

# OA Designer Module 1

---

Prepared by:	Documentation Team
Prepared for:	Learning Resources
Module:	Modules
Date:	2016
Document Ref:	LMDSY0050
Version:	4.0

Construction Industry Solutions Ltd.  
11 St. Laurence Way  
SL1 2EA

# Contents

<b>1 Database Overview</b>	<b>6</b>
1.1 Relational Database Model .....	7
1.2 Tables and Modules .....	8
1.2.1 Database Structure .....	8
1.2.2 Company Specific Modules .....	9
1.2.3 Process Specific Modules .....	9
1.3 Summary and Detail Tables .....	11
1.4 Open Items .....	12
1.5 Indexes .....	13
1.6 Record Service Procedures .....	14
1.7 Read Only Fields .....	17
<b>2 OA Query Language</b>	<b>19</b>
2.1 FOR EACH .....	20
2.2 WHERE .....	21
2.3 Joining Tables .....	23
2.3.1 EACH .....	23
2.3.2 FIRST .....	23
2.3.3 LAST .....	23
2.3.4 OF .....	23
2.4 Curly Braces .....	25
<b>3 OA and BI Utilities</b>	<b>26</b>
3.1 Database Enquiry .....	27
3.2 Query Editor .....	29
3.2.1 Search and Replace .....	33
3.2.2 Exporting from the Query Editor .....	34
3.2.3 Saving Queries in Query Editor (10.27) .....	34
3.3 Calculation Editor .....	36
<b>4 Functions and Sections</b>	<b>37</b>
4.1 Function Naming .....	38
<b>5 Datasets - Overview</b>	<b>39</b>
5.1 Creating a Data Set .....	40
5.1.1 Data Set Fields .....	43
5.2 Using Data Sets in the Query Editor .....	45
5.3 Using the Data Set in Queries .....	46
5.3.1 In-Line Reports .....	47
5.4 Granting Access to Data Sets for Other Users .....	49
5.5 Rules for Keys .....	50
5.6 Best Practice .....	51
5.7 Cross Modular Reporting .....	53
6.1 Report Pre-Processing (Syuds) .....	54
6.1.1 Syuds.Calc .....	55
6.1.1.1 syuds.calc example .....	55
6.1.2 Syuds.Debug .....	58

6.1.3 Syuds.Delete .....	59
6.1.4 Syuds.Filter .....	60
6.1.5 Syuds.GroupQuery .....	61
6.1.6 Syuds.Merge .....	63
6.1.7 Syuds.Pivot .....	65
6.1.7.1 Report Design .....	67
6.1.7.2 Sample Output .....	69
6.1.8 Repeat .....	71
6.1.9 Syuds.Store .....	72
6.1.10 Sum .....	73
6.1.11 TableAlias .....	74
6.1.12 Syuds.TimeSlice() .....	75
6.1.13 Top .....	77
6.1.14 Union .....	78
<b>7 Data Set Exercises .....</b>	<b>80</b>
<b>    7.1 Using a Dataset in a Report .....</b>	<b>81</b>
7.1.1 Create the Function and Menu .....	81
7.1.2 Set up the Page .....	81
7.1.3 Add the Page Section Forms .....	82
7.1.4 Add the Fields .....	83
7.1.5 Create the report section .....	84
7.1.6 Sorting the Report .....	85
7.1.7 Add the Fields .....	87
7.1.8 Running the Report .....	87
<b>    7.2 Using multiple Data Sets .....</b>	<b>89</b>
7.2.1 Create the Function and Menu .....	89
7.2.2 Creating the new Data Set .....	89
7.2.3 Adding the Fields .....	90
7.2.4 Create the Report .....	91
7.2.5 Running the report .....	92
<b>    7.3 Using multiple Data Sets to merge data .....</b>	<b>94</b>
7.3.1 Create the Function and Menu .....	94
7.3.2 Creating the first new Data Set .....	94
7.3.3 Adding the Fields .....	94
7.3.4 Adding the Fields .....	95
7.3.5 Running the report .....	98
<b>    7.4 Exercise 1c - Using Datasets for Exporting Data .....</b>	<b>99</b>
7.4.1 Create the Function and Menu .....	99
7.4.2 Create the Report Section .....	99
7.4.3 Running the report .....	100
<b>    7.5 Exercise 1d - Dynamically switching between multiple Data Sets .....</b>	<b>101</b>
7.5.1 Create the Function and Menu .....	101
7.5.2 Build the datasets .....	101
7.5.3 Create the page .....	101
7.5.4 Create the Report .....	102
7.5.5 Running the report .....	103
<b>    7.6 Exercise 2 - Generating Pages from a Data Set .....</b>	<b>104</b>
7.6.1 Create a Browse Page from a Data Set .....	104
7.6.2 Add New Functions for the Browse Page .....	105
7.6.3 Create the Page .....	106
7.6.4 Running the Function .....	106
<b>8 Report Builder .....</b>	<b>108</b>
<b>    8.1 Initial Setup .....</b>	<b>109</b>
8.1.1 Browser Version .....	109

8.1.2 Report Builder System Parameters .....	110
8.1.3 Report Data Sources - Pre-Built Datasets .....	111
8.1.4 Security .....	112
8.1.5 Query Results Licence .....	113
<b>8.2 Report Builder .....</b>	<b>114</b>
8.2.1 Canvas .....	115
8.2.2 Button and Options Bar .....	116
8.2.2.1 Report Actions .....	116
8.2.2.2 Field Formatting .....	116
8.2.2.3 Canvas View .....	117
8.2.2.4 Actions .....	117
8.2.2.5 Run .....	117
8.2.3 Opening an Existing Report .....	118
8.2.4 Creating a New Report .....	119
8.2.4.1 Fields .....	119
8.2.4.2 Data Source .....	121
8.2.5.3 Existing Data Source .....	121
8.2.6.4 Custom Data Source .....	121
8.2.7.5 Data Frame .....	123
8.2.7.6 Adding Fields .....	124
8.2.7.7 Groups .....	127
8.2.8.8 Fields .....	127
8.2.8.9 Filters .....	129
8.2.8.10 Run Report .....	131
8.2.8.11 Refining the Report - Fields .....	132
8.2.8.12 Field Positioning .....	134
8.2.8.13 Text, Count, Calculations .....	135
8.2.8.14 Dataset Field Update .....	137
8.2.8.15 Adding Fields to Datasets .....	138
8.2.9 Report Builder List .....	141
<b>9 Page Design Exercises .....</b>	<b>142</b>
<b>9.1 Creating a Contract Browse Screen .....</b>	<b>143</b>
9.1.1 Create the Function .....	143
9.1.2 Create the Page .....	144
9.1.3 Create a Body Page Form .....	144
9.1.4 Add the Fields .....	145
9.1.5 Adding a Sort to a Column .....	146
9.1.6 Linking a Field to another Enquiry .....	146
9.1.7 Running The Function .....	147
<b>9.2 Enhancing the Contract Browse Screen .....</b>	<b>148</b>
9.2.1 Modifying the Page Section .....	148
9.2.2 Adding Totals to the Page .....	148
9.2.3 Adding an Excel Output Function to the Page .....	150
<b>9.3 Adding Sections to the Contract Browse Screen .....</b>	<b>152</b>
9.3.1 Adding a Body Span Section .....	152
9.3.2 Adding a Body Detail Section .....	153
9.3.3 Adding a Context Section .....	154
<b>9.4 Adding Filters to the Contract Browse Screen .....</b>	<b>156</b>
9.4.1 Adding a Page Browse Filter .....	156
9.4.2 Using Populates with a Browse Filter .....	156
<b>9.5 Adding a Body Selector .....</b>	<b>158</b>
9.5.1 Adding a Body Selector Section .....	158
9.5.2 Using the Body Selector value .....	158

<b>10 Creating A Simple listing Report (supplier)</b>	.....	<b>160</b>
10.1 Create the Function and Menu .....		160
10.2 Set up the Page .....		161
10.3 Add the Page Section Forms .....		161
10.4 6.4 Add the Fields .....		162
10.5 Create the report section .....		162
10.6 Sorting the Report .....		164
10.7 Create the Report Form .....		164
10.8 Add the Fields .....		165
10.9 Running the Report .....		166
<b>11 Creating a Contract Listing Report</b>	.....	<b>168</b>
11.1 Create the Function .....		168
11.2 Create the Page .....		169
11.3 Add the Page Section Forms .....		169
11.4 Add the Page Fields .....		170
11.5 Create the report section .....		170
11.6 Sorting the Report .....		171
11.7 Create the Report Forms .....		172
11.8 Add the Fields .....		173
11.9 Add Totals to the Report .....		174
11.10 Run the report .....		174
11.11 Summarising Reports .....		176
11.11.1 Copy the Report from BI0004 .....		176
11.11.2 Select the Sort Level .....		176
11.11.3 Select the correct Header and Footer Forms .....		176
11.11.4 Amend the fields on the report .....		177
11.12 Calculated Fields .....		178
11.12.1 Copy the Report created in BI0004 .....		178
11.12.2 Add Calculated Fields .....		178
11.13 Using Report Selection Criteria in the Report Query .....		181
11.13.1 Amend the Page Fields .....		181
11.13.2 Modify the Report Query .....		182
11.14 Using Report Selection Criteria to Pass Values to RO Fields .....		182
11.14.1 Create the Function .....		182
11.14.2 Add the Page Fields .....		183
11.14.3 Modify the Report Fields .....		183

## 1 Database Overview

A database is a collection of records stored in a computer in a systematic way, such that a computer program can consult it to answer questions. For better retrieval and sorting, each record is usually organized as a set of data elements (facts). The items retrieved in answer to queries become information that can be used to make decisions. The computer program used to manage and query a database is known as a database management system (DBMS). The properties and design of database systems are included in the study of information science.

The central concept of a database is that of a collection of records, or pieces of knowledge. Typically, for a given database, there is a structural description of the type of facts held in that database: this description is known as a schema. (You can view the coins schema using the database enquiry screen in coins OA - see accompanying documentation).

The schema describes the objects that are represented in the database, and the relationships among them. There are a number of different ways of organizing a schema, that is, of modelling the database structure: these are known as database models (or data models).

## 1.1 Relational Database Model

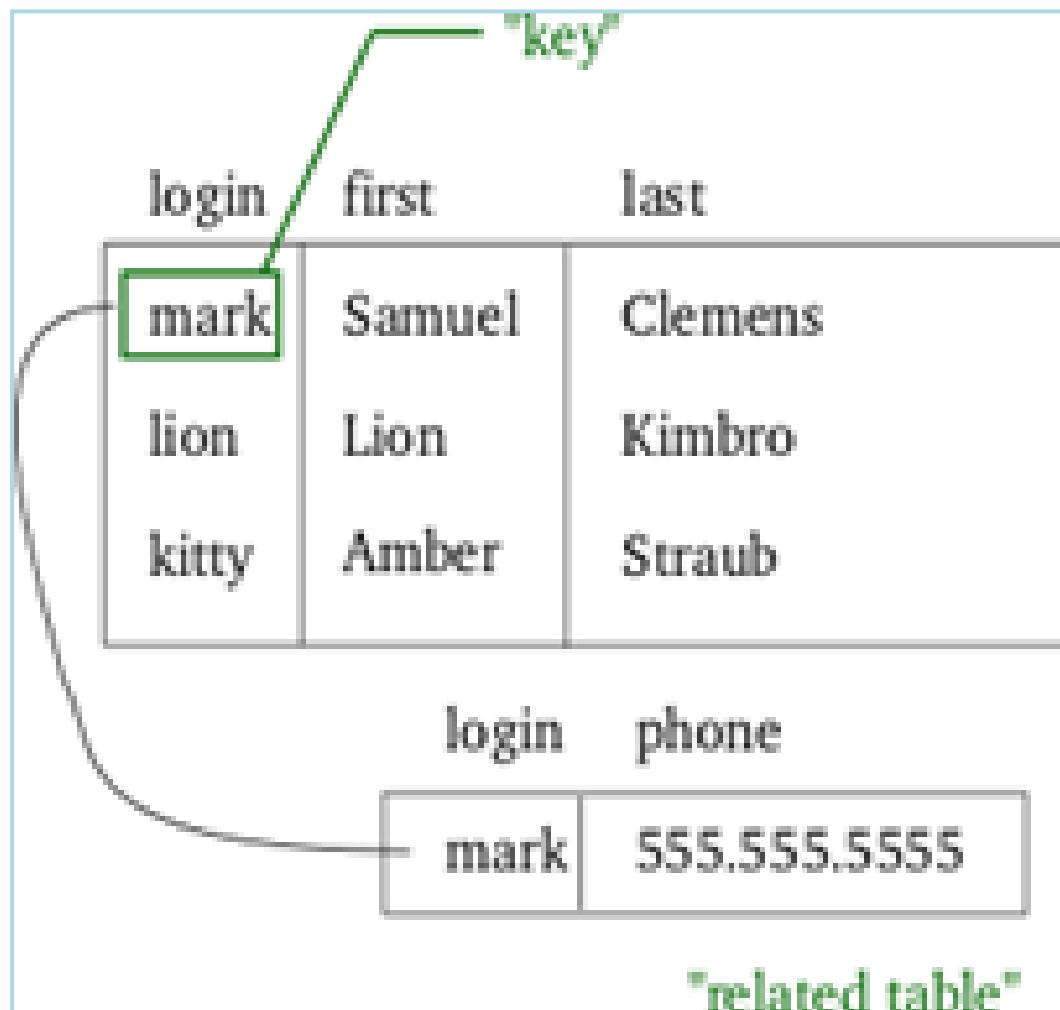
The model in most common use today is the relational model, which represents all information in the form of multiple related tables each consisting of rows and columns.

A relational database is a database based on the relational model. Strictly speaking the term refers to a specific collection of data but it is invariably employed together with the software used to manage that collection of data. That software is more correctly called a relational database management system, or RDBMS.

An important feature of relational systems is that a single database can be spread across several tables. This differs from flat-file databases, in which each database is self-contained in a single table.

On Relational Database Model represents relationships by the use of values common to more than one table

In the relational model some bit of information was used as a "key", uniquely defining a particular record. When information was being collected about a user, information stored in the optional (or related) tables would be found by searching for this key. For instance, if the login name of a user is unique, addresses and phone numbers for that user would be recorded with the login name as its key.



## 1.2 Tables and Modules

The table structure of the coins database has been designed to be organized with a direct relation to the business processes and modules of the system. One of the main attributes of this design is the naming convention used on the schema.

Coins have aimed to use a naming convention that would make it easy to identify which tables are used by which module. On most cases the first two letters of the table will point to the module code of the system.

Examples:

ap\_Purchase Ledger  
ar\_Sales Ledger  
cb\_Cash Book  
ci\_Central Repository



The above is the standard convention but there are tables in the database which do not conform to this convention of which users should be aware - main examples are payroll and all system information which either do not use underscores but use hyphens or do not break the table names. The Database Enquiry contains all the information as required

The table name will also contain a descriptive element, for example ap\_invoice is the Purchase Ledger invoice table and ap\_invdist the table which contains its associated distribution records.

Each table has a three letter ID, this is used as reference throughout coins - and is often used as the prefix of a field name. In the example of ap\_invoice this ID is ain, therefore the field name for the Purchase Ledger Invoice balance is ain\_balance.

To reference a field the syntax is:

{tablename}.{fieldname}

For example :

ap\_invoice.ain\_balance

### 1.2.1 Database Structure

The coins database is based on various levels, the top level being the Central Repository. Information held in the Central Repository is not COINS Company specific and is available across the system.

The main pieces of information held in the Central Repository are :

CI Company Information  
PI Project Information  
TI Technical Information  
PP People Information

In addition to the Central Repository, system information such as Users, Functions, Printers are also held at this top level.

**SY** System

**MS** Menus and Functions

**PM** Print Manager

**XL** Translations and Language

**IB** Insurances and Bonds

**MK** Marketing

Most data in the Coins database is actually held at COINS Company level. Even though only one company may be used the company details will need to be used to access the data. In each instance the company number is held on each table in the kco (current logged in company) field.

There are a set of tables which relate directly to company information - configuration table etc, in addition to generic tables such as Batches - these are held in the co module.

**CO** Company

### 1.2.2 Company Specific Modules

**GL** General Ledger

**JC** Contract Status Ledger (Job Costing)

**CB** Cash Book

**AP** Purchase Ledger (Accounts Payable)

**AR** Sales Ledger (Accounts Receivable)

**SC** SubContract Ledger

### 1.2.3 Process Specific Modules :

**CS** Contract Sales

**FM** Facilities Management

**SM** Valuations (Site Manager)

**SW** Small Works

**House Builders Modules**

**BQ** Bill of Quantities

**HS** House Sales

**LA** Land Appraisal

**VP** Valuations and Payments

**WF** Workflow

**Payroll and HR**

**CR** Credit Control

**HR** Human Resources

**EX** Expenses

**PR** Payroll

**Plant, Assets and Stock**

**CM** Components

**FA** Fixed Assets

**FL** Fleet

**PC** Plant Control

**SO** Sales Orders

**ST** Stock

## Procurement

**PO** Procurement

Other Modules :

**BP** Professional Billing

**CR** Credit Control

**DC/DM** Document Control / Management

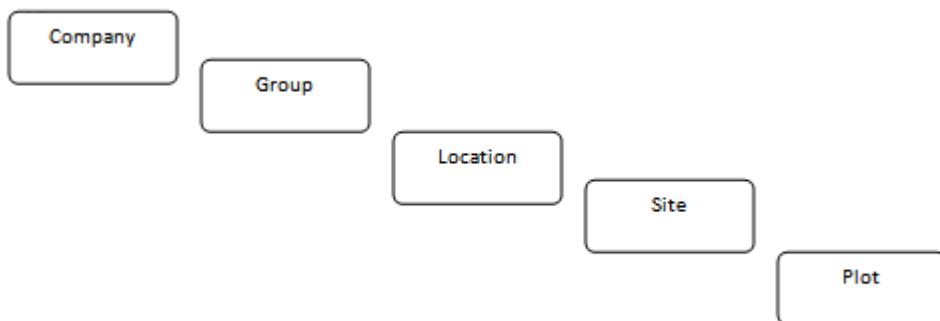
**IB** Insurance and Bonds

## 1.3 Summary and Detail Tables

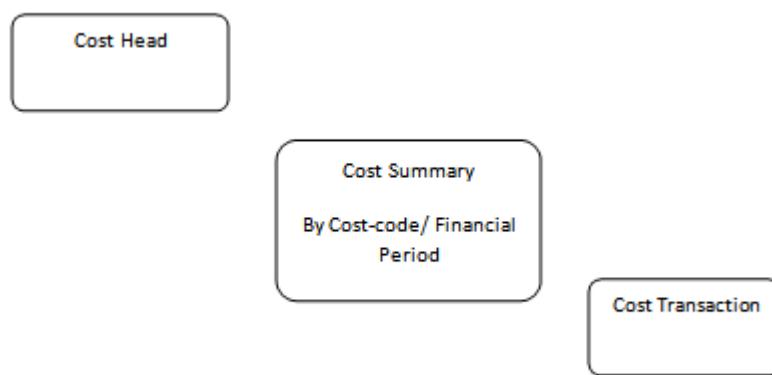
Coins has designed its database to match the processes of the industry and therefore the tables mirror the process components. Each of the modules will cover different process and within each module we can find different sub-processes. The organizational structure of the company is also related to the structure of the tables.

Each of these sub-processes are also divided in tables that will hold information down to the lowest level of detail and there will also be tables that will summarize that information at different levels (depending to the process these could be dates, codes etc).

Company Organizational Structure & Contract Structure Example (House Builders):



Cost Transactions Structure Example (House Builders):



## 1.4 Open Items

In addition to the Summary and detail information, COINS also has specific tables for open transactions in the database. This is to enhance performance when maintain and reporting on current data.

For example every PL Invoice that has not been paid, or has been part paid will have an associate record in the PL Invoice Open Item table. Once an invoice has been fully paid the open item record is deleted.

It is therefore recommended that when enquiring or reporting on open items that it is the open item record which is used as the basis of the query.

For example :

```
FOR EACH ap_invopen WHERE ap_invopen.kco = {kco},  
  EACH ap_invoice OF ap_invopen
```

Each of the tables which contain transactional data will have an associate open item table.

## 1.5 Indexes

Databases can take advantage of indexing to increase their speed (Dataset retrieval using queries). The most common kind of index is a sorted list of the contents of some particular table column, with pointers to the row associated with the value. An index allows a set of table rows matching some criterion to be located quickly.

The order that columns are listed in the index definition is important. It is possible to retrieve a set of row identifiers using only the first indexed columns. However, it is not possible or efficient (on most databases) to retrieve the set of row identifiers using only the second or greater indexed column.

For example, imagine a phone book that is organized by city first, then by last name, and then by first name. If given the city, you can easily extract the list of all phone numbers for that city. However, in this phone book it would be very tedious to find all the phone numbers for a given last name. You would have to look within each city's section for the entries with that last name.

Each coins table has one or more Indices. An Index is built up of several fields in a record which in combination will assist the query in narrowing down the number of records which will be read to determine which meet the query requirements.

Each Table has a Primary (the index used by default unless you determine otherwise in your query) and a Unique key (the combination of these fields in a single record is always unique).

However coins OA will use the most appropriate index for your query.

An example of how an index would work is to use the Current Logged in Company (kco) in addition to Contract Number (job\_num) to search for a particular Contract. Another would be to use Current Logged in Company (kco) plus Order Type (tip\_type) where you would query only where tip\_type = "TRADE", the query would immediately know only to search through Subcontract Orders to find orders which matched the other criteria rather than search every single order.

## 1.6 Record Service Procedures

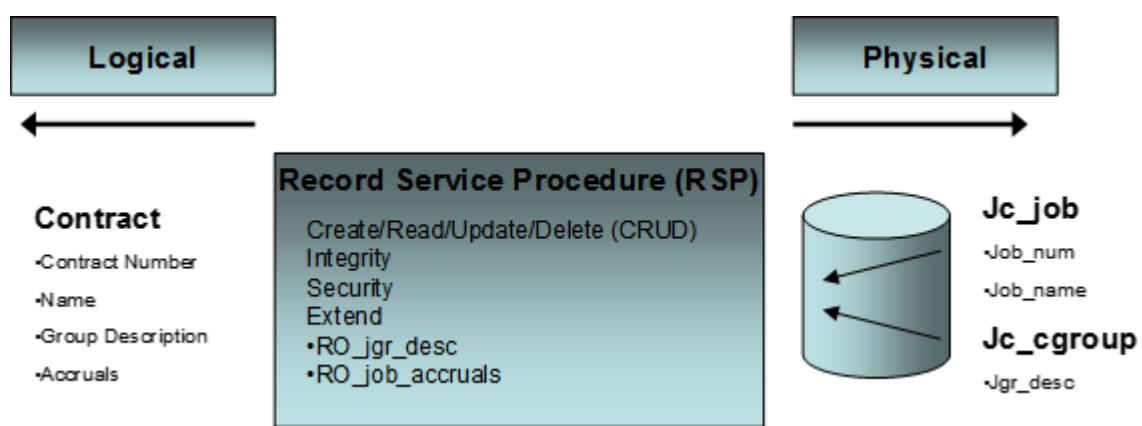
Every table in the coins database has a RSP (Record Service Procedure). Each of these RSP's provides the OA Reporting tool with the business logic required to extract the appropriate data from the database.

In the Database Enquiry you can see the RSP under the Table Code and its Label.  
The RSP's have a naming convention -

{table-ID}-rsp.p.

Where the Table ID is as shown in the Database Enquiry (you may also hear this referred to as the Table Acronym or TLA).

RSP's control, amongst other things, the basic table update functions for that table. Each RSP has a common set of methods that define standard behaviour for the object. They control record scope and locking, security, default values on creation, data integrity rules etc.



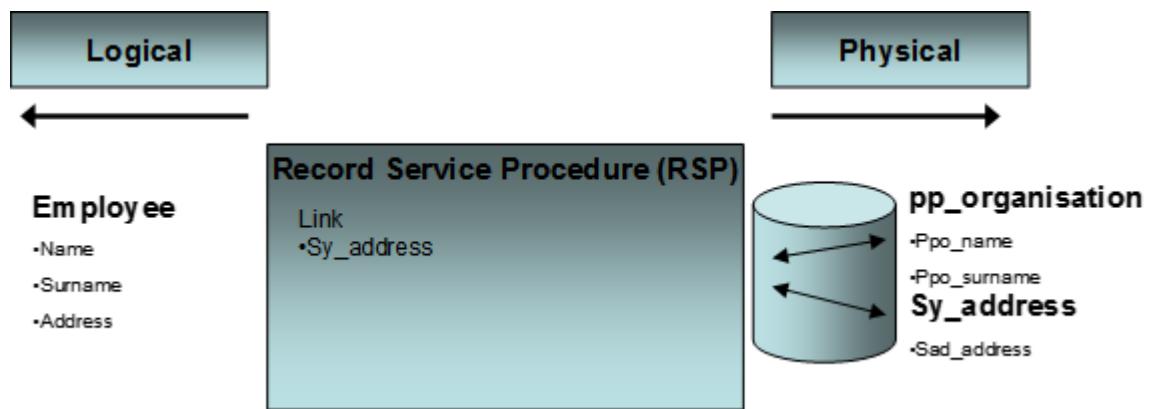
The RSP can also extend the database table to produce a logical view of data for the business logic to work with. A by-product of this is that we are to provide an XML field on every record in the database in which the user can configure their own extended

fields for use in the presentation layer. This enables client-specific fields to be added to screens and included on reports. The RSP handles the translation of the data from the logical record buffer to the physical fields in the physical database.

The RSP is also able to de-normalise the database for the purposes of logical data access. For example, the contract record (jc\_job) in the database is linked to many other tables including the contract group table (jc\_cgroup). To show the group description on a screen or report it might be expected that the interface designer would have to build a query to link the contract record with the appropriate group record and then display the group description from that record. The RSP extends (for read-only purposes) the table and de-normalises the data and makes the contract group description available as just another field on the contract table. The data remains physically in the jc\_cgroup table in the database, but to the business logic and presentation layers it is shown in a more logical place which is on the contract record.

The same principle is applied to more complex calculation fields. For example, the value of accruals on a contract is a highly complex calculation involving many tables (purchase orders, order lines, goods received notes, etc). Again, this data is made available in the logic buffer in the RSP so that to the interface designer it is just another field on the contract record. They are simply able to paste, say, the contract number, contract name, group description, and accrual value on to a page without any need to know where each of the bits of data is coming from.

The RSP is also able to simplify the database for updates.



There are instances where common database tables are reused in many areas of the application. An example is the address table sy\_address. This holds the address details for an employee record in the HR system. In this instance the RSP is able to link the two records together, presenting a single logical table to the business logic and presentation layers. The updates are performed on this logical buffer in the RSP and it is only the RSP that knows that the data is split into two separate records when written back to the database.

The RSP also has an audit layer so that as the logical record is committed back to the database, changes on the logical table can be recorded in the audit records. Auditing can be performed as part of the managed data source through the use of triggers, but in this instance the audit records created are an audit of the physical data and it is much more difficult to reconstruct the separate physical table and field audit records in to a logical view of the record at a later stage.

The RSP controls all data access through to the database. COINS can insert bespoke trigger code in that RSPs that can act on data as it is committed back to the database. For example, this can be used to push changes in business data in COINS out to a

data warehouse application by creating XML messages from COINS that are then consumed by a data warehouse load interface. In this way it is possible to keep a data warehouse up to date with live business data.

## 1.7 Read Only Fields

In addition to the standard tables and fields in the coins database, Open Architecture also uses the RSP's to provide access to certain calculated and non standard fields. These are known as "RO" or Read Only fields and are also fully documented in the Database Enquiry.

RO_avm_RoutingCode	International Bank Routing Code	character	X(5)	RO
RO_avm_sclab_details	Supplier Tax Details	character	X(60)	RO  If the supplier provides subcontract labour (avm_sclab=TRUE) then this is a composite description of the CIS details for the subcontractor.
RO_avm_Tide_Age	Aging: Tide Age	character	X(10)	RO  Tide "Age"
RO_avm_Tide_Amt	Aging: Tide Amount	character	X(10)	RO  Tide "Amount"
RO_avm_Tide_Key	Aging: Tide Key	character	X(10)	RO  Tide "Key"
RO_avm_Tide_QryAmt	Aging: Tide Query Amount	character	X(12)	RO  Tide "Query Amount"
RO_avm_Total	Outstanding Total	decimal	.X,X>,.X>,.X>9.99	RO  Total outstanding invoice/payments for the supplier. Sum of the open invoices and payments in account currency
RO_avm_unleared	Uncleared To Pay	decimal	.X,X>,.X>,.X>9.99	RO  The total of invoices that are due to be paid and are held.
RO_avm_offset +>Workflow Variable Value	Active Workflow Variable Value	character	X(30)	RO
RO_BalD[2]	Discount Lost	decimal	.X,X>,.X>,.X>9.99	RO  Discount lost in the year taken from ap_vendbal for the current GL year, Array 1=current GL year, 2=next year.
RO_BalT[2]	Discount Taken	decimal	.X,X>,.X>,.X>9.99	RO  Discount taken in the year taken from ap_vendbal for the current GL year, Array 1=current GL year, 2=next year.
RO_BalPay[2]	Payments	decimal	.X,X>,.X>,.X>9.99	RO  Payments made in the year taken from ap_vendbal for the current GL year, Array 1=current GL year, 2=next year.
RO_BalPv[2]	Purchases	decimal	.X,X>,.X>,.X>9.99	RO  Purchases (Invoices) in the year taken from ap_vendbal for the current GL year, Array 1=current GL year, 2=next year.
RO_BalYear[2]	Financial Year	character	X(4)	RO  Current GL year: Array 1=current GL year, 2=next year.
RO_coc_store +>[StoreType]>>[SiteId]> +>PeriodType>>[DateFrom]>> +>ContractNumbers>> +>ExcludeCostHd	Company Data Store	decimal	.X,X>,.X>,.X>9.99	RO  Standard store options. See co_config RW_coc_store
RO_contractuals_sum +>ContractNumber>> +>ExcludeCostHd	Contract Accounts by Contract	decimal	.X,X>,.X>,.X>9.99	RO  The contract accounts for the supplier. +>The contract for which accounts are required. +>CAN-DO list of cost heads to exclude
RO_contractuals_avm +>ContractNumber>> +>ExcludeCostHd>> +>PeriodOffset>> +>FromDate>>	Contract Costs by Contract	decimal	.X,X>,.X>,.X>9.99	RO  The contract costs for the supplier. +>The contract for which accounts are required. +>CAN-DO list of cost heads to exclude PeriodType, Offset, Date as for co_config RW_coc_store
RO_coc_encode	Reporting Currency	character	X(4)	RO   If RS_Jean is set to YES then the company base currency otherwise the supplier account currency
RO_Flag_Notes	File for Red/Clear Flag or no Flag picture	character	X(60)	RO  Whether notes have been entered against this company. If notes have been entered against the company, this field shows a flag icon:  . If at least one note is marked by a Red Flag, the flag will appear here as a Red Flag  . To see the notes, click the flag icon. See Company Notes
RO_glo_1	Turnover Q1	decimal	.X,X>,.X>,.X>9.99	RO  Turnover in Q1 for the year held in variable RS_avb_year
RO_glo_2	Turnover Q2	decimal	.X,X>,.X>,.X>9.99	RO  Turnover in Q2 for the year held in variable RS_avb_year
RO_glo_3	Turnover Q3	decimal	.X,X>,.X>,.X>9.99	RO  Turnover in Q3 for the year held in variable RS_avb_year
RO_glo_4	Turnover Q4	decimal	.X,X>,.X>,.X>9.99	RO  Turnover in Q4 for the year held in variable RS_avb_year
RO_ng_changed	Record Changed	character	X(24)	RO  The date and time of the most recent change to the record, and the user ID of the user who changed it.
RO_ng_created	Record Created	character	X(17)	RO  The last changed date, time and user id.
RO_ncn_code	Subcontractor Account	character	X(10)	RO  The nc_mand record exists for this supplier then this shows the subcontractor account code otherwise blank.
RO_supplier	Group Supplier	character	X(5)	RO  The company the plant item is held from.
RO_wfbm_branch	Bank Branch (Workflow)	character	X(30)	RO  If variable RS_cons = YES then the group account (avm_gracc) is non blank then the group account otherwise the supplier account code.
RO_wfbm_name	International Bank Name (Workflow)	character	X(30)	RO
RW_avm_an_ndx	Notes	character		RW  Any notes pertaining to the supplier.
RW_CVRSite +>ContractNumber>>				

Although these fields have certain restrictions, they are incredibly powerful when used in enquiries and reports.

In most instances RO\_fields will provide information from related tables to the main queried table - for example summary cost information at Contract Level, or descriptions from an associated Lookup Table without the Page or Report designer having to query and access many tables from the coins database.

Many of the calculated fields reflect similar fields to the coins + Configurable Reporter, such as Accruals, Costs and Revenue fields. These fields can then be passed parameters to enhance the information returned to a report. Typically these fields can be limited by dates, values and financial periods as well simply parameters such as "TD" for a To Date value.

In the Database Enquiry RO\_fields are shown in a format as the example below. Any parameters immediately after the caret are mandatory; each parameter is then separated by a pipe. Any parameters which are encapsulated in square brackets are optional.

RO\_ContractCosts^<PeriodType>[]<PeriodOffset>[]<FDate>[]|PhaseMasks  
[|CostcodeMasks[|CategoryMasks[|AnalysisMask]]]]]

## 2 OA Query Language

COINS OA uses a simplified version of the Progress 4GL query language in combination with RSP's (Record Service Procedures) to extract the data for reports and enquiries (for further information on RSP's- see later in this guide).

COINS OA uses the query to decide which records are accessed from the coins database from the database. In response to a query, the database returns a result set, which is just a list of rows containing the answers. The Page/Report Design will determine which fields from these records are displayed (either on screen or in a report).

The simplest query is just to return all the rows from a table, but more often, the rows are filtered in some way to return just the answer wanted.

The flexibility of relational databases allows programmers to write queries that were not anticipated by the database designers. As a result, relational databases can be used by multiple applications in ways the original designers did not foresee, which is especially important for databases that might be used for decades. This has made the idea and implementation of relational databases very popular with businesses.

## 2.1 FOR EACH

To begin a query in OA, the first statement must begin FOR EACH followed by a table name.

Example Query on the coins database to retrieve all contracts (jc\_job)

FOR EACH jc\_job

1. jc\_job is the name of the table in the COINS database
2. The FOR EACH statement starts a block of code that iterates once for each contract record (hence the syntax FOR EACH)

## 2.2 WHERE

Simply specifying the table with a FOR EACH statement in a query is okay, assuming we want every record from the selected table, but in practice we would normally want to restrict the number of records returned in some way. In COINS, transactional data is held at company level.

Even though you may only have one company in your organisation, the data is still recorded with a company identifier. COINS uses the field kco to identify the company number.

Most queries will need to specify the kco values to ensure that the records returned relate specifically to the company you are reporting on.

The WHERE statement is used to add a constraint to the query and may refer to a constant, filed name, variable name or expression whose value you want to use to select records

Example Query on the coins database to retrieve all contracts (jc\_job) that belong to company 1:

FOR EACH jc\_job WHERE kco = 1

In the example above we have used '=' as the comparison operator. There are a number of others than may be used with the WHERE statement. These are listed in the table below:

Keyword	Symbol	Description
EQ	=	Equal to
NE	<>	Not equal to
GT	>	Greater than
LT	<	Less than
GE	>=	Greater than or equal to
LE	<=	Less than or equal to
BEGINS	Not applicable	A character value that begins with this substring.
MATCHES	Not applicable	A character value that matches this substring, which can include wild card characters

The expression you use to the right of the MATCHES keyword can contain the wild card characters:

An asterisk (\*) represents one or more missing characters.

A period (.) represents exactly one missing character.

Keyword	Symbol	Description
CONTAINS	Not applicable	A database text field that has a special kind of index called a WORD-INDEX  The WORD-INDEX indexes all the words in a field's text strings, for all the records of the table, allowing you to locate individual words or associated words in the database records, much as you do when you use an Internet search engine to locate text in documents on the web..

The WHERE statement can be followed by any expression that identifies a subset of the data using AND/OR to join multiple tests.

Example Query on the coins database to retrieve a specific contract (field job\_num) for Company 1 from table jc\_job

```
FOR EACH jc_job WHERE jc_job.kco = 1  
AND jc_job.job_num = '123456'
```

## 2.3 Joining Tables

Often, data from multiple tables gets combined into one, by doing a join. Conceptually, this is done by taking all possible combinations of rows (the "cross-product"), and then filtering out everything except the answer.

To begin each join a comma should end the previous statement before beginning the next one. DO NOT add a comma to the end of the last statement as this will result in an error.

### 2.3.1 EACH

FOR is only used for the first table in the query, all subsequent tables must be accessed with EACH to start an iterating query that will find a single record on each pass

To establish a join, the table(s) you are adding to the query must have some relation to one or more tables already in the query.

Example Query on the coins database to retrieve all costheads (jc\_costcode) that belong to contracts (jc\_job) that belong to the logged in Company

```
FOR EACH jc_job WHERE jc_job.kco = {kco},  
EACH jc_costcode WHERE jc_costcode.kco = jc_job.kco  
AND jc_costcode.job_num = jc_job.job_num
```

If you do not use the EACH keyword for a subsequent table then you must use one of the following to obtain a single record:

### 2.3.2 FIRST

Uses the criteria in the record-phrase to find the first record in the table that meets that criterion.

