**Executive Reports**

There are two main parts to executive reports:

1. The data collection process written in Perl. This should be set to run once every month, ideally around 12:01 AM on the first of every month. By default it collects and consolidates the data needed to generate the executive reports for that month.

(PINES Notes: It was taking about 45-60 minutes to run the process for a month

The code is on next-brick-01-head/openils/bin/executivereports)

1. The user interface written in PHP. It generates the report from data already collected by the monthly data collection process.

**Executive Reports UI**

The PHP UI code resides in the /report-creator directory along with the rest of quick reports. There are two main parts to the UI – the report menu where users select reports to view, and the report output.

Access to the report menu page is controlled by permissions.

Executive reports permissions overview (see below for configuration details):

1. Any user with access to quick reports will see Executive Reports if it is enabled.
2. You can create a permission that only allows access to the Executive reports menu but not other functions within quick reports.
3. The HTML report output has no security. Anyone can see it. Therefore it is possible to share reports simply by copying the report output URL and emailing to someone.

The new PHP files are:

* executiveReport.php
* config/executiveReports.config.php
* controllers/executiveReportsMenu.controller.php
* controllers/executiveReportGenerator.controller.php
* css/executiveReport\_v1.0.css
* css/executiveReportMenu\_v1.0.css
* models/executiveReport.class.php
* views/executiveReportsMenu.view.php
* views/executiveReportGenerator.view.php

The CSS files can be modified to customize the look and feel.

1. The CSS file executiveReportMenu\_v1.0.css provides the styles for the report menu interface. It is unlikely this would need to be modified.
2. The CSS file executiveReport\_v1.0.css provides the styles for the report that is generated. While anything can be modified, the main items to customize are:

* the logo file
* the background color for zero values
* the color of negatives

**PHP Configuration**

There are two configuration files that have to be modified for each specific installation of the executive reports PHP code. One file is new and one already exists as part of quick reports.

1. In the existing quick reports config/production.config.php file, you need to set the following:

|  |  |  |
| --- | --- | --- |
| Parameter | Function | Values |
| QR\_EXECUTIVE\_REPORTS\_ENABLED | Enables the quick reports UI | true | false |
| QR\_EXECUTIVE\_REPORTS\_START\_YEAR | First year that is displayed in the month/year list | 4 digit year (e.g.) 2016 |
| QR\_EXECUTIVE\_REPORTS\_ADDITIONAL\_ ALLOWED\_PERMISSIONS | Optional setting. Allows you to grant users access to Executive reports but not other parts of quick reports. Works in conjunction with existing parameter  QR\_USERS\_ALLOWED\_PERMISSIONS. By default users with access to quick reports will have access to executive reports. Only use this if you want to provide a permission that allows access only to the executive report menu but not other parts of quick reports. | List of integer perm values or an empty string '' |
| QR\_EXECUTIVE\_REPORTS\_DESCRIPTION\_URL | URL for the Report Data Definitions link | A valid URL |
| QR\_EXECUTIVE\_REPORTS\_TITLE\_TAG\_TEXT | The HTML TITLE tag value of the report page | Text |
| QR\_EXECUTIVE\_REPORTS\_OUTPUT\_ HEADER\_TITLE | The report title. It’s the first line in the upper left corner of a report. | Text |
| QR\_EXECUTIVE\_REPORTS\_CONSORTIUM\_ COLUMN\_HEADING | The name of the consortium that will be displayed on the report below the title. | Text |
| QR\_EXECUTIVE\_REPORTS\_ZERO\_VALUE | How zero will be displayed on reports. | Text |
| QR\_EXECUTIVE\_REPORTS\_TOTAL\_KEY | This is used internally to store column totals for grouped reports. It likely does not have to be modified unless there is a data value that happens to be the same. | Text string that must be unique from any data. |
| QR\_EXECUTIVE\_REPORTS\_SUBREPORT \_PADDING | Reports that are grouped have indented values. Set the number of spaces here. | String of &nbsp;  (HTML spaces) |
| QR\_EXECUTIVE\_REPORTS\_MISSING\_ SUBREPORT\_LABEL | Some grouped reports returning empty values. Use this to replace the empty cell on the report with another value. | Text |

1. The new configuration file config/executiveReports.config.php contains the following values which define the reports for PHP. Note there is a separate Perl configuration file that defines the queries for the reports that the Perl process uses to generate data. In PHP each report requires the following information:
   1. id – uniquely defines the report.
   2. category – groups reports together. There is space on the menu for 6 groups, 2 rows of 3 columns. Currently there are only 5 columns in use.
   3. name – this is what appears in the report menu
   4. description – this is what appears on the report output
   5. format – this is only defined when the report output should be currency. Otherwise the data format will default to integer.

