ESR Business Intelligence Quick Reference Guide for BI Administrators

Author: James Haddon
Reviewers: NHS Development Team

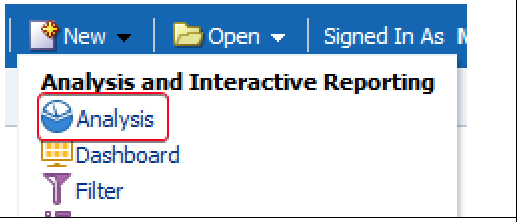
Change Record

Date	Author	Version	Change Reference
28/12/2012	James Haddon	0.1	Initial draft
19/03/2013	James Haddon	0.2	Updated to include useful calculations
22/03/2013	James Haddon	1.0	Initial release
07/05/2013	James Haddon	2.0	Update to include HTML
20/06/2019	Chris Holroyd	4.0	Update following developments

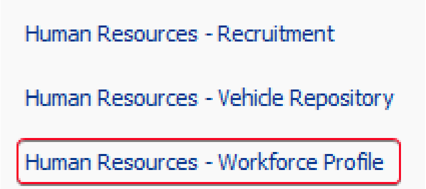
NHS ESR Business Intelligence Quick Reference Guide

Creating a new Analysis

1 Select New -> Analysis



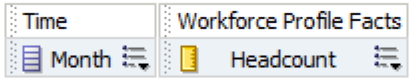
2 Select the relevant Subject Area


Human Resources - Recruitment

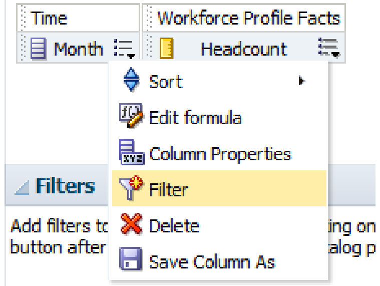
Human Resources - Vehicle Repository

Human Resources - Workforce Profile

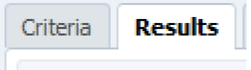
3 Double-Click (or drag & drop) required fields into analysis.



4 Create a filter to filter the results



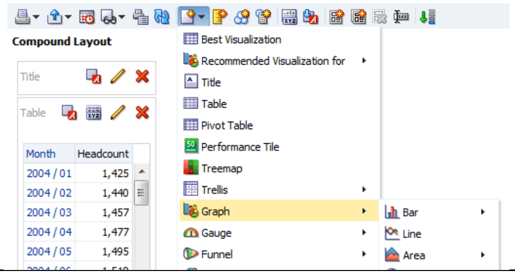
5 Click the results tab to see the results.



6 Save your analysis for later use.

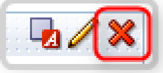
Adding a Chart to an Analysis

1 From the Result tab of an Analysis click the new icon, select graph then the type of graph needed.

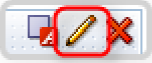


Month	Headcount
2004 / 01	1,425
2004 / 02	1,440
2004 / 03	1,457
2004 / 04	1,477
2004 / 05	1,495

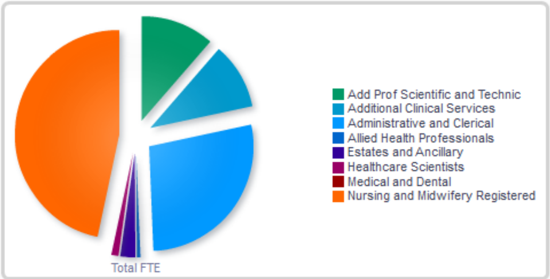
2 The chart will be displayed beneath your current data. Press the 'x' icon on the table of data to hide it if needed. Don't worry – you can easily retrieve this later.



3 Your chart is now displayed on its own. To alter the chart options click on the edit icon



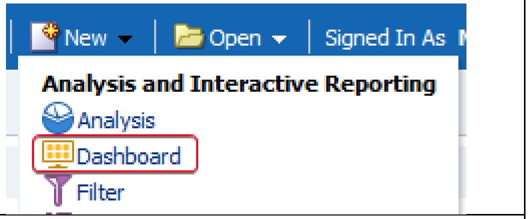
4 You may now edit the size, data included and colours of the chart. All charts should follow the NHS colour scheme guidelines available here.