Progress finds the first record before any sorting.

### 2.3.3 LAST

Uses the criteria in the record-phrase to find the last record in the table that meets that criterion.

Progress finds the last record before sorting.

The FIRST and LAST keywords are especially useful when you are sorting records in a table in which you want to display information. Often, several related records exist in a related table, but you only want to display the first or last related record from that table in the sort. You can use FIRST or LAST in these cases.

### 2.3.4 OF

Some of the tables in the COINS database share a relationship based on common field names between record and table that also participate in a UNIQUE index for either record or table. All OF relationships within the coins database are detailed in the database enquiry and appear for each table in the form similar to:

From	To	Join To	Documentation	Code
1	*	ap_invdat	ap_invdat OF jc_job	ap_invdat.kco=jc_job.kco AND ap_invdat.job_num=jc_job.job_num
*	1	ar_cussum	Links to the Customer Summary record for the customer of this Contract.	ar_cussum.kco=jc_job.kco AND ar_cussum.cdm_num=jc_job.cdm_num

Where such a relationship exists, the OF statement may be used to relate one table to another. So in our earlier example we used the query:

```
FOR EACH jc_job WHERE jc_job.kco = {kco},
  EACH jc_costcode WHERE jc_costcode.kco = jc_job.kco
    AND jc_costcode.job_num = jc_job.job_num
```

An OF relationship exists between jc\_job and jc\_costcode as can be seen in the database enquiry for jc\_job:

1	*	jc_costcode	jc_costcode OF jc_job	jc_costcode.kco=jc_job.kco AND jc_costcode.job_num=jc_job.job_num
---	---	-------------	-----------------------	--

So we can re-write this query as:

```
FOR EACH jc_job WHERE jc_job.kco = {kco},
  EACH jc_costcode OF jc_job
```

## 2.4 Curly Braces

The functionality of {}'s is to specify a place holder in fields and calculations into which a value can be passed.. When using {}'s around a field the use of quotes is required if the field is a character field. The use of double or single quotes is acceptable.



The only thing to be aware of is that when using '{field}' replacement on a character field is that if the information within the field could contain an apostrophe (for example- J O'Connor) then the apostrophe would cause close to the single quote and you will get a symbol not found(Connor) Error. To overcome this error the use of double quotes "{field}" is the answer.

The use of {}'s in calculations is possible on all field values **except within the DataSets and the calculate conditions on a report**. In these instances it is necessary to always qualify out the field with the table name.

```
{RO_ContractCosts^TD|0|{RS_glp_fdate__2}}  
would be written as:  
jc_job.RO_ContractCosts^TD|0|{RS_glp_fdate__2}.
```



The use of the table name is allowed in all calculations but whereas in most instances the formatting of the result is suppressed, within the calculate condition it is not and therefore the comma in a result of a figure in excess of 1,000 may result in an error in syntax in a calculation. (NB. Please note that the replacement on parameters of an RO field is still acceptable).

Within the OA reporter/screens we use curly braces {} as a method to pass values to a query or a report or a page. Enclosed within the curly braces you specify the commands, RS\_fields, or other data you need to communicate across or within objects. {kco} is a common usage, and is used to place the current logged in company number into the query.

The next example gets information from jc\_job and inherits the Company Number from the system, retrieving the company number the user is logged into.

```
FOR EACH jc_job WHERE jc_job.kco = {kco}
```

## 3 OA and BI Utilities

To assist developers in creating and testing OA Queries and calculations, a number of utilities are available with the OA & BI Reporting Module.

The commonly used utilities are:

- Database Enquiry
- Query Editor
- Calculation Editor
- Object Enquiry

### 3.1 Database Enquiry

To assist users in Open Architecture to understand and exploit the coins database schema when creating enquiries and reports a powerful tool has been developed which provides information on all coins tables, fields and formats.

The Database Enquiry provides detailed information on the structure of every table in the coins database.

Table	Description	DB	ID
abi_dfa	ABI Data dfa file	coins	abi
abi_dfc	ABI Data dfc file	coins	abc
abi_ddf	ABI Data ddf file	coins	abd
abi_flds	ABI field mapping	coins	afld
abi_load		coins	abl
abi_tran	ABI transaction log	coins	atr
abi_tran_det	ABI transfer details	coins	atd
ac_scheme	Scheme	coins	acc
ac_schtyps	Scheme Types	coins	ach
al_asset	Asset Repository	coins	als
ap_check	P/I Payment	coins	acs
ap_chooseon	P/I Open Payment	coins	aco
ap_cinvline	Capital Invoice Line	coins	aci
ap_config	Configuration File	coins	act
ap_invdit	Invoice Distribution File	coins	aid
ap_invhist	Invoice History	coins	ahst
ap_invme	P/I Invoice Lines	coins	ail
ap_invoic	P/I Invoice	coins	ain
ap_invoiem	P/I Open Invoice	coins	aop
ap_invoisy	Invoice queries	coins	aiq
ap_itdist	P/I IntraSite Distribution Line	coins	api
ap_paz	P/I Payment Allocation	coins	apy
ap_pcards	PCards	coins	app
ap_pcardline	PCard Transaction Lines	coins	apd
ap_pcardtran	PCard Transactions	coins	apt

Filters at the top of the page allow you to search for specific tables, fields and table types.

Table	Description	DB	ID
	Description	Type:	Field:

It also provides field information, descriptions, formats as well as documentation supplied by the coins Development Team to support users in creating their queries. This includes all calculated fields available via the RSP's (Record Service Procedures).

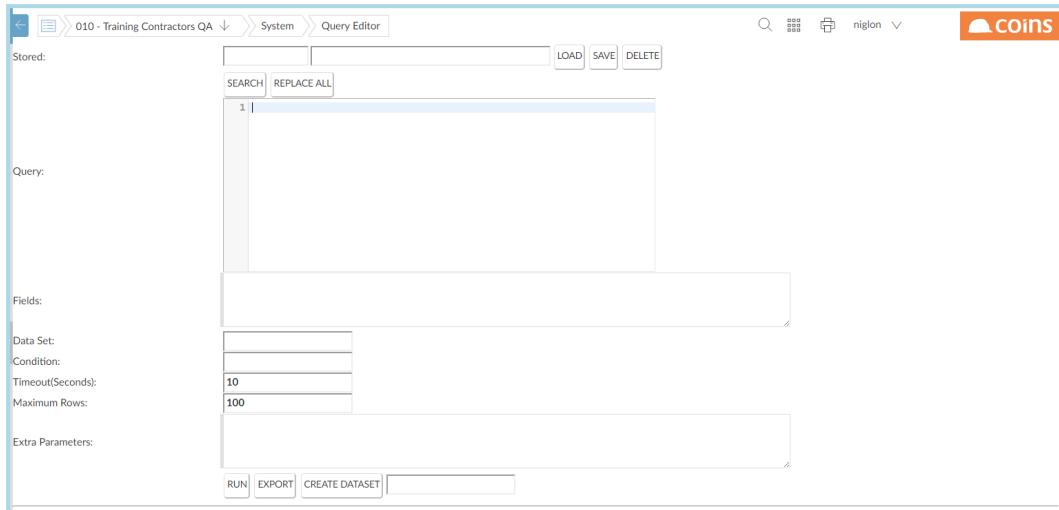
Table: PL_Invoice					Number of Rows: 18
Index	Primary	Unique	Field	Documentation	
ain_key	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	lco+avm_num+ain_inv+		
ain_key1		<input checked="" type="checkbox"/>	lco+avm_num+ain_inv+		
ain_key10		<input checked="" type="checkbox"/>	lco+avm_num+ain_inv+		
ain_key2		<input checked="" type="checkbox"/>	lco+avm_num+ain_inv+		
ain_key3		<input checked="" type="checkbox"/>	lco+avm_num+ain_supref+		
ain_key4		<input checked="" type="checkbox"/>	lco+avm_num+ain_supref+		
ain_key5		<input checked="" type="checkbox"/>	lco+avm_num+ain_supref+		
ain_key6		<input checked="" type="checkbox"/>	lco+avm_num+ain_supref+		
ain_key7		<input checked="" type="checkbox"/>	lco+avm_num+ain_supref+		
ain_key8		<input checked="" type="checkbox"/>	lco+avm_num+ain_supref+		
ain_key9		<input checked="" type="checkbox"/>	lco+avm_num+ain_supref+		
Field	Label	Data Type	Format	Documentation	
ain_achcur	Supplier Alternative Currency	logical	yes/no	RW <small>Indicates whether alternative currencies i.e. different to the account currency (ap_vendor.cur_code) are allowed to be entered on the account.</small>	
ain_amount	Gross Amount	decimal	>>>>>9.99	DB <small>The gross amount of the invoice.</small>	
ain_anal	Analysis (Contract or Dept)	character	x[8]	DB <small>The main analysis code for the invoice in base currency. Equivalent to ain_cur_gross[2].</small>	
ain_apacct	P/L Control	character	X[19]	DB <small>The main contract or department to which the invoice is assigned. Depends on ain_entry.</small>	
ain_atotaxpay	ATO Reporting	character	x[4]	DB <small>The option to credit the creditor's control account posted to when the invoice was committed.</small> <small>The option which allows to select behaviour for specific invoice in order to providing correct data to ATO (Australian Taxation Office). It can take following options:</small> <ul style="list-style-type: none"><li>- Include in the ATO Taxable Payments report</li><li>- Exclude from the ATO Taxable Payments report</li><li>- Blank Default to the Supplier setting or if not set default to the Company workbench setting for this company (check PL/USECIS parameter).</li></ul>	

In addition the Database Enquiry will provide the links available to associated tables and also provide the syntax required to build a query to create these links within Page and Report Designer.

From	To	Join To	Documentation	Code	
1	*	ap_cinvline	ap_cinvline OF ap_invoice	ap_cinvline.kco-ap.invoice.kco AND ap_cinvline.invnum=ap.invoice.invnum AND ap_cinvline.invdate=ap.invoice.invdate	
1	*	ap_invlist	ap_invlist OF ap_invoice	ap.invlist.kco-ap.invoice.kco AND ap.invlist.invnum=ap.invoice.invnum AND ap.invlist.invdate=ap.invoice.invdate	
1	*	ap_invline	ap_invline OF ap_invoice	ap.invline.kco-ap.invoice.kco AND ap.invline.invnum=ap.invoice.invnum AND ap.invline.invdate=ap.invoice.invdate	
1	1	ap_invoisen	ap_invoisen OF ap_invoice	ap.invoisen.kco-ap.invoice.kco AND ap.invoisen.invnum=ap.invoice.invnum AND ap.invoisen.invdate=ap.invoice.invdate	
1	1	ap_inquiry	ap.inquiry OF ap_invoice	ap.inquiry.kco-ap.invoice.kco AND ap.inquiry.invnum=ap.invoice.invnum AND ap.inquiry.invdate=ap.invoice.invdate	
1	*	ap_invstat	ap_invstat OF ap_invoice	ap.invstat.kco-ap.invoice.kco AND ap.invstat.invnum=ap.invoice.invnum AND ap.invstat.invdate=ap.invoice.invdate	
*	1	ap_vendor	ap.vendor OF ap_invoice	ap.vendor.kco-ap.invoice.kco AND ap.vendor.vendorid=ap.invoice.vendorid AND ap.vendor.vendorname=ap.invoice.vendorname	
*	1	ap_vendoun	ap.vendoun OF ap_invoice	ap.vendoun.kco-ap.invoice.kco AND ap.vendoun.vendorid=ap.invoice.vendorid AND ap.vendoun.vendorname=ap.invoice.vendorname	
*	1	ar_invoice	ar.invoice OF ap_invoice	ar.invoice.kco-ap.invoice.kco AND ar.invoice.invoiceid=ap.invoice.invoiceid AND ar.invoice.invoicecobnum=ap.invoice.invoicecobnum	
*	1	cb_tdet	cb.tdet OF ap_invoice	cb.tdet.kco-ap.invoice.kco AND cb.tdet.invoiceid=ap.invoice.invoiceid AND cb.tdet.invoicecobnum=ap.invoice.invoicecobnum	
*	1	cb_topen	cb.topen OF ap_invoice	cb.topen.kco-ap.invoice.kco AND cb.topen.invoiceid=ap.invoice.invoiceid AND cb.topen.invoicecobnum=ap.invoice.invoicecobnum	
1	co_currency	co.currency OF ap_invoice	co.currency.kco-ap.invoice.kco AND co.currency.curcode=ap.invoice.cur_code	co.currency.curcode=ap.invoice.cur_code co.curcode.kco-ap.invoice.kco	
*	1	co_vat	co.vat OF ap_invoice	co.vat.kco-ap.invoice.kco AND co.vat.invoiceid=ap.invoice.invoiceid AND co.vat.invoicevatcode=ap.invoice.vat_code	co.vat.invoiceid=ap.invoice.invoiceid co.invoicevatcode.kco-ap.invoice.kco
*	1	cs_certificate	cs.certificate OF ap_invoice	cs.certificate.kco-ap.invoice.kco AND cs.certificate.certnum=ap.invoice.certnum AND cs.certificate.certdate=ap.invoice.certdate	cs.certnum.kco-ap.invoice.kco cs.certdate.kco-ap.invoice.kco
*	1	fa_invoice	fa.invoice OF ap_invoice	fa.invoice.kco-ap.invoice.kco AND fa.invoice.invoiceid=ap.invoice.invoiceid AND fa.invoice.invoicecobnum=ap.invoice.invoicecobnum	fa.invoice.invoiceid=ap.invoice.invoiceid fa.invoice.invoicecobnum=ap.invoice.invoicecobnum
*	1	hs_devsnum	hs.devsnum OF ap_invoice	hs.devsnum.kco-ap.invoice.kco AND hs.devsnum.jobnum=ap.invoice.jobnum AND hs.devsnum.phase=ap.invoice.phase	hs.devsnum.jobnum=ap.invoice.jobnum hs.phase.kco-ap.invoice.kco
*	1	jc_sb	jc.job OF ap_invoice	jc.job.kco-ap.invoice.kco AND jc.job.invoiceid=ap.invoice.invoiceid AND jc.job.jobnum=ap.invoice.job_num	jc.job.invoiceid=ap.invoice.invoiceid jc.jobnum.kco-ap.invoice.kco
*	1	jc_lobnum	jc.lobnum OF ap_invoice	jc.lobnum.kco-ap.invoice.kco AND jc.lobnum.jobnum=ap.invoice.job_num	jc.lobnum.jobnum=ap.invoice.job_num
*	1	jc_phase	jc.phase OF ap_invoice	jc.phase.kco-ap.invoice.kco AND jc.phase.jobnum=ap.invoice.job_num AND jc.phase.phase=ap.invoice.phase	jc.phase.jobnum=ap.invoice.job_num jc.phase.phase=kco-ap.invoice.kco
*	1	jc_phssnum	jc.phssnum OF ap_invoice	jc.phssnum.kco-ap.invoice.kco AND jc.phssnum.jobnum=ap.invoice.job_num AND jc.phssnum.phase=ap.invoice.phase	jc.phssnum.jobnum=ap.invoice.job_num jc.phase=kco-ap.invoice.kco

## 3.2 Query Editor

The Query Editor allows you to try out 4GL queries against the COINS database and sample the data returned. This function can be useful to test queries before being used in reports or enquiries.



To use the query editor simply enter the query, and (optionally) any fields required - space separated - and click RUN. The system will return an error if any part of the query is incorrect - and a sample of data if the query compiles OK.



If the fields section was left blank, all fields will be displayed. If field names were specified, only those fields will be shown.

D010 - Training Contractors QA																			
System > Query Editor																			
Stored:																			
SEARCH REPLACE ALL																			
1 For EACH ap_vendor WHERE kco > 10																			
Query:																			
Fields:																			
Data Set:																			
Condition:																			
Timeout(Seconds):																			
10																			
Maximum Rows:																			
100																			
Extra Parameters:																			
RUN EXPORT CREATE DATASET																			
1	10	ABB001	Abbey Glass	42 Bramall Lane	Sheffield		S25 4DL	01642 887766	1	30	0.00	1	30	N	01642 887766	N	Abbey	Abbey Glass	
2	10	ABSO004	Absolute Invoice Finance Limited	J ST JAMES HOUSE	CHARLOTTE STREET	MANCHESTER	M1 4ZQ		1	30	0.00	1	30	Y			N	ABSO	Absolute Invoice Finance Limited
3	10	AGGP001	Aggregate Supplies	22 Ashton Gates	Bristol		BS30 5SJ	01675 556644	1	30	0.00	1	30	N	01675 556643	N	Aggregate Supplies	Aggregate Supplies	
4	10	AGGR001	Aggregate Industries UK Ltd	BARDON HILL	COALVILLE	LEICESTERSHIRE	LE67 1TL	01530 511956	1	30	0.00	1	30	N	01530 815180	N	Aggregate Industries Ltd	Aggregate Industries Ltd	
5	10	AMSP001	Amsterdam Imports Limited	21 Plymouth Docks	Plymouth	Devon	PL1 XYZ		1	30	0.00	1	30	N			N	Amsterdam	Amsterdam Imports Limited
6	10	APL001	A Plant Hire	Colliers Industrial Estate	The High Street	Maidenhead	Berkshire	SL6 3ND	1	30	0.00	1	30	N			Y	A Plant Hire	A Plant Hire
7	10	B&Q001	B&Q plc	Copysol Rd.	Plymouth	Devon	PL7 4SS		1	30	0.00	1	30	N			N	B&Q Ltd	B&Q plc
8	10	BR001	Berts Bricks &	The High					*	**	**	*	**	*	**	*	**	Berts Bricks &	Berts Bricks &

Field	Description
Data Set	A Data Set definition can be entered here to display the information created in the data set (No query or fields are required for this).
Condition Field	A function that determines whether a record should be included or not. The function returns a logical value: yes to include the record, no to exclude it.
Maximum Rows	Allows the query to run faster by only displaying a maximum number of rows per query. 10.23 onwards, this defaults to 10

Field	Description
Extra Parameters (10.23 onwards)	<p>Where a dataset has been specified, this field allows entry of parameters (URL) that are needed by the dataset query.</p> <p>e.g.</p> <p>The parameterised fields are so that you don't have to 'hard code' queries in the dataset to get it to run in the query editor – particularly if there are date replacements etc with fields like {RS_glp_fdate_2}. Or another useful reason for using these parameters is so that you can test results in a efficient way for instance: if you have a query on the dataset which reads:</p> <pre>FOR EACH jc_job WHERE jc_job.kco = {kco} {jobSelect}</pre> <p>You could call the dataset from within the query editor and in the parameters say jobSelect=and jc_job.job_num = 'XXXX' (where XXXX is a vailid contract number).</p> <p>That way the dataset would run but for only contract XXXX - This is good to save time in checking the validation of fields in the dataset as you don't have to wait till the whole dataset evaluates prior to getting a response back.</p> <p>If you have more than one {} replacement in your dataset then you would separate the parameters with a &amp; symbol Eg: Dataset query might read :</p> <pre>FOR EACH jc_job WHERE jc_job.kco = {kco} {jobSelect}, EACH jc_costcode of jc_job WHERE TRUE {jccSelect}</pre> <p>You could call the dataset from within the query editor and in the parameters say jobSelect=and jc_job.job_num = 'XXXX' &amp;jccSelect= and jc_costcode.jcc_cc = 'YYYY' (where XXXX is a vailid contract number and YYYY is a valid Costcode).</p>

DB Fields defined as array in the Database Enquiry such as avm\_add[4] can be entered without the index of elements to display all elements (previously the element would need to be specified such as avm\_add\_1, avm\_add\_2 etc.)

010 - Training Contractors QA → System → Query Editor

Stored:  LOAD SAVE DELETE

SEARCH REPLACE ALL

```
1 For EACH ap_vendor WHERE kco = 10
```

Query:

Fields:

Data Set:

Condition:

Timeout(Seconds):

Maximum Rows:

Extra Parameters:

RUN EXPORT CREATE DATASET

	kco	avm_add_1	avm_add_2	avm_add_3	avm_add_4
1	10	42 Bramall Lane		Sheffield	
2	10	ST JAMES HOUSE	7 CHARLOTTE STREET	MANCHESTER	
3	10	22 Ashton Gates		Bristol	
4	10	BARDON HILL	COALVILLE	LEICESTERSHIRE	
5	10	21 Plymouth Docks	Plymouth	Devon	
6	10	Colliers Industrial Estate	The High Street	Maidenhead	Berkshire
7	10	Coypool Rd.	Plympton	Plymouth	Devon
8	10	The High Street			Surrey
9	10	Slough Trading Estate	107 - 113 Farnham Road	Slough	Berkshire
10	10	15 High Street			
11	10	12 High Row		Darlington	
12	10	Northfield Retail Park	Rotherham Rd	Parkgate	Rotherham, South Yorkshire
13	10	14 Jesmond road		Newcastle	
14	10	90	Greengairs Rd	Greengairs	Lanarkshire
15	10	45 High Street		Bromley	Kent
16	10	Station Road	Bristol		
17	10	Stairfoot Business Park	Bleachcroft Way	Barnsley	South Yorkshire
18	10	The High Street		Cambridge	Cambridgeshire

Wildcards can be used in “Fields” on Query Editor screen to mask fields

010 - Training Contractors QA → System → Query Editor

Stored:  LOAD SAVE DELETE

SEARCH REPLACE ALL

```
1 For EACH ap_vendor WHERE kco = 10
```

Query:

Fields:

Data Set:

Condition:

Timeout(Seconds):

Maximum Rows:

Extra Parameters:

RUN EXPORT CREATE DATASET

	kco	avm_add_1	avm_add_2	avm_add_3	avm_add_4	avm_badd_1	avm_badd_2	avm_badd_3	avm_badd_4	RO_avm_printadd_1	RO_avm_printadd_2	RO_avm_printadd_3	RO_avm_printadd_4	RO_avm_bprintadd_1
1	10	42 Bramall Lane		Sheffield						42 Bramall Lane	Sheffield	S25 4DL		
2	10	ST JAMES HOUSE	7 CHARLOTTE STREET	MANCHESTER						ST JAMES HOUSE	7 CHARLOTTE STREET	MANCHESTER	M1 4DZ	
3	10	22 Ashton Gates	Bristol							22 Ashton Gates	Bristol	BS30 5SJ		
4	10	BARDON HILL	COALVILLE	LEICESTERSHIRE						BARDON HILL	COALVILLE	LEICESTERSHIRE	LE67 1TL	
5	10	21 Plymouth Docks	Plymouth	Devon						21 Plymouth Docks	Plymouth	PL1 5YZ		
6	10	Colliers Industrial Estate	The High Street	Maidenhead	Berkshire					Colliers Industrial Estate	The High Street	Maidenhead	Berkshire, SL6 3ND	
7	10	Coypool Rd.	Plympton	Devon						Coypool Rd.	Plympton	PL7 4SS		
8	10	The High Street		Surrey						The High Street	Surrey	GU1 5BH		
9	10	Slough Trading Estate	107 - 113 Farnham Road	Slough	Berkshire					Slough Trading Estate	107 - 113 Farnham Road	Slough	Berkshire, SL1 4UN	
10	10	15 High								15 High Street	GU14 5LN			

### 3.2.1 Search and Replace

Clicking Search allows you to search for a character string within the Query. This is particularly useful for large queries.

```

1 for EACH co_batch WHERE co_batch.kco = 'GLJOUR' no-lock,
2 first sysuser WHERE sysuser.su-userid = co_batch.su-userid no-lock

```

If found, the string will be highlighted within the query.

```

1 for EACH co_batch WHERE co_batch.kco = 'GLJOUR' no-lock,
2 first sysuser WHERE sysuser.su-userid = co_batch.su-userid no-lock

```

Clicking Replace All will prompt for the search string to be replaced

```

1 for EACH co_batch WHERE co_batch.kco = 'GLJOUR' no-lock,
2 first sysuser WHERE sysuser.su-userid = co_batch.su-userid no-lock

```

Enter the required string and click Enter. The system will then prompt for the replacement string.

```

1 for EACH co_batch WHERE co_batch.kco = 'GLJOUR' no-lock,
2 first sysuser WHERE sysuser.su-userid = co_batch.su-userid no-lock

```

Enter the replacement string and press Enter. The system will then replace all instances of original text with the replacement string.



### 3.2.2 Exporting from the Query Editor

To export the query and associated results to excel simply click Export . The query editor will then open the data set in a new screen. This information can then be exported to Excel by right-clicking anywhere in the data table.

The spread sheet created will contain the appropriate query and links to coins so that the data can be refreshed at any point.

### 3.2.3 Saving Queries in Query Editor (10.27)

To prevent the need to keep re-typing regularly used queries, save and load options are available as of v10.27 and are located at the top of the Query Editor page.

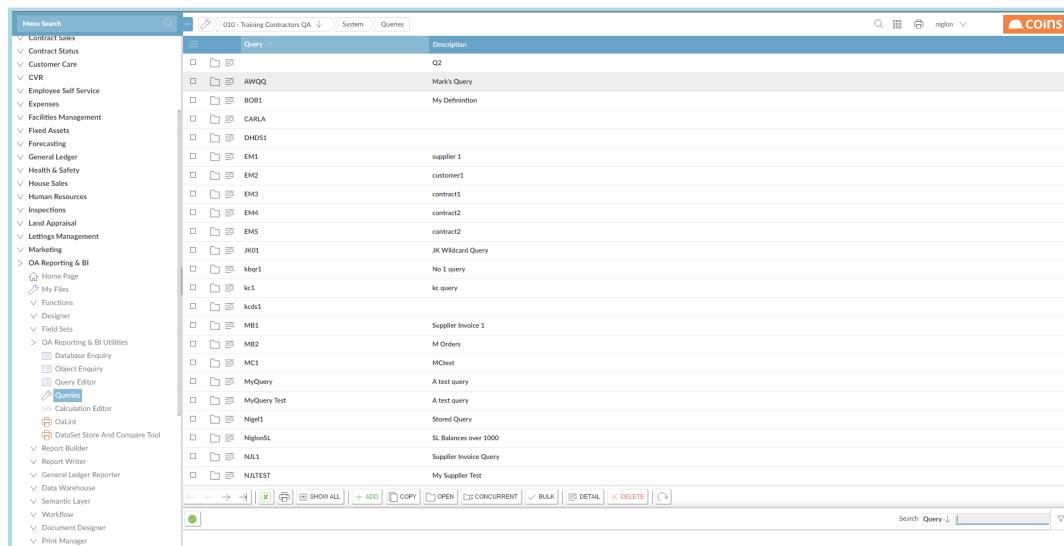
Once you have set up your query and selected the required fields, in the Stored fields, specify a name for the query and a description to further identify it.

Click Save

The Query will now be stored and can be retrieved at any time by running Query Editor, specifying the Query name and clicking Load button.

To delete a saved query, specify the name and select the Delete button.

The full list of stored queries may be viewed from the new Queries option from the OA Reporting and BI Utilities Menu



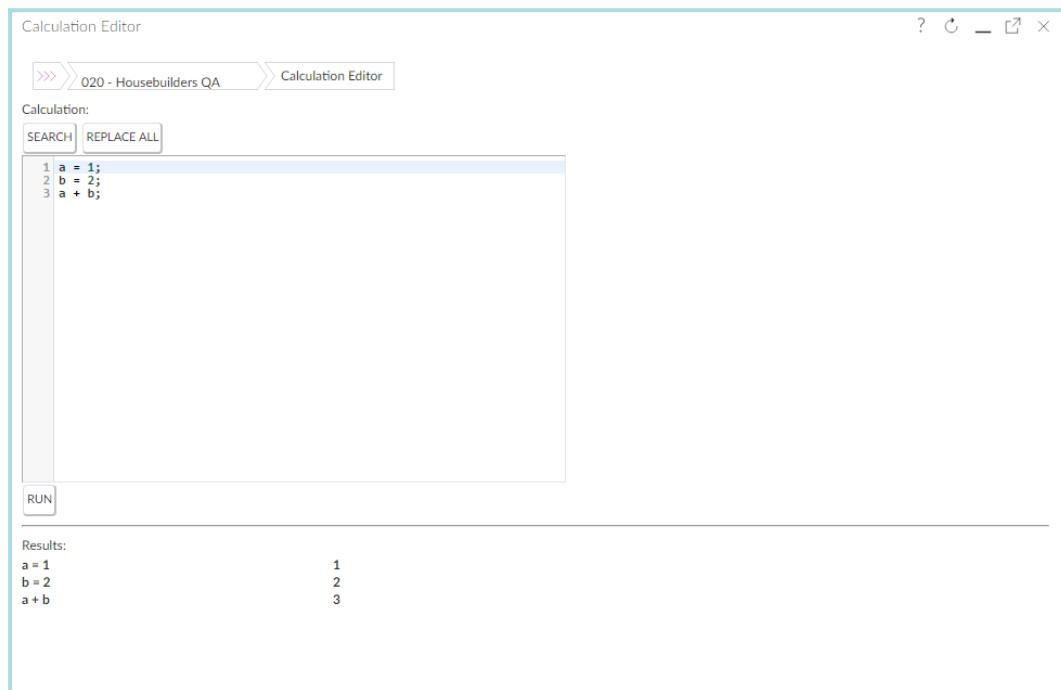
This screen will allow the creation, deletion and maintenance of the stored queries.



To see the results of the queries, you will need to return to Query Editor and Load the appropriate query.

### 3.3 Calculation Editor

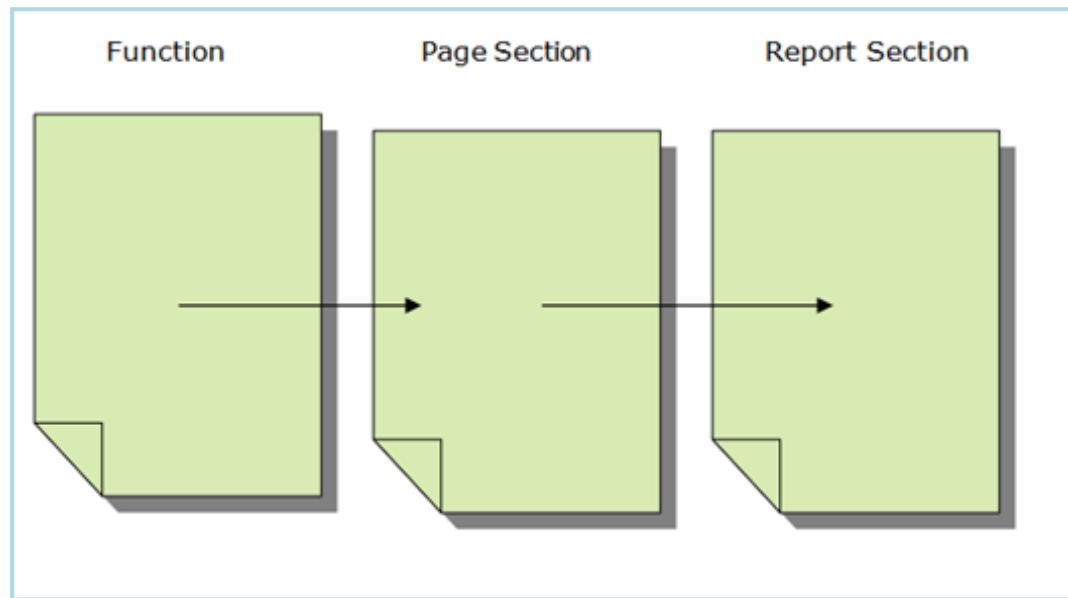
The Calculation Editor allows you to try out calculations using the coins business logic calculation methods (see accompanying documentation). This function can be useful to test calculations before being used in reports and enquiries.



Simply enter in the calculations using the appropriate syntax and click run to test the results.

## 4 Functions and Sections

To generate a Page or Report in coins OA there are three development components :



A function in OA simply runs a procedure (unless defined as a menu).

This procedure is nearly always wou005.p and it simply builds a webpage for display in coins OA.

To determine what is displayed and how it is displayed, coins OA (wou005.p) uses Pages.

A Page Design contains Form definitions for the webpage. These Forms definitions may include headers, bodies, footers, totals etc. Each of these Form Types will also have Fields defined for display.

In addition, Pages also allow for the definition of Filters which will be available on the webpage when the function is run.

When you are running a Report in coins OA, the Function will still reference a Page, however this webpage will contain only the information (layout and fields required) for the report selection criteria. The actual definition of fields, layout, headers and totals are then maintained in the Report Design.

When you create a new function in coins OA - until you have completed the Page Design nothing will be returned to screen. If this function is to be a report, it is also necessary to define the Report Design.

## 4.1 Function Naming

Below is the current COINS standard naming conventions for OA functions

example: %WPL5100BAVMT1

%WXXnnnmCTTT[Y]Tn

%	% Denotes a coins standard function (which will be updated and maintained by coins and will be overwritten as standard data during processes such as environment upgrades). If you wish to amend a standard Coins function, it is strongly suggested, as with Coins+ you copy the function and rename as with the original but replacing the % with a +.
XX	Module code (i.e. GL, JC or PO)
nnn	functions sequence number (Enquiries generally begin 5nn and Reports 3nn)
m	sub-sequence number (often 0 but used for imbedded/inline reports)
C	Class of page (Browse, Form, List, Report, Summary, Utility, Web Page)
TTT	Table ID (From the main table used - see Database Enquiry)
Y	(optional) Sub-function type (for buttons etc.) (Add, Update, Delete, More, eXport, Generate, Bulk, Options Menu, Link/Actions Menu), Concurrent Update
T	Tab
N	Sequence

## 5 Datasets - Overview

A Data Set is a pre-defined Table which is created at the time of the query. Once the table is created then it can be accessed via either Report Writer or Report Designer. The benefits of using Data Sets are:

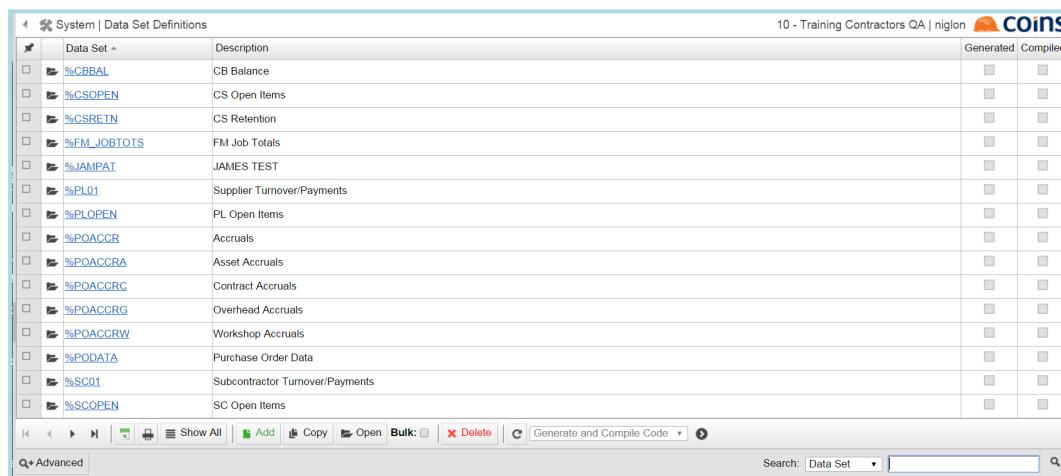
- Provide access to PROGRESS temporary table functionality
- Build single table of data from various COINS tables
- Build Data Sets for use in multiple reports
- Allow sorting by virtual fields
- Allow filtering by virtual fields
- Allow union of several Data Sets ( Useful for Cross Modular Reporting)
- Summarisation of data
- Simplification of data views for users

## 5.1 Creating a Data Set

Once the fields which are required have been identified the query to obtain those fields must be created. In this example the fields required have been identified as

- Contract Number -(job\_num)
- Contract Name - (job\_name)
- Contract Cost - (RO\_ContractCosts^TD)
- Contract Revenue - (RO\_ContractRevenue^TD)
- Contract Profit (RO\_ContractRevenue^TD - RO\_ContractCosts^TD)

To create a new Data Set go to the Data Set Definition option under the Designer Menu.



	Data Set	Description	Generated	Compiled
	%CBAL	CB Balance		
	%CSOPEN	CS Open Items		
	%CSRETN	CS Retention		
	%FM_JOBTOTALS	FM Job Totals		
	%JAMPAT	JAMES TEST		
	%PL01	Supplier Turnover/Payments		
	%PLOPEN	PL Open Items		
	%POACCR	Accruals		
	%POACCRA	Asset Accruals		
	%POACCRC	Contract Accruals		
	%POACCRG	Overhead Accruals		
	%POACCRW	Workshop Accruals		
	%POADATA	Purchase Order Data		
	%SC01	Subcontractor Turnover/Payments		
	%SCOPEN	SC Open Items		

Data Set	Description	Generated	Compiled	Locked By
%SCBBAL	CB Balance			
%CSOPEN	CS Open Items			
%CSRETN	CS Retention			
%NFM_JOBTOTS	FM Job Totals			
%SPL01	Supplier Turnover/Payments			
%SLOPEN	PL Open Items			
%POACCR	Accruals			
%POACCRA	Asset Accruals			
%POACCRC	Contract Accruals			
%POACCRG	Overhead Accruals			
%POACCRW	Workshop Accruals			
%PODATA	Purchase Order Data			
%SC01	Subcontractor Turnover/Payments			
%SCOPEN	SC Open Items			
%SCRETN	SC Retention			
%SLOPEN	SL Open Items			
%SYMEXIC	Mobile mExec			
%WCBS100CCBD	CB Receipts Last 7 Days			
%WCBS200CCBD	CB Payments Last 7 Days			
%WCBCB010	mExec Cash Paid Today			
%WCBCB020	mExec Cash Received Last & Days			
%WCBCB030	mExec Cash Received Today			

Click



Select a Data Set Name and Description for the Data Set in this example is has been named 'NLWJC\_PROF' with a description of 'Standard JC Profit/Loss DataMart'.