**Database Configuration**

Executive reports requires two additional tables. Use the scripts below to create them. The database schema must match the schema you define in the configuration for PHP and Perl. In the scripts below the schema is set to quick\_reports:

create table quick\_reports.executive\_reports\_data\_collection

(

id serial NOT NULL,

time\_stamp timestamp without time zone NOT NULL DEFAULT now(),

year\_month integer NOT NULL,

org\_unit integer NOT NULL,

report character varying(10) NOT NULL,

sub\_report character varying(100),

data numeric(11,2),

CONSTRAINT pkey PRIMARY KEY (id)

)

create table quick\_reports.executive\_reports\_data

(

id serial NOT NULL,

create\_time timestamp with time zone NOT NULL DEFAULT now(),

year\_month integer NOT NULL,

org\_unit integer NOT NULL,

p1 text,

p2 text,

p3 text,

p4 text,

b1 numeric(11,2),

b2 numeric(11,2),

b3 numeric(11,2),

c1 integer,

c2 text,

c3 text,

c4 text,

c5 integer,

h1 integer,

h2 integer,

h3 integer,

h4 integer,

h5 integer,

h6 integer,

h7 integer,

i1 integer,

i2 numeric(11,2),

i3 integer,

i4 integer,

CONSTRAINT executive\_reports\_data\_pkey PRIMARY KEY (id)

);

**Perl Scripts**

There are 3 Perl script files:

1. erqueries.pl – defines the SQL queries
2. erconfig.pl – configuration details
3. erprocess.pl – the main script that you execute

The erqueries.pl file requires 3 queries for each report. One query generates the data for the branches, another for the systems and a third for the consortium as a whole. Each query is assigned to a variable. The name of the variable is not important. For simplicity we named them according to the id used in the PHP config file. For example in PHP there is a report id P1. Therefore 3 queries are defined in the erqueries.pl file – p1, p1consortium and p1system.

One note though is that they will be sorted prior to execution so they will not be executed in the order listed. Because of this it is best to name the 3 versions of the query with the same first few letters that are unique from other reports. This will ensure that the 3 queries for each report run together. This is necessary to allow for the process to be restarted with a certain query.

The following have to be set in erconfig.pl:

* db\_schema (the database schema containing the executive reports tables)
* log\_file (path to a detailed log file that will be written)
* status\_file (path to an html file for Nagios monitoring)
* email\_notify (email address to send completion email to)
* consortium\_ou (what is the id of the consortium record in the table actor.org\_unit)

Additionally you have to specify the database connection and email server information. This can be done either by specifying the path to the opensrf\_core.xml file:

* config

or specifying the following values individually:

* db\_driver
* db\_host
* db\_port
* db\_name
* db\_user
* db\_pw
* db\_timeout
* email\_host (email host for completion email notification)
* sender\_address (Value to use as the From in the email)

**Perl Script Execution**

The monthly data collection process consists of a collection phase and a consolidation phase. The data collection executes the 3 queries for each report and saves the data in the database table called executive\_reports\_data\_collection. When all of the queries are done executing, the script consolidates the data into the table called executive\_reports\_data. This table is where the data comes from when a user generates a report. The collection table is only used as temporary storage while the Perl script executes.

To execute the scripts you need to set execute permissions. Also, 2 files will be generated – one for the detailed log file and the HTML file for Nagios. You need to have write permissions on these.

To execute the full process for the previous month (hence why you should run it on the first of every month):

* perl erprocess.pl

To run it for a different month specify the year and month as a 6 digit integer in YYYYMM format. For example to collect data for January, 2015 you would use:

* perl erprocess.pl --period=201501

If you want to restart the process at a specific report you specify where to start. Remember that reports run in sorted order. Let’s say that all of the reports prior to p1system ran successfully. You could resolve the data issue and resume the data collection process. You have to start at p1 so you would use:

* perl erprocess.pl --start=p1

**(Optional) Distributed Perl Execution**

The script for generating one month of data executes in 45-60 minutes in the PINES test environment. Therefore it is unlikely that the process will ever need to be distributed. However it can be done. The process consists of a collection phase and a consolidation phase. You could run multiple copies of the scrips concurrently, even on separate hardware. To do this you would divide up the queries into multiple erqueries.pl files and execute each of the subgroups of queries with a different copy of erprocess.pl. However be sure to split such that all 3 queries for each report go in the same config file. Don’t put p1 in one file, p1consortium in another.

Now you need to execute the scripts but only do the data collection phase. Once all of the data has been collected by all of the processes, you can then execute the consolidation phase. To do this, first collect the data without consolidating on each of the smaller divided groups of queries using:

* perl erprocess.pl --noconsolidate

Then when all of the processes are done you run the consolidation phase only. You only run consolidation one time. It will consolidate all of data collected from all of the processes:

* perl erprocess.pl --nocollect

All parameters above can be combined. However, combining noconsolidate and nocollect together results in nothing being done.

**Process Output Files**

The detailed log file contains:

* a timestamp indicating the start of the script
* a timestamp indicating the start and completion of each query
* a timestamp indicating the start of the consolidation phase
* a timestamp indicating when the script completed

The simple status file provides a way for Nagios to monitor the status of the perl script. The values written to the file are:

* The name of the query currently executing (e.g. b1, p1, etc.)
* Consolidation (the consolidation phase started)
* Done (the process completed)

**Updating Queries**

You can modify queries simply by updating all 3 versions of the query in erqueries.pl. However be aware that the data already collected will be based on the version of the queries at the time the data was collected. Changes will only be reflected in the data going forward.

**Adding Reports**

To add reports you have to modify the following:

1. Define the report in PHP in config/ executiveReports.config.php
2. Add the 3 queries for the report in Perl to erqueries.pl
3. Add the new report to the consolidate query in erqueries.pl
4. Add a column to the executive\_reports\_data table to hold the new value that will be generated each month. No changes are needed to the executive\_reports\_data\_collection table.