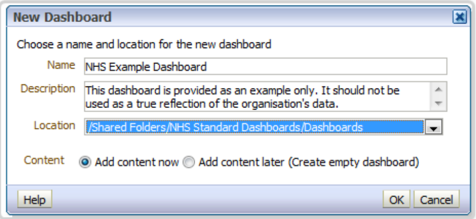
5 Click the 'Done' button, then save your analysis for later use.

Creating a new Dashboard

1 Select New -> Dashboard



2 Complete the name and description fields, and set the location of your new dashboard.


New Dashboard

Choose a name and location for the new dashboard

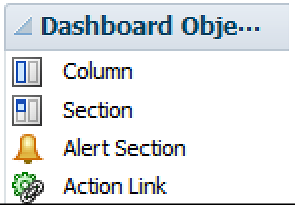
Name: NHS Example Dashboard

Description: This dashboard is provided as an example only. It should not be used as a true reflection of the organisation's data.

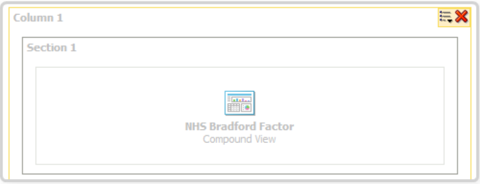
Location: /Shared Folders/NHS Standard Dashboards/Dashboards

Content: ● Add content now ○ Add content later (Create empty dashboard)

3 Drag Columns and Sections into the dashboard to arrange your analyses.



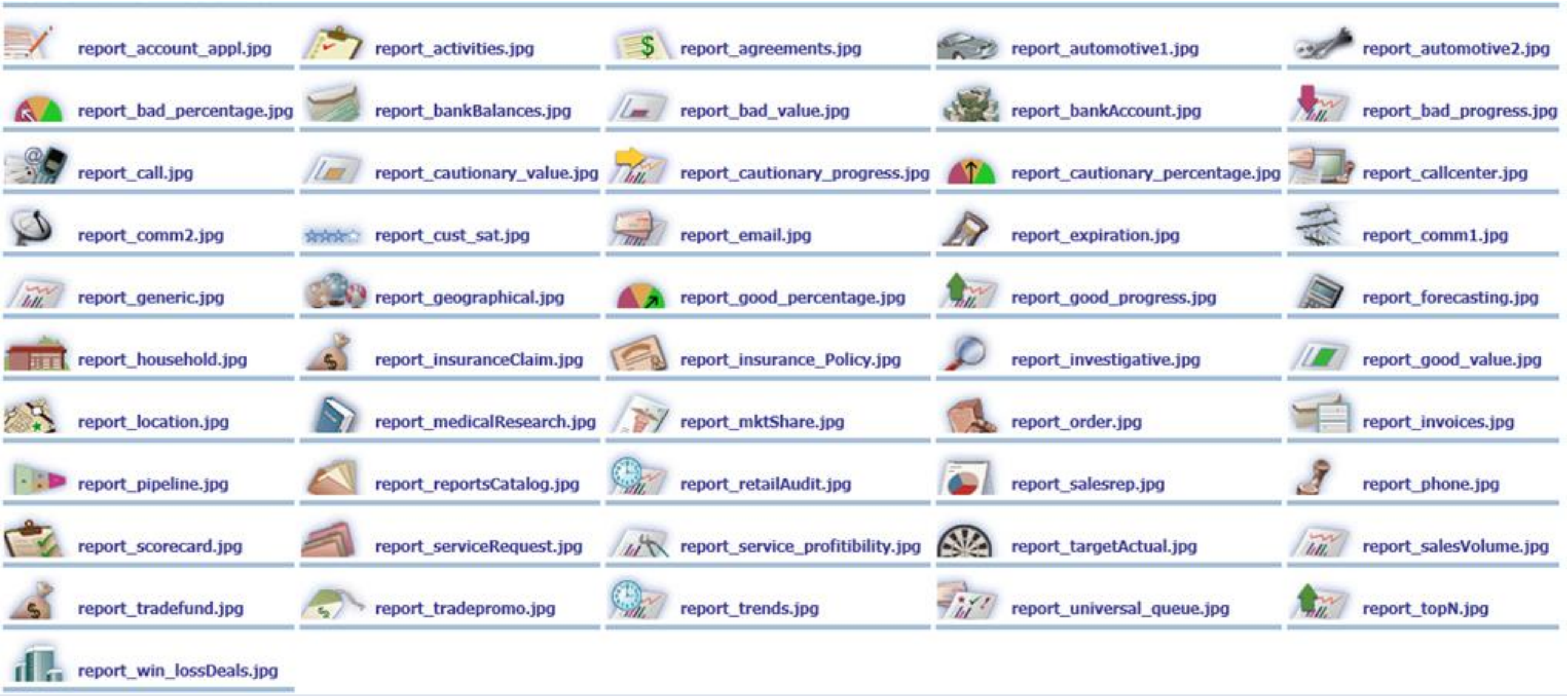
4 Drag your analyses into the sections created above.