A Query must be assigned to the Data set, which in the following example is a simple query of each Contract which takes advantage of a 'Contract Selection' made at run time.

```
FOR EACH jc_job WHERE jc_job.kco = {kco}{jobSelect}
```

Create an identifier for the Data Set Table Name. In this example it is 'contract'.

Once created, the table name is always prefixed with a 'tt' when used, so in this case the table will be accessible in a report using a 'FOR EACH ttcontract' query.

The screenshot shows the 'Definition' tab of the Data Set Summary screen. It includes sections for 'Generate Program', 'Condition', 'Raw DB Query', 'Table Name', 'Initial Calculation', 'Pre Calculation', and 'Post Calculation'. Each section contains a code editor window with specific SQL or COINS logic. At the bottom, there are buttons for 'OPEN', 'Generate and Compile Code', and 'NEXT'.

Field	Description
Generate Program:	It is now possible to call COINS standard generate programs and include them into your Data Set. – Sample generates scr399, plr399, slr399, csr399. (Open item records).
Raw DB Query	This will bypass the Business Logic and get data directly from the Database so care should be taken on who is given access to Data Set Queries where this selection is selected. (ie: This will ignore all security).

There are three calculation fields which can be applied to the data set.

Field	Description
Initial calculation	<p>This will calculate prior to the query executing thus allowing the variable to be used within the Query, e.g.</p> <pre>\$sdate = date\$(datestring(co_config.glp_fdate^-12 {RS_glp_fdate}));</pre> <p>Would allow a query to be created such as:</p> <pre>FOR EACH jc_job WHERE jc_job.job_condate &gt;= '{eval.sdate}'</pre>
Pre calculation	This will take place after the query but prior to the fields calculating
Post calculation:	:This will take place prior to the data set closing



Variables used in a Dataset are NOT available in the report.

To move onto the Data Set fields, click



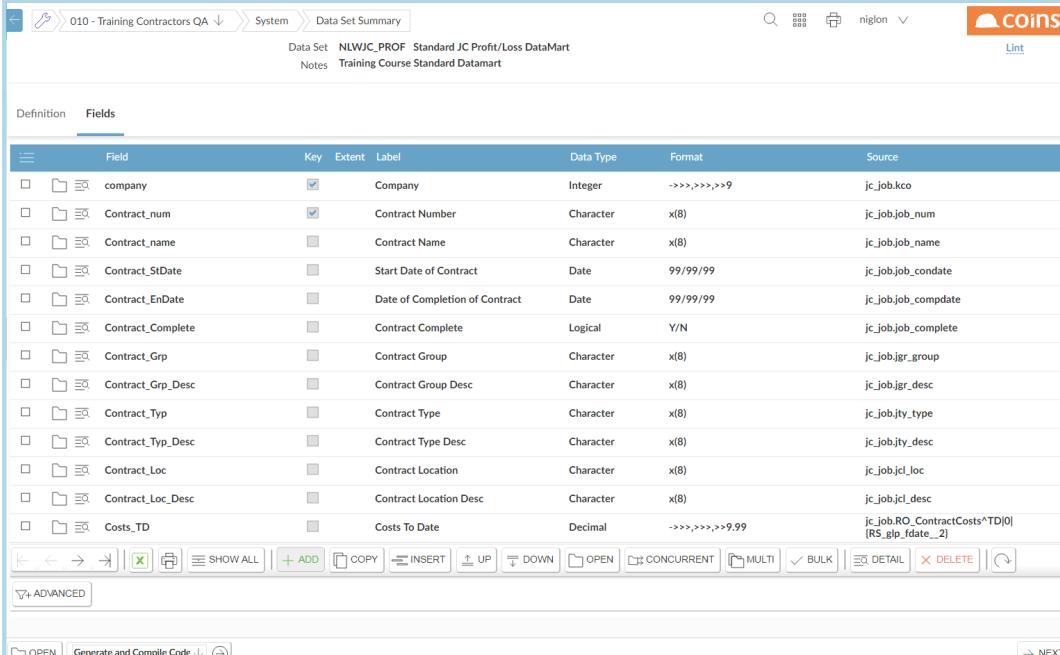
## 5.1.1 Data Set Fields

When Adding fields to the table, you can give the field a Name that can either be same field name as the field in the source table or you can name the field something that will help Report Writer/Designer users identify the information in the field more easily. (Eg: location\_desc is assigned to the field jc\_desc)

Each field should be given:

Field	Description
Label	This will become the name of the field when using the Data Set in queries/Reports etc. You may either use the same name as the source field or assign a more user-friendly name. Spaces should not be used
Data Type	(eg Character, Date, Decimal, Integer, Logical)
Default Format	(NB. All formats are only defaults and can be overridden in reporting)
Source	The source can be any field (Database or RO) from the tables accessible via the query or it could be a calculation.

As an example, the sample of the fields which have been added to the Data Set are seen in the diagrams below.



Field	Key	Extent	Label	Data Type	Format	Source
company	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Company	Integer	->>,>>,>>9	jc_job.kco
Contract_num	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Contract Number	Character	x(8)	jc_job.job_num
Contract_name	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Contract Name	Character	x(8)	jc_job.job_name
Contract_StartDate	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Start Date of Contract	Date	99/99/99	jc_job.job_condate
Contract_EndDate	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Date of Completion of Contract	Date	99/99/99	jc_job.job_comdate
Contract_Complete	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Contract Complete	Logical	Y/N	jc_job.job_complete
Contract_Grp	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Contract Group	Character	x(8)	jc_job.jgr_group
Contract_Grp_Desc	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Contract Group Desc	Character	x(8)	jc_job.jgr_desc
Contract_Typ	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Contract Type	Character	x(8)	jc_job.jty_type
Contract_Typ_Desc	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Contract Type Desc	Character	x(8)	jc_job.jty_desc
Contract_Loc	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Contract Location	Character	x(8)	jc_job.jcl_loc
Contract_Loc_Desc	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Contract Location Desc	Character	x(8)	jc_job.jcl_desc
Costs_TD	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Costs To Date	Decimal	->>,>>,>>9.99	jc_job.RO_ContractCosts^TD 0 [RS_glp_fdate_2]

010 - Training Contractors QA > System > Data Set Summary

Data Set: NLWJC\_PROF Standard JC Profit/Loss DataMart  
Notes: Training Course Standard Datamart

Definition Fields

Field	Key	Extent	Label	Data Type	Format	Source
Costs_TD			Costs To Date	Decimal	->>>,>>>,>>9.99	jc_job.RO_ContractCosts^TD[0] [RS_glp_fdate_<2]
Costs_TP			Costs This Period	Decimal	->>>,>>>,>>9.99	jc_job.RO_ContractCosts^TP[0] [RS_glp_fdate_<2]
Costs_TY			Costs This Year	Decimal	->>>,>>>,>>9.99	jc_job.RO_ContractCosts^TY[0] [RS_glp_fdate_<2]
Revenue_TD			Revenue To Date	Decimal	->>>,>>>,>>9.99	jc_job.RO_ContractRevenue^TD[0] [RS_glp_fdate_<2]
Revenue_TP			Revenue This Period	Decimal	->>>,>>>,>>9.99	jc_job.RO_ContractRevenue^TP[0] [RS_glp_fdate_<2]
Revenue_TY			Revenue This Year	Decimal	->>>,>>>,>>9.99	jc_job.RO_ContractRevenue^TY[0] [RS_glp_fdate_<2]
Profit_TD			Profit To Date	Decimal	->>>,>>>,>>9.99	
Profit_TP			Profit This Period	Decimal	->>>,>>>,>>9.99	
Profit_TY			Profit This Year	Decimal	->>>,>>>,>>9.99	

SHOW ALL ADD COPY INSERT UP DOWN OPEN CONCURRENT MULTI BULK DETAIL DELETE ADVANCED

OPEN Generate and Compile Code

010 - Training Contractors QA > System > Data Set Summary

Data Set: NLWJC\_PROF Standard JC Profit/Loss DataMart  
Notes: Training Course Standard Datamart

Definition Fields

Field	Key	Extent	Label	Data Type	Format	Source
Profit_TD			Profit To Date	Decimal	->>>,>>>,>>9.99	
Profit_TP			Profit This Period	Decimal	->>>,>>>,>>9.99	
Profit_TY			Profit This Year	Decimal	->>>,>>>,>>9.99	

Calculation: 1 Profit\_TD=Rev\_TD - Costs\_TD;

Recalculate Summary

Calculation: 1 Profit\_TP=Rev\_TP - Costs\_TP;

Recalculate Summary

Calculation: 1 Profit\_TY=Rev\_TY - Costs\_TY;

Recalculate Summary

OPEN Generate and Compile Code

## 5.2 Using Data Sets in the Query Editor

The Query Editor allows fast access to information in the COINS Database via the Business Logic. It also allows export to Microsoft Excel.

It is possible to reference a Data Set from the query editor. Simply referencing the data set and running will return the complete contents of the Data Set - No query is required.

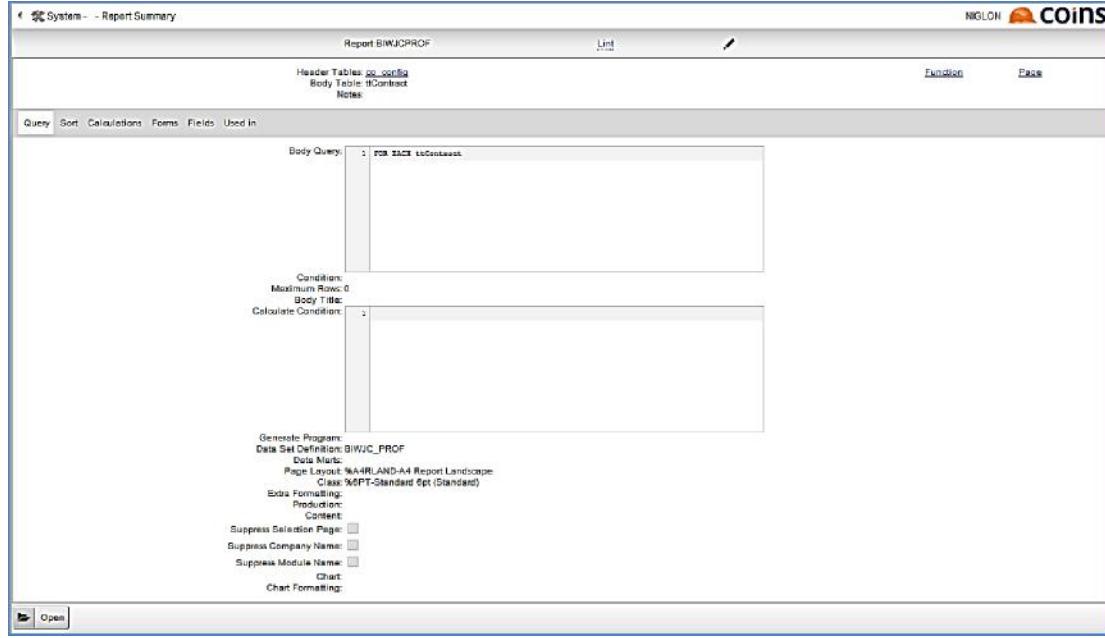
If the Dataset query takes a parameter replacement field (i.e. {RS\_year}) then you can set the parameters by adding them to the URL on the Query Editor Page. (i.e. &RS\_year=2009)

NOTE : the full Data Set will be generated before any results are returned so be warned - a large data set may be slow.

This has allowed the extract to Excel to be more flexible as limits in the information being passed limited both the query length and number of fields that could be accessed through the editor. Referencing a dataset code greatly enhances the ability to extract data this way.

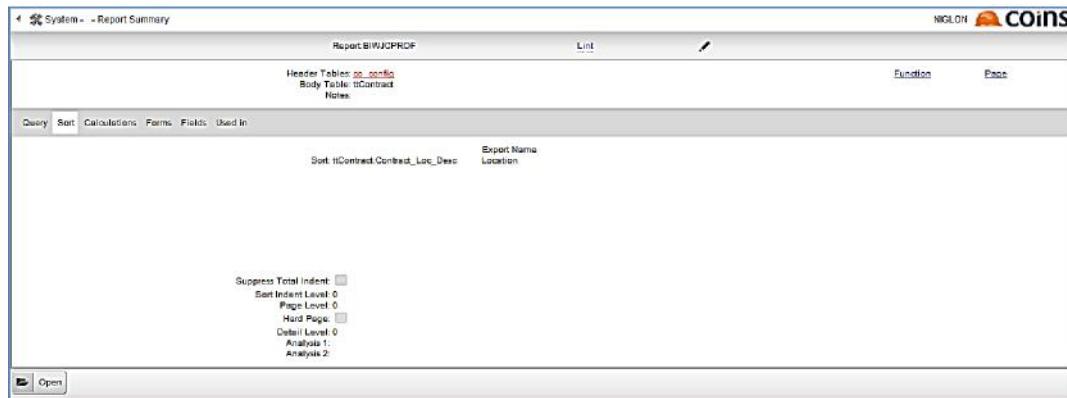
## 5.3 Using the Data Set in Queries

The Data Set can now be used in Queries in Report Designer. Using our example dataset, the body table should be ttContract, the body query should be - FOR EACH ttContract and the Data Set Definition would be BIWJC\_PROF.



Note that you can use more than one dataset in a query. To do this enter each dataset name in the Data Set Definition field separated by a comma. You will need to reference each ttTablelename as appropriate in the body query.

It is possible to sort on any field from within the Data Set - In the example below we are using a field which has been populated with the virtual field jcl\_desc with is the Location Description. This will put the contract in alphabetical order of the locations to which they belong. (NB. It is always necessary to qualify the field name with the Data Set table name in the sort)



Fields can then be added to the report in the standard way. It is always good practice to qualify the field names also with the Data Set table name although it is not mandatory.

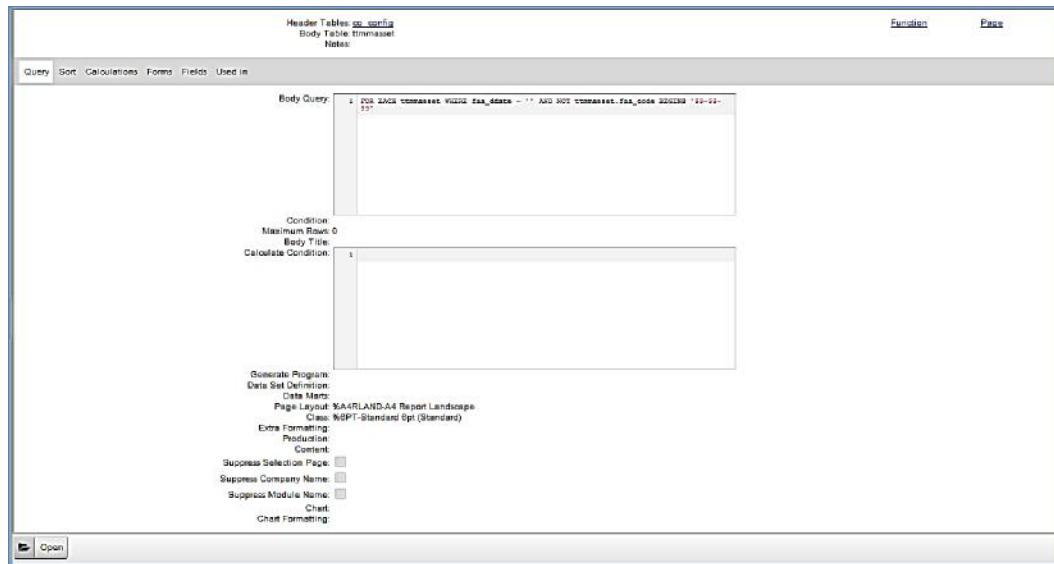
The screenshot shows the 'Report Summary' interface for report BIW/JCPROF. The header includes 'Header Tables: m\_config' and 'Body Table: tContract'. The main area displays a table of fields with columns: Field, Label, Width, Height, Function, and View As. Fields listed include nContract.job\_num, Name, Cost TD, Rev TD, Profit TD, Cost TP, Rev TP, and Profit TP. Below the table are various toolbar buttons for report management. At the bottom, there are dropdown menus for 'Form: Body' and 'View: Standard'.

### 5.3.1 In-Line Reports

For in-line reports, specify the dataset(s) to be used on the container report e.g.:

The screenshot shows the 'Report Summary' interface for report BIW/JCPROF. The header includes 'Header Tables: m\_config' and 'Body Table: tContract'. The main area displays a table of fields with columns: Field, Label, Width, Height, Function, and View As. Fields listed include nContract.job\_num, Name, Cost TD, Rev TD, Profit TD, Cost TP, Rev TP, and Profit TP. Below the table are various toolbar buttons for report management. At the bottom, there are dropdown menus for 'Form: Body' and 'View: Standard'.

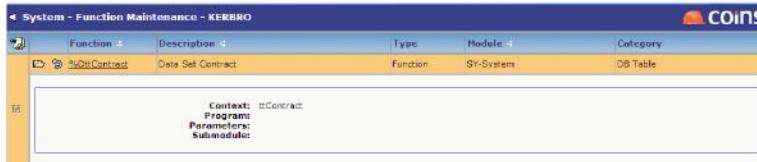
Then for each inline report, only use the ttTablename in the body query of each report but do not specify the dataset e.g.



This will ensure that the datasets are only built once and then data then shared across each report that uses it. If you specify the dataset name in the in-line reports, the dataset has to be rebuilt for that report which will have an impact on performance.

## 5.4 Granting Access to Data Sets for Other Users.

Once the Data Set setup has been completed you can allow access to the relevant users by creating a %DttContract Function via Function Maintenance.



Access to this function is then granted via the standard Function Security procedure.

The Data Set can be enabled to be accessed by Report Writer by the creation of a Report Writer query accessing the ttContract table.

## 5.5 Rules for Keys

It is possible to summarise DataSets by adding a 'KEY'. If the query is looking at all Contract (jc\_job) records but the requirement in the report is to be by Contract Location then a 'KEY' can be set at kco/jcl\_loc (ie: Tick the Key box on both of these fields). In this instance only one record will be created on a unique find of Company/Contract Location. Any numerical fields are accumulated whilst character fields are assigned where there is common data where records share the same 'Key' details. (If it finds character fields which differ then the value of the field will be blank).

If there is a requirement to do a calculation once all of the records for the 'Key' are accumulated then you can tick the 'Recalculate Summary'. (Eg: An example of where this might be required is when a percentage is required).

Note: Any field which is used as a key CANNOT have a value of blank.

## 5.6 Best Practice

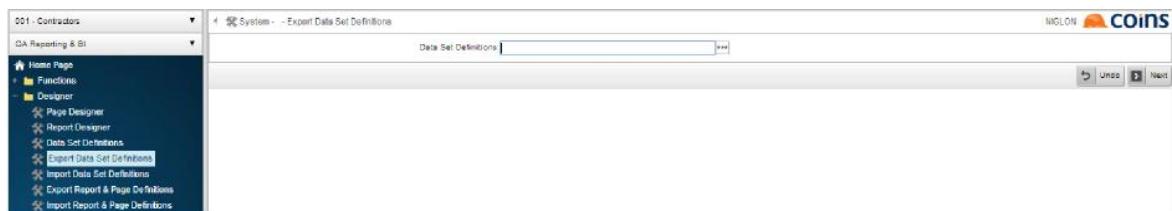
- Only create fields in a DataSet that are going to be used in the equivalent report.
- It is better to 'Source' a field rather than to 'Calculate' a field for efficiency.
- It is recommended that any calculations that can be carried out on the report/page should be done at that stage.
- For performance reasons, {}'s are now replaced once at the start of the dataset (v10.22 onwards) so as to not replicate the process for every field in the dataset. This is not a problem in most instances as the {}'s used in RO fields are usually something along the lines of {RS\_glp\_fdate\_2} which is consistent across all records. It will mean however that you cannot put {}'s around field names; for instance '{job\_num}' is not possible because this replacement will differ on each record. It is imperative that fields like this are referenced with Tablename.Fieldname e.g. jc\_job.job\_num - It is always best practice to fully qualify fields in a calculation on a dataset as the field will not strip out formatting etc so the value 10,000 would cause a problem previously in an if statement as the comma would be read into the if statement causing the syntax to have too many parameters.
- To maximise efficiency, any field on a DataSet which is not at the lowest level of the DataSet query, should be taken out of the DataSet and initiated on the report. For example if you had a query on the DataSet which reads

**FOR EACH jc\_job WHERE jc\_job.kco = {kco},  
EACH jc\_costcode OF jc\_job.**

- Then fields from jc\_costcode should be included in the DataSet but any RO fields etc from the jc\_job table should be called on the report. To save unnecessary replication of calculations.
- It is good practice to Summarise at DataSet level, rather than at Report level if Summarisation is required. This is so that multiple records are not created and passed across to the report unnecessarily. This will reduce the load on the network traffic, especially where report servers are in use, and will reduce the amount of time the report takes to generate.

**Debug(1);** - Can be turned on to debug a dataset but should be removed when setting a DataSet live. Debug(1) should be defined as a calculation in the Initial Calculation box. Once debug is turned on then all calculated fields will be verbose in the Report Log File.

Data Sets can be exported from one environment to another by simply choosing the Export Data Set Definitions option from the Designer menu.



Enter the Data Set Name or use the lookup facility to find the relevant Data Sets and click Next

Once the Data Set has been exported to the Definition Data Window use the standard windows select all(Ctrl-A) and copy(Ctrl-C) functions to copy the data from the Definition Data Window and then Paste(Ctrl-V) to the Import Data Sets Definitions window also found Menu below in the environment you are wanting to add the Data Set after which you should then click the save icon.



A Data Set cannot pre-exist so if there is amendment required to a Data Set it must first be removed prior to importing

## 5.7 Cross Modular Reporting

To achieve Cross Modular Reporting two independent Data Sets can be created using different Source information but containing the following commonalities

- Same Table Name
- Same Field Names (in same order)
- Fields must have the same Data Type (ie. Char, Int, Dec, Logical, Date)

The datasets can then be both called on the report and the common table name will unionise the data. An example of such an application might be a Purchase Ledger and SubContract Ledger Open Invoice Dataset.

## 6.1 Report Pre-Processing (Syuds)

Reports in OA have mainly been banded or grouped row reports. There are a few hard coded exceptions where a grid or matrix of data is presented over a number of pages of the report.

Pre-Processing provides a generic mechanism to pivot a set of data in to columns and if required to page over multiple pages to allow any dataset to be processed in to further datasets that are suitable for printing in OA as matrix of data.

Further pre-processing methods are also provided to manipulate the data set (temp tables), however they are produced, and create further temp tables with the processed data. Methods are provided for Union, Merge, and Sum. There is also a debug option provided to dump the contents of a dataset to the log file and a Store option to store a dataset in a datamart extract.

### 6.1.1 Syuds.Calc

**syuds.calc** allows you to create a new table with calculated fields in it from other fields in the query.

You use **calcTable** to set the input query (multiple table) and output table name (e.g. Calc to produce a table called ttCalc)

You then run **calc** for each field you want to add to ttCalc and pass the following parameters:

name,

data type,

extent,

label,

format,

calc string

**Calceexec** then builds the ttCalc table, adds the fields you defined and runs the query and calculates for each row.

Can be very useful for adding extra stuff alongside existing records

The ttCalc record is created with rowid fields for each of the tables in the query so you can then join to it in a report query

#### 6.1.1.1syuds.calc example

Dataset Query:

```
FOR EACH jc_job WHERE kco = {kco}
```

Dataset Post Calculation:

```
method('syuds.calcTable','FOR EACH ttcontract','calc');  
method('syuds.calc','lastYTD','decimal',0,'Last YTD','','ttcontract.job_costsTD  
- ttcontract.job_costsTY');  
method('syuds.calc','lastPTD','decimal',0,'Last PTD','','ttcontract.job_costsTD  
- ttcontract.job_costsTP');  
method('syuds.calceexec');
```

Dataset Fields:

Data Set: syudscalc Syuds Exercise 1  
Notes

Definition Fields

	Field	Key	Extent	Label	Data Type	Format	Source
<input type="checkbox"/>	<input type="checkbox"/> job_num	<input type="checkbox"/>		Contract	Character	x(8)	jc_job.job_num
<input type="checkbox"/>	<input type="checkbox"/> job_name	<input type="checkbox"/>		Name	Character	x(8)	jc_job.job_name
<input type="checkbox"/>	<input type="checkbox"/> job_costsTD	<input type="checkbox"/>		Costs TD	Decimal	>>>,>>>9.99	jc_job.RO_ContractCosts^TD
<input type="checkbox"/>	<input type="checkbox"/> job_costsTY	<input type="checkbox"/>		Costs TY	Decimal	>>>,>>>9.99	jc_job.RO_ContractCosts^TY
<input type="checkbox"/>	<input type="checkbox"/> job_costsTP	<input type="checkbox"/>		Costs TP	Decimal	>>>,>>>9.99	jc_job.RO_ContractCosts^TP

### Output: ttContract

010 - Training Contractors QA System Query Editor

Stored:

Query:

```
1 FOR EACH ttcontract
```

Fields:

Data Set: syudscalc

Condition: 10

Timeout(Seconds): 10

Maximum Rows: 10

Extra Parameters:

RUN EXPORT CREATE DATASET

job_num	job_name	job_costsTD	job_costsTY	job_costsTP
10000	Penny Hill Estates	12550	12225	12000
100010	Purchase Land	0	0	0
1001	PF Training Project	0	0	0
10010	New Wing	1200	0	0
1005	Baxter Building Refurbishment	0	0	0
1007	Hamptons Hospital Construction	57600	0	0
11000		0	0	0
11001		0	0	0
11002		0	0	0
1111	Milton Keynes Building	4300	0	0

### Output: ttCalc

010 - Training Contractors QA > System > Query Editor

Stored:

Query:

Fields:

Data Set: syudscalc

Condition: 10

Timeout(Seconds): 10

Maximum Rows: 10

Extra Parameters:

RUN EXPORT CREATE DATASET

ttcontractRowid	lastYTD	lastPTD
0x0000000000027900	325	550
0x0000000000027901	0	0
0x0000000000027902	0	0
0x0000000000027903	1200	1200
0x0000000000027904	0	0
0x0000000000027905	57600	57600
0x0000000000027906	0	0
0x0000000000027907	0	0
0x0000000000027908	0	0
0x0000000000027909	4300	4300

## 6.1.2 Syuds.Debug

This method causes the contents of a dataset to be exported to the log file. Useful for seeing the results at various stages of pre-processing.

The parameters are table name, condition, fields (defaults to all fields in the dataset) and the number of records (if zero specified then 10 will be output).

### Dataset Post Calculation

In this example, the first 10 contents of dataset table ttContract will be output

**Method('syuds.debug','Contract','','',0);**

Output of the data in the log file is in CSV format suitable for pasting in to EXCEL.

### Example output:

```

① 195.40.14.50/cgi-bin/oatrainbi/JC001295.log?kco=10&pvCIlevel=1&pvCISibling=10&TopMenu=%25WHOME&pvFrame=F%2C1583%2C620
PM01
costs_td = jc_job.RO_ContractCosts^TD
1787
rev_td = jc_job.RO_ContractRevenue^TD
0
prof_td = rev_td - costs_td
-1787
costs_td = jc_job.RO_ContractCosts^TP|0|
0
03/11/2016 11:27:57.266+00:00 PostRunQuery
03/11/2016 11:27:57.266+00:00 PostDataSet
Method('syuds.debug','Contract','','',0)
DEBUG=niglon METHOD SIGNATURE=PROCEDURE, INPUT pcTable CHARACTER,INPUT pcCondition CHARACTER,INPUT pcFields CHARACTER,INPUT piCount INTEGER
DEBUG=niglon SET-PARAMETER 1 INPUT CHARACTER Contract
DEBUG=niglon SET-PARAMETER 2 INPUT CHARACTER
DEBUG=niglon SET-PARAMETER 3 INPUT CHARACTER
DEBUG=niglon SET-PARAMETER 4 INPUT INTEGER 0
03/11/2016 11:27:57.267+00:00 Debug: Contract 0
03/11/2016 11:27:57.268+00:00 Query: FOR EACH ttContract
kco, job_num, job_name, costs_td, rev_td, prof_td, costs_td
"10" "10000", "Penny Hill Estates", "12550", "156300", "143750", "12000"
"10" "100010", "Purchase Land", "0", "0", "0"
"10" "1001", "PF Training Project", "0", "0", "0"
"10" "10010", "New Wing", "1200", "0", "-1200"
"10" "1005", "Baxter Building Refurbishment", "0", "0", "0"
"10" "1007", "Hamptons Hospital Construction", "57600", "0", "-57600", "0"
"10" "11000", "", "0", "0", "0"
"10" "11001", "", "0", "0", "0"
"10" "11002", "", "0", "0", "0"
"10" "1111", "Milton Keynes Building", "4300", "0", "-4300", "0"
03/11/2016 11:27:57.268+00:00 Dataset Complete:NLM2DS1
03/11/2016 11:27:57.269+00:00 Generate End
03/11/2016 11:27:57.269+00:00 Setup Query Start
03/11/2016 11:27:57.271+00:00 Query Tables:ttcontract
03/11/2016 11:27:57.271+00:00 Query: .001s for EACH ttcontract no-lock
by ttcontract.job_num
indexed-reposition
03/11/2016 11:27:57.272+00:00 Report Content Start
03/11/2016 11:27:57.272+00:00 ReportContent
03/11/2016 11:27:57.277+00:00 Report Content Prepared
03/11/2016 11:27:57.549+00:00 Rows Processed: 75
03/11/2016 11:27:57.550+00:00 ReportSelection
03/11/2016 11:27:57.586+00:00 Generate PDF Start
Nov 3, 2016 11:27:58 AM org.apache.fop.layoutmgr.table.TableLayoutManager getNextKnuthElements
INFO: table-layout="fixed" and width="auto", but auto-layout not supported => assuming width="100%"
Nov 3, 2016 11:27:59 AM org.apache.fop.layoutmgr.table.TableLayoutManager getNextKnuthElements
INFO: table-layout="fixed" and width="auto", but auto-layout not supported => assuming width="100%"
03/11/2016 11:27:59.708+00:00 Generate PDF End
03/11/2016 11:27:59.715+00:00 Report Content End
03/11/2016 11:28:00.145+00:00 Email accepted for delivery. Srv smtpuk.coins.local. Port 25; To: nigel.longley@coins-global.com
03/11/2016 11:28:00.147+00:00 Delete FSP: jcfrp.p
End Time:03/11/2016 11:28:00.147+00:00
Timings: Startup- 29 Generate- 839 Prepare- 3 Production- 314 PDF- 2122 Completion- 439 Total- 3746

```

### 6.1.3 Syuds.Delete

This method deletes a dataset/temp tables from the report. It should be used if the dataset that has been generated is no longer required but might be rebuilt or reused later in the report (typically on inline reports).

```
Method('syuds.delete','CostRev,Budget');
```

The parameter is table names. The example would delete ttCostRev and ttBudget.

## 6.1.4 Syuds.Filter

This method causes the contents of a dataset to be copied to an identical dataset except that the records in the output dataset are filter based on a condition passed to the method.

```
Method('syuds.filter','CostRev','tdate="31/01/13"','CostRevJan');
```

The parameters are table name, condition, output table name.

The above example would take records from ttCostRev and filter on a condition clause where field tdate was equal to 31/01/13. The output records would be in ttCostRevJan which would have the same fields as ttCostRev.

This method is useful on inline reports to filter out a set of records from the containing report for processing or display on the inline. See also delete() method to delete this dataset once it has been used.

Also useful to filter a set of records in to a new temp table before storing using store().

## 6.1.5 Syuds.GroupQuery

**GroupQuery** came about because of report builder and is similar to sum except that they do multiple sums at different levels.

For example:

group by kco by jgr\_group by job\_num would create a summary record for:

level 0 (grand total),  
level 1 (kco),  
Level 2 (group),  
Level 3 (job\_num).

It equates the footer forms in the report builder and the report footer (level 0).

**GroupQuery** allows multi table query and takes the form:

```
Method('syuds.GroupQuery','[query string]', '[output table]', '[keys]', '[sum fields]')
```

**Group** is just a simpler form of GroupQuery (for a single table).

**Group** takes the form:

```
Method('syuds.Group','[Source Table]', '[Condition]', '[output table]', '[keys]', '[sum fields]')
```

It actually then runs groupQuery with “FOR EACH XXX WHERE condition”

Example:

```
Method('syuds.group','Source','drev>1000','pcTable','pcKeys','SumFields');
```



Leaving fields to sum blank will do all decimals

Example:

Dataset Query:

```
FOR EACH jc_job WHERE kco = {kco} AND CAN-DO('RS_job_num_3', job_num)
```

Dataset Post Calculation:

```
method('syuds.calcTable','FOR EACH ttContract','calc');
method('syuds.calc','lastYTD','decimal',0,'Last YTD','');
method('syuds.calc','lastPTD','decimal',0,'Last PTD','');
method('syuds.calexec');
method('syuds.sum','Contract','','Location','jcl_loc','jcl_desc','');
method('syuds.group','Contract','','Group1','jcl_loc','job_costsID');
method('syuds.groupquery','FOR EACH ttContract, EACH ttCalc WHERE ttCalc.ttContractRowid=ROWID(ttContract)','Group2','jcl_loc','');
```

Dataset Fields:

Definition Fields				
Field	Key Extent Label	Data Type	Format	Source
job_num	Contract	Character	x(8)	jc_job.job_num
job_name	Name	Character	x(30)	jc_job.job_name
jc_loc	Location	Character	x(8)	jc_job.jc_loc
jc_desc	Location Description	Character	x(30)	jc_job.jc_desc
jgr_group	Group	Character	x(8)	jc_job.jgr_group
jgr_desc	Group Description	Character	x(30)	jc_job.jgr_desc
job_costsTD	Costs TD	Decimal	>>>,>>>9.99	jc_job.R0_contractcosts^TD
job_costsTY	Costs TY	Decimal	>>>,>>>9.99	jc_job.R0_contractcosts^TY
job_costsTP	Costs TP	Decimal	>>>,>>>9.99	jc_job.R0_contractcosts^TP

### Output:

20 - Housebuilders QA - System - Query Editor

Stored:  Search Replace all Load Save Delete

Query:

```
1 |FOR EACH ttGroup1
```

Fields:

Data Set: Timjob  
Condition:  
Timeout(Seconds): 10  
Maximum Rows: 100  
Extra Parameters: &RS\_job\_num\_3=1000

Run Export Create Dataset

level	kjcl_loc	count	job_costsTD	job_costsTD_max	job_costsTD_min	job_costsTD_avg
0		1	-71124.4	-71124.4	-71124.4	-71124.4
1	WARKS	1	-71124.4	-71124.4	-71124.4	-71124.4

20 - Housebuilders QA - System - Query Editor

Stored: syudscalc Syuds Calc Example Load Save Delete

Query:

```
1 |FOR EACH ttGroup2
```

Fields:

Data Set: Timjob  
Condition:  
Timeout(Seconds): 10  
Maximum Rows: 100  
Extra Parameters: &RS\_job\_num\_3=1000

Run Export Create Dataset

level	kjcl_loc	count	job_costsTD	job_costsTD_max	job_costsTD_min	job_costsTD_avg	job_costsTY	job_costsTY_max	job_costsTY_min	job_costsTY_avg	job_costsTP	job_costsTP_max	job_costsTP_min	id
0		1	-71124.4	-71124.4	-71124.4	-71124.4	4273	4273	4273	4273	200	200	200	20
1	WARKS	1	-71124.4	-71124.4	-71124.4	-71124.4	4273	4273	4273	4273	200	200	200	20

## 6.1.6 Syuds.Merge

This method will combine selected columns from two or more dataset (with a common key) in to a new dataset.

Suppose we have a dataset ttCost

Kco	Job_num	dCosts
100	1001	100
100	1002	150

And a dataset ttRev

Kco	Job_num	dRev
100	1001	200
100	1002	250

Then the resulting dataset (ttCostRev) might be

Kco	Job_num	dCosts	dRev
100	1001	100	200
100	1002	150	250

This is achieved in a report initialisation calculation as follows:

```
Method('syuds.mergeKeys','kco,job_num');
Method('syuds.mergeTable','cost',"");
Method('syuds.mergeTable','rev',"");
Method('syuds.mergeExec','CostRev');
```

mergeKeys is used to specify the unique keys used to merge the data.

mergeTable is called once for each table to be merged. The first parameter is the dataset name (without the tt prefix), the second parameter is the condition to be applied to this set of records, the third parameter is the key field names in this table (defaults to the mergeKeys), the fourth parameter is the field names (in this table) to be combined, the fifth parameter is the name of the fields in the output dataset.

e.g. `Method('syuds.mergeTable','cost','WHERE dCosts > 0','kco,job_num','dCosts','dMyCosts');`

would take only records with costs greater than zero and using kco and job\_num from this record write dCosts in to a combined field called dMyCosts.

