [159, 247.85, 621.69] }

{ROWS_product_value_EMPTY_name_data} >> { “name”: “Pearl Chardonnay Wine”, “data”: [210, 54.28, 136.5] }, { “name”: “Pearl Imported Beer”, “data”: [175, 63.12, 159.25] }, { “name”: “Walrus Light Beer”, “data”: [187, 191.12, 484.33] }, { “name”: “Good White Zinfandel Wine”, “data”: [159, 247.85, 621.69] }

The name EMPTY is very important, because it is an escape word.

3.10 Saiku Tips And Tricks

3.11 How To Articles

CHAPTER 4

Project Information

4.1 Saiku Project Information

Saiku was created in 2008 as a test project to replace the clunky JPivot, since then we've been through a number of name and design changes until we settled upon Saiku the light weight and flexible OLAP server.

Saiku provides a fully functioning OLAP server out of the box, but can also been seen as a toolkit or framework allowing you to manipulate and extend upon it to fit your specific use case.

Saiku offers a user friendly, web based analytics solution that lets users, quickly and easily analyse corporate data and create and share reports. The solution connects to a range of OLAP Servers including Mondrian, Microsoft Analysis Services, SAP BW and Oracle Hyperion and can be deployed rapidly and cost effectively to allow users to explore data in real time.

CHAPTER 5

Development

5.1 Development

5.1.1 Building Saiku

Get the source and create a local working branchClone the repository and setup a ‘master’ local branch.

```
git clone git://github.com/OSBI/saiku.git
```

or if you are an authenticated committer, you would use..

```
git clone git@github.com/OSBI/saiku.git
```

if developing off of a non-master branch, create local branch that tracks the remote branch

```
git checkout -b somebranch origin/somebranch
```

Build all saiku projects (core libs + webapps for UI and backend)

```
cd saiku && mvn clean install -DskipTests
```

The most commonly used package is the saiku-server package. It includes the saiku webapps deployed on a tomcat (with or without sample data). The server package with foodmart can be found at:

./saiku/saiku-server/target/dist/saiku-server/

cd there and run the script *start-saiku.sh* / *start-saiku.bat* depending on your platform. Saiku server will then be available (in Tomcat) at [http://localhost:8080{]}(http://localhost:8080)

Warning It’s important to have *JRE_HOME* set to the java install you want to use. Ubuntu and probably other linux distros by default install open-jdk (which will not run Saiku).

All other packages are available at :

./saiku/saiku-server/target/

5.1.2 Developing Saiku in Eclipse

Once you have cloned the public git repo (see step 1 above), you are left with a number of maven projects. These projects are setup for inclusion in Eclipse. If you want any Saiku going in any other IDE you need to use the according maven goal (like mvn eclipse)

Note In either case you may see warnings in Eclipse indicating a bad JRE reference in your build paths. If so, you need to adjust the JRE used in the Eclipse build path (for each project) to point it to a valid 1.6 JRE.

Import into Eclipse with Maven plugin

If you have an Eclipse plugin installed for Maven integration, such as [http://maven.apache.org/eclipse-plugin.html{]}(http://maven.apache.org/eclipse-plugin.html), you can import Maven projects directly into Eclipse and the plugin will generate the correct Eclipse project metadata for you.

```
File->Import...
Select Maven->Existing Maven Projects
In "Root Directory" specify the root saiku folder
Select which projects to import and then Finish
```

Let Maven generate Eclipse project files. Change directory into each of the three top level saiku projects and execute the ‘eclipse’ goal of the ‘eclipse’ Maven plugin like so:

```
cd saiku-core && mvn eclipse:eclipse
cd saiku-webapp && mvn eclipse:eclipse
cd saiku-server && mvn eclipse:eclipse
```

Once you’ve done that, import the projects into your Eclipse workspace:

```
File->Import...
Select General->Existing Projects into Workspace
Choose "Select Root Directory" and specify the root saiku folder
Select which projects to import and then Finish
```

5.2 Server Documentation

If you are interested in finding out more about the inner workings of the Saiku server please checkout our Javadocs and inbuilt Enunciate documentation.

Our project uses Enunciate for documentation and some maintenance of endpoints. Most of our configuration is just based around on what is supplied out of the box but we may look to extend this in the future.

In the mean time you will find much documentation about our REST endpoints here: [http://demo.analytical-labs.com/saiku/serverdocs/{]}(http://demo.analytical-labs.com/saiku/serverdocs/) or if you have your own installation: [http://localhost/saiku/serverdocs/index.html{]}(http://localhost/saiku/serverdocs/index.html) Just hit login on any login screens that may appear.

CHAPTER 6

Getting Help

6.1 IRC

If you prefer more “real time” help and support. We have a very active and helpful community on the Saiku IRC channel. This can be found at ##saiku on the Freenode IRC network. Feel free to drop in and introduce yourself we’ll be more than willing to answer any questions or queries you may have.

6.2 Commercial Support

For customers who require Enterprise Support we offer a complete technical support solution. Support is available on an annual subscription basis. Support is provided via a dedicated support portal where customers can raise issues and support requests, track issue progress and receive responses from our full time support team.

For more information on our support services and pricing please visit our [website](http://saiku.meteorite.bi)

CHAPTER 7

Licensing

Saiku Analytics is licenced under the Apache Licence. Prior to version 2.4 the Saiku server component was licensed under the GPL v2 license and the UI under the LGPL license. From version 2.4 onwards we have switched the entire stack to the Apache license V2. We believe that using a more permissive license will help Saiku grow as a product as it allows people to use the software more freely.