5 Click 'Run' to run your dashboard

NHS ESR Business Intelligence Quick Reference Guide

Available Analysis Logos. Usage: fmap:images/image_name.jpg



report_account_appl.jpg	report_activities.jpg	report_agreements.jpg	report_automotive1.jpg	report_automotive2.jpg
report_bad_percentage.jpg	report_bankBalances.jpg	report_bad_value.jpg	report_bankAccount.jpg	report_bad_progress.jpg
report_call.jpg	report_cautionary_value.jpg	report_cautionary_progress.jpg	report_cautionary_percentage.jpg	report_callcenter.jpg
report_comm2.jpg	report_cust_sat.jpg	report_email.jpg	report_expiration.jpg	report_comm1.jpg
report_generic.jpg	report_geographical.jpg	report_good_percentage.jpg	report_good_progress.jpg	report_forecasting.jpg
report_household.jpg	report_insuranceClaim.jpg	report_insurance_Policy.jpg	report_investigative.jpg	report_good_value.jpg
report_location.jpg	report_medicalResearch.jpg	report_mktShare.jpg	report_order.jpg	report_invoices.jpg
report_pipeline.jpg	report_reportsCatalog.jpg	report_retailAudit.jpg	report_salesrep.jpg	report_phone.jpg
report_scorecard.jpg	report_serviceRequest.jpg	report_service_profitibility.jpg	report_targetActual.jpg	report_salesVolume.jpg
report_tradefund.jpg	report_tradepromo.jpg	report_trends.jpg	report_universal_queue.jpg	report_topN.jpg
report_win_lossDeals.jpg				

Reference: http://obiee101.blogspot.co.uk/2011/08/obiee11g-report-logos.html

NHS ESR Business Intelligence Quick Reference Guide

Useful Functions. Full documentation available here.

Conversion Functions

Function	Syntax	Example
CAST: Changes the data type of an expression to another data type. data_types = CHARACTER, VARCHAR, INTEGER, FLOAT, SMALLINT, DOUBLE PRECISION, DATE, TIME,TIMESTAMP, BIT, BIT VARYING	CAST(expr AS data_type)	CAST(staffgroup AS CHAR)
IFNULL: Tests if an expression evaluates to a null value, and if it does, assigns the specified value to the expression.	IFNULL(expr, value)	IFNULL(FTE,0)

Date/Time Functions

Function	Syntax	Example
CURRENT_DATE: Returns the current date.	CURRENT_DATE	CURRENT_DATE
DAYNAME: Returns the name of the day of the week for a specified date.	DAYNAME(dateExpr)	DAYENAME(startDate)
MONTHNAME: Returns the name of the month for a specified date.	MONTHNAME(dateExpr)	MONTHNAME(startDate)
WEEK_OF_YEAR: Returns a number (between 1 and 53) corresponding to the week of the year for the specified date.	WEEK_OF_YEAR(dateExpr)	WEEK_OF_YEAR(startDate)
DAYOFMONTH: Returns the number corresponding to the day of the month.	DAYOFMONTH(dateExpr)	DAYOFMONTH(startDate)
TIMESTAMPADD Adds a specified number of intervals to a specified timestamp, and returns a single timestamp. Adding a week translates to adding seven days, and adding a quarter translates to adding three months. A negative integer value results in a subtraction (such as going back in time).	TIMESTAMPADD(interval, intExpr, timestamp) Intervals = SQL_TSI_SECOND SQL_TSI_MINUTE SQL_TSI_HOUR SQL_TSI_DAY SQL_TSI_WEEK SQL_TSI_MONTH SQL_TSI_QUARTER SQL_TSI_YEAR	This example asks for the resulting timestamp when 3 days are added to 2000-02-27 14:30:00. Since February, 2000 is a leap year, the query returns a single timestamp of 2000-03-01 14:30:00. TIMESTAMPADD(SQL_TSI_DAY, 3, TIMESTAMP'2000-02-27 14:30:00')
TIMESTAMPDIFF: Returns the total number of specified intervals between two timestamps.	TIMESTAMPDIFF(interval, timestamp1, timestamp2) Intervals = SQL_TSI_SECOND SQL_TSI_MINUTE SQL_TSI_HOUR SQL_TSI_DAY SQL_TSI_WEEK SQL_TSI_MONTH SQL_TSI_QUARTER SQL_TSI_YEAR	Difference in days between timestamps 1998-07-31 23:35:00 and 2000-04-01 14:24:00. It returns a value of 610. Notice that the leap year in 2000 results in an additional day. TIMESTAMPDIFF (SQL_TSI_DAY, TIMESTAMP'1998-07-31 23:35:00',TIMESTAMP'2000-04-01 14:24:00')