A short version is also available with default options.

```
Method('syuds.merge','cost,rev','kco,job_num','CostRev');
```

This is equivalent to the series of calls above. Datasets ttCost and ttRev are combined using kco and job\_num (in all tables) and all fields from the two datasets are combined and returned in ttCostRev.

Decimal values will be summed in the combined dataset. All other field types will be written from the last record to be combined. It is expected that the record to be merged would already be unique on the keys.

## 6.1.7 Syuds.Pivot

A new program has been introduced that is designed to be called in the initialize calculation of an OA report. It is handed a dataset name and a series of criteria and transforms the named dataset in to three dynamic datasets which are designed to be easier to print in a matrix.

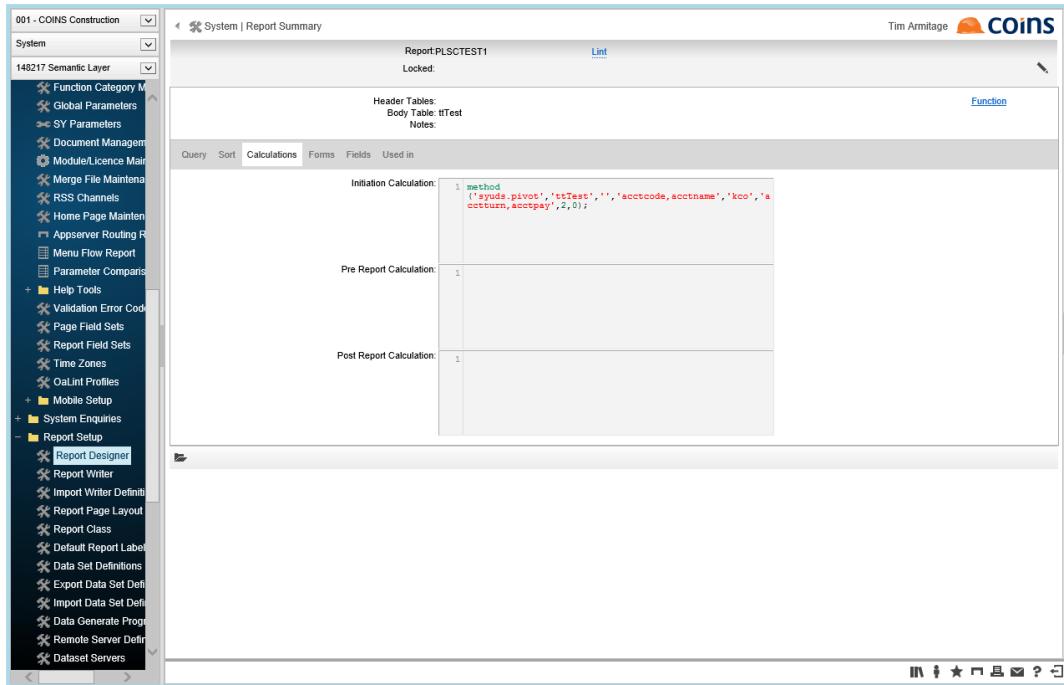
Syuds.Pivot take the following form:

```
method('syuds.pivot',[Source],'sumtype=""',[Row Fields],[Column Fields]','  
[Sum Fields],[Columns],[Total Column]);
```

Source	the name of the temp table source data that is required to be pivoted
SumType	sumtype="" is a condition to extract only selected records. SumType=" " will extract all records
Row Fields	the fields that will be used to create unique rows in the resulting pivoted dataset
Column Fields	The field(s) which will be used to create the columns of the array of data in the resulting pivoted dataset
Sum Fields	the fields to be summed and added to the resulting pivot dataset
Columns	the number of columns per page
Total Column	1 means add a total column (the sum of all the columns on the report), 0 means no total column and instead a total field will be added

The following calculation might be used in the initiation calculation of a report which contains a dataset ttTest. The dataset ttTest can be any type of temp table generated and prepared for a report.

```
method  
('syuds.pivot','Test','sumtype="","",acctcode,acctname','kco','acctturn,acctpay',2  
,1);
```



Suppose we had the following data :

Table ttTest

Kco	Acctcode	Acctname	Acturn	Acctpay	Sumtype
1	A	Supplier A	100	50	
2	A	Supplier A	150	0	
3	A	Supplier A	200	100	
	A	Supplier A	450	150	TOT
1	B	Supplier B	300	300	
	B	Supplier B	300	300	TOT

Running the pivot method above would result in three temp tables being created.

Table	Description
ttTestPage	containing page and column information
ttTestRow	containing row data
ttTestCol	containing column data

Table ttTestPage

iPageSequence	iColumn	cColumnLabel_1	cColumnLabel_2	bUsed_1	bUsed_2	bFirst	bLast	Kco_1	Kco2
1	1	1	2	Yes	Yes	Yes	No	1	2
2	3	3	Total	Yes	Yes	No	Yes	3	?

Table ttTextCol

iColumn	iPageSequence	iPageColumn	cColumnLabel	bTotalColumn	Kco
1	1	1	1	No	1
2	1	2	2	No	2
3	2	1	3	No	3
4	2	2	4	Yes	?

Table: ttTestRow

Acctnum	Acct- name	iCount_1	iCount_2	iCount_3	iCount_4	Acct- turn_1	Acct- turn_2	Acct- turn_3	Acct- turn_4	Acctpay_1	Acctpay_2	Acctpay_3	Acctpay_4
		_1	_2	_3	_4	turn_1	turn_2	turn_3	turn_4	_1	_2	_3	_4
A	Supplier A	1	1	1	3	100	150	200	450	50	0	100	150
B	Supplier B	1	0	0	1	300	0	0	300	300	0	0	300

If the total options is set to zero (no) then the final array entry is suppressed and separate total fields are created iCountTotal, AcctTurnTotal and AcctPayTotal with the values as you would expect.

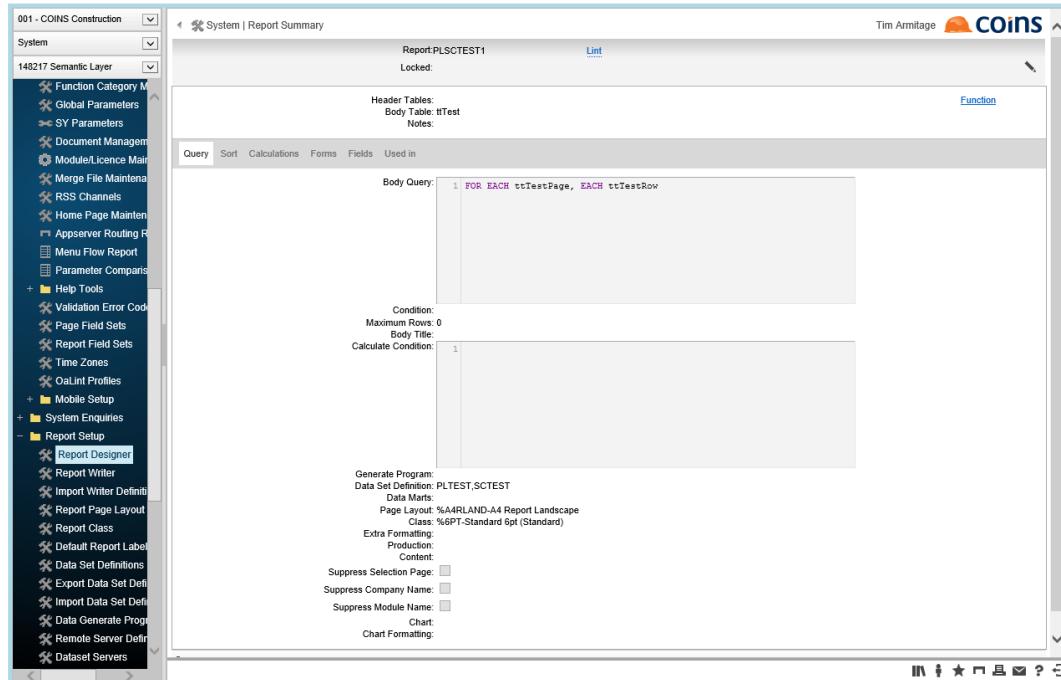
The records with SumType=TOT would not be processed because they were excluded with the query condition.

If a page size of 0 is specified then no paging will take place and a single ttPage record (iPageSequence=1) will be produced with the extent of the value fields being the same as the number of columns.

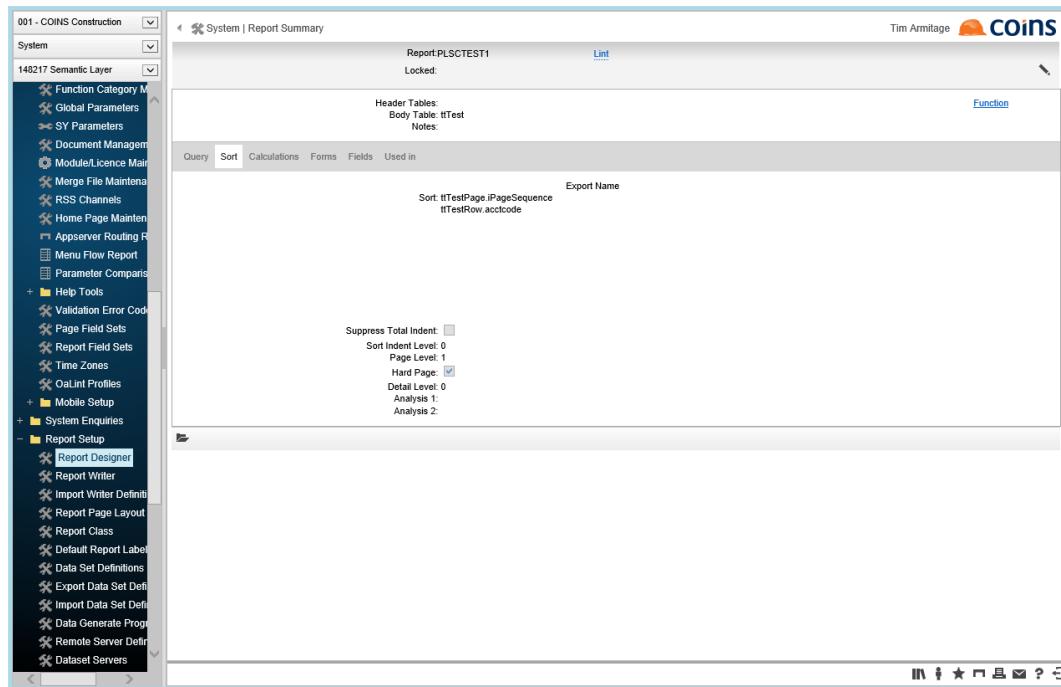
#### 6.1.7.1 Report Design

The temp tables (ttTestRow and ttTestPage) produced are designed to be used in OA designer and there are supporting methods and techniques to allow a matrix report to be built.

The query used on the report should be FOR EACH ttTestPage, EACH ttTestRow. This will repeat all the rows on each page sequence (assuming there are multiple sequences of pages for all the columns to be fitted on).

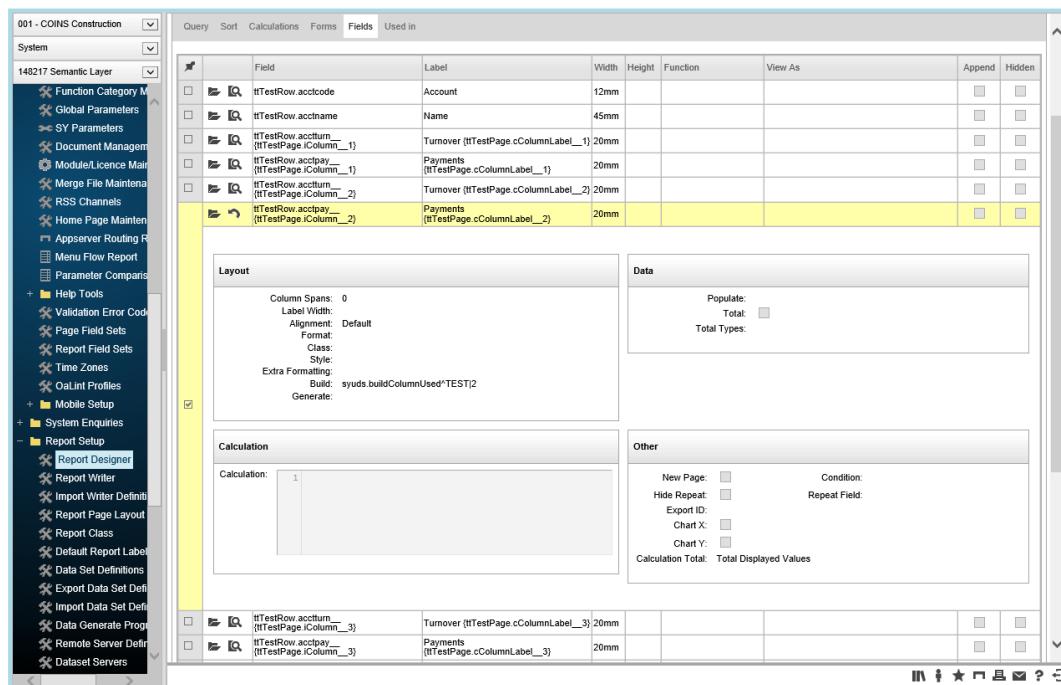


The sort order should include the page sequence at the top to allow the pages to come out in the correct sequence.



You should apply a hard page to the iPageSequence level.

The fields of the report should be set out as you require using fields from the ttTestRow temp table. The columns of the matrix can be specified using {ttTestPage.iColumn\_n} where n is the page column number. The columns can be labeled using {ttPage.cColumnNameLabel\_n} where n is the page column number. The total column will have a label of "TOTAL" in this field. The column fields are also replicated in ttPage.cKco\_n except that the total column will have a ? value in it.



A build condition should be used on the column fields so that they are not built when not used. The syuds.buildColumnUsed condition takes two parts to the parameter. The table name TEST (without the tt) and the page column number.

### 6.1.7.2 Sample Output

With a total field

Purchase Ledger - PLSC Combined1						
COINS Construction						
Account	Name	Turnover 1	Payments 1	Turnover 2	Payments 2	Total Turnover
A1	Abba Sealants Ltd	0.00	0.00	0.00	0.00	0.00
A100	ABC Project Controls Services	0.00	0.00	0.00	0.00	0.00
A10	A & E Projects Ltd	1,998.00	57.75	0.00	0.00	1,998.00
A1000	Airport Builders MerchantsX	14,000.00	10,829.10	0.00	0.00	14,000.00
A1001	Albion Services Ltd	1,998.00	0.00	0.00	0.00	1,998.00
A1001	ADM Contracts & Plant Hire	15,650.00	16,000.00	0.00	0.00	15,650.00
A1001	Abba Tiling Ltd**Co	9,762.50	10,000.00	0.00	0.00	9,762.50
A1002	Alba Andrews Brown, Cast Steel	112,040.00	0.00	0.00	0.00	112,040.00
A1002	Invoicing A1002	0.00	0.00	0.00	0.00	0.00
A1002	Albion Projects Ltd	4011.00	0.00	0.00	0.00	4,011.00
A1002	Albion Projects Ltd**Co	0.00	2,176.65	0.00	0.00	2,176.65
A1004	ARC Quarry Products Ltd	280.00	0.00	0.00	0.00	280.00
A1005	ARC East Scotland Oil Supply	695.00	190.00	0.00	0.00	695.00
A1007	Carter Inc	0.00	0.00	0.00	0.00	0.00
A1008	Carter Inc 2	0.00	0.00	0.00	0.00	0.00
A101	A & J Controls (g)	1,219.00	689.00	0.00	0.00	1,219.00
A11	A & J Beverage Ltd	-100.00	0.00	0.00	0.00	-100.00
A11	Albion Projects Ltd	0.00	0.00	0.00	0.00	0.00
A1100	Micro Controls & Eng Serv Ltd	0.00	0.00	0.00	0.00	0.00
A1101	ATS Scotland Tyre Service	0.00	0.00	0.00	0.00	0.00
A1101	ADM Contracts & Eng Serv Ltd	255.00	100.00	0.00	0.00	255.00
A1101	Albion Projects Ltd	0.00	0.00	0.00	0.00	0.00
A1178	Apex Industrial Ltd	0.00	0.00	0.00	0.00	0.00
A1181	John Consulting	0.00	0.00	0.00	0.00	0.00
A1234	Alba Fencing Ltd	0.00	0.00	0.00	0.00	0.00
A123405	ASD Andrews Brown, Cast Steel	0.00	0.00	0.00	0.00	0.00
A123405	Albion Projects Ltd**Co	0.00	0.00	0.00	0.00	0.00

With a total column

Purchase Ledger - PLSC Combined1						
COINS Construction						
Account	Name	Turnover 1	Payments 1	Turnover 2	Payments 2	Total Turnover
A1	Abba Sealants Ltd	0.00	0.00	0.00	0.00	0.00
A100	ABC Project Controls Services	0.00	0.00	0.00	0.00	0.00
A10	A & E Projects Ltd	1,998.00	57.75	0.00	0.00	1,998.00
A1000	Airport Builders MerchantsX	14,000.00	10,829.10	0.00	0.00	14,000.00
A1001	Albion Services Ltd	1,998.00	0.00	0.00	0.00	1,998.00
A1001	ADM Contracts & Plant Hire	15,650.00	16,000.00	0.00	0.00	15,650.00
A1001	Abba Tiling Ltd**Co	9,762.50	10,000.00	0.00	0.00	9,762.50
A1002	Alba Andrews Brown, Cast Steel	112,040.00	0.00	0.00	0.00	112,040.00
A1002	Invoicing A1002	0.00	0.00	0.00	0.00	0.00
A1002	Albion Projects Ltd	4011.00	0.00	0.00	0.00	4,011.00
A1002	Albion Projects Ltd**Co	0.00	2,176.65	0.00	0.00	2,176.65
A1004	ARC Quarry Products Ltd	280.00	0.00	0.00	0.00	280.00
A1005	ARC East Scotland Oil Supply	695.00	190.00	0.00	0.00	695.00
A107	Carter Inc	0.00	0.00	0.00	0.00	0.00
A108	Carter Inc 2	0.00	0.00	0.00	0.00	0.00
A101	A & J Controls (g)	1,219.00	689.00	0.00	0.00	1,219.00
A11	A & J Beverage Ltd	-100.00	0.00	0.00	0.00	-100.00
A11	Albion Projects Ltd	0.00	0.00	0.00	0.00	0.00
A1100	Micro Controls & Eng Serv Ltd	0.00	0.00	0.00	0.00	0.00
A1101	ATS Scotland Tyre Service	0.00	0.00	0.00	0.00	0.00
A1101	ADM Contracts & Eng Serv Ltd	255.00	100.00	0.00	0.00	255.00
A1101	Albion Projects Ltd	0.00	0.00	0.00	0.00	0.00
A1178	Apex Industrial Ltd	0.00	0.00	0.00	0.00	0.00
A1181	John Consulting	0.00	0.00	0.00	0.00	0.00

**Purchase Ledger - PLSC Combined1**  
COINS Construction



Account	Name	Total Turnover	Total Payments	Total Turnover	Total Payments
A1	Abba Geelani & Ltd	0.00	0.00	0.00	0.00
A10	Abbas Brothers	0.00	0.00	0.00	0.00
A10	A2 C Russell Ltd	1,989.00	572.75	0.00	0.00
A100	Airport Builders Merchant Ltd	14,000.00	10,829.10	0.00	0.00
A101	Airbus	1,000.00	1,000.00	0.00	0.00
A101	AGB Contract & Parallel	16,650.00	16,000.00	0.00	0.00
A101	Alba Tiling Ltd*co	9,762.50	0.00	0.00	0.00
A101	Alba Tiling Ltd*co Castlefield	11,240.00	0.00	0.00	0.00
A102	Albucip A103	0.00	0.00	0.00	0.00
A102	AT16	4,011.00	0.00	0.00	0.00
A102	Alba Contracting Ltd*co	0.00	0.00	0.00	0.00
A104	Alba Roofing Ltd*2*	0.00	2,176.65	0.00	0.00
A104	ARC Oilyarn Products Ltd	280.00	0.00	0.00	0.00
A105	Arden East Scotland Oil Supply	985.00	194.00	0.00	0.00
A107	Arnefex	0.00	0.00	0.00	0.00
A108	Carriano	1,219.00	1,738.00	0.00	0.00
A108	Carronite G	0.00	689.00	0.00	0.00
A109	AS & J Best Edge Ltd	-100.00	0.00	0.00	0.00
A11	Asheer Ltd	0.00	0.00	0.00	0.00
A110	ATS Scotland Tyre Sales Ltd	0.00	0.00	0.00	0.00
A110	ATS Scotland Tyre Sales Ltd	285.00	100.00	0.00	0.00
A111	ATP Contracting Ltd	0.00	0.00	0.00	0.00
A117	Apex Industrial Ltd	0.00	0.00	0.00	0.00
A118	Joe K Costings	0.00	0.00	0.00	0.00
A120	ASD Contracting Ltd	0.00	0.00	0.00	0.00
A124	Alba Fencing Ltd	0.00	0.00	0.00	0.00
A125	ASD Contracting Ltd Castlefield	0.00	0.00	0.00	0.00
A126	ASD Contracting Ltd Castlefield	0.00	0.00	0.00	0.00
A130	Ashtead Plant Factor Ltd	0.00	0.00	0.00	0.00
A130	Ashtead Platform & Co. Ltd	0.00	0.00	0.00	0.00
A131	Ashtead Platform & Co. Ltd	0.00	0.00	0.00	0.00
A138	ASL Building Co. C	0.00	0.00	0.00	0.00
A141	Andrew Weatherhead Ltd	0.00	0.00	0.00	0.00
A145	Angie Decking Co Ltd	0.00	0.00	0.00	0.00
A160	ACS Building Ltd	0.00	0.00	0.00	0.00

## 6.1.8 Repeat

This method removes repeated fields from a temp table. Typical usage is if the data set contains header and detail data and the repeating header details only want to be shown (and aggregated) once. This method will allow you to manipulate the dataset so that repeated values only appear once.

```
Method('syuds.repeatsource','Source','kco,job_num');  
Method('syuds.repeatfields','1','kco,coc_name');  
Method('syuds.repeatfields','2','job_num,job_name,RO_job_costs^TD');  
Method('syuds.repeatExec');
```

The ttSource table will be replaced with a copy where the repeating fields kco, coc\_name and job\_num,job\_name are shown once per kco and job\_num sort sequence.

repeatSource specifies the source table and the sort fields to be used.

repeatFields specifies for each of the sort sequence the fields that should be shown just once

In this example RO\_job\_costs^TD (and job\_num and job\_name) are shown only once for each job\_num within kco. This would allow this column to be aggregated on the report.

repeatExec actually performs the repeat field blanking and saves the temp table overwriting the original source data.

## 6.1.9 Syuds.Store

This option is identical to the data mart writing on the tail end of a report except that you control the date/time of the extract.

The datamart must be defined in the usual way and the mapping of the fields in the datamart to the fields in the dataset must be configured.

For example

```
Method('syuds.store',100,'job,cost','JOBDM,COSTDM',TODAY,0);
```

Would store the dataset ttJob in datamart JOBDM and similarly dataset ttCost in datamart COSTDM with an extract date of midnight on the day of running in company 100. The extract date would be expected to be a report input/selection value.

## 6.1.10 Sum

This method will combine and sum rows from two or more dataset in to one or more new datasets.

Suppose we have a dataset ttCostRev

Kco	Job_num	dCosts	dRev
100	1000	100	150
100	1001	200	250
100	1002	0	100
100	1003	100	0

Then the resulting dataset ttKcoSum

Kco	dCosts	dRev
100	400	500

would be produced with the following commands

```
Method('syuds.sumSource','CostRev','');
Method('syuds.sumTable','KcoSum','kco','','dCosts,dRev');
Method('syuds.sumExec');
```

SumSource specifies the input dataset and a condition to apply to that set of records.

sumTable specifies the output summary required. The first parameter is the output dataset name (without the tt prefix), the second parameter is the key fields, the third parameter is other fields to be assigned (similar to key fields but not used to find uniqueness e.g. keys=kco, fields=coc\_name), the fourth parameter is the fields to sum.

sumExec executes the summing and creates the required output datasets. Multiple sumTable methods may be used to create multiple summaries on a single pass through the source data.

A short version with defaults exists

```
Method('syuds.sum','CostRev','','KcoSum','kco','','dCosts,dRev');
```

With the following parameters input table, condition, output table, keys, fields, sum fields.

### 6.1.1 TableAlias

```
Method('syuds.tableAlias','source','output');
```

This takes a table and renames it for the purposes of using a table multiple times.

1<sup>st</sup> parameter is the source dataset table and the 2<sup>nd</sup> parameter is the new dataset table.

## 6.1.12 Syuds.TimeSlice()

This is a new method in syuds.p to manipulate an input dataset or datasets in to a new summarized dataset which has a date/time element.

The method can be called like any other post processing method on a dataset, for example

```
method('syuds.timeslice','FOR EACH ttStats','TimeStats','{fromtime}','{totime}',  
'{interval}','{timeunit}','moe_key1,moe_key2','moc_snapshot','')
```

The parameters are as follows:

1	Query	The query to be run against existing dataset(s) which will return a set of record containing the data that is to be time sliced
2	TableName	The output table name
3	From Date/Time	A string containing a valid from date or date/time (depending on the units being used). If omitted then the time range will not be completed and records will only exist for the data being processed
4	To Date/Time	Same as From date/time
5	Interval	An integer value to be used in conjunction with the time unit field to allow the creation of the complete set of time records
6	Time Unit	"S"econds (for date/time), "D"ay, "W"eek, "M"onth, "Y"ear
7	Keys	The key fields used to summarise the data. Can be left blank and just the time element will be used
8	Time Key	The field in the source data that contains the date/time field to be used to slice the data
9	Fields	The fields from the source data to be aggregated. If left blank then ALL decimal fields in the source tables will be aggregated.
		For each field that is aggregated the total, max, min and average values for the time slice will be evaluated.

If the Time Unit field is "S" for seconds then the Interval is the number of seconds for each time slice and the start and end date/time values are date/time format strings. E.g. From=15/01/16 08:00&To=15/01/16 18:00&Interval=3600 would produce hourly slices from 8am until 6pm.

If the Time Unit field is "D" for days then the interval is not used and single day records are produced between the from and to dates.

If the Time Unit field is "W" for weeks then the interval indicates the day of the week for the week ending i.e. 1 for Sunday to 6 for Saturday. Week end dates for that day are then produced between the start and end dates.

If the Time Unit is "M" for months then the interval is ignored and calendar month end dates are used between the start and end dates.

If the Time Unit is “Y” for years then the interval is ignored and calendar year end dates are used between the start and end dates.

The results dataset will contain four fields for each of the aggregated fields (one of the same name for the total and then extensions min, max and avg for the other three values). A count is always added plus the unique grouping key and key date/time.

### 6.1.13 Top

This method returns the Top n records within a dataset when a sort is defined.

```
Method('syuds.top','FullList','drev>100000','by drev descending',10,'Top10Rec');
```

The 1<sup>st</sup> parameter is the source temp-table. The 2<sup>nd</sup> parameter is the condition applied to the query, the 3<sup>rd</sup> parameter is the sort order, the 4<sup>th</sup> parameter is the number of records returned , the 5<sup>th</sup> parameter is the output table name.

In the example above it is taking all the records from the ttFullList table and for records which revenue exceeds 100000 then it sorts it by the revenue field (highest to lowest) and returns the top 10 records into a table called ttTop10Rec.

## 6.1.14 Union

This method will combine rows from two or more dataset in to a new dataset.

Suppose we have a dataset ttCost

Kco	Job_num	dCosts
100	1001	100
100	1002	150

And a dataset ttRev

Kco	Job_num	dRev
100	1001	200
100	1002	250

Then the resulting dataset (ttCostRev) might be

Kco	Job_num	dCosts	dRev
100	1001	100	0
100	1002	150	0
100	1001	0	200
100	1002	0	250

This would be achieved with the following method calls.

```
Method('syuds.unionFields','kco,job_num,dCosts,dRev');
Method('syuds.unionTable','cost','','kco,job_num,dCosts,');
Method('syuds.unionTable','rev','','kco,job_num,,dRev');
Method('syuds.unionExec','CostRev');
```

unionFields specifies the fields in the returned dataset.

unionTable specifies the source dataset to combine. The first parameter is the dataset name (without the tt prefix), the second parameter is a condition for the records to select from this source dataset, the third parameter is the fields to combine.

unionExec specifies the output dataset and executes the union.

If the fields specified are the same then the values are combined in to the same field

e.g.

```
Method('syuds.unionFields','kco,job_num,dValue');
Method('syuds.unionTable','cost','','kco,job_num,dCosts');
Method('syuds.unionTable','rev','','kco,job_num,dRev');
Method('syuds.unionExec','CostRev');
```

Would produce the output dataset

Kco	Job_num	dValue
100	1001	100
100	1002	150
100	1001	200
100	1002	250

There is also a short version with default options.

```
Method('syuds.union','cost,rev','kco,job_num,dCosts,dRev','CostRev');
```

ttCost and ttRev would be combined with fields kco,job\_num,dCosts,dRev in to dataset CostRev. This is equivalent to the first example above.

## 7 Data Set Exercises

This section will run through several exercises exploring how Data Sets can be used in reporting.

# Discovery BI0011

## 7.1 Using a Dataset in a Report

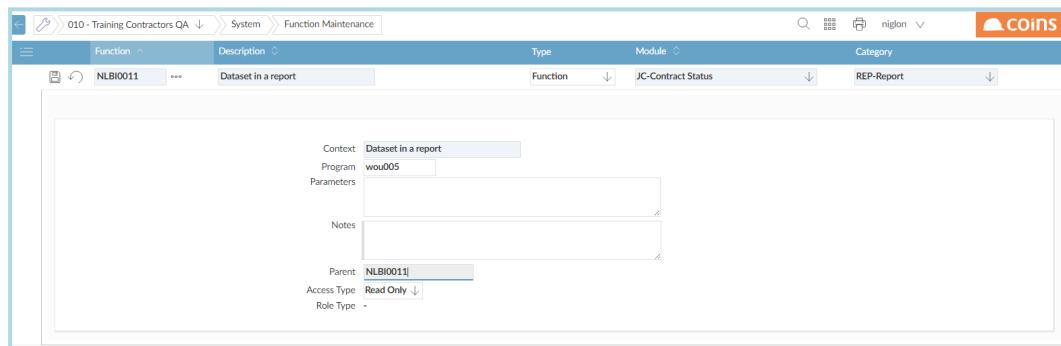


You should complete Discovery BI0009 before attempting this exercise.

### 7.1.1 Create the Function and Menu

Create the function that will be used to run the report then add this function to an appropriate menu.

Function Code	Initials + BI0011 e.g. NLBI0011
Description	A relevant description
Function Module	Contract Status
Function Category	Report
Function Context	Leave as defaulted from the Function Description.
Function Program	wou005



Click

### 7.1.2 Set up the Page

- Navigate to OA Reporting & BI > Designer > Page Designer

- Click to create the new Page and fill in the fields as follows (leave the others blank):
  - Give the page section the same name as the function created.
  - Form Service Procedure: **jcfrep.p**



Click to save the new Page. COINS returns to the summary for the page section being created.

### 7.1.3 Add the Page Section Forms

Select the Form tab create an Update form.

Click and

and enter the following information to create the Form :

Form	Select Update
Field Type	Select Update
Linked To	Leave blank

Click to create the form.

## 7.1.4 Add the Fields

 In the Field Tab go to the FORM selector at the bottom of the page, select Update and click Apply Filter.



You should always do this to ensure you are adding fields to the correct form - adding fields to an incorrect form will mean that they do not display.

Click



Add a Field, leaving Field and Label blank :

In the Generate field, enter jcfrep.jobSelectionGenerate. This is the standard selection generator for contracts.

Click to create the field.

Test the report selection by adding &mainarea=xxBI0011 (where xx are your initials) to the end of the URL

COINS should display the selection criteria:

The screenshot shows the COINS OA Designer Module 1 interface. On the left is a navigation menu with various modules like Home, Bills & Quantities, Bonds & Guarantees, Business Planning & Forecasting, Cash Book, Cash Profiles, Central Repository, Commercial Manager, Commercials, Contract Sales, Contract Status, Customer Care, CVR, Employee Self Service, Expenses, Facilities Management, Fixed Assets, Forecasting, General Ledger, Health & Safety, House Sales, Human Resources, Inspections, Land Appraisal, Lettings Management, Marketing, OA Reporting & BI, Payroll, Plant Control, Procurement, Project Management, Purchase Ledger, Sales Ledger, and Service Management. The main area is titled '010 - Training Contractors QA' and 'Contract Status'. It shows a 'Dataset in a report' section with 'Main' and 'Output Options' tabs. Under 'Main', there are filters for 'From' and 'To' fields, and a 'Matches' section with dropdowns for Contract, Group, Type, Location, Manager, Customer, and Status of Contract. Below these are 'Active' and 'Complete' dropdowns set to 'Both'. An 'Include Future Contracts' checkbox is also present. At the bottom right are 'UNDO' and 'NEXT' buttons.

If you were to click to generate the report it will not produce any output, because the report section has not yet been created. If you refer to the log file you will see an error message similar to the one below.

The log file screenshot shows the following error message:

```

Start Time:09/01/2017 15:27:54.856+00:00
Database User :9
Report Process:4751
09/01/2017 15:27:54.869+00:00 COINSSystemError: Report does not exist: NLBI0011 Stack: syurep.p,pm99.p,pm99b.p
09/01/2017 15:27:54.870+00:00 Report does not exist: NLBI0011 Stack: syurep.p,pm99.p,pm99b.p
End Time:09/01/2017 15:27:54.872+00:00
Timings: Startup- ? Generate- ? Prepare- ? Production- ? PDF- ? Completion- ? Total- 16

```

## 7.1.5 Create the report section



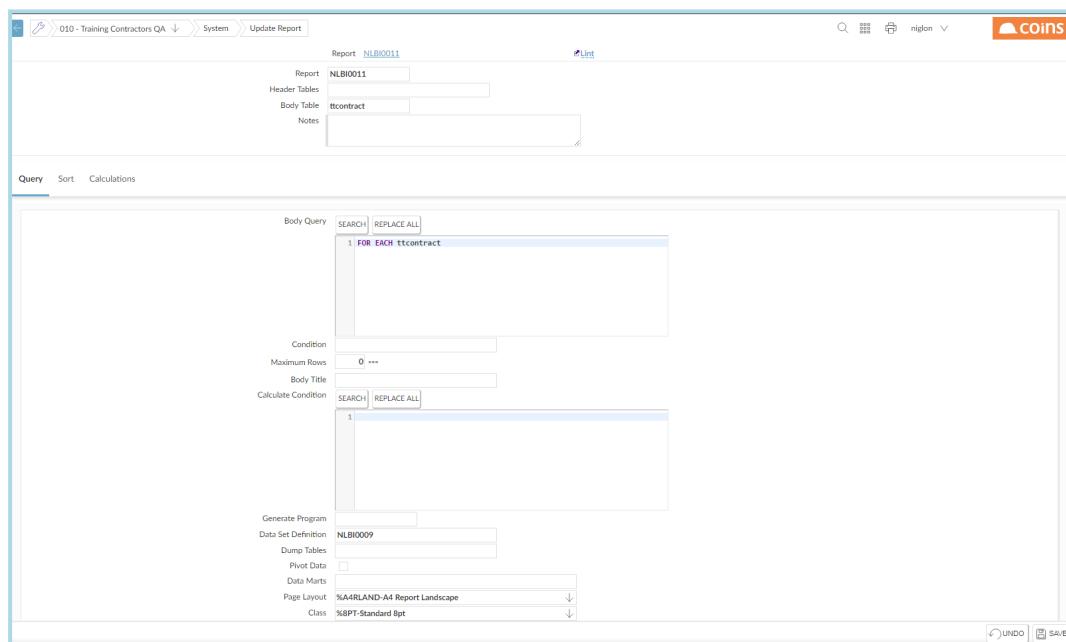
Navigate to OA Reporting & BI > Designer > Report Designer



Click to create a new report section and fill in the details as follows:

Field	Details
Report Section	Give the report section the same name as the function created.
Body Table	Enter the name of the database table being reporting on; in this case our temporary table from the dataset ttcontract

Field	Details
Body Query	<p>Enter the query. This will select which records are shown on the report. Enter the following:</p> <pre>FOR EACH ttcontract</pre> <p>This query selects all records in the dataset. Note that in this exercise we do not need to specify the kco or the contract selection {jobSelect} as we have in earlier exercises because these have all been done in the dataset query.</p>
DataSet	Specify the name of the Data Set that contains the temporary table. In this case xxBI0009 (replace xx with your initials).
Page Layout	<p>Select %A4RLAND-A4 Report Landscape .</p> <p>The page layout determines the orientation (whether the report format is landscape or portrait), the margins, and the standard headers and footers. For COINS reports, with a logo, a title, and a report footer, choose a report layout (with an R in the name: %A4RLAND or %A4RPORT).</p>
Class	Select the font class to use. Since the user listing report doesn't have many fields on, use Arial 8pt (a larger font than Arial 6pt).



Click to save the section and present additional tabs.

### 7.1.6 Sorting the Report

On the sort tab, enter ttcontract.job\_num+ in the Sort field

The screenshot shows the 'Sort' tab of the OA Designer interface. It includes fields for 'Sort' (containing 'ttcontract.job\_num+') and 'Export Name'. There are also sections for 'Suppress Total Indent' and 'Sort Indent Level' (set to 0), 'Page Level' (set to 0), 'Hard Page', 'Detail Level' (set to 0), 'Analysis 1', and 'Analysis 2'. At the bottom right are 'UNDO' and 'SAVE' buttons.

Click **Save** **SAVE**

Select the Form tab create a Body form. This will be used to display the report line fields.

Click **Add** **+ ADD** and enter the following information to create the Form:

Form	Body
Field Type	Body
Linked To	Leave blank

The screenshot shows the 'Forms' tab of the OA Designer interface. It displays a table with columns for 'Form', 'Field Type', 'Linked To', and 'Class'. The 'Form' column has a dropdown set to 'Body'. The 'Field Type' column is also set to 'Body'. The 'Linked To' and 'Class' columns are empty. Below the table are sections for 'Extra Formatting', 'Build', 'Use', 'Hidden', 'Pre Form Calculation' (with 'SEARCH' and 'REPLACE ALL' buttons), and 'Post Form Calculation' (with 'SEARCH' and 'REPLACE ALL' buttons).

Click to create the form.

## 7.1.7 Add the Fields

In the Field Tab go to the FORM selector at the bottom of the page, select Body and click Apply Filter.



You should always do apply the filter to ensure you are adding fields to the correct form - adding fields to an incorrect form will mean that they do not display.

For each of the fields in our dataset to be shown on the report, fill in the following:

- |       |  |
|-------|--|
| Field | The name of the field.                     |
| Label | The label to appear in the column heading. |
| Width | The width of the column on the report.     |

Field	Label	Width
job_num	Contract	15mm
job_name	Contract Name	60mm
costs_td	Costs TD	20mm
rev_td	Revenue TD	20mm
prof_td	Profit TD	20mm

Click  after entering each field.

The report is now ready to run.

## 7.1.8 Running the Report

Run the report selection by adding &mainarea=xxBI0011 (where xx are your initials) to the end of the URL

- Enter selection criteria.
- Click on the Output Options tab and select the appropriate options.
- Click  to run the report.

### Example Output:

Contract	Contract Name	Costs TD	revenue TD	Profit TD
1000	Oxford Road Office.	2,521,712.24	10,801,715.45	8,280,003.21
1001	Holiday Inn Slough	5,418,730.39	7,643,419.54	2,224,689.15
1002	Terminal 5 retail fitout	6,488,047.81	11,633,810.01	4,745,763.20
1100	A406 Ealing underpass	857,097.61	2,047,277.06	1,683,129.45
1101	Wembley stadium replacement	300,000.00	870,000.00	570,000.00
1102	Eton River Thames bank rebuilding	555,629.23	930,159.67	374,530.44
1103	Heathrow T4 hydrant fuel insta	387,821.36	622,336.05	234,514.69
1104	M34 Northampton - Southampton	21,870,051.32	21,408,916.57	-461,134.75
1111	Mxd Development at Precinct 14	15,000.00	0.00	-15,000.00
1115	M61 Extension	6,325.00	46,342.67	40,017.67
1200	Bluewater shopping M & E	5,061.30	1,289,733.19	1,284,671.86
1401	NTL Cabling installation	926.00	67,879.00	66,953.00
1406	FM - Wembley National Stadium	200.00	120.00	-72.00
500	test005	0.00	0.00	0.00
501	ed	0.00	0.00	0.00
6000	FM Contract	-400.00	80.00	480.00
6100	FM Property Maintenance	44.00	4,848.00	4,804.00
6110	Oxford CC Maintenance	0.00	0.00	0.00
6200	A465 Vale of Neath Bridge Hirwaun	395.00	6,775.60	6,380.60
987	987	0.00	0.00	0.00
BA1100	CONTRACT PHASE	0.00	0.00	0.00
EBB001	LeverEdge Test Contract	0.00	0.00	0.00
EBB002	Euro Contract	69.45	0.00	-69.45
FSSO	FSS Overheads	0.00	0.00	0.00
L1000	Frenchay Refurb Project	0.00	0.00	0.00
P001	Discontinued contract	0.00	0.00	0.00
PLANT	Central Plant Re-Hire Contract	0.00	0.00	0.00
PS001	ps001	0.00	0.00	0.00
SAMTEST	Sam's Test Contract	0.00	0.00	0.00
T1	T1	0.00	0.00	0.00
T2	T222	0.00	0.00	0.00
TM099	Tim D Test	0.00	0.00	0.00
W100	Singapore Airport	0.00	0.00	0.00
Z001	Trial for OA reporting xxx	99,937.50	0.00	-99,937.50
z004	test auto number sequence	150.00	0.00	-150.00

Printed using COINS V10.23.110802+ by Nigel Longley at 14:33:59 on 24/01/12 (NLM2Exercise)

Page 1

# Discovery BI0012

## 7.2 Using multiple Data Sets

This exercise will demonstrate how multiple Data Sets can be used in a report.



You will need to have completed Discoveries BI0009 and BI0011 before attempting this exercise.

### 7.2.1 Create the Function and Menu

Create the function that will be used to run the report then add this function to an appropriate menu.

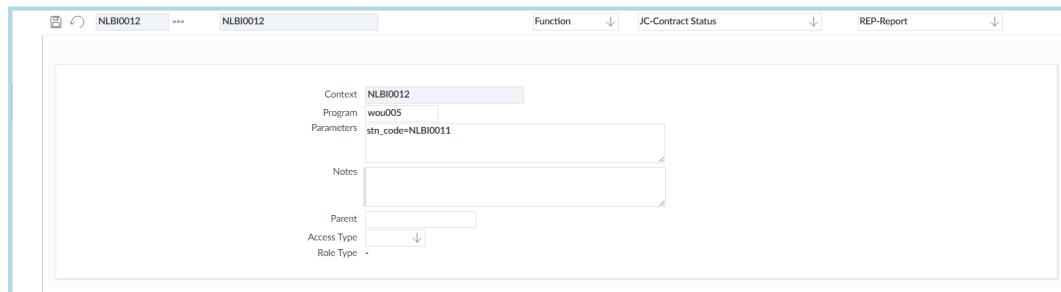


Replace xx with your initials.

Function Code	xxBI0012
Function Name	xxBI0012
Function Module	Contract Status
Function Category	Report
Function Context	Leave as defaulted from the Function Description
Function Program	wou005

We already have a page that we can re-use from Discovery BI0011 so the following parameter will allow us to use this:

Function Parameters



Click

### 7.2.2 Creating the new Data Set

In Designer, navigate to Data Set Definitions.

Click

Enter a Data Set name and description for the data set. For this exercise use your initials followed by BI0012.

In the Query box, enter:

```
FOR EACH jc_job WHERE jc_job.kco = {kco} {jobSelect},
  EACH ap_invoice OF jc_job .
```

We need to create an identifier for the Data Set table name. In this example call your table plinvoice

Click



### 7.2.3 Adding the Fields

On the Fields Tab, click



and add the following fields:

Field	Label	Data Type	Format	Source
kco	Company	Integer	->>>,>>>,>>9	jc_job.kco
job_num	Contract	Character	x(8)	jc_job.job_num
plgross	PL Gross	Decimal	->>>,>>>,>>9.99	ap_invoice.ain_amount

For both kco and job\_num tick the field called Key. This will summarise the Data Set by company and contract.

Field	Key	Extent	Label	Data Type	Format	Source
kco	<input checked="" type="checkbox"/>	Company	Company	Integer	->>,>>>,>>9	jc_job.kco
job_num	<input checked="" type="checkbox"/>	Contract	Contract	Character	x(8)	jc_job.job_num
plgross		PL Gross	PL Gross	Decimal	->>,>>>,>>9.99	ap_invoice.ain_amount

## 7.2.4 Create the Report

Copy the report definition for xxBI0011 and rename it xxBI0012.

For this new report, modify the query to read our two Data Set tables:

```
FOR EACH ttcontract,
FIRST ttPLinvoice WHERE ttPLinvoice.kco = ttcontract.kco
AND ttPLinvoice.job_num = ttcontract.job_num
```



There must be a relationship between the fields in each of the Data Sets.

In the Data Set Definition field, list the Data Sets being used separated by commas

For example:

NLB10009,NLB10012

Click

On the Fields Tab, on the Body Form add an additional column for plgross

Field	Label	Width	Height	Function	View As	Append	Hidden
job_num	Contract	20mm				<input type="checkbox"/>	<input type="checkbox"/>
job_name	Contract Name	60mm				<input type="checkbox"/>	<input type="checkbox"/>
costs_td	Costs TD	20mm				<input type="checkbox"/>	<input type="checkbox"/>
rev_td	revenue TD	20mm				<input type="checkbox"/>	<input type="checkbox"/>
prof_td	Profit TD	20mm				<input type="checkbox"/>	<input type="checkbox"/>
plgross	PL Gross	20mm				<input type="checkbox"/>	<input type="checkbox"/>

Click

## 7.2.5 Running the report

Run the function xxBI0012. You should get output similar to the following:

**Contract Status - NLB0012**  
Training Contractors QA



Contract	Contract Name	Costs TD	revenue TD	Profit TD	PL Gross
10000	Penny Hill Estates	12,550.00	156,300.00	143,750.00	37,222.76
1001	PF Training Project	0.00	0.00	0.00	0.00
10010	New Wing	1,200.00	0.00	-1,200.00	2,054.00
1007	Hamptons Hospital Construction	57,600.00	0.00	-57,600.00	71,380.00
1111	Milton Keynes Building	4,300.00	0.00	-4,300.00	8,315.00
11111	Y2Project	12,280.00	0.00	-12,280.00	14,736.00
12007	Cambridge Fens	0.00	0.00	0.00	0.00
12500	buy bricks	1,000.00	0.00	-1,000.00	1,187.50
20014	Riverside Brassiere The	3,085.00	0.00	-3,085.00	3,702.00
20015	Eastside Locks	1,355.00	0.00	-1,355.00	1,490.50
20016	Midshires Business Park	1,355.00	0.00	-1,355.00	1,626.00
20017	Beeston Business Centre	1,355.00	0.00	-1,355.00	1,626.00
2012	The End Of The World	1,500.00	0.00	-1,500.00	1,800.00
20777	Bykings Shopping Centre	2,490.00	0.00	-2,490.00	2,968.00
21007	Masons Estate	11,282.50	0.00	-11,282.50	13,539.00
21107	Grenville Business Estate	230,628.28	0.00	-230,628.28	174,314.30
21111	Meadows Parks Estates	466.57	0.00	-466.57	559.88
21122	Windsor House Derby	8,410.00	0.00	-8,410.00	14,052.00
211231	Windsor Homes	13,200.00	0.00	-13,200.00	15,840.00
21124	Lace	1,060,000.00	0.00	-1,060,000.00	1,812,000.00
21126	Leisure Centre	18,810.00	0.00	-18,810.00	27,480.00
21130	Castle warehousing	2,249.50	0.00	-2,249.50	3,783.20
21135	Hanger Lane Estates	400.00	0.00	-400.00	1,200.00
21150	Slough Warehousing	3,600.00	0.00	-3,600.00	4,660.00
21160	Vision Cineplex	4,240.00	0.00	-4,240.00	7,836.00
4000	Circle Line Upgrade	666,000.00	0.00	-666,000.00	878,400.00
8000	Cal State University	0.00	0.00	0.00	155,078.24
8888C	tes8888	8,890.00	0.00	-8,890.00	12,492.50
9000	Cal State Industrial park	22,610.00	0.00	-22,610.00	30,445.85
B5555	New Build School	11,708.45	0.00	-11,708.45	478.70
FM57	Berkshire Schools FM Co Locations	49,529.68	579.20	-48,950.48	39,891.26
ng1	ng1	870.00	0.00	-870.00	1,104.00
PM01	Project Management Test Project	1,787.00	0.00	-1,787.00	9.45

Printed using COINS V11.06.161022SP1911-30/12/16 by Nigel Longley at 10:34:45 on 10/01/17 (NLB0012)

Page 1

# Discovery BI0013

## 7.3 Using multiple Data Sets to merge data



You will need to have completed Discovery BI0009 before carrying out this exercise

This exercise will demonstrate how multiple Data Sets can be used to merge information from different tables in a report.

### 7.3.1 Create the Function and Menu

Create the function that will be used to run the report then add this function to an appropriate menu.

Function Code	xx(Initials) + BI0013 e.g. NLBI0013
Function Name	xxBI0013
Function Module	Contract Status
Function Category	Report
Function Context	Leave as defaulted from the Function Description.
Function Program	wou005

We already have a page that we can re-use from the first exercise so the following parameter will allow us to use this:

Function Parameters

stn\_code=NLCI0009

### 7.3.2 Creating the first new Data Set

In Designer, navigate to Data Set Definitions.

Select the add icon .

Enter a Data Set name and description for the data set. For this exercise use your initials followed by BI0013.

In the Query box, enter:

```
FOR EACH jc_job WHERE jc_job.kco = {kco} {jobSelect},  
  EACH ar_invoice OF jc_job .
```

We need to create an identifier for the Data Set table name. In this example call your table SLCSinvoice

### 7.3.3 Adding the Fields



On the Fields Tab, click  and key in the following field:

Field	Label	Data Type	Format	Source
kco	Company	Integer	->>,>>,>>9	jc_job.kco
job_num	Contract	Character	x(8)	jc_job.job_num
slgross	SLGross	Decimal	->>,>>,>>9.99	ar_invoice.rin_amount

For both kco and job\_num tick the field called Key. This will summarise the Data Set by company and contract.

The third field is another direct source field:

Definition Fields							
	Field	Key	Extent	Label	Data Type	Format	Source
<input type="checkbox"/>	kco	<input checked="" type="checkbox"/>		Company	Integer	->>,>>,>>9	jc_job.kco
<input type="checkbox"/>	job_num	<input checked="" type="checkbox"/>		Contract	Character	x(8)	jc_job.job_num
<input type="checkbox"/>	SLGross	<input type="checkbox"/>		SL Gross	Decimal	->>,>>,>>9.99	ar_invoice.rin_amount

#### 1.4.4 Creating the second new Data Set

In Designer, navigate to Data Set Definitions.

Click Add.

Enter a Data Set name and description for the data set. For this exercise use your initials followed by M2DS4.

In the Query box, enter:

```
FOR EACH jc_job WHERE jc_job.kco = {kco} {jobSelect},
EACH cs_certificate OF jc_job .
```

We need to create an identifier for the Data Set table name. In order for the data to be merged, our data sets must all share the SAME table name, so we will use SLCSSinvoice here too.

Data Set: NLM2DS4    Data Set Example - Merge  
Notes: Contract Sales Transactions

Definition	Fields
Generate Program:	Query: 1 FOR EACH jc_job WHERE jc_job.kco = {kco} {jobSelect}, 2 EACH cs_certificate OF jc_job
Condition:	
Raw DB Query:	
Table Name:	SLCSSinvoice

#### 7.3.4 Adding the Fields

On the Fields Tab, click Add and key in the following field: Note again that although the source data and/or calculations may be different it is vital that the same field names and labels etc are used across all the datasets to be merged. Failure to do this will cause the report to fail.

FieldLabelData TypeFormatSource

kcoCompanyInteger->>,>>,>>9jc\_job.kco

job\_numContractCharacterx(8)jc\_job.job\_num

For both kco and job\_num tick the field called Key. This will summarise the Data Set by company and contract.

The third field is another direct source field:

FieldLabelData TypeFormatSource

slgrossSL GrossDecimal->>>,>>>,>>9.99cs\_certificate.cst\_cur\_gross\_\_1

	Field	Key	Event	Label	Data Type	Format	Source
<input type="checkbox"/>	kco	<input checked="" type="checkbox"/>		Company	Integer	>>>,>>>,>>9	jc_job.kco
<input type="checkbox"/>	job_num	<input checked="" type="checkbox"/>		Contract	Character	x(8)	jc_job.job_num
<input type="checkbox"/>	SLGross		<input type="checkbox"/>	Gross	Decimal	>>>,>>>,>>9.99	cs_certificate.cst_cur_gross__1

#### 1.4.6 Create the Report

Copy the report definition for xxM2exercise1 and rename it xxM2exercise1b.

For this new report, modify the query to read The contract table jc\_job and then link it to our datasets - note that we only reference the single table name.

```
FOR EACH jc_job WHERE kco = {kco}{jobSelect},
EACH ttSLCSInvoice WHERE ttSLCSInvoice.kco = jc_job.kco
AND ttSLCSInvoice.job_num = jc_job.job_num
```

In the Data Set Definition field, list the Data Sets being used separated by commas.

NLM2DS3,NLM2DS4

ReportNLM2Exercise1b

Header Tables: co\_config  
Body Table: jc\_job  
Notes: DATASET MULTIPLE & MERGE

Query Sort Calculations

Body Query:

```

1. FOR EACH jc_job WHERE kco = {kco}{jobSelect},
2. EACH ttSLCSinvoice WHERE ttSLCSinvoice.kco = jc_job.kco
   AND ttSLCSinvoice.job_num = jc_job.job_num

```

Condition:   
Maximum Rows: 0  
Body Title:   
Calculate Condition:

Generate Program:   
Data Set Definition: NLM2DS1,NLM2DS4  
Data Marts:   
Page Layout: NARLAND-A4 Report Landscape  
Class: %SPT-Standard 8pt  
Extra Formatting:   
Production:   
Content:   
Suppress Selection Page:   
Suppress Company Name:   
Suppress Module Name:   
Chart:   
Chart Formatting:

On the Body fields, delete all the columns after job name and replace with the following:

FieldLabelwidthCalculation  
 RO\_ContractCosts^TDCosts TD20mmCostsTD = this;  
 RO\_ContractRevenueRevenue TD20mmRevTD = this;  
 Profit TD20mmRevTD - CostsTD;  
 ttSLCSinvoice.SLGrossSales Balance20mm

	Field	Label	Width	Height	Function	View As	Append	Hidden
<input type="checkbox"/>	jc_job_num	Contract	20mm				<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	jc_job_name	Contract Name	60mm				<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	RO_ContractCosts^TD	Costs TD	20mm				<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	RO_ContractRevenue^TD	Revenue TD	20mm				<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Profit TD	Profit TD	20mm				<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	ttSLCSinvoice.SLGross	Sales Balance	20mm				<input type="checkbox"/>	<input type="checkbox"/>

Query Sort Calculations Forms Fields Used in

Header Tables: co\_config  
Body Table: jc\_job  
Notes: DATASET MULTIPLE & MERGE

Add Copy Insert Up Down Open Multi Bulk:  Detail Delete Transfer Fields

Q+ Advanced Selectors Form: Body View: Grouped

### 7.3.5 Running the report

Run the function xxM2Exercise1b. You should get output similar to the following:

**JC Contract Status - NL M2 Exercise 1b**  
Contractors QA



Contract	Contract Name	Costs TD	Revenue TD	Profit TD	Sales Balance
10000	Penny Hill Estates	0.00	155,000.00	155,000.00	182,125.00
20014	Riverside Brassiere The	3,085.00	0.00	-3,085.00	35,250.00
20015	Eastside Locks	1,355.00	0.00	-1,355.00	35,250.00
20017	Beeston Business Centre	1,355.00	0.00	-1,355.00	35,250.00
FM107	Hampshire Housing Estates	750.00	2,393.00	1,643.00	2,877.85
FM55	Berkshire Schools FM	2,720.66	151.98	-2,568.68	178.58
FM57	Berkshire Schools FM Co Locations	49,529.68	579.20	-48,950.48	688.95

Printed using COINS V10.23.110502 by Nigel Longley at 12:15:20 on 10/05/12 (NLMDExercise1b)

Page 1

The figures Sales balance figure will be a sum of the SL and CS figures. To prove this, run each data set in the Query Editor to see the values returned.

## 7.4 Exercise 1c – Using Datasets for Exporting Data

This exercise will demonstrate how a dataset can be used to export data in Excel format only without the need to create a detailed report definition.

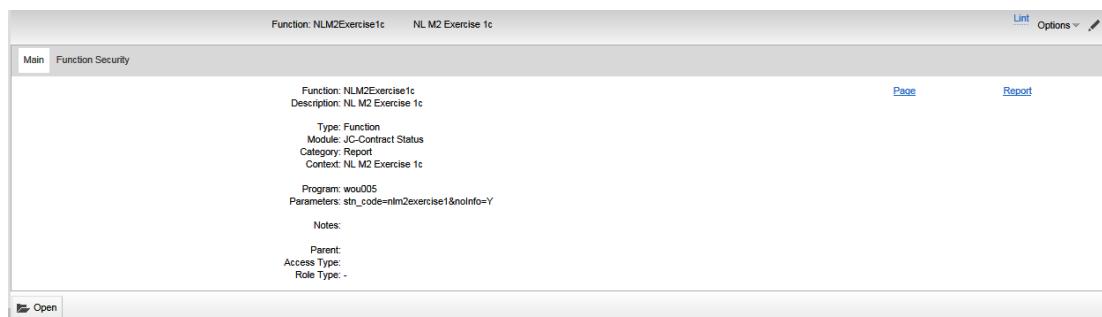
### 7.4.1 Create the Function and Menu

Create the function that will be used to run the report then add this function to an appropriate menu.

Function Code -Initials + M2 + Exercise No.e.g. NLM2Exercise1c  
 Function Name -Description identifier.E.g. NLM2EX1c  
 Function Module - Contract Status  
 Function Category - Report  
 Function Context - Leave as defaulted from the Function Description.  
 Function Program - wou005

We already have a page that we can re-use from the first exercise so the following parameter will allow us to use this:

Function Parameters strn\_code=nlm2exercise1



### 7.4.2 Create the Report Section

For this exercise we only want to create a dump of the data from the dataset straight to Excel so we only need to specify the dataset to be used - we do not need to define any Forms .

Set up the report section:

**Report Section** Give the report section the same name as the function created.

**Header Tables** Leave Blank

**Body Table** Enter the name of the database table being reporting on; in this case our temporary table from the Exercise 1 dataset ttcontract

**Body Query** FOR EACH ttcontract

**DataSet** Specify the name of the Data Set that contains the temporary table. In this case xxM2DS1 (replace xx with your initials).

**Page Layout** Select %A4RLAND-A4 Report Landscape .

**ClassSelect** the font class to use. Since the user listing report doesn't have many fields on, use Arial 8pt (a larger font than Arial 6pt).

Report:NLM2Exercise1c

[Lint](#)

Header Tables:  
Body Table: ttcontract  
Notes:

[Function](#)

Query	Sort	Calculations	Forms	Fields	Used in
<p>Body Query: 1 <code>FOR EACH ttcontract</code></p> <p>Condition: Maximum Rows: 0</p> <p>Body Title: Calculate Condition: 1</p> <p>Generate Program: Data Set Definition: <a href="#">NLM2DS1</a> Data Marts: Page Layout: %AARLAND-A4 Report Landscape Class: %SPT-Standard 8pt</p> <p>Extra Formulas: Productions: Content: Suppress Selection Page: <input type="checkbox"/> Suppress Company Name: <input type="checkbox"/> Suppress Module Name: <input type="checkbox"/> Chart: Chart Formatting:</p>					

**Do NOT define any forms or fields.**

### 7.4.3 Running the report

Run the function xxM2Exercise1c. On the output queue you will not get a pdf file generated only an Excel and XML.

Opening the Excel you should get output similar to:

	A	B	C	D	E	F	G
1	Report:	NLM2Exercise1c-Contract Status - NL M2 Exercise 1c					
2	Date/Time:	25/06/12	14:16				
3	Selection:	&RS_job_num_1=&RS_job_num_2=&RS_job_num_3=*&RS_jgr_group_1=&RS_jgr_group_2=&RS_jgr_group_3=*					
4	Company:	10					
5	User:	niglon					
6							
7	RecordType	Company	Contract	Name	Costs	Revenue TD	Profit TD
8	ttContract	10	10000	Penny Hill Estates	0	155000	155000
9	ttContract	10	100010	Purchase Land	0	0	0
10	ttContract	10	1001	PF Training Project	0	0	0
11	ttContract	10	10010	New Wing	1200	0	-1200
12	ttContract	10	1007	Hamptons Hospital	57500	0	-57500
13	ttContract	10	11000		0	0	0
14	ttContract	10	11001		0	0	0
15	ttContract	10	11002		0	0	0
16	ttContract	10	1111	Milton Keynes Buildir	0	0	0
17	ttContract	10	11111	YZProject	12280	0	-12280
18	ttContract	10	1119	material	0	0	0
19	ttContract	10	12007	Cambridge Fens	0	0	0
20	ttContract	10	12345	Brick buy	0	0	0
21	ttContract	10	12500	buy bricks	1000	0	-1000
22	ttContract	10	20014	Riverside Brassiere	13085	0	-3085
23	ttContract	10	20015	Eastside Locks	1355	0	-1355

## 7.5 Exercise 1d – Dynamically switching between multiple Data Sets

This exercise will demonstrate how you can dynamically select the data set to be used in a report

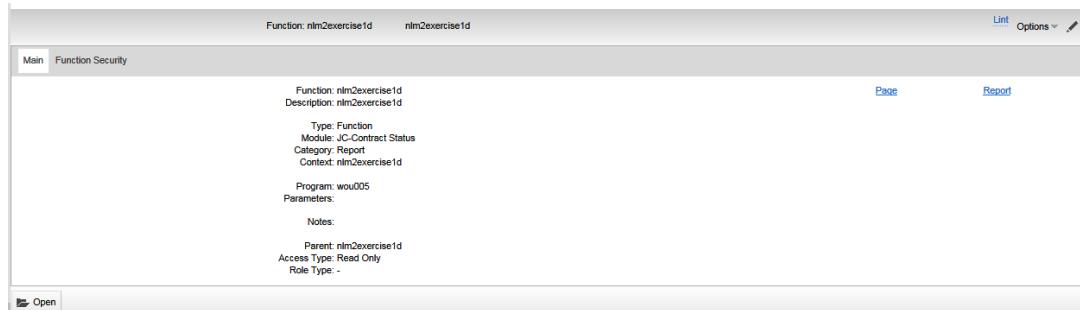
### 7.5.1 Create the Function and Menu

Create the function that will be used to run the report then add this function to an appropriate menu.

Function Code -Initials + M2 + Exercise No.e.g. NLM2Exercise1d  
 Function Name -Description identifier.E.g. NLM2EX1d  
 Function Module - Contract Status  
 Function Category - Report  
 Function Context - Leave as defaulted from the Function Description.  
 Function Program - wou005

We already have a page that we can re-use from the first exercise so the following parameter will allow us to use this:

Function Parameters stn\_code=nlm2exercise1



### 7.5.2 Build the datasets

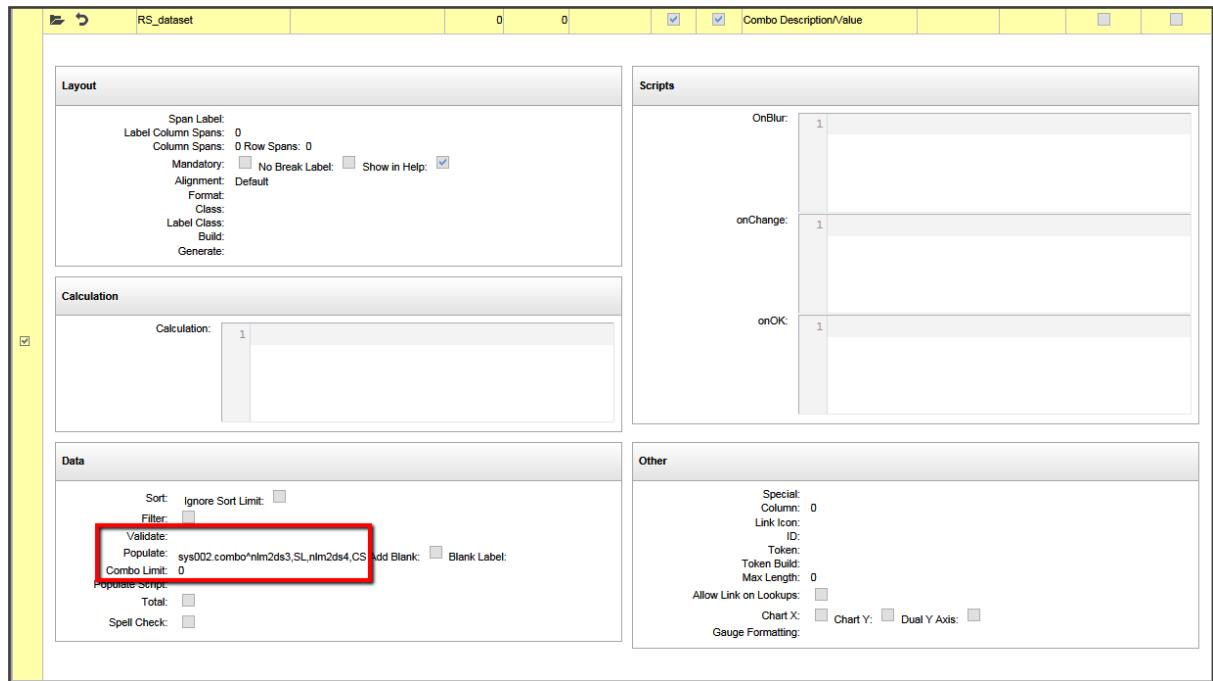
For this example, we will use the datasets nlm2ds3 and nlm2ds4 that we created in the previous exercise so there is no need to build new ones.

### 7.5.3 Create the page

Copy the page nlm2exercise1 and call it nlm2exercise1d

Add a new field to the update form called RS\_Dataset with a view as of Combo Description/Value. The populate field should be as follows:

sys002.combo^nlm2ds3,SL,nlm2ds4,CS



This will create a dropdown selection field that will allow us to select either SL or CS and the corresponding name of the datasets (either nlm2ds3 or nlm2ds4) will be returned.

#### 7.5.4 Create the Report

Copy the report definition for xxM2exercise1b and rename it xxM2exercise1d.

For this new report, the query can be left as it is, since the query tables are correct, however we don't want to specify the dataset names to be used, instead we want the selection page to control this

In the Data Set Definition field, replace the dataset names with dataset selection field from the page i.e. {RS\_Dataset}.

```

Report.NLM2Exercise1d
Lint
Header Tables: co_config
Body Table: jc_job
Notes: DATASET MULTIPLE & MERGE
Function Page
Query Sort Calculations Forms Fields Used in
Body Query:
1 FOR EACH jc_job WHERE kco = (kco){jobSelect},
2 EACH ttSLCSInvoice WHERE ttSLCSInvoice.kco = jc_job.kco
AND ttSLCSInvoice.job_num = jc_job.job_num
Condition: Maximum Rows: 0
Body Title: Calculate Condition: 1
Generate Program:
DataSet Definition: (RS_Dataset)
Data Marts:
Page Layout: %ARLANDA Report Landscape
Color: %PPT-Standard 8pt
Extra Formatting:
Production:
Content:
Suppress Selection Page: 
Suppress Company Name: 
Suppress Module Name: 
Chart:
Chart Formatting:

```

### 7.5.5 Running the report

Run the function xxM2Exercise1d. The page should offer the dropdown field to let you change the datasets

Run the report twice with each a different dataset selection and verify that the figures are different.

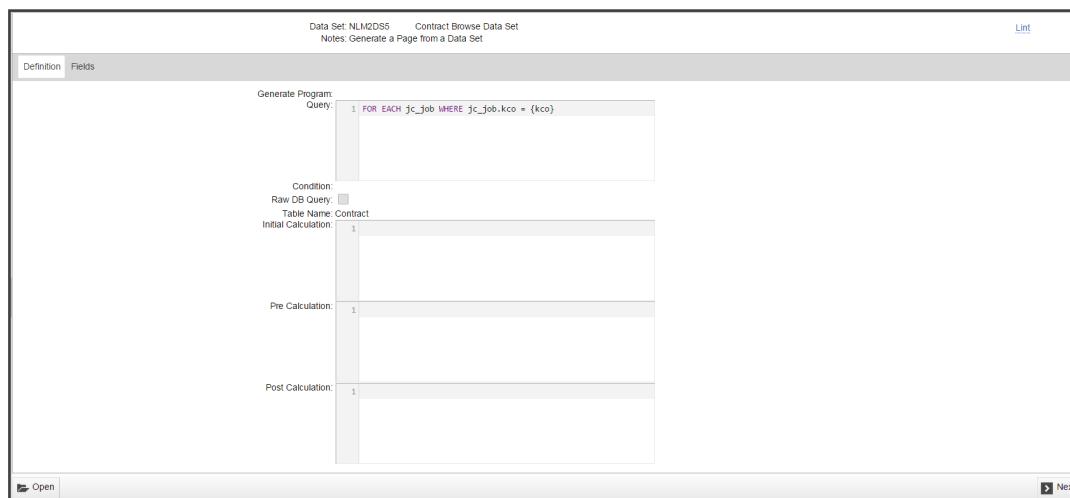
## 7.6 Exercise 2 - Generating Pages from a Data Set

### 7.6.1 Create a Browse Page from a Data Set

Create a New Data Set called xxM2DS5 with simple contract details (replace xx with your initials).

Query FOR EACH jc\_job WHERE jc\_job.kco = {kco}

Table NameContract



Add fields Add fields to the data set and name the field names for the relevant sy\_temp fields, where:

smp\_key1 is the key field of the data set and

smp\_cha[20] is used for character fields,

smp\_dat[20] = Date fields,

smp\_int[20] = Integer fields,

smp\_dec[20] = Decimal fields and

smp\_log[20] = Logical fields.

On the Fields Tab, click Add and the following fields:

Field	Label	Data Type	Format	Source
smp_key1	Contract	Character	x(8)	jc_job.job_num
smp_cha_1	Name	Character	x(8)	jc_job.job_name
smp_cha_2	Location	Character	x(8)	jc_job.jcl_desc
smp_dat_1	Date	Date	99/99/99	jc_job.job_condate
smp_cha_3	Status	Character	x(8)	jc_job.job_active
smp_dec_1	Costs	Decimal	>>>,>>>,>>9.99	jc_job.RO_contractcosts^TD
smp_dec_2	Revenue	Decimal	>>>,>>>,>>9.99	jc_job.RO_ContractRevenue^TD

## 7.6.2 Add New Functions for the Browse Page

## Step 1:

Create a new function and add this to a test menu under the Development Workspace on the OA Reporting and BI Menu.

Function Code -Initials + Exercise No.e.g NLM2Exercise2

Function Name -Description identifier.e.g. NL M2 Exercise 2

## Function Type - Function

## Function Module - Contract Status

## Function Category - Browse

**Function Context** - Leave as defaulted from the Function Description.

Function Program - wou005

## Step 2:

Create a Generate Function for the Page i.e. suffix G with the following parameters:

Program: wougen

Parameters: prog=sygxxx&DataSet=aaM2DS5 (where aa = your initials).

Note: There must be no spaces in a parameter string and for generate pages the program name is sygxxx, the x's are not replacement characters as in previous exercises

	Function ▾	Description ▾	Type	Module ▾	Category
<input checked="" type="checkbox"/>	<a href="#">NL M2 Exercise 2</a>	NL M2 Exercise 2	Function	JC-Contract Status	REP-Report
<input checked="" type="checkbox"/>	<a href="#">NL M2 Exercise 2G</a>	NL M2 Exercise 2 Generate	Function	JC-Contract Status	REP-Report

### 7.6.3 Create the Page

Add a New Page. The Query should refer to table sy\_temp whilst the body table should refer to the body table of the data being returned

## Body Tablejc\_job

Body QueryFOR EACH sy\_temp WHERE syl\_procid = '{syl\_procid}' AND mf\_function = '{mf\_function}' AND smp\_txn = '{smp\_txn}'

Page NLM2Exercise2

[Lint](#)

Header Tables: Body Table: [sy\\_temp](#) ↗ Function  
Description: G-Function  
Notes:

Header Body Script Forms Fields Named Filters Browse Filters Fields Enquiry Used in

Body Program:  
Body Rows: 12

Body Query:

```
1 FOR EACH sy_temp
2 WHERE syl_procid = '{sy1_procid}'
3 AND mf_function = '{mf_function}'
4 AND smp_txn = '{smp_txn}'
```

Query Condition:  Ignore Security:

Sort Limit: 0  
Body Title:  
Body Sum:  
Actions:  
Frozen Columns: 0  
Dialogue Width: 0  
Dialog Update:   
Matrix Columns:  
Fixed Columns:  
Auto Add:  Condition:   
Single Add:   
Sequenced Update:   
Graph Formatting:  
Mandatory Columns:  
Essential Fields:

[Open](#) [Next](#)

## FormAdd a Body form

## FieldsAdd Dataset fields to the Body form

#### 7.6.4 Running the Function.

If you run the main function, the screen will not contain any records.