String Functions

Function	Syntax	Example
CHAR_LENGTH: Returns the length, in number of characters, of a specified string. Leading and trailing blanks are not counted in the length of the string.	CHAR_LENGTH(strExpr)	CHAR_LENGTH(orgName)
CONCAT: Concatenates two character strings	CONCAT(strExpr1, strExpr2)	CONTACT(firstName,lastName)
INSERT: Inserts a specified character string into a specified location in another character string.	INSERT(strExpr1, integer1, integer2, strExpr2)	In the first string, starting at the second position, three characters (the numbers 2, 3, and 4) are replaced by the string abcd. INSERT('123456', 2, 3, 'abcd') Result: 1abcd56
LEFT: Returns a number of characters from the left of a string.	LEFT(strExpr, integer)	LEFT('123456', 3) Result:123
LENGTH: Returns the length, in number of characters, of a string.	LENGTH(strExpr)	LENGTH('abcd') Result: 4

LOCATE: Returns the numeric position of a character string in another character string. If the string cannot be found, 0 is returned.	LOCATE(strExpr1, strExpr2 [, integer])
	strExpr1 = needle
	strExpr2 = haystack
		LOCATE('d', 'abcdef')
		Result: 4
		LOCATE('g', 'abcdef')
		Result: 0

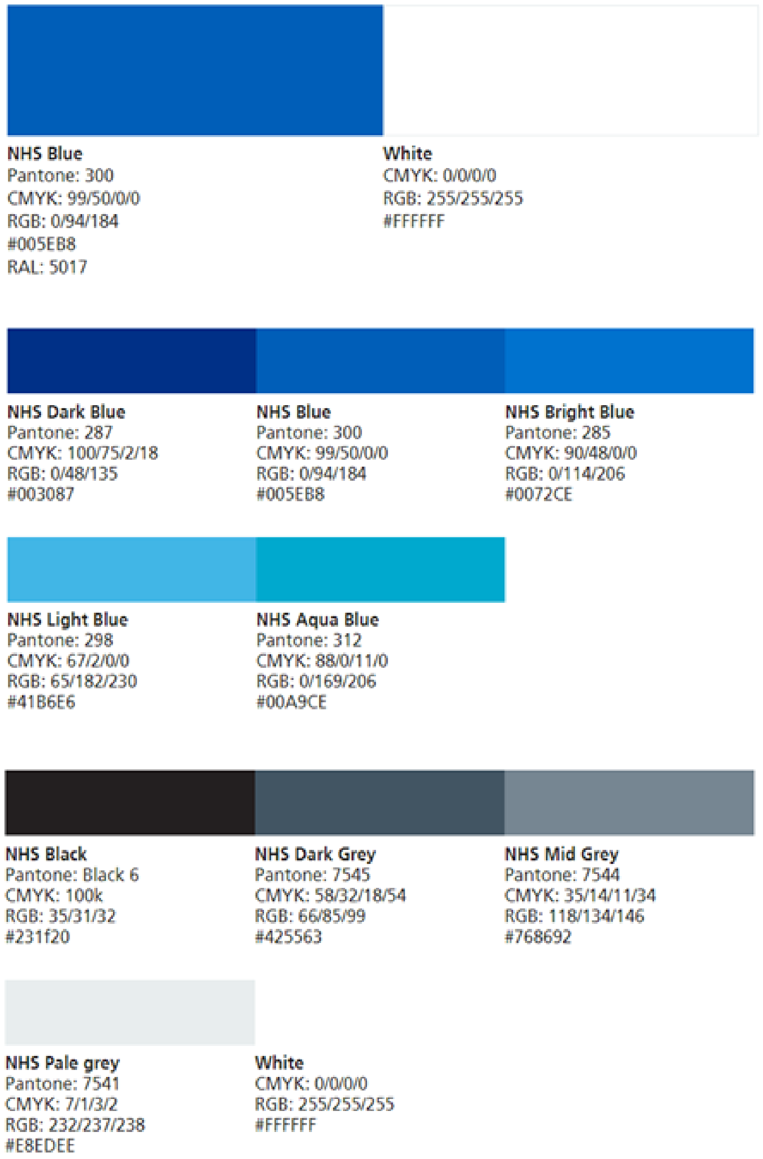
REPLACE: Replaces one or more characters from a character expression with one or more other characters.	REPLACE(strExpr1, strExpr2, strExpr3)	Replace('abcd1234', '123', 'zz') Result: abcdzz4
SUBSTRING: Creates a new string starting from a fixed number of characters into the original string.	SUBSTRING(strExpr FROM starting_position)	SUBSTRING('ABCDE' FROM 2) Result: BCDE

NHS ESR Business Intelligence Quick Reference Guide

Aggregate Functions

Function	Syntax	Example
AGGREGATE AT: Aggregates columns based on the level or levels you specify.	AGGREGATE(expr AT level [, level1, levelN])	AGGREGATE(sales AT Year)
AVG: Calculates the average (mean) value of an expression in a result set.	AVG(numExpr)	AVG(FTE)
BOTTOMN: ranks the lowest n values of the expression argument from 1 to n, 1 = the lowest numeric value.	BOTTOMN(numExpr, integer)	BOTTOMN(absenceDays, 10)
COUNT: Calculates the number of rows having a nonnull value for the expression	COUNT(expr)	Count(empNo)
COUNTDISTINCT: Count the number of distinct values in a result set.	COUNT(DISTINCT expr)	COUNT(DISTINCT empNo)
COUNT(*): Counts the number of rows.	COUNT(*)	COUNT(*)
MIN: Calculates the minimum value (lowest numeric value) of the rows.	MIN(numExpr)	MIN(absDays)
MAX: calculates the maximum value (highest numeric value) of the rows.	MAX(numExpr)	MAX(absDays)
MEDIAN: Calculates the median (middle) value of the rows satisfying the numeric expression argument. When there are an even number of rows, the median is the mean of the two middle rows.	MEDIAN(numExpr)	MEDIAN(absDays)
RANK: calculates the rank for each value satisfying the numeric expression argument. The highest number is assigned a rank of 1, and each successive rank is assigned the next consecutive integer (2, 3, 4,...). If certain values are equal, they are assigned the same rank (for example, 1, 1, 1, 4, 5, 5, 7...).	RANK(numExpr)	Rank(empScore)
STDDEV: returns the standard deviation for a set of values. If ALL is specified, the standard deviation is calculated for all data in the set.	STDDEV([ALL | DISTINCT] numExpr)	STDDEV(empScore)
SUM: Calculates the sum obtained by adding up all values	SUM(numExpr)	SUM(empScore)
TOPN: Ranks the highest n values of the expression argument from 1 to n, 1 = the highest numeric value.	TOPN(numExpr, integer)	Top 10 rows by absence days: TOPN(absDays, 10)