Use the Regenerate button on the Page to generate data

Contract Status | NL M2 Exercise 2

1 - COINS Construction | NIGLON 

Contract	Name	Location	Contract Date	Costs	Revenue	Status
1000	Oxford Road Office	London 01	01/01/05	2,229,668.89	11,422,833.73	A
1001	Holiday Inn Slough	North East 02	01/02/05	3,607,897.45	7,643,419.54	A
1002	Terminal 5 retail fitout	North West 02	01/02/05	6,836,247.40	11,658,811.01	A
1003	Castle View House Slough	South East 01	01/01/09	2,720.00	0.00	A
1004	Re-surfacing Jnc 1 - Jnc 3	North East 02		50,000.00	0.00	A
1100	A406 Ealing underpass	South East 01	01/02/05	854,830.22	2,540,167.50	A
1101	Windsor Rail bridge replacement	South East 01	01/03/05	302,662.54	870,809.08	A
1102	Eton River Thames bank rebuilding	Scotland 02	18/01/05	555,629.23	930,159.67	A
1103	Heathrow T4 hydrant fuel insta	Scotland 02	15/01/05	387,821.36	622,336.05	A
1104	M34 Northampton - Southampton	Overseas 00	01/01/05	21,942,729.91	22,142,061.92	A
1110	LOR Terminal 5	London 01		0.00	0.00	A
1111	Mxd Development at Precinct 14	London 01	01/01/07	15,000.00	0.00	A
1115	M61 Extension	Overseas 00	01/01/06	3,325.00	46,342.67	A
1200	Bluewater shopping M & E	Overseas 00	01/02/05	961.33	1,289,733.19	A
1401	NTL Cabling installation	North East 02	01/01/05	926.00	67,879.00	A

To avoid this additional step to see the data, instead of calling the main function, instead call the G function, e.g. In this exercise call function xxM2Exercise 2G

## 8 Report Builder

Report Builder is intended for use by end users who are comfortable with using Excel. It is a drag and drop report writing tool which allows the user to lay out banded reports, with groupings, subtotals and totals. It also provides a powerful way to extract data into Excel.

Users can use two types of data source: prebuilt datasets, or custom-built datasets using the semantic layer. Calculated fields and report calculations can do many of the calculations Excel users are familiar with, both on character fields and on values.

A video demonstrating the use of Report Builder is available:

<http://vimeo.com/coinslms/coins-qa-report-builder-1103>

## 8.1 Initial Setup

### 8.1.1 Browser Version

Please note that Report Builder has been tested on IE 11 and Chrome. It is NOT available for use with IE 9.

## 8.1.2 Report Builder System Parameters

There are two new SY parameters

Parameter ID	Description
RBFUNCS	A comma-separated list of the calculation functions that are available in Report Builder.  If this is blank, the default list is: absolute, can-do, date\$, datestring, day, decimal, entry, entry\$, if, if\$, ifexec, ifexec\$, index, inlist, integer, left\$, length, max, min, month, nonzero, numentries, range, right\$, round, string\$, substring\$, sum, today, trim, trimleft\$, trimright\$, weekday, year.  * means all functions are available.
RBPAGES	A comma-separated list of the report page layouts that are available on a report builder report. For example: %A4RLAND,%A4RPORT,%A3RLAND,%A3RPOR T.  Blank will show all page layouts.
RBSTYLES	A comma-separated list of the report styles that are available on a report builder report. For example: %8PT,%6PT,%10PT.  This allows just a few select styles to be shown rather than all that are defined for other reasons in COINS. Blank will show all styles.

These parameters are not essential to the running of report builder and can be left blank.

## 8.1.3 Report Data Sources – Pre-Built Datasets

The function previously called Report Writer Query has been renamed Report Data Sources and now allows suitable data sources to be flagged as enabled for Report Builder.

	Data Source	Description	Tables	Report Builder
□	NLM2DS1	NL Exercise M2	ttContract	<input checked="" type="checkbox"/>
□	NLM2Exercise2	NLM2Exercise2	jc_job.tttslcsinvoice	<input checked="" type="checkbox"/>
□	nitrans	nl trans test	tttrans	<input checked="" type="checkbox"/>
□	PJ_AIN	All PL Invoice File for current company	ap_invoice.ap_vendor	<input type="checkbox"/>
□	PQT_Collins	All pp_orgtime	pp_orgtime.pp_organisation	<input type="checkbox"/>
□	PXGL_GLT	GL Transactions with Account and Source Selection by Date Range	gl_acct.gl_trans	<input type="checkbox"/>
□	r1	robtest1		<input type="checkbox"/>
□	RB_M2DS1	RB_M2DS1	ttContract	<input checked="" type="checkbox"/>
□	RHCVS	All CVR Header for current company	cv_cvrhead.jc_job	<input type="checkbox"/>
□	RIN_JOB	All SL Invoice File for current company with Contract	ar_invoice.jc_job	<input type="checkbox"/>
□	SLEX5	SL Exercise 5	ttExercise5NL	<input type="checkbox"/>
□	Suppliers	Suppliers	ttSuppliers	<input checked="" type="checkbox"/>
□	syudscal	Syuds Exercise1	ttcontract	<input checked="" type="checkbox"/>
□	syudscal2	syudscal2	ttContract.ttCalc	<input checked="" type="checkbox"/>
□	TESTBIGQ	Test Big Query	co_config.jc_job.sc_certificate.sc_mast.sc_certdet	<input type="checkbox"/>
□	TESTBIGQ2	Testing Big query with combo dropdowns	co_config.jc_job.sc_certificate.sc_mast.sc_certdet	<input type="checkbox"/>
□	type 1	rob type 1		<input type="checkbox"/>

Report Builder queries MUST use only datasets to extract data; they cannot perform direct queries on the database (which was allowed with Report Writer queries).



Only those data sources selected as Report Builder enabled will be shown in the data sources combo in Report Builder.

## 8.1.4 Security

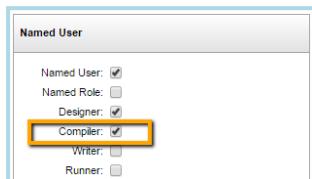
The access that users have in Report Builder to data sources (existing report data sources and collections for custom data sources) relies on function security. This requires new functions to be set up.

- To provide access to a report data source, you must create a function with a function code in the form DSRC\_<data-source-name>. For example, to provide access to data source %JOB, create a function DSRC\_%JOB.
- To provide access to a semantic layer collection, you must create a function with a function code in the form DCOL\_<collection-name>. For example, to provide access to collection %JCT, create a function DCOL\_%JCT.

Then use standard function security to grant access to users.

## 8.1.5 Query Results Licence

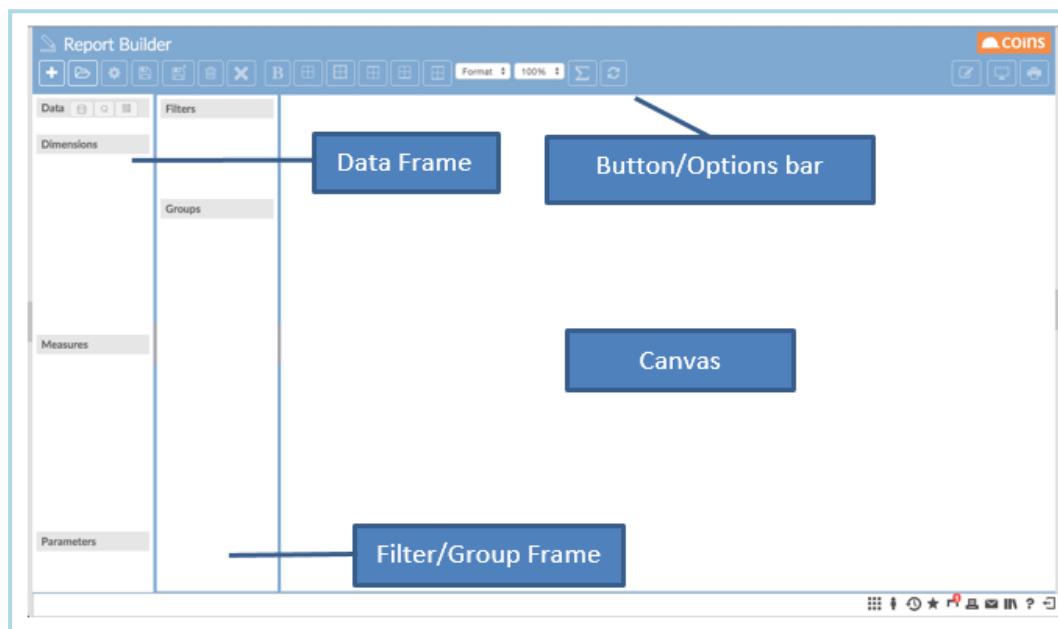
To use custom data sources, you must have a Progress Query Results licence, and each named user needs to have the Complier option on their user record ticked.



## 8.2 Report Builder

When you first run Report Builder (%WSY1150WRTN), it shows an empty report screen, which consists of:

- Canvas
- Button/options bar
- Data Frame
- Filter/Group Frame



The page is disabled apart from the “New Report” and “Open Existing Report” buttons.



## 8.2.1 Canvas

The canvas shows the current report (work in progress - WIP). When you open a report or start a new one it is the WIP report that you are looking at and changing. Each change is saved immediately; you can leave the WIP report, quit the system and log back in and it will still be there. The WIP report can be discarded or overwritten with the report action buttons as described below; it is only when you Save or Save As that the report is saved permanently.

The canvas is disabled if the Report Type is Spreadsheet Only (see Creating a New Report).

## 8.2.2 Button and Options Bar

### 8.2.2.1 Report Actions



These buttons run actions that affect the report as a whole.

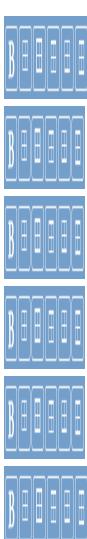


- New Report - discard the current report and create a new one
- Open Existing Report - discard the current report and open an existing one
- Properties of Current Report - change the properties of the current report
- Save Current Report - save the current report with the existing name
- Save Current Report As - save the current report with a new name. The previous save of the current report will still exist
- Delete Current Report - discard the current report and delete the saved version
- Clear Current Report - discard the current report but retain the last saved version

### 8.2.2.2 Field Formatting



These buttons apply formatting to fields selected on the canvas. Select a field by clicking on it, or select multiple fields using CTRL+click (Windows) or ALT+click (Mac). Then use the buttons to format the field or fields.



- Bold
- No border
- Full border
- Top border
- Bottom border
- Top and Bottom border

### 8.2.2.3 Canvas View



The View combo allows you to show the field names, the field labels or the field format on the canvas.

The Size combo allows you to set the scale of the canvas.

### 8.2.2.4 Actions



These buttons apply actions to the report canvas.



The Sum button allows you to automatically build total forms for the fields in the body. You can select which forms are built. The forms selected will be deleted and rebuilt using the configuration of fields in the body. Any changes that you made to the total forms will be lost.



The Refresh button rebuilds the canvas from the saved report data in COINS.

### 8.2.2.5 Run



These buttons relate to running the report.



The Designer button takes you to the report designer page for the current report. This is useful for users who have report designer skills and want to see the underlying data that is being created by the report writer.



The Preview button generates a sample report based on up to 100 records from the data source and immediately shows the result in a PDF in a frame. This preview may not be available until mandatory parameters have been set.

#### TECHNICAL NOTE



Each time a preview is run, the report output log file is written (overwritten) in \$BASE/var/diag/preview.log. This log is shared between all users but may be useful for debugging.

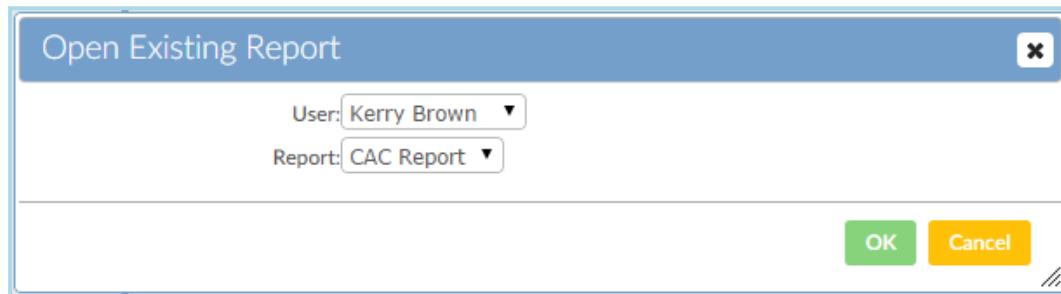


The Report button shows a report selection form (in a dialog) for you to complete and to run the report as any other normal COINS OA report. The resulting PDF and Excel/CSV output will be available on the Report Status Workbench.

### 8.2.3 Opening an Existing Report

You can open an existing report created by you (or by another user if you have the appropriate permissions).

Press the Open Existing Report button  This loads the Open Existing Report dialog.

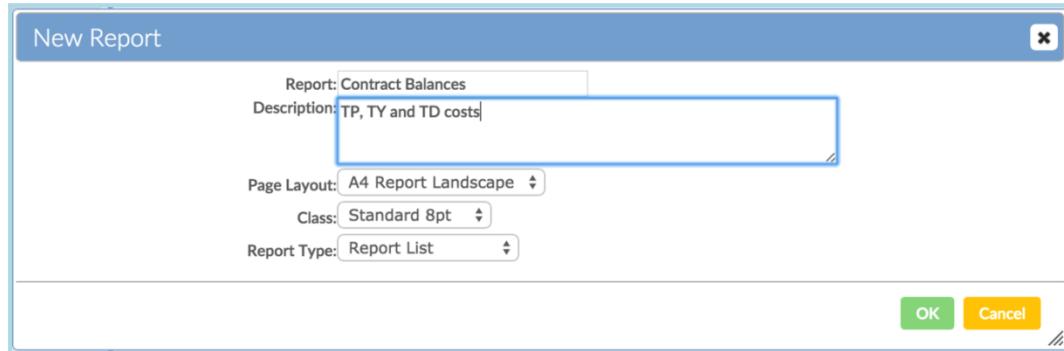


Select the user who created the report and the report you want to open.

To create a new report it may be easiest to open an existing report (which could be another user's report) as above, then use the Save As button  to save it as a new report. This also allows easier report testing - one user can create the report, then another user can copy using this method and test it.

## 8.2.4 Creating a New Report

Press the New Report button  to load the new report dialog



### 8.2.4.1 Fields

Field	Description
Report	The name of the report for you to refer to it. It should be treated like a file name of a document on a disk. Do not use special characters (such as  , ", ~\ or ^) in the name.
Description	A fuller description of what the report does.
Page Layout	The page sizes and layout for the report. The list of possible sizes and layouts is controlled by SY parameter RBPAGES.
Class	The font class to use. The list controlled by SY parameter RBSTYLES.
Report Type	The type of report. This can be one of: <ul style="list-style-type: none"><li>● Report List (normal column listing report)</li><li>● Report Form (the body of the report is a form rather than a column layout)</li><li>● Spreadsheet Only (if only a spreadsheet file is required and no PDF is produced). The canvas is disabled with this option; all fields, including calculated fields, will be exported.</li></ul>

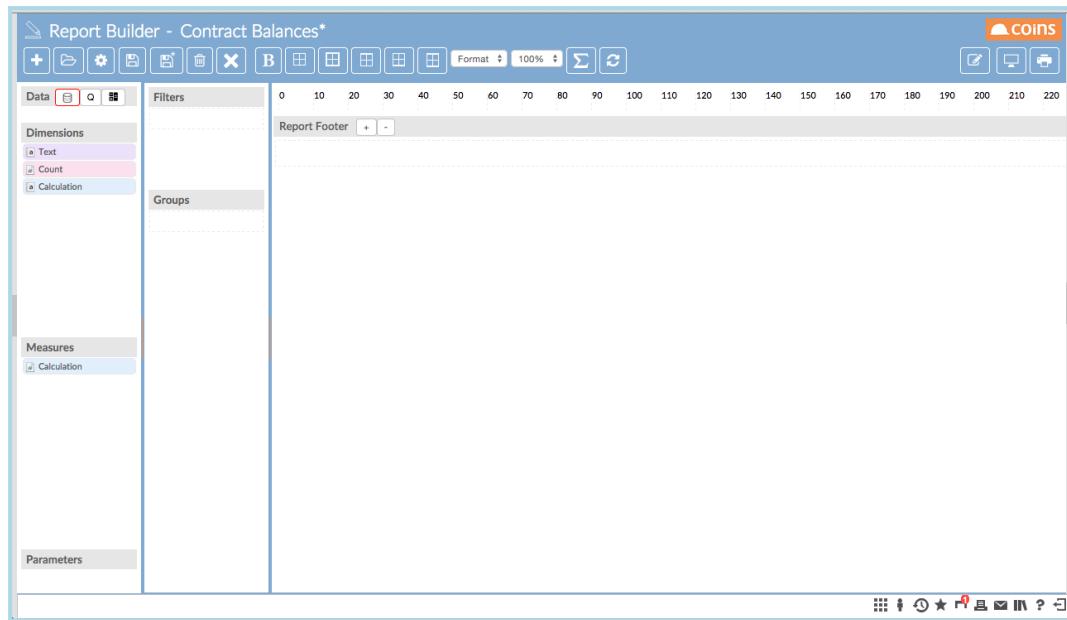
Clicking Cancel would discard the new report definition. Any current report on the canvas would be retained.

When you press the OK button, the busy icon will be shown in the top left hand corner. This is used whenever anything is being sent to the server.



If necessary, you can change the properties of the report using the  button.

After creating the new report, a blank canvas will be shown. The buttons are now enabled.

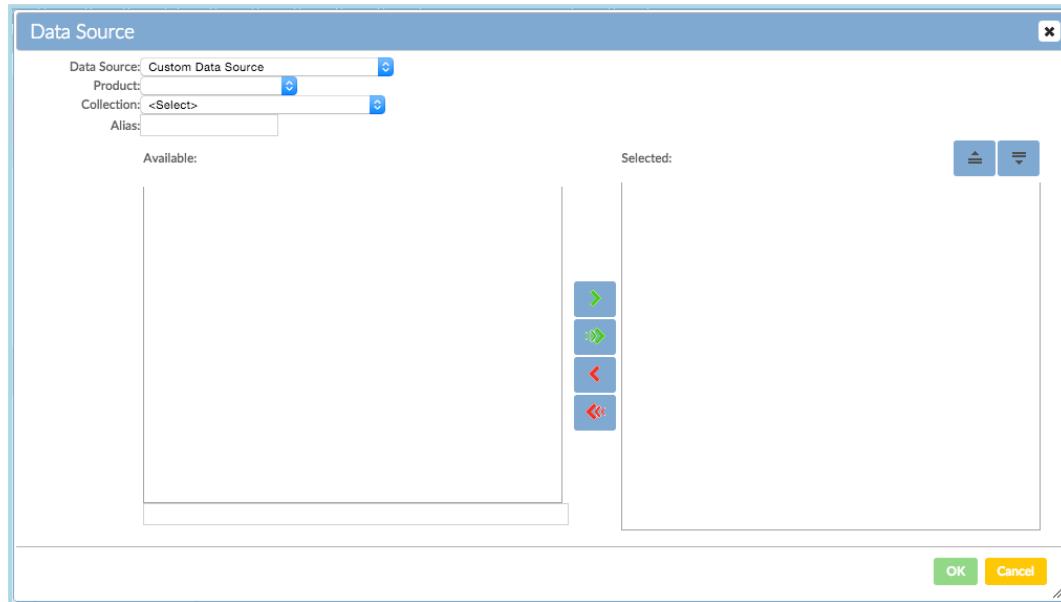


The data source button is highlighted because you have yet to define a data source.



### 8.2.4.2 Data Source

Press the data source button to display the Data Source dialog.

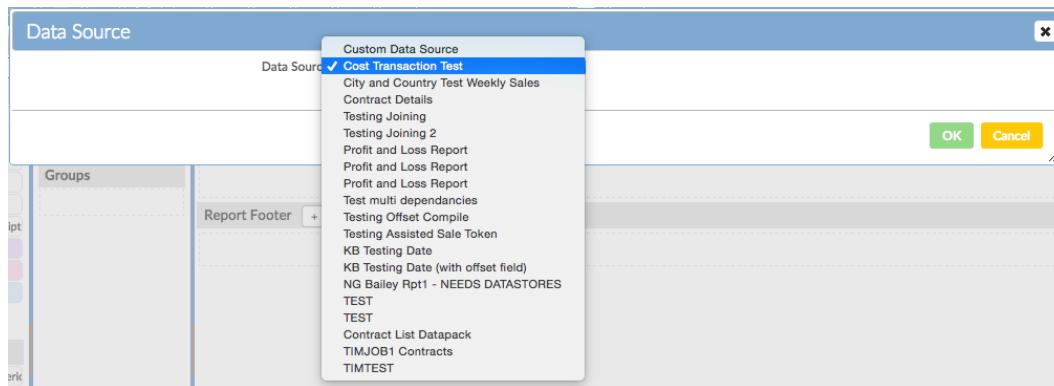


You can select an existing data source or (if you have permission) create a custom data source.

#### 8.2.5.3 Existing Data Source

An existing data source is a data source that has been set up to be available in Report Builder (see 2.3, Report Data Sources).

If you select a pre-defined data source then you simply select from the list of available report data sources.

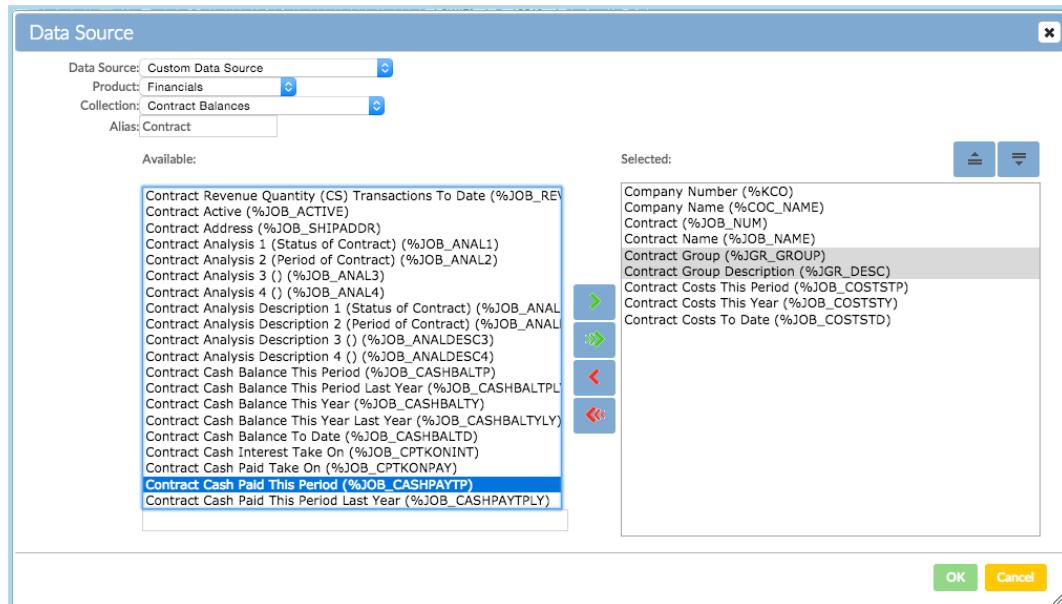


#### 8.2.6.4 Custom Data Source

A custom data source is a data set you create 'on the fly' from the semantic layer. You need a compiler licence which allows you to compile data sets from the semantic layer (see 8.1.5 , Query Results Licence).

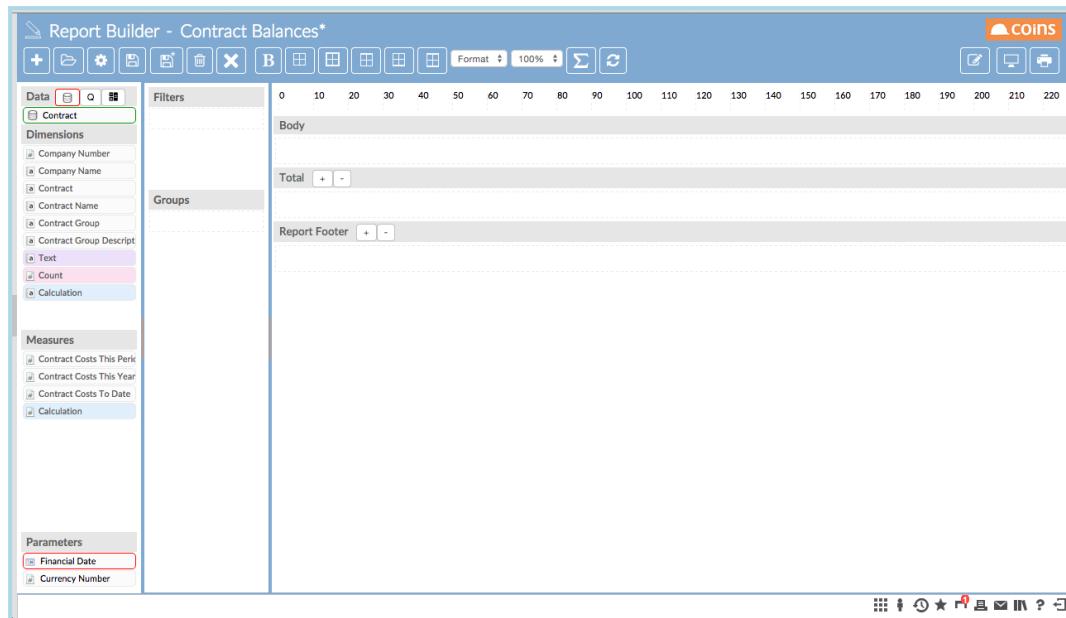
Select a product to filter the list of collections. When you select a collection, the default alias for the resulting table will be copied from the semantic layer and the available fields will be shown.

Select the fields that are required for the report.



Press OK to compile the dataset. You can then define the output.

#### 8.2.7.5 Data Frame



The data frame shows:

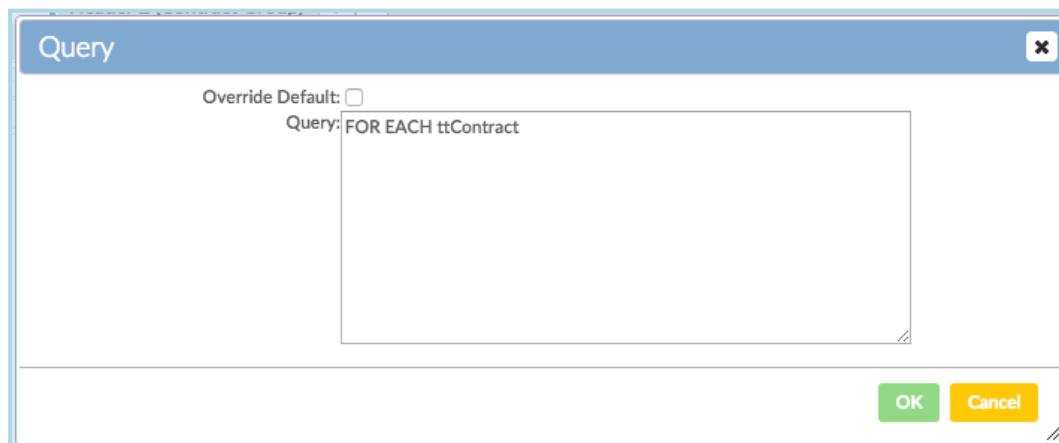
- The tables of the data source (Contract in the example above); there may be multiple tables when using pre-defined data sources. The tables used in the query are shown highlighted in green.
- The fields of the data source, split into dimensions (values - strings, dates, integers) and measures (decimals - things that can be aggregated).

Dimensions also includes text, count and calculations (for character strings) which you can define when you drag them onto the report canvas (see 3.1.5.3, Text, Count, Calculations). There is a calculation for decimal values in Measures.

- If the data source requires parameters then they are shown at the bottom. Highlighted parameters (Financial Period in the example) are mandatory and yet to be completed.

The canvas has empty body and total forms added to it.

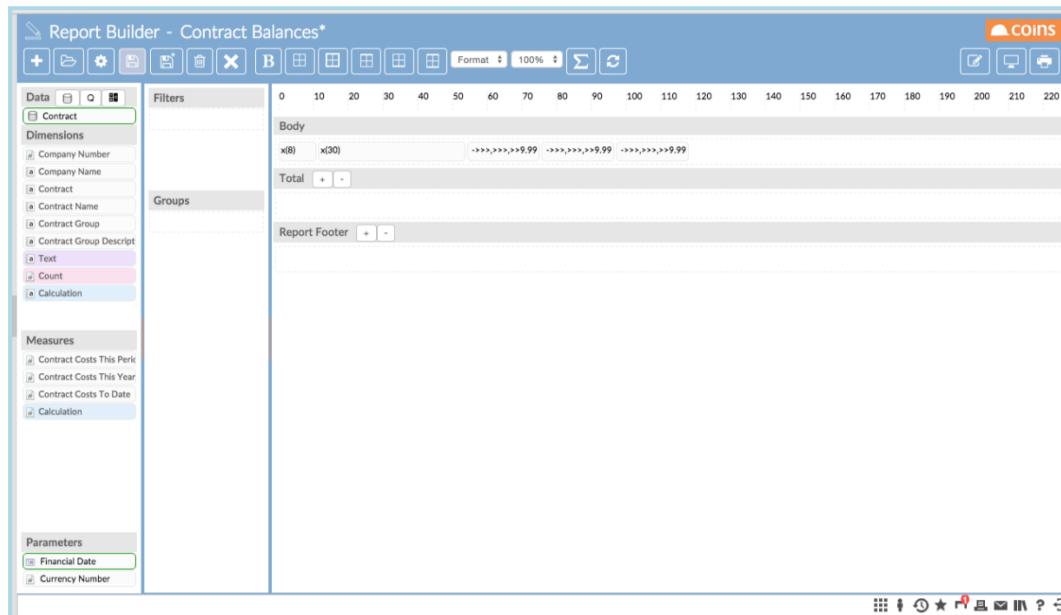
The query associated with the data source is automatically applied. You can view it (and change it if required) by pressing the Query  button.



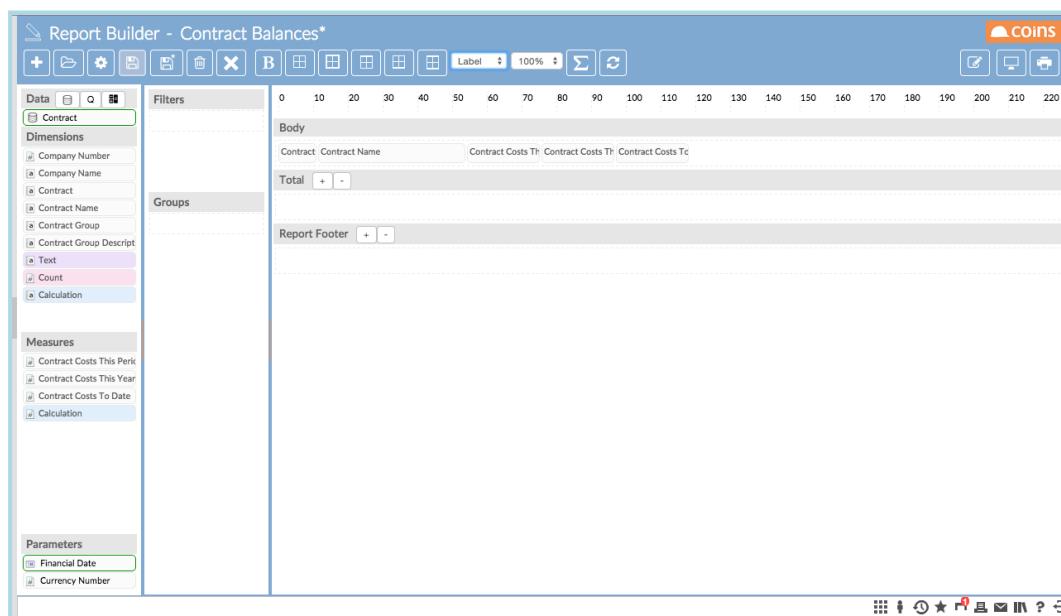
## 8.2.7.6 Adding Fields

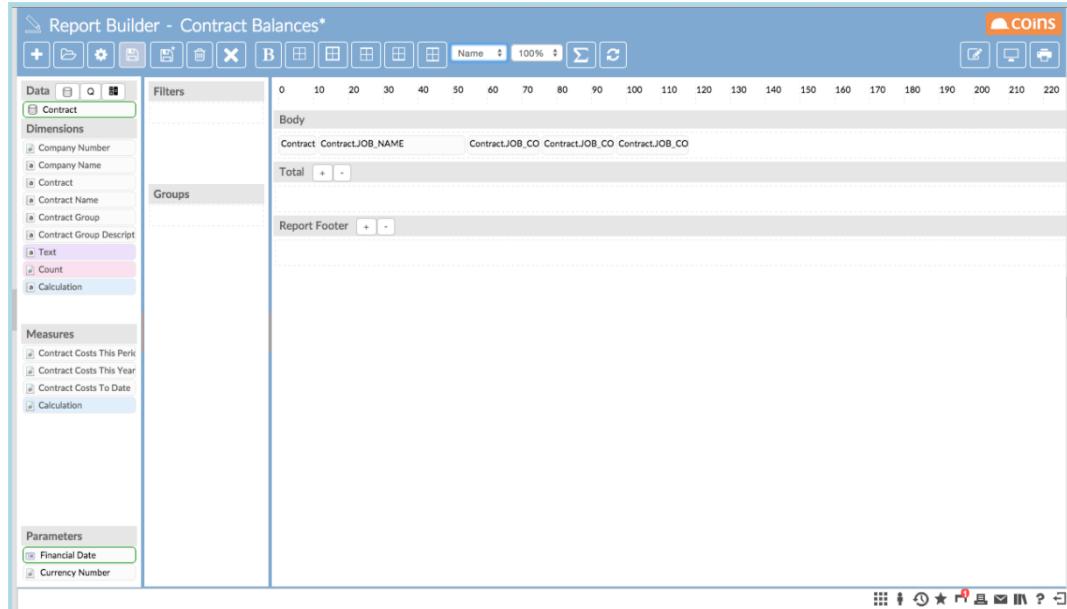
To add fields to the report, either:

- Drag a field from the dimensions or measures frame to the canvas and drop it. This allows you to position the fields in the order you want.
- Double-click a field; it will be added to the end of the body form.

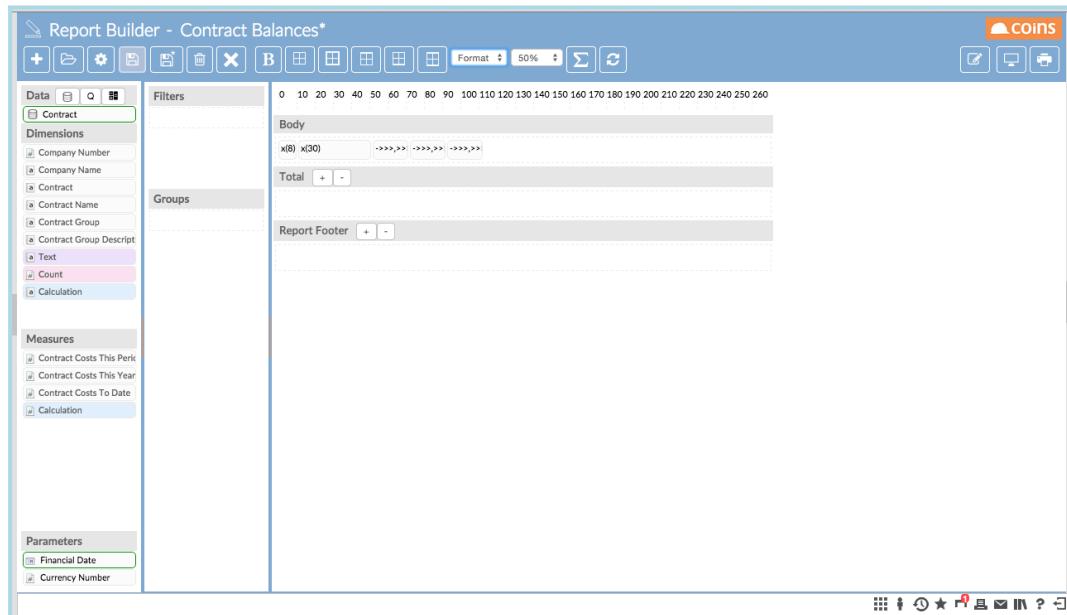


You can change the show combo in the header shows labels or names.





You can change the scale of the canvas using the Scale selector  :



You may notice the Save button pulsing. This indicates that you have made changes to the current WIP report but they have not yet been saved. If you press the Save (or Save As) button, the report will be saved and the pulsing will stop.

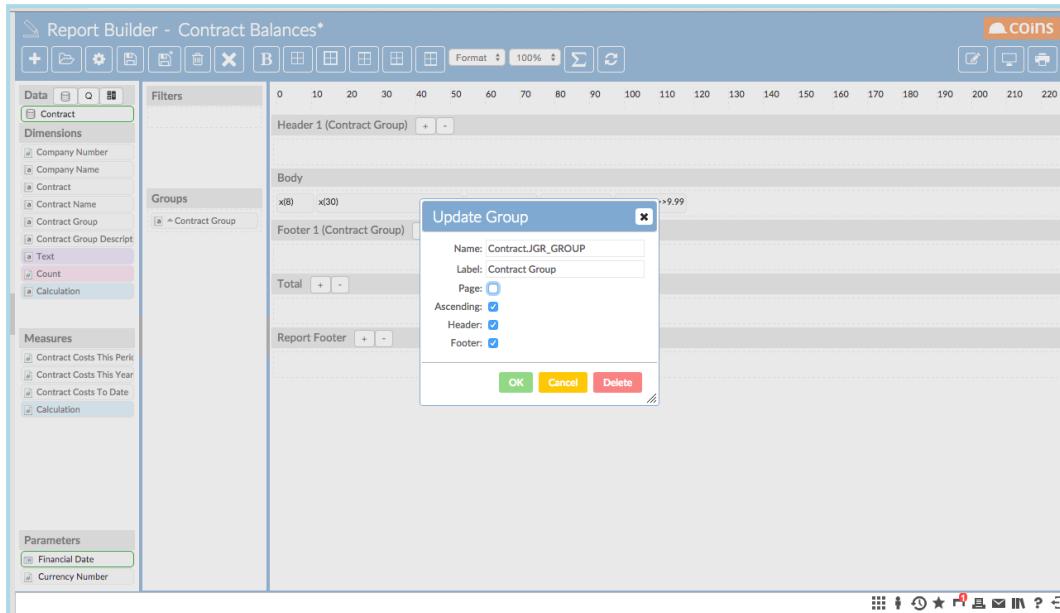


Pressing the Preview button  shows the report so far.



## 8.2.7.7 Groups

To sort data, and (optionally) create header and footer forms, drag fields from the data frame to the groups frame. You can drag the group fields within the groups frame to change the sort/group order. Double-click a group to show the group dialog.



### 8.2.8.8 Fields

#### Page

If you tick this, the report will have a new page for each different value of this field.

#### Ascending

If you tick this, the sort order for this field will be ascending. Otherwise it will be descending.

#### Header/Footer

Whether a header and/or footer form for this field should be shown on the canvas. If you are only using the group to control sorting, you may prefer not to show the forms. If a header or footer form is shown on the canvas but is empty then it will not appear on the report.

You can now drag fields to the new forms.

Report Builder - Contract Balances\*

Format 100%  $\Sigma$  

0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200 210 220

**Data**                        

**Filters**

**Dimensions**

Company Number  
 Company Name  
 Contract  
 Contract Name  
 Contract Group  
 Contract Group Description  
 Text  
 Count  
 Calculation

**Groups**

Company Number  
 Contract Group

**Measures**

Contract Costs This Period  
 Contract Costs This Year  
 Contract Costs To Date  
 Calculation

**Parameters**

Financial Date  
 Currency Number

Header 1 (Company Number)    
x40

Header 2 (Contract Group)    
x8 x30

Body

x8 x30 >>>,>>,>>9.99 >>>,>>,>>9.99 >>>,>>,>>9.99

Footer 2 (Contract Group)  

Footer 1 (Company Number)  

Total  

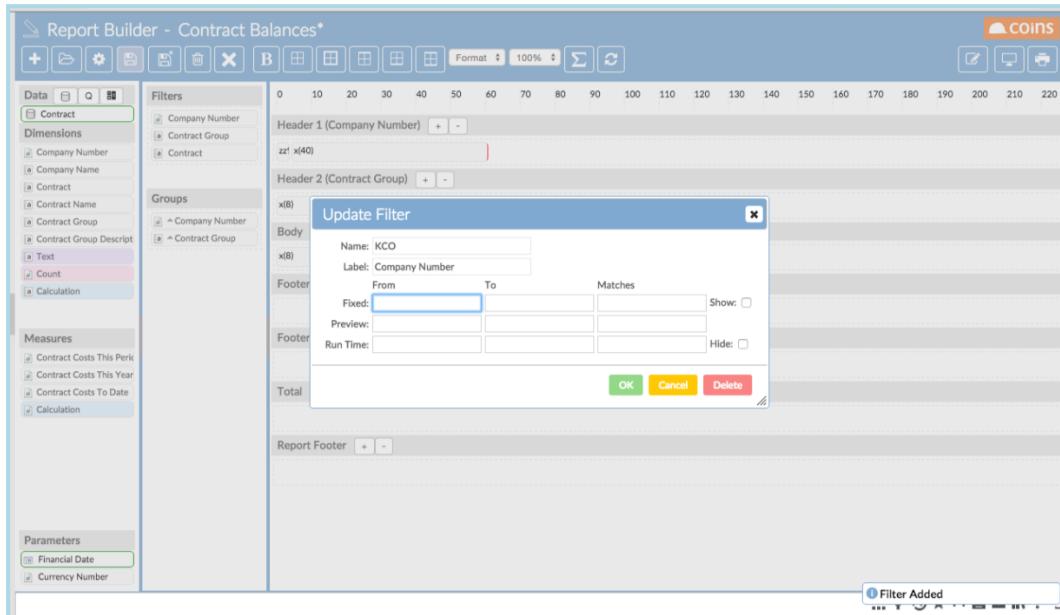
Report Footer  

### 8.2.8.9 Filters

To create a filter, drag fields to the filters frame. The order shown is the order they will appear on the input form for the user when running the report. You can re-order the filters by dragging them within the filters frame.

Double-click a filter to show the filter dialog.



The From, To and Matches columns are for the filters and act the same as normal from, to and matches filters in COINS.

#### Fixed

If you tick this, the filter will be fixed (users cannot override it at run time).

For example, you might use this to omit a specific joint venture company by putting !9,\* in the matches column. This would mean not company 9, but all other companies. The user then might still override which of the other companies they want to see at run time but they would never see company 9.

The Show option allows this fixed filter to be shown on the report selection screen (if desired).

#### Preview

The values that will be used at preview time only. They are not used at run time. This allows you to pick some useful data to be shown on the preview report.

#### Run Time

Default values for the report selection screen.

If you do not want to allow the user to change the filter, tick the Hide option (in this case you will need to set a fixed filter).

### Update Filter

Name:	<input type="text" value="KCO"/>	
Label:	<input type="text" value="Company Number"/>	
From	To	Matches
Fixed:		<input type="text" value="!9,*"/> Show: <input type="checkbox"/>
Preview:		<input type="text" value="10,20"/> Hide: <input type="checkbox"/>
Run Time:		

OK Cancel Delete

Pressing Preview again will show the report so far.

Report Builder - Contract Balances\*

Preview

**Contract Balances**  
**CONTRACTOR MASTER**

Contract Name	Contract Costs This Period	Contract Costs This Year	Contract Costs To Date
10 Contractors OA	0.00	0.00	32,732.72
00 Sector 00XXXXyy1234567	0.00	0.00	0.00
A1005 Northgate High Rail Station	0.00	0.00	800.00
A1004 Optiville Town Hall	0.00	0.00	200.00
000020 Aevem Contract Status	0.00	0.00	0.00
000022 Ihor Stenyk Test	0.00	0.00	0.00
A1001 Brockway School	0.00	0.00	13,702.00
A1000 Chelmsford Hospital	0.00	0.00	32,732.72
9001 Irish Contract (no wbs)	0.00	0.00	0.00
8001 The Edge	0.00	0.00	150.00
8000 The Promenade	0.00	0.00	0.00
886 Project 1 or 1	0.00	0.00	0.00
1002 Forseco Estates	0.00	0.00	100.00
1004 Little Aston Park	0.00	0.00	13,419.00
5000 ASDA South Critical Refurb	-685.00	815.00	0.00
1007 New Cross Hospital Construction	0.00	0.00	92,413.92
314159 BABYLON TOWER	0.00	0.00	0.00
1009 Test WBS Default v2	0.00	0.00	20.00
106753 Test WBS	0.00	0.00	0.00
18 1225 Major Contracts	0.00	0.00	350.00
21122 Winsor House Derby	0.00	0.00	890.00
2012 Jinmao Tower	0.00	0.00	3,630.00
2013 Jinmao Tower	0.00	0.00	0.00
52463 Cverres Creek Library	0.00	0.00	4,830.00
A1003			

Financial Date      Currency Number

The sort/group and (preview) filtering have been applied.

### 8.2.8.1 Run Report

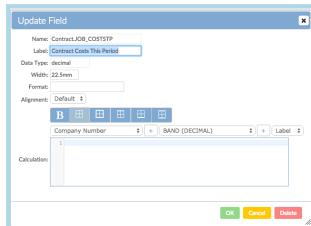
Press the Run Report button to run the report.

The report selection form is shown in a frame like any other COINS report.

Fixed parameters or filters are shown (as defined) and you can change the run time selection fields. Press the Next button to submit the report to the background queue to be generated. The result will appear on the Report Status Workbench.

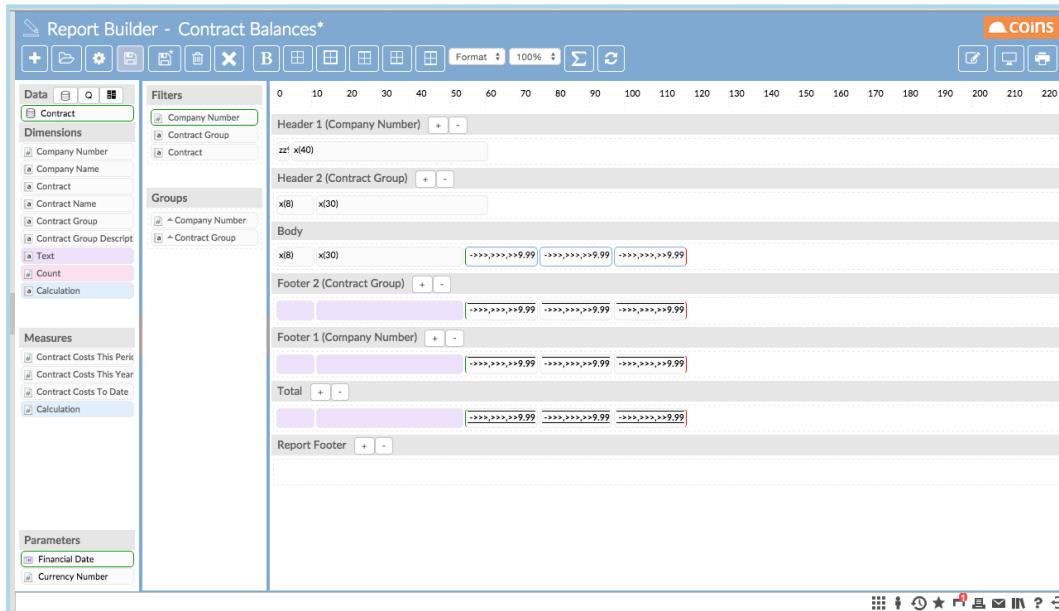
### 8.2.8.1 Refining the Report - Fields

Double-click a field to launch the field dialog.



This allows you to set the label, width, format, alignment, bold and border and specify a calculation. (The bold and border options are also available by selecting one or more fields and pressing the corresponding buttons in the button bar.)

After changing the values in the field dialog, press the OK button to have them reflected on the canvas. You can also press the Delete button to delete the field.



To change more than one field, select one field by clicking on it then use CTRL (or ALT for Mac) to select other fields. They are highlighted with a blue outline. Continue to hold CTRL (or ALT) and double-click one of the fields to bring up the multi field update dialog.



This allows you to set the label, width, format and alignment for all the fields you have selected (similar to multi-update in COINS OA). Similarly the Delete button will delete all the selected fields.

You can also select one or more fields on the canvas and press the Delete key to delete fields. You will be asked to confirm the fields should be deleted.



### 8.2.8.1 Field Positioning

You can drag fields on the canvas to change the order in a form, and you can resize them by dragging the right hand end of the field.

When you select a field (or fields), all fields that match the left side of the selected field (or fields) will be shown with a green left border and all fields that match the right side will be shown with a red right border. This allows you to see when fields are aligned.



When resizing a field, matching fields will show amber right border when close and red when an exact match.

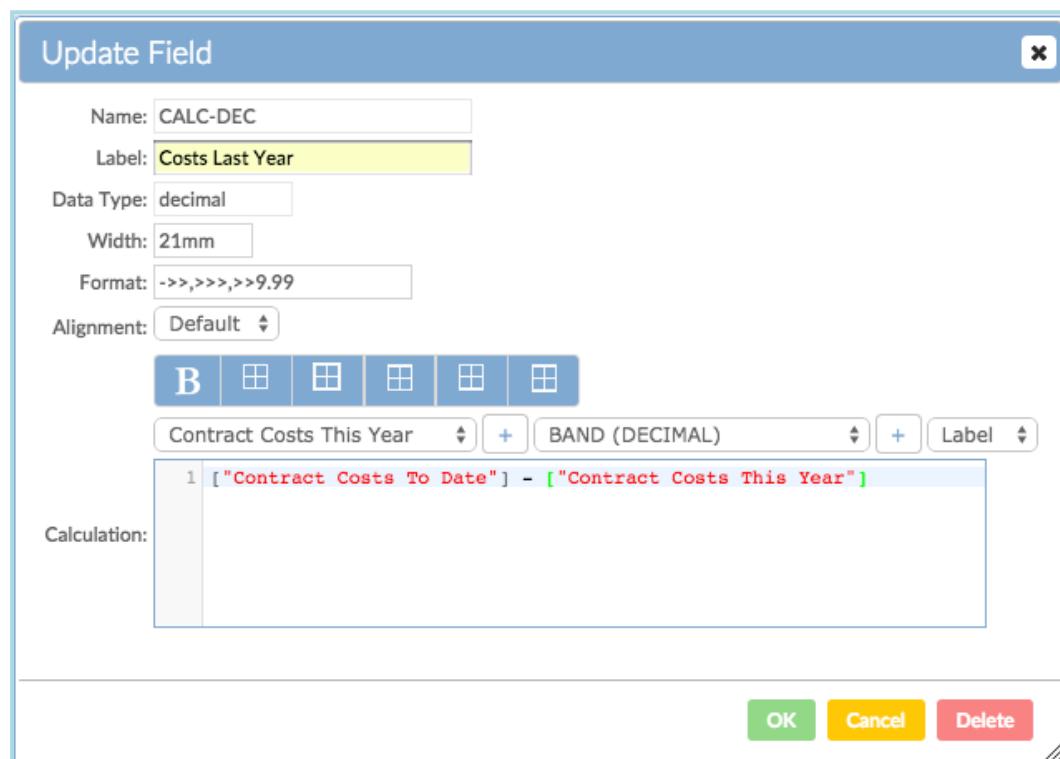
### 8.2.8.1 Text, Count, Calculations

If you drag a text, count or calculation field on to the canvas then you will be able to set its properties by double-clicking like any other field.

For Text fields, the Label is the text that is shown on the report. You can use text fields with no label as padding to position other fields.

Counts can be added to header and footer forms and are evaluated automatically.

Calculations you can specify the calculation to be performed.



The selectors above the calculation area show the fields and functions that are available; you can add these to the calculation by pressing the + button.

The calculation helper allows you to show labels or field names (the image above shows labels).

Calculations on a body form use the values from the data set. Calculations defined on a header or footer use the automatically-aggregated values for the group. If you add a calculation to the body form then generate automatic totals for the report, the calculation is adjusted for the group forms. For example, if you add a calculation like body.profit / body.costs to the body line, and automatically add it to the group form

 using the  button, the calculation on the group form will be group.total-profit / group.total-costs.

For more information about calculations, see [Calculations in the BI Toolset](#).

**Report Builder - Contract Balances\***

Filters: Company Number, Contract Group, Contract

Groups: Company Number, Contract Group

Measures: Contract Costs This Period, Contract Costs This Year, Contract Costs To Date, Calculation

Parameters: Financial Date, Currency Number

Header 1 (Company Number): x40

Header 2 (Contract Group): x8 x30

Body: x8 x30

Footer 2 (Contract Group): x8 x30

Footer 1 (Company Number): Contract Costs This Year

Total:

Report Footer:

**Report Builder - Contract Balances\***

Preview: 220

Contract Balances  
CONTRACTOR MASTER

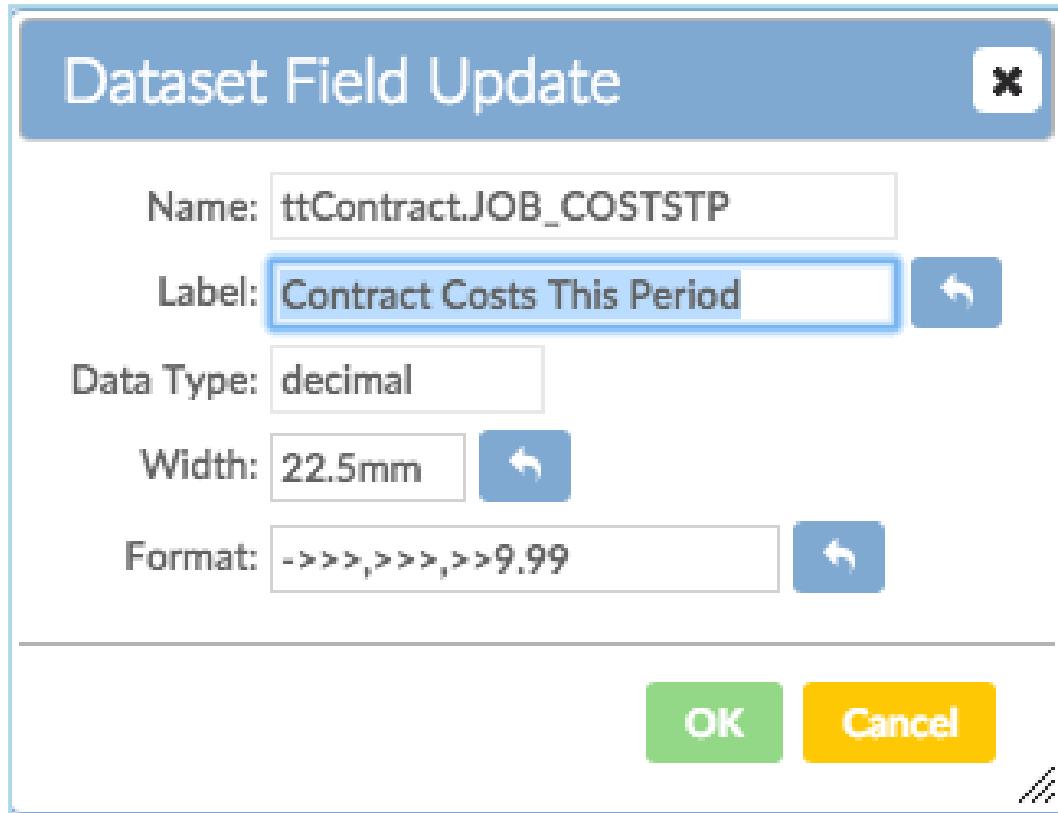
COINS logo

Contract	Contract Name	Contract Costs This Period	Contract Costs This Year	Contract Costs To Date	Costs Last Year
000010	650 Apartments/Office/Leisure	9,616.75	12,152.20	12,421.17	288.97
5588	Apex Offices	0.00	0.00	0.00	0.00
1001	Apex Offices	0.00	60,196.99	91,177.39	31,199.99
10000	Penny Hill Estate	0.00	413.79	58,811.32	58,197.53
000023	Steve Cund Contract 000023	0.00	69,212.22	69,212.22	0.00
21009	Hamptons House	0.00	69,212.22	0.00	0.00
		<b>9,616.75</b>	<b>142,066.94</b>	<b>233,084.69</b>	<b>91,017.66</b>
BET	Material Betterment	0.00	0.00	311.00	311.00
000014	Aleem	0.00	0.00	311.00	311.00
EL	Sector EL	0.00	0.00	0.00	0.00
1008	New Test	0.00	0.00	0.00	0.00
WEST	West contracts	0.00	0.00	8,166.67	8,166.67
22610	Site Store Contract	0.00	6,050.00	110,738.80	104,888.80
000622	Jewel of The Creek	0.00	4,000.00	4,093.04	93.04
1994	BabyonTower2	0.00	10,050.00	122,998.51	112,948.51
ZZ	Sector ZZ	94,147.99	466,016.37	988,297.22	522,280.85
000150	Ohta Center	94,147.99	466,016.37	988,297.22	522,280.85
		<b>103,764.74</b>	<b>617,248.31</b>	<b>6,564,663.35</b>	<b>5,947,415.04</b>
		<b>103,764.74</b>	<b>617,248.31</b>	<b>6,564,663.35</b>	<b>5,947,415.04</b>

Parameters: Financial Date, Currency Number

### 8.2.8.1 Dataset Field Update

You can change the default settings on a data source field (for this report) by pressing CTRL (Alt for Mac) and double-clicking the field in the data frame. This brings up the Dataset Field Update dialog.

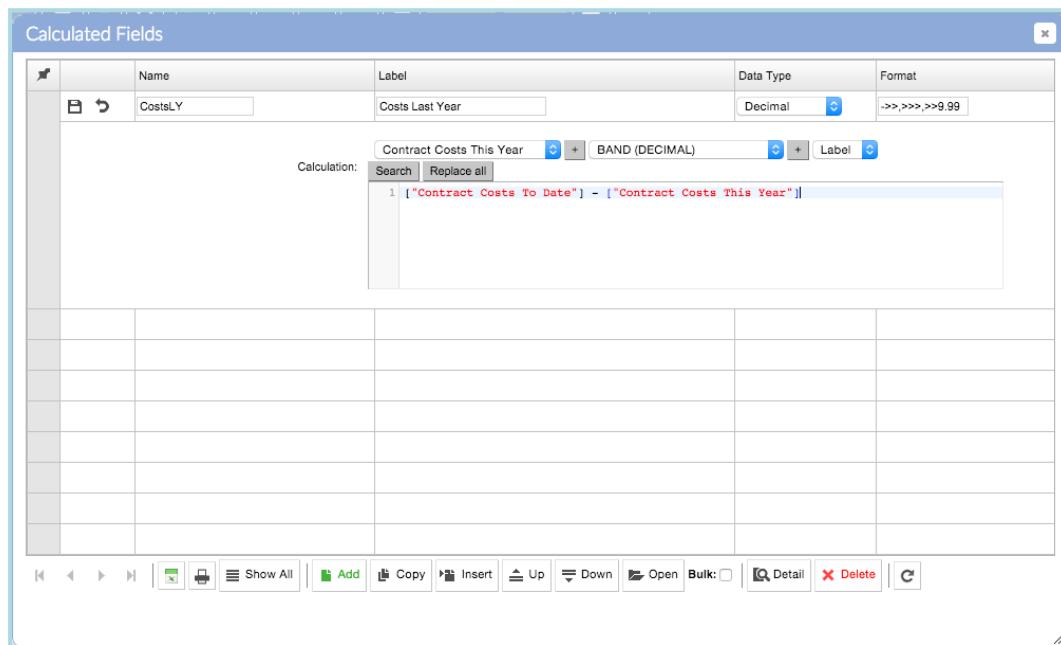


You can override the label, width and format. These will then be used as defaults if you drag the field on to the canvas. The original data source is not affected by these amendments but if the report is copied then they will be retained.

### 8.2.8.1 Adding Fields to Datasets

You can add fields to the dataset as part of the report.

Pressing the Calculated Field button  allows you to add calculated fields to the dataset.



Enter a field name (this must be unique within the dataset), a label, data type and format and the calculation to evaluate to populate the field with data.

When you close the calculated fields dialog, the dimensions and measures will be updated with any calculated fields shown in yellow.

The screenshot shows the COINS Report Builder interface for 'Contract Balances'. The left sidebar lists dimensions (Company Number, Contract Group, Contract), groups (Company Number, Contract Group), measures (Costs Last Year, Contract Costs This Perk, Contract Costs This Year, Contract Costs To Date, Calculation), and parameters (Financial Date, Currency Number). The main area displays a report structure with sections: Header 1 (Company Number), Body (repeating row with calculation), Footer 2 (Contract Group), Footer 1 (Company Number), Total, and Report Footer. The calculation field in the Body section is highlighted.

They can now be added to the report like any other field (the example above duplicates the calculation shown in 3.1.5.3).

This screenshot shows the same Report Builder interface as the previous one, but with a different calculation result. In the Body section, the calculation field now displays a value of 9.99 instead of the previous 9.999. The rest of the report structure remains identical to the first screenshot.

**Report Builder - Contract Balances\***

Preview

**Contract Balances**  
**CONTRACTOR MASTER**

Contract Name	Contract Costs This Period	Contract Costs This Year	Contract Costs To Date	Costs Last Year	Costs Last Year
000010 850 Apartments/Office/Leisure	9,317.75	12,340.20	12,421.17	289.97	289.97
5588 Apex Offices	0.00	0.00	0.00	0.00	0.00
1001 Bovis Test	0.00	60,030.39	91,135.35	31,104.96	31,104.96
10000 Penny Hill Estate	0.00	413.79	58,811.32	58,197.53	58,197.53
000023 Steve Curd Contract 000023	0.00	69,212.22	69,212.22	0.00	0.00
21009 Hamptons House	0.00	0.00	0.00	0.00	0.00
	<b>9,616.75</b>	<b>142,066.94</b>	<b>233,084.60</b>	<b>91,017.66</b>	<b>91,017.66</b>
BET Material Betterment	0.00	0.00	311.00	311.00	311.00
000014 Alern	0.00	0.00	311.00	311.00	311.00
EL Sector EL	0.00	0.00	0.00	0.00	0.00
1008 New Test	0.00	0.00	0.00	0.00	0.00
WEST West contracts	0.00	0.00	8,166.67	8,166.67	8,166.67
22610 Site Stock Contract	0.00	6,050.00	110,738.80	104,688.80	104,688.80
000622 Jewel of The Creek	0.00	4,000.00	4,093.04	93.04	93.04
1984 BabylonTower2	0.00	10,050.00	122,998.51	112,948.51	112,948.51
ZZ Sector ZZ	94,147.99	466,016.37	988,297.22	522,280.85	522,280.85
000150 Otsa Center	<b>94,147.99</b>	<b>466,016.37</b>	<b>988,297.22</b>	<b>522,280.85</b>	<b>522,280.85</b>
	<b>103,764.74</b>	<b>617,246.31</b>	<b>6,564,663.35</b>	<b>5,947,415.04</b>	<b>5,947,415.04</b>
	<b>103,764.74</b>	<b>617,246.31</b>	<b>6,564,663.35</b>	<b>5,947,415.04</b>	<b>5,947,415.04</b>

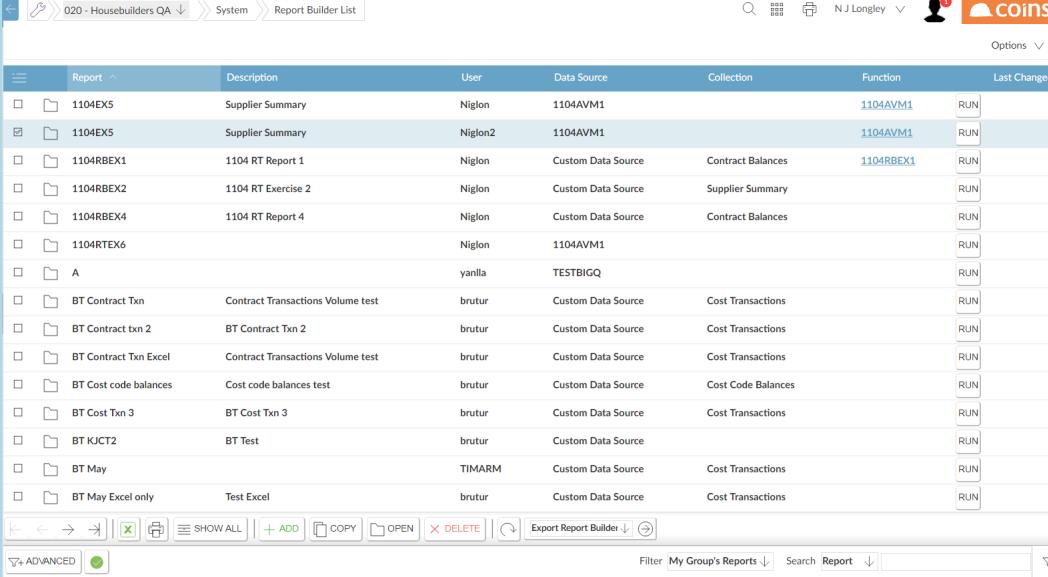
Parameters

Financial Date  
Currency Number

Print, Refresh, Save, Help, Exit

## 8.2.9 Report Builder List

Report Builder List is a new function that allows you to look at all your reports (and if you have the appropriate group permissions, those of other users in your group).



Report	Description	User	Data Source	Collection	Function	Last Changed
1104EX5	Supplier Summary	Niglon	1104AVM1		1104AVM1	RUN
1104EX5	Supplier Summary	Niglon2	1104AVM1		1104AVM1	RUN
1104RBEX1	1104 RT Report 1	Niglon	Custom Data Source	Contract Balances	1104RBEX1	RUN
1104RBEX2	1104 RT Exercise 2	Niglon	Custom Data Source	Supplier Summary		RUN
1104RBEX4	1104 RT Report 4	Niglon	Custom Data Source	Contract Balances		RUN
1104RTEx6		Niglon	1104AVM1			RUN
A		vanilla	TESTBIGQ			RUN
BT Contract Txn	Contract Transactions Volume test	brutur	Custom Data Source	Cost Transactions		RUN
BT Contract txn 2	BT Contract Txn 2	brutur	Custom Data Source	Cost Transactions		RUN
BT Contract Txn Excel	Contract Transactions Volume test	brutur	Custom Data Source	Cost Transactions		RUN
BT Cost code balances	Cost code balances test	brutur	Custom Data Source	Cost Code Balances		RUN
BT Cost Txn 3	BT Cost Txn 3	brutur	Custom Data Source	Cost Transactions		RUN
BT KJCT2	BT Test	brutur	Custom Data Source			RUN
BT May		TIMARM	Custom Data Source	Cost Transactions		RUN
BT May Excel only	Test Excel	brutur	Custom Data Source	Cost Transactions		RUN

You can run reports directly from this workbench or delete reports that are no longer required. (Deleting is possible in the Report Builder but only one at a time.) Here multiple reports can be deleted at the same time.

You can also update the report record and give the report a function code. This will create a normal COINS function (you will need to give it a suitable function code that is unique) that can be put on a menu for other users to run the report. Once you have given it a function code you will need to grant access permissions to the function and add it to a suitable menu for users to access.

## 9 Page Design Exercises

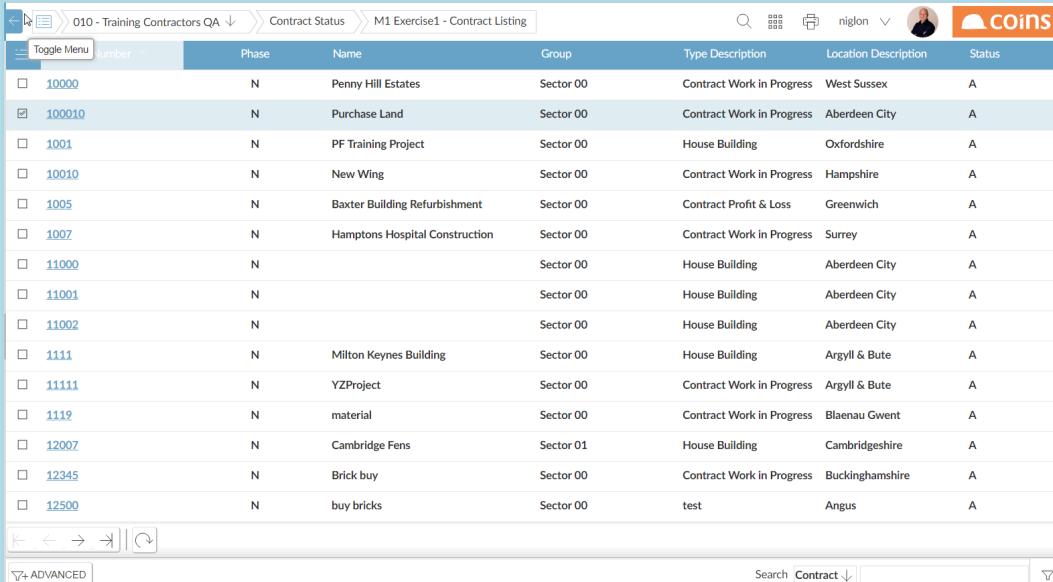
In the following exercises we will work through the creation of a Contract Browse.

The exercises have been designed to introduce all the main components of designing a browse page. In practice, not all the features covered would be needed on every page you design, but including them in the exercises may help you to understand how standard pages in COINS work.

# Discovery BI0002

## 9.1 Creating a Contract Browse Screen

This exercise is to create a simple contract enquiry screen. The screen will select contracts and return basic information from the jc\_job record.

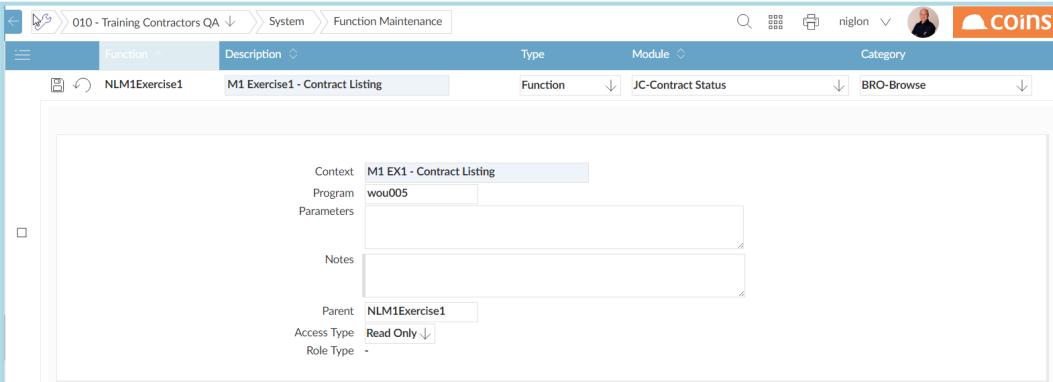


Number	Phase	Name	Group	Type Description	Location Description	Status
10000	N	Penny Hill Estates	Sector 00	Contract Work in Progress	West Sussex	A
100010	N	Purchase Land	Sector 00	Contract Work in Progress	Aberdeen City	A
1001	N	PF Training Project	Sector 00	House Building	Oxfordshire	A
10010	N	New Wing	Sector 00	Contract Work in Progress	Hampshire	A
1005	N	Baxter Building Refurbishment	Sector 00	Contract Profit & Loss	Greenwich	A
1007	N	Hamptons Hospital Construction	Sector 00	Contract Work in Progress	Surrey	A
11000	N		Sector 00	House Building	Aberdeen City	A
11001	N		Sector 00	House Building	Aberdeen City	A
11002	N		Sector 00	House Building	Aberdeen City	A
1111	N	Milton Keynes Building	Sector 00	House Building	Argyll & Bute	A
11111	N	YZProject	Sector 00	Contract Work in Progress	Argyll & Bute	A
1119	N	material	Sector 00	Contract Work in Progress	Blaenau Gwent	A
12007	N	Cambridge Fens	Sector 01	House Building	Cambridgeshire	A
12345	N	Brick buy	Sector 00	Contract Work in Progress	Buckinghamshire	A
12500	N	buy bricks	Sector 00	test	Angus	A

### 9.1.1 Create the Function

Create a new function

Field	Value
Function Code	Initials + M1 + Exercise No.e.g. NLM1Exercise1
Function Name	Description identifier.e.g. NLM1Exercise 1
Function Type	Function
Function Module	Contract Status
Function Category	Browse
Function Context	Leave as defaulted from the Function Description.
Function Program	wou005



Function: M1 Exercise1 - Contract Listing

Context: M1 EX1 - Contract Listing  
Program: wou005  
Parameters:  
Notes:  
Parent: NLM1Exercise1  
Access Type: Read Only  
Role Type:

## 9.1.2 Create the Page

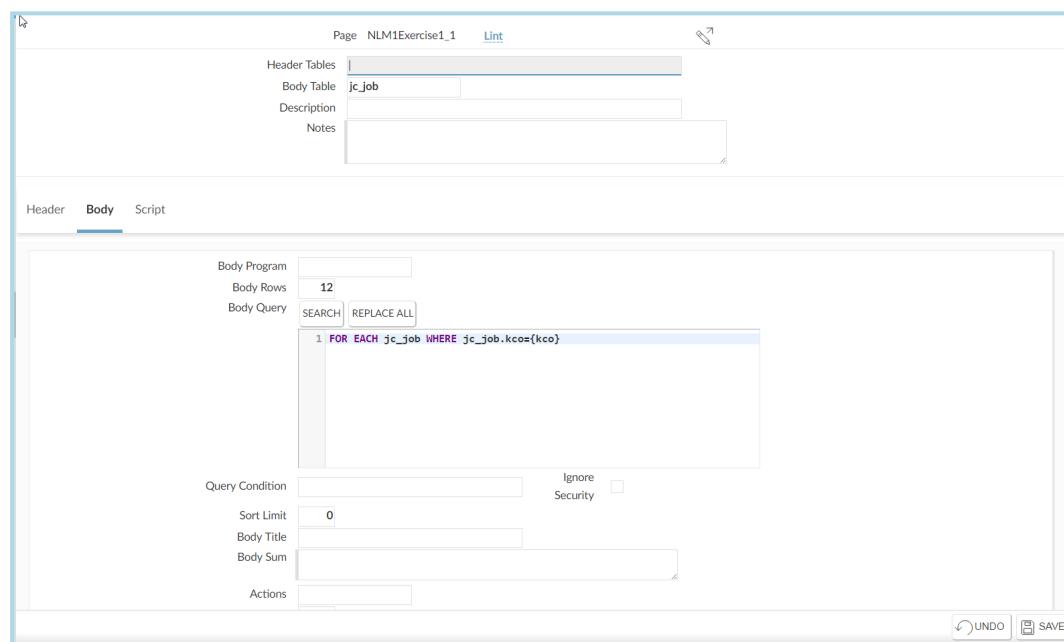
Create the page section (with the same name as the function).

Enter the information into the Page Header as follows:

Field	Value
Header Table	
Body Table	jc_job

On the Body Tab enter the following:

Field	Value
Body Rows	12
Body Query	FOR EACH jc_job WHERE jc_job.kco = {kco}



Click **Save** **SAVE** to complete the Page Summary.