Colour Palette



NHS Blue
Pantone: 300
CMYK: 99/50/0/0
RGB: 0/94/184
#005EB8
RAL: 5017

White
CMYK: 0/0/0/0
RGB: 255/255/255
#FFFFFF

NHS Dark Blue
Pantone: 287
CMYK: 100/75/2/18
RGB: 0/48/135
#003087

NHS Blue
Pantone: 300
CMYK: 99/50/0/0
RGB: 0/94/184
#005EB8

NHS Bright Blue
Pantone: 285
CMYK: 90/48/0/0
RGB: 0/114/206
#0072CE

NHS Light Blue
Pantone: 298
CMYK: 67/2/0/0
RGB: 65/182/230
#41B6E6

NHS Aqua Blue
Pantone: 312
CMYK: 88/0/11/0
RGB: 0/169/206
#00A9CE

NHS Black
Pantone: Black 6
CMYK: 100k
RGB: 35/31/32
#231f20

NHS Dark Grey
Pantone: 7545
CMYK: 58/32/18/54
RGB: 66/85/99
#425563

NHS Mid Grey
Pantone: 7544
CMYK: 35/14/11/34
RGB: 118/134/146
#768692

NHS Pale grey
Pantone: 7541
CMYK: 7/1/3/2
RGB: 232/237/238
#E8EDEE

White
CMYK: 0/0/0/0
RGB: 255/255/255
#FFFFFF

NHS ESR Business Intelligence Quick Reference Guide

Colour Palette

NHS Dark Green
Pantone: 342
CMYK: 93/10/75/43
RGB: 0/103/71
#006747

NHS Green
Pantone: 355
CMYK: 91/0/100/0
RGB: 0/150/57
#009639

NHS Light Green
Pantone: 368
CMYK: 65/0/100/0
RGB: 120/190/32
#78BE20

NHS Aqua Green
Pantone: 3272
CMYK: 94/0/48/0
RGB: 0/164/153
#00A499

NHS Purple
Pantone: 2685
CMYK: 90/99/0/8
RGB: 51/0/114
#330072

Dark Pink
Pantone: 683
CMYK: 26/99/12/50
RGB: 124/40/85
#7C2855

NHS Pink
Pantone: 675
CMYK: 18/100/0/8
RGB: 174/37/115
#AE2573

NHS Dark Red
Pantone: 1955
CMYK: 9/100/54/43
RGB: 138/21/56
#8A1538

Emergency Services Red
Pantone: 485
RAL 3020 Traffic Red
CMYK: 0/95/100/0
RGB: 218/41/28
#DA291C

NHS Orange
Pantone: 144
CMYK: 0/51/100/0
RGB: 237/139/0
#ED8B00

NHS Warm Yellow
Pantone: 1235
CMYK: 0/31/98/0
RGB: 255/184/28
#FFB81C

NHS Yellow
Pantone: Process Yellow
CMYK: 0/0/100/0
RGB: 250/225/0
#FAE100

Useful Calculations

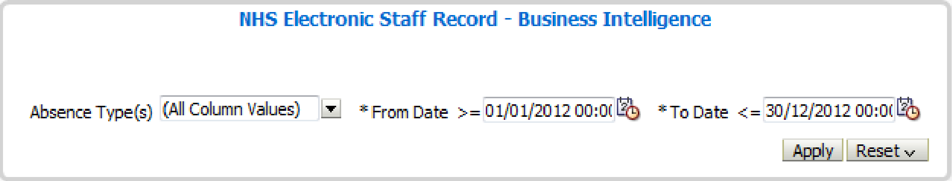
Calculation	Description
Repository Variable: CURRENT_DATE	Returns the current system date
TIMESTAMPADD(SQL_TSI_DAY, -1, CURRENT_DATE)	Yesterday
TIMESTAMPADD(SQL_TSI_MONTH, -1, TIMESTAMPADD(SQL_TSI_DAY , DAYOFMONTH(CURRENT_DATE) * -(1) + 1, CURRENT_DATE))	First day of previous month
TIMESTAMPADD(SQL_TSI_DAY , DAYOFMONTH(CURRENT_DATE) * -(1) + 1, CURRENT_DATE)	First day of current month
TIMESTAMPADD(SQL_TSI_MONTH, 1, TIMESTAMPADD(SQL_TSI_DAY , DAYOFMONTH(CURRENT_DATE) * -(1) + 1, CURRENT_DATE))	First day of next month
TIMESTAMPADD(SQL_TSI_DAY , -(1), TIMESTAMPADD(SQL_TSI_DAY , DAYOFMONTH(CURRENT_DATE) * -(1) + 1, CURRENT_DATE))	Last day of previous month
TIMESTAMPADD(SQL_TSI_DAY , -(1), TIMESTAMPADD(SQL_TSI_MONTH , 1, TIMESTAMPADD(SQL_TSI_DAY , DAYOFMONTH(CURRENT_DATE) * -(1) + 1, CURRENT_DATE)))	Last day of current month
TIMESTAMPADD(SQL_TSI_DAY , -(1), TIMESTAMPADD(SQL_TSI_MONTH , 2, TIMESTAMPADD(SQL_TSI_DAY , DAYOFMONTH(CURRENT_DATE) * -(1) + 1, CURRENT_DATE)))	Last day of next month
SELECT CASE WHEN 1=0 THEN "Time"."Date" ELSE TIMESTAMPADD(SQL_TSI_MONTH, -12, CURRENT_DATE) END FROM "Human Resources - Workforce Profile"	Default a date in prompt (today – 12 months in this example)