## 9.1.3 Create a Body Page Form

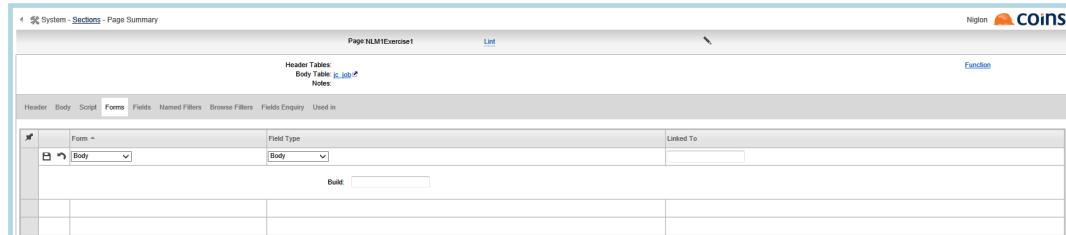
Select the Form tab and create a Body form to use Body Fields.

Click the **Add** **+ ADD** and enter the following information to create the Form :

Field	Value
FormSelect	Body

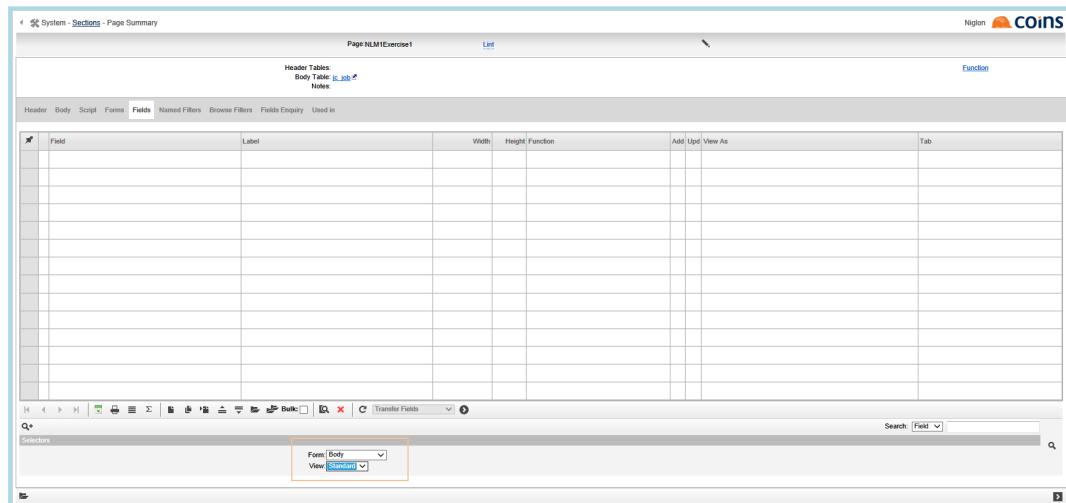
Field	Value
Field Type	Body
Linked To	Leave blank

Click the Save Button to create the form.



### 9.1.4 Add the Fields

In the Fields Tab go to the FORM selector at the bottom of the page:



Select Body and click Apply Filter. This will determine which form you will work with.

The option below for Standard or Grouped will change the way the field details are displayed on the form. At this stage it does not matter whether you set Standard or Grouped as long as you have set the correct form.



You should always do this to ensure you are adding fields to the correct form - adding fields to an incorrect form will mean that they do not display.

Add the following Fields :

Field	Label	Width
job_num	Contract Number	10
job_phase	Phase	5
job_name	Name	20
jgr_desc	Group	20
jty_desc	Type	20

Field	Label	Width
jcl_desc	Location	20
job_active	Status	5

You should check that your field widths add up to 100. The screen will still work if they do not, but the results concerning column widths may not be as expected.

Field	Label	Width	Height	Function	Add	Upd	View As	Tab
job_num	Contract Number	10	0					
job_phase	Phase	5	0					
job_name	Name	20	0					
grp_desc	Group	20	0					
jt_desc	Type	20	0					
jct_desc	Location	20	0					
job_active	Status	5	0					

Click Save after entering each field.

### 9.1.5 Adding a Sort to a Column

A column header can be used to sort the data.

To add a sort enter the field name followed by either a + for ascending or - for descending.

Edit the job\_num field by clicking on the Open button and in the Sort field add:  
**kco+,job\_num+**

This will also provide a simple search filter to the Enquiry screen.

### 9.1.6 Linking a Field to another Enquiry

A field on an enquiry can be linked to another enquiry - for example to dig deeper into further information. This can be an existing standard enquiry or another user defined enquiry. Use the Function option in Field Maintenance to do this.

To link the job\_num field to the contract record:

On the Function field enter **%WJC5014SJ0B** (this is a standard coins enquiry).

Save the field.

### 9.1.7 Running The Function.

Run the Function by adding &MainArea=yourfunctionname to the URL

Contract Number	Phase	Name	Group	Type	Location	Status
0001	N	Kingsley Meadows	Head Office	House Building	Oxfordshire	A
0002	N	The Building	Head Office	House Building	Vale of Glamorgan	A
0003	N	Ullswater Golf Course	Head Office	House Building	Staffordshire	A
0004	N	Hilton Hotel	Head Office	House Building	Lancashire	A
0005	N	Highview Road	Head Office	House Building	Tyne & Wear	A
0006	N	Millfield	Head Office	House Building	Warwickshire	A
1001	N	Contractors Contract	Head Office	Commercial	Bristol	A
1018	N	Bovis Test	Head Office	Commercial	Aberdeen City	A
1011	N	New Contract	Head Office	Commercial	Aberdeen City	A
1012	N	Draft Cott Test	Head Office	House Building	Buckinghamshire	A
1111	N	QA CVR Testing	Head Office	Other	Aberdeen City	A
1112	N	QA PROJECT	Head Office	House Building	Aberdeen City	A
2000	N	Evergreen	Head Office	House Building	Warwickshire	A
2001	N	The Swallows	Head Office	House Building	Cambridgeshire	A
22286	N	Test PDR	Head Office	Commercial	Aberdeen City	A
22286A	N	Test PDR	Head Office	Commercial	Aberdeen City	A

Contract Status - Contract Status	
Contract: 0001	Kingsley Meadows
Contract Currency: GBP	Current Financial Period: 26/02/13
<input checked="" type="radio"/> Main <input type="radio"/> Purchase Orders <input type="radio"/> Subcontract <input type="radio"/> Material <input type="radio"/> Plant	
Contract Number: 0001 Name: Kingsley Meadows Customer: EB8091 Group: 00 Type: HSE Location: OXON Manager: CM Site Address: 34 Green Park Stowe on the Wold Oxfordshire OX106BN Telephone: 01345 866444 Fax: 01345 866445 Status: Active Complete: 0 Original Contact Period: 0 weeks Extension of Time: 0 weeks	

## Discovery BI0002.2

### 9.2 Enhancing the Contract Browse Screen

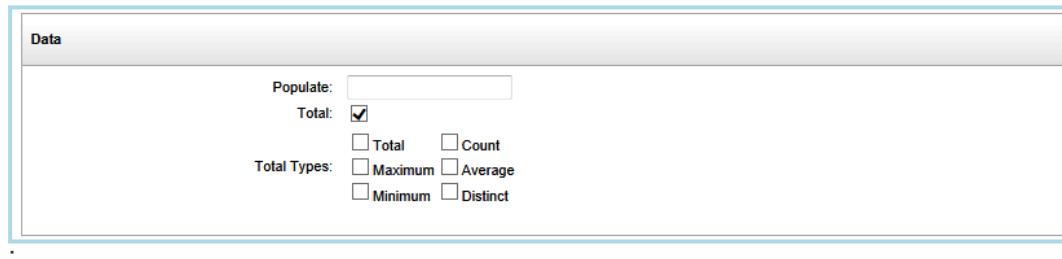
This exercise is to modify the contract browse screen created in Discovery BI0002 to include financial information.

#### 9.2.1 Modifying the Page Section

Edit the page section by clicking on the Open button .

Select Field Tab, In the FORM selector at the bottom of the page, select Body and click Apply Filter.

Click Add to add some more fields to the Enquiry screen. Add the following fields - for each of these also tick the Total box



The screenshot shows the 'Data' tab of a page section configuration. It includes fields for 'Populate' (a dropdown menu), 'Total' (checkbox checked), and 'Total Types' (checkboxes for Total, Count, Maximum, Average, Minimum, and Distinct). Below this is a table listing three fields added to the page:

Field	Label	Width
RO_job_sln_debtors	Extended SL Net debtors	10
RO_job_pln_register	Extended PL Net Registered Creditors	10
RO_job_pln_costed	Extended PL Net Costed Creditors	10

:

Field	Label	Width
RO_job_sln_debtors	Extended SL Net debtors	10
RO_job_pln_register	Extended PL Net Registered Creditors	10
RO_job_pln_costed	Extended PL Net Costed Creditors	10

Click Save after adding each field.

Now adjust the field widths of the other fields to get back to 100 as the total width of all fields.

#### 9.2.2 Adding Totals to the Page

To add totals to a page, a Total Form is required. Create a new form on the Page by

clicking  on the Forms Tab and enter the following details :

Field	Type
Form	Select Totals.
Field Type	Select Totals.
Linked To	Leave blank

Form	Field Type	Linked To
Body	Body	
Totals	Totals	

Click Save to save the new form.

The next stage is to add Fields to the Total Form; Select the Fields Tab from the drop-down menu.

In the Fields selector at the bottom of the page, select Body and click Apply Filter.

**NOTE :** You should always do this to ensure you are adding fields to the correct form - adding fields to an incorrect form will mean that they do not display.

Select each of the fields listed below as a group

Fields
RO_job_sln_debtors
RO_job_pln_register
RO_job_pln_costed

With the three fields highlighted select to action Transfer Fields and select the icon to apply the action.

Field	Label	Width	Height	Function	Add Upd	View As	Tab
job_num	Contract Number	10	0	0%WIC50145JOB			
job_phase	Phase	5	0				
job_name	Name	20	0				
jr_desc	Group	10	0				
py_desc	Type	10	0				
jr_desc	Location	10	0				
job_active	Status	5	0				
RO_job_sln_debtors	SL Net Debtors	10	0				
RO_job_pln_register	PL Net Reg. Creditors	10	0				
RO_job_pln_costed	PL Net Costed Creditors	10	0				

Select Copy, and specify the Totals Form as the destination

Action:  Copy    Move

Other Section:

Form: TOTAL-Totals

Do Not Transfer If Already Exists:

Run the function again.

Contract Number	Phase	Name	Group	Type	Location	Status	SL Net Debtors	PL Net Reg. Creditors	PL Net Costed Creditors
0001	N	Kingsley Meadows	Head Office	House Building	Oxfordshire	A	1,002.38	6,700.00	3,652.25
0002	N	The Bullring	Head Office	House Building	Vale of Glamorgan	A	690.00	15,960.00	17,548.00
0003	N	Ulster Golf Course	Head Office	House Building	Staffordshire	A	0.00	0.00	1,092.50
0004	N	Hilton Hotel	Head Office	House Building	Lancashire	A	0.00	0.00	11,150.00
0005	N	Highview Road	Head Office	House Building	Tyne & Wear	A	0.00	750.00	3,700.00
0006	N	Millets	Head Office	House Building	Warwickshire	A	0.00	0.00	3,512.00
1001	N	Contractors Contract	Head Office	Commercial	Bristol	A	543.62	100.00	956.30
1010	N	Bovis Test	Head Office	Commercial	Aberdeen City	A	0.00	0.00	406.90
1011	N	New Contract	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00
1012	N	Draft Cost Test	Head Office	House Building	Buckinghamshire	A	0.00	0.00	0.00
1111	N	QA CVR Testing	Head Office	Other	Aberdeen City	A	0.00	0.00	0.00
1112	N	QA PROJECT	Head Office	House Building	Aberdeen City	A	0.00	0.00	0.00
2000	N	Evergreen	Head Office	House Building	Warwickshire	A	0.00	0.00	0.00
2001	N	The Swallows	Head Office	House Building	Cambridgeshire	A	0.00	0.00	0.00
22786	N	Test PDR	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00
22786A	N	Test PDR	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00

A sum  $\Sigma$  icon will be displayed on the footer of the browse, click on it to get totals for the fields entered on the whole record set on the browse (or the record set within the current Filter).

Contract Number	Phase	Name	Group	Type	Location	Status	SL Net Debtors	PL Net Reg. Creditors	PL Net Costed Creditors
0001	N	Kingsley Meadows	Head Office	House Building	Oxfordshire	A	1,002.38	6,700.00	3,652.25
0002	N	The Bullring	Head Office	House Building	Vale of Glamorgan	A	690.00	15,960.00	17,548.00
0003	N	Ulster Golf Course	Head Office	House Building	Staffordshire	A	0.00	0.00	1,092.50
0004	N	Hilton Hotel	Head Office	House Building	Lancashire	A	0.00	0.00	11,150.00
0005	N	Highview Road	Head Office	House Building	Tyne & Wear	A	0.00	750.00	3,700.00
0006	N	Millets	Head Office	House Building	Warwickshire	A	0.00	0.00	3,512.00
1001	N	Contractors Contract	Head Office	Commercial	Bristol	A	543.62	100.00	956.30
1010	N	Bovis Test	Head Office	Commercial	Aberdeen City	A	0.00	0.00	406.90
1011	N	New Contract	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00
1012	N	Draft Cost Test	Head Office	House Building	Buckinghamshire	A	0.00	0.00	0.00
1111	N	QA CVR Testing	Head Office	Other	Aberdeen City	A	0.00	0.00	0.00
1112	N	QA PROJECT	Head Office	House Building	Aberdeen City	A	0.00	0.00	0.00
2000	N	Evergreen	Head Office	House Building	Warwickshire	A	0.00	0.00	0.00
2001	N	The Swallows	Head Office	House Building	Cambridgeshire	A	0.00	0.00	0.00
22786	N	Test PDR	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00
22786A	N	Test PDR	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00

### 9.2.3 Adding an Excel Output Function to the Page

It is possible to add additional functionality on a page. To do this create or copy the Function and apply the same name as the original function; however add an "X" to the end of the function name and change the Program from wou005 to X (this is required for any clients using system Parameter MENUSEC, but good practice for all clients.)

Example

FunctionName	xxM1Exercise1
Excel Export Function	xxM1Exercise1X

Function	Description	Type	Module	Category
NLM1Exercise1	M1 Exercise1 - Contract Listing	Function	JC-Contract Status	BRO-Browse
	Contact: M1 Exercise1 - Contract Listing Program: wou005 Parameters: Notes: Parent: NLM1Exercise1 Access Type: Read Only Role Type: -			
NLM1Exercise1X	M1 Exercise1 - Contract Listing	Function	JC-Contract Status	BRO-Browse
	Contact: M1 Exercise1 - Contract Listing Program: X Parameters: Notes: Parent: NLM1Exercise1 Access Type: Read Only Role Type: -			

COINS will automatically add an Excel export button, the PDF Report Button and a Show All icon to the footer of the browse screen.

Contract Number	Phase	Name	Group	Type	Location	Status	SL Net Debtors	PL Net Reg. Creditors	PL Net Costed Creditors
0001	N	Kingsley Meadows	Head Office	House Building	Oxfordshire	A	1,002.36	6,700.00	3,652.25
0002	N	The Bullring	Head Office	House Building	Vale of Glamorgan	A	690.00	15,900.00	17,546.00
0003	N	Ullswater Golf Course	Head Office	House Building	Staffordshire	A	0.00	0.00	1,002.50
0004	N	Hilton Hotel	Head Office	House Building	Lancashire	A	0.00	0.00	11,150.00
0005	N	Highview Road	Head Office	House Building	Tyne & Wear	A	0.00	750.00	3,700.00
1000	N	Millets	Head Office	House Building	Warwickshire	A	0.00	0.00	3,512.00
1001	N	Contractors Contract	Head Office	Commercial	Bristol	A	543.62	100.00	956.30
1010	N	Bovis Test	Head Office	Commercial	Aberdeen City	A	0.00	0.00	406.90
1011	N	New Contract	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00
1012	N	Draft Cest Test	Head Office	House Building	Buckinghamshire	A	0.00	0.00	0.00
1111	N	QA CIV Testing	Head Office	Other	Aberdeen City	A	0.00	0.00	0.00
1112	N	QA PROJECT	Head Office	House Building	Aberdeen City	A	0.00	0.00	0.00
2000	N	Evergreen	Head Office	House Building	Warwickshire	A	0.00	0.00	0.00
2001	N	The Seawalls	Head Office	House Building	Cambridgeshire	A	0.00	0.00	0.00
22788	N	Test PDR	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00
22788A	N	Test PDR	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00

[Navigation icons: back, forward, search, etc.]
Search: Contract Number

# Discovery BI0002.3

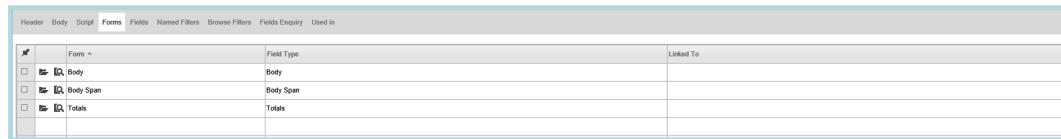
## 9.3 Adding Sections to the Contract Browse Screen

This exercise is to modify the contract browse screen to include additional sections.

### 9.3.1 Adding a Body Span Section

Edit the page section by clicking on the Open button  and save OR follow the link on the Page Name.

Select the Form tab create a Body Span form to use Body Span Fields.

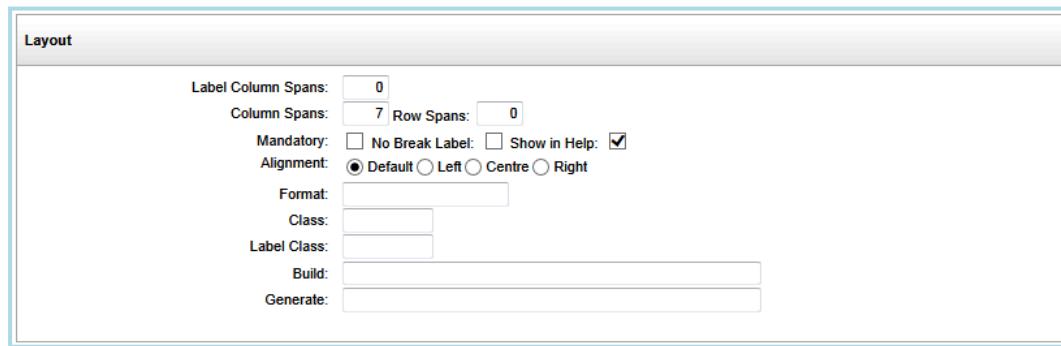


	Form	Field Type	Linked To
<input checked="" type="checkbox"/>	Body	Body	
<input checked="" type="checkbox"/>	Body Span	Body Span	
<input checked="" type="checkbox"/>	Totals	Totals	

In the Fields Tab go to Form selector at the bottom of the page, select Body Span and click Apply Filter.

Click  to add some more fields to the Body Span form.

Field	Label	Column Span
(Leave Blank)	Contract Details	7
	Values	3



Click Save  after adding each field.

Run the function:

Contract Status - NLMExercise1							Values		
Contract Number		Contract Name		Address		Values			
<input checked="" type="checkbox"/> 2001	N	Kingsley Meadows	Head Office	House Building	Oxfordshire	A	1,002.36	6,700.00	3,652.25
<input type="checkbox"/> 0002	N	The Bullring	Head Office	House Building	Vale of Glamorgan	A	690.00	15,980.00	17,544.00
<input type="checkbox"/> 0003	N	Ulsterstar Golf Course	Head Office	House Building	Staffordshire	A	0.00	0.00	1,090.50
<input type="checkbox"/> 0004	N	Hilton Hotel	Head Office	House Building	Lancashire	A	0.00	0.00	11,150.00
<input type="checkbox"/> 0005	N	Highbury Road	Head Office	House Building	Tyne & Wear	A	0.00	750.00	3,700.00
<input type="checkbox"/> 1009	N	Milfields	Head Office	House Building	Warwickshire	A	0.00	0.00	3,512.00
<input type="checkbox"/> 1011	N	Contractors Contract	Head Office	Commercial	Bristol	A	543.62	100.00	958.30
<input type="checkbox"/> 1019	N	Bovis Test	Head Office	Commercial	Aberdeen City	A	0.00	0.00	406.90
<input type="checkbox"/> 1011	N	New Contract	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00
<input type="checkbox"/> 1012	N	Draft Cost Test	Head Office	House Building	Buckinghamshire	A	0.00	0.00	0.00
<input type="checkbox"/> 1111	N	QA CVR Testing	Head Office	Other	Aberdeen City	A	0.00	0.00	0.00
<input type="checkbox"/> 1112	N	QA PROJECT	Head Office	House Building	Aberdeen City	A	0.00	0.00	0.00
<input type="checkbox"/> 2009	N	Evergreen	Head Office	House Building	Warwickshire	A	0.00	0.00	0.00
<input type="checkbox"/> 2001	N	The Swallows	Head Office	House Building	Cambridgeshire	A	0.00	0.00	0.00
<input type="checkbox"/> 2276	N	Test PDR	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00
<input type="checkbox"/> 2278A	N	Test PDR	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00

### 9.3.2 Adding a Body Detail Section

Edit the page section by clicking on the Open button

Select the Form tab create a Body Detail form to use Body Detail Fields.

The screenshot shows the 'Fields' tab selected in the top navigation bar. Below it is a table with a single row containing five items under the 'Forms' column. The first item is selected, indicated by a checked checkbox. The columns are labeled 'Form' and 'Field Type'. The rows are: 
 

- Form: Body, Field Type: Body
- Form: Body Detail, Field Type: Body Detail
- Form: Body Span, Field Type: Body Span
- Form: Totals, Field Type: Totals

In the Fields Tab go to the Form selector at the bottom of the page, select Body Detail and click Apply Filter.

Click to Add some more fields to the Enquiry screen. For example :

Field	Label
job_shipaddr_1	Address
job_shipaddr_2	(Blank)
job_shipaddr_3	(Blank)
job_shipaddr_4	(Blank)

The screenshot shows the 'Fields' tab selected in the top navigation bar. Below it is a table with four rows, each containing a field name and its label. The first row is selected, indicated by a checked checkbox. The columns are labeled 'Field' and 'Label'. The rows are:
 

- Field: job\_shipaddr\_1, Label: Address
- Field: job\_shipaddr\_2, Label: (Blank)
- Field: job\_shipaddr\_3, Label: (Blank)
- Field: job\_shipaddr\_4, Label: (Blank)

Click Save  after adding each field.

Now run the function.

Niton COINS

Contract Status - NLM1Exercise1							Values			
	Contract Number ~	Phase	Name	Group	Type	Location	Status	SL Net Debtors	PL Net Reg. Creditors	PL Net Costed Creditors
<input checked="" type="checkbox"/>	1_0001	N	Kingsley Meadows	Head Office	House Building	Oxfordshire	A	1,002.36	6,700.00	3,652.00
<input type="checkbox"/>	1_0002	N	The Bullring	Head Office	House Building	Vale of Glamorgan	A	690.00	15,980.00	17,548.00
<input type="checkbox"/>	1_0003	N	Ulsterstar Golf Course	Head Office	House Building	Staffordshire	A	0.00	0.00	1,092.00
<input type="checkbox"/>	1_0004	N	Hilton Hotel	Head Office	House Building	Lancashire	A	0.00	0.00	11,150.00
<input type="checkbox"/>	1_0005	N	Highbury Road	Head Office	House Building	Tyne & Wear	A	0.00	750.00	3,700.00
<input type="checkbox"/>	1_0009	N	Millets	Head Office	House Building	Wanacshire	A	0.00	0.00	3,512.00
<input type="checkbox"/>	1_0011	N	Contractors Contract	Head Office	Commercial	Bristol	A	543.62	100.00	956.30
<input type="checkbox"/>	1_0010	N	Bowes Test	Head Office	Commercial	Aberdeen City	A	0.00	0.00	406.00
<input type="checkbox"/>	1_0011	N	New Contract	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00
<input type="checkbox"/>	1_0012	N	Draft Cct Test	Head Office	House Building	Buckinghamshire	A	0.00	0.00	0.00
<input type="checkbox"/>	1_0011	N	QA CVR Testing	Head Office	Other	Aberdeen City	A	0.00	0.00	0.00
<input type="checkbox"/>	1_0012	N	QA PROJECT	Head Office	House Building	Aberdeen City	A	0.00	0.00	0.00
<input type="checkbox"/>	2_0009	N	Evergreen	Head Office	House Building	Wanacshire	A	0.00	0.00	0.00
<input type="checkbox"/>	2_0001	N	The Swallows	Head Office	House Building	Cambridgeshire	A	0.00	0.00	0.00
<input type="checkbox"/>	2_0004	N	Test PDR	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00
<input type="checkbox"/>	2_0004A	N	Test PDR	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00

Search: Contract Number

Niton COINS

Contract Status - NLM1Exercise1							Values			
	Contract Number ~	Phase	Name	Group	Type	Location	Status	SL Net Debtors	PL Net Reg. Creditors	PL Net Costed Creditors
<input checked="" type="checkbox"/>	1_0001	N	Kingsley Meadows	Head Office	House Building	Oxfordshire	A	1,002.36	6,700.00	3,652.00
<input type="checkbox"/>	1_0002	N	The Bullring	Head Office	House Building	Vale of Glamorgan	A	690.00	15,980.00	17,548.00
<input type="checkbox"/>	1_0003	N	Ulsterstar Golf Course	Head Office	House Building	Staffordshire	A	0.00	0.00	1,092.00
<input type="checkbox"/>	1_0004	N	Hilton Hotel	Head Office	House Building	Lancashire	A	0.00	0.00	11,150.00
<input type="checkbox"/>	1_0005	N	Highbury Road	Head Office	House Building	Tyne & Wear	A	0.00	750.00	3,700.00
<input type="checkbox"/>	1_0009	N	Millets	Head Office	House Building	Wanacshire	A	0.00	0.00	3,512.00
<input type="checkbox"/>	1_0011	N	Contractors Contract	Head Office	Commercial	Bristol	A	543.62	100.00	956.30
<input type="checkbox"/>	1_0010	N	Bowes Test	Head Office	Commercial	Aberdeen City	A	0.00	0.00	406.00
<input type="checkbox"/>	1_0011	N	New Contract	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00
<input type="checkbox"/>	1_0012	N	Draft Cct Test	Head Office	House Building	Buckinghamshire	A	0.00	0.00	0.00
<input type="checkbox"/>	1_0011	N	QA CVR Testing	Head Office	Other	Aberdeen City	A	0.00	0.00	0.00
<input type="checkbox"/>	1_0012	N	QA PROJECT	Head Office	House Building	Aberdeen City	A	0.00	0.00	0.00
<input type="checkbox"/>	2_0009	N	Evergreen	Head Office	House Building	Wanacshire	A	0.00	0.00	0.00
<input type="checkbox"/>	2_0001	N	The Swallows	Head Office	House Building	Cambridgeshire	A	0.00	0.00	0.00
<input type="checkbox"/>	2_0004	N	Test PDR	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00
<input type="checkbox"/>	2_0004A	N	Test PDR	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00

Search: Contract Number

### 9.3.3 Adding a Context Section

Edit the page section by clicking on the Open button 

Select the Form tab and create a Context form to use Context Fields.

In the Fields Tab go to the Form selector at the bottom of the page, select Context and click Apply Filter.

Click  to add some more fields to the Enquiry screen. For example:

Field	Label	Function
Text^ Contract Costs	Run Report	WJCRJOB

Click Save after adding the field.

The screenshot shows the 'Fields' tab in the OA Designer interface. A new field named 'Contract Costs' has been added to the 'Run Report' section. The 'Function' column for this field contains the value '0%WJCRJOB'. Other fields in the table include 'Field' (with icons), 'Label', 'Width', 'Height', 'Add Upd', 'View As', and 'Tab'.

This will create a Context section on the Page with a link to a Contract Status report.

The screenshot shows the 'Contract Status - NLM1Exercise1' report. At the top, there is a link labeled 'Run Report: Contract Costs'. Below it is a grid of contract details. The columns include Contract Number, Phase, Name, Group, Type, Location, Status, and several financial values (Sl. Net Debtors, Pt. Net Reg. Creditors, Pt. Net Costed Creditors). A search bar at the bottom is set to 'Search: Contract Number'.

Contract Number	Phase	Name	Group	Type	Location	Status	Values
0001	N	Kingsley Meadows	Head Office	House Building	Oxfordshire	A	1,903.38
0002	N	The Bulding	Head Office	House Building	Vale of Glamorgan	A	690.00
0003	N	Ulverster Golf Course	Head Office	House Building	Staffordshire	A	0.00
0004	N	Hilton Hotel	Head Office	House Building	Lancashire	A	0.00
0005	N	Highview Road	Head Office	House Building	Tyne & Wear	A	0.00
1000	N	Miffields	Head Office	House Building	Warwickshire	A	0.00
1001	N	Contractors Contract	Head Office	Commercial	Bristol	A	545.62
1010	N	Bovis Test	Head Office	Commercial	Aberdeen City	A	0.00
1011	N	New Contract	Head Office	Commercial	Aberdeen City	A	0.00
1012	N	Orat Cert Test	Head Office	House Building	Buckinghamshire	A	0.00
1111	N	QA CVS Testing	Head Office	Other	Aberdeen City	A	0.00
1112	N	QA PROJECT	Head Office	House Building	Aberdeen City	A	0.00
2000	N	Evergreen	Head Office	House Building	Warwickshire	A	0.00
2001	N	The Swallows	Head Office	House Building	Cambridgeshire	A	0.00
22786	N	Test PDR	Head Office	Commercial	Aberdeen City	A	0.00
22786	N	Test PDR	Head Office	Commercial	Aberdeen City	A	0.00

## Discovery BI0002.4

### 9.4 Adding Filters to the Contract Browse Screen

This exercise is to modify the contract browse screen to include advanced filters.

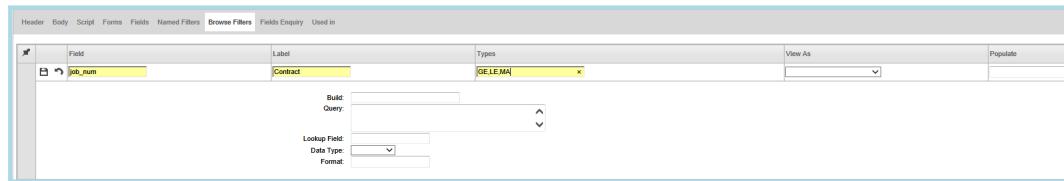
#### 9.4.1 Adding a Page Browse Filter

Edit the page section by clicking on the Open button  and save or follow the link on the Page Name.

Select the Browse Filters Tab.

Click  to Add fields to the filter as follows:

Field	Label	Types
job_num	Contract	GE,LE,MA



Click  to Add some more fields to the filter as follows:

Field	Label	Types
jcl_loc	Location	GE,LE,MA
jgr_group	Group	GE,LE,MA
jty_type	Type	GE,LE,MA

Click Save  after adding each field.

#### 9.4.2 Using Populates with a Browse Filter

Click  to add some more fields to the Enquiry screen. Add in a filter as follows (watch out for the mixture of underlines and dashes in the populate field!):

FieldLabelTypesView AsPopulate

job\_foreManagerMACCombo job-rsp.job\_fore



Click Save  after adding each field.

Now run the function.



Use the  button to access the advanced filter. Experiment with your new filters - remember to apply the filter after you change the criteria.



# Discovery BI0002.5

## 9.5 Adding a Body Selector

This exercise is to modify the contract browse screen to include Body Selectors and to apply the selection to fields

### 9.5.1 Adding a Body Selector Section

Edit the page section by clicking on the Open button .

Select the Form tab create a Body Selector form to use Body Selector Fields.

Header			Body	Script	Forms	Fields	Named Filters	Browse Filters	Fields Inquiry	Used in
	Form		Field Type							
<input checked="" type="checkbox"/>	Form +		Body							
<input type="checkbox"/>	Body		Body							
<input type="checkbox"/>	Body Detail		Body Detail							
<input type="checkbox"/>	Body Selector		Body Selector							
<input type="checkbox"/>	Body Span		Body Span							
<input type="checkbox"/>	Conted		Conted							
<input type="checkbox"/>	Totals		Totals							

In the Field Tab go to the Form selector at the bottom of the page, select Body Selector and click Apply Filter.

Click  to Add a new field to this screen. Add the following field :

FieldLabelPopulateView-As

EndPeriodTo Periodglp-rsp.glp\_fdateCombo

Data	
Sort:	<input type="text"/> Ignore Sort Limit: <input type="checkbox"/>
Validate:	
Populate:	<input type="text"/> Add Blank: <input type="checkbox"/> Blank Label: <input type="text"/>
Combo Limit:	<input type="text"/> 0
Populate Script:	<input type="text"/>
Total:	<input type="checkbox"/>



EndPeriod & StartPeriod have default behaviour in that by selecting the view-as as combo the description/value will always appear. If you did choose Combo description/value then the description would be duplicated.

Click Save  after adding the field.

### 9.5.2 Using the Body Selector value

In the Field Tab go to the FORM selector at the bottom of the page, select the Body Form Type and click Apply Filter.

NOTE : You should always do this to ensure you are adding fields to the correct form - adding fields to an incorrect form will mean that they do not display.

In the Body Form create the fields :

Field	Label
RO_ContractCosts^TD 0 {EndPeriod}	Costs to Date
RO_ContractRevenue^TD 0 {EndPeriod}	Revenue to Date



This will display the costs and Revenue to date for the date entered when the report is run.

The fields take three parameters:- the type of cost (TD = to date; TP = this period; TY = this year; TO = total)

- the offset from the period selected. For example: 0 = the period selected (no offset); -1 = the period before the one selected; !12-0 = period twelve in the year of the period selected; !1203 = period 12 in 2003.

- the period to report on. This is the value selected on the report selection.

Field	Label	Width	Height	Function	Add	Upd	View As	Tab	Layout	Append	Hidden
job_num	Contract Number	5	0	0%WJC014\$JOB	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
job_phase	Phase	5	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
job_name	Name	15	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
gr_desc	Group	10	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
ty_desc	Type	10	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
jc_desc	Location	10	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
job_active	Status	5	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
RO_job_st_debtors	SL Debtors	10	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
RO_job_pl_register	PL Registered	10	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
RO_job_pl_costed	PL Costed	10	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
RO_ContractCosts^TD 0 {EndPeriod}	Costs to Date	10	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
RO_ContractRevenue^TD 0 {EndPeriod}	Revenue to Date	10	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					

You will need to adjust your Body Span fields to reflect the new columns and adjust your field widths

Now run the function.

Contract Number	Phase	Name	Group	Type	Location	Status	SL Net Debtors	PL Net Reg. Creditors	PL Net Costed Creditors	Costs TD	Rev TD
0001	N	Kingsley Meadows	Head Office	House Building	Oxfordshire	A	1,002.38	6,700.00	3,652.25	5,239.00	5,239.00
0002	N	The Basing	Head Office	House Building	Vale of Glamorgan	A	690.00	15,980.00	17,548.00	0.00	0.00
0003	N	Uttweler Golf Course	Head Office	House Building	Staffordshire	A	0.00	0.00	1,092.50	0.00	0.00
0004	N	Hilton Hotel	Head Office	House Building	Lancashire	A	0.00	0.00	11,150.00	0.00	0.00
0005	N	Highview Road	Head Office	House Building	Tyne & Wear	A	0.00	750.00	3,700.00	0.00	0.00
1000	N	Mifields	Head Office	House Building	Worcsshire	A	0.00	0.00	3,512.00	0.00	0.00
1001	N	Contractors Contract	Head Office	Commercial	Bristol	A	543.62	100.00	956.30	-35.00	-35.00
1010	N	Bovis Test	Head Office	Commercial	Aberdeen City	A	0.00	0.00	406.90	0.00	0.00
1011	N	New Contract	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00	0.00	0.00
1012	N	Draft Cert Test	Head Office	House Building	Buckinghamshire	A	0.00	0.00	0.00	0.00	0.00
1111	N	QA CVR Testing	Head Office	Other	Aberdeen City	A	0.00	0.00	0.00	0.00	0.00
1112	N	QA PROJECT	Head Office	House Building	Aberdeen City	A	0.00	0.00	0.00	0.00	0.00
2000	N	Evergreen	Head Office	House Building	Warrickshire	A	0.00	0.00	0.00	0.00	0.00
2001	N	The Swallows	Head Office	House Building	Cambridgeshire	A	0.00	0.00	0.00	0.00	0.00
22788	N	Test POR	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00	0.00	0.00
22266	N	Test POR	Head Office	Commercial	Aberdeen City	A	0.00	0.00	0.00	0.00	0.00

Change the To Period value and check your final two column figures change.

## Discovery BI0003

### 10 Creating A Simple listing Report (supplier)

This exercise is to create a simple vendor listing. Each vendor is shown on a single line and there are no totals or calculations.

Purchase Ledger - NLM1Exercise6					
Housebuilders QA					
Supplier no	Supplier Name	Address			Postcode
A8B001	Abbey Glass	42 Bramall Lane	Sheffield		S2 4DL
AGG001	Aggregate Supplies	22 Ashton Gates	Bristol		BS30 5SJ
BOD001	BOD Plumbing & Heating Supplies	12 High Row	Darlington		DL37QQ
BR1001	British Gas	14 Jesmond road	Newcastle		NE11
CH1001	Chester Building