NHS ESR Business Intelligence Quick Reference Guide

Best Practice

Dashboard Layout

Title / Prompts: Give each dashboard page a title, in title case. Ensure prompts are provided horizontally across the top of the dashboard taking up as little space as possible.


NHS Electronic Staff Record - Business Intelligence

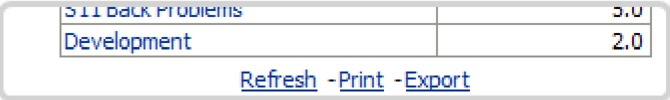
Absence Type(s) (All Column Values) ▼ * From Date >= 01/01/2012 00:0(* To Date <= 30/12/2012 00:0(

Apply | Reset ⌄

If you have a large number of prompts, you may wish to make the section collapsible. Always include an 'Apply' and 'Reset' button for prompts.

Screen Resolution: Always build / configure your dashboards with a resolution of 1024x768 in mind as this is the most common resolution for desktop PCs. If you know your organisation uses a different resolution, ensure you conform to this.

Report Links: Include links that users will actually need to use, and not all links. Try to keep the number of 'options' a user has to a minimum in a simple dashboard.



S11 Back Problems	3.0
Development	2.0

Refresh - Print - Export

Supporting Text: Dashboard developers are able to add 'Static Text' areas to dashboards. If you have complex analyses in a dashboard, think about including a static text area to help users understand what is being shown to them.

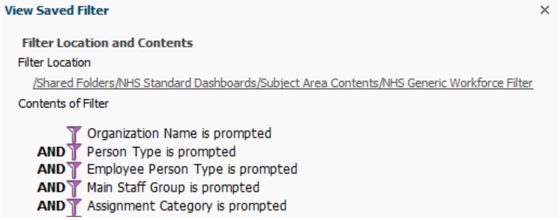
HTML: HTML can be added to dashboards and analyses to support users in using the system. For example –hyperlinks can be added to a page to direct users to an external site or document.

There are no restrictions on the links that can be added to BI – BI Administrators must ensure any HTML used has been checked and where external links are used they are checked regularly to ensure they do not misdirect users to potentially unsafe sites. Organisations are responsible for the HTML used in any locally created dashboards and must be aware that the use of HTML in dashboards is not supported.

Analysis Design

Colour Scheme: Always use the colour scheme to base your charts on. Try to ensure that colours used have the same or similar meanings on all analyses.

Prompts: Where possible, try not to include prompts for individual analyses. Users will find it easier to complete all prompts in one place (i.e. at the top of the dashboard) rather than having to complete them several times for each analysis.


View Saved Filter ×

Filter Location and Contents

Filter Location

/Shared Folders/NHS Standard Dashboards/Subject Area Contents/NHS Generic Workforce Filter

Contents of Filter

Organization Name is prompted
AND Person Type is prompted
AND Employee Person Type is prompted
AND Main Staff Group is prompted
AND Assignment Category is prompted

Filters: Where possible, try to create a 'Saved' filter that you can apply to all of your dashboards, rather than creating the same filter multiple times. This also helps when drilling from one dashboard to another as any 'Prompted' filters take values from the drilled-from dashboard.

Action Links: By default, some data items will have 'drillable' functionality included. This may not always be right for your analysis (for example, you may wish the user to be navigated to a different dashboard rather than drilling on the same analysis). Try to create a 'Repository' of action links to use in your dashboards rather than creating a new action link for each one.

Formatting: Where possible, use the 'Format Copy' functionality to copy formatting from one analysis to another. Not only does this reduce your workload, it also ensures analyses are formatted in the same way to make them easy to understand.

Analysis Criteria

Reduce Data with Filters: ESRBI is provided to enable users to see an easy view aggregated data to enable them to make business decisions. ESRBI is not provided as an 'Export' tool to export large amounts of data. When creating analyses, limit and appropriately aggregate your data to ensure your dashboards are usable and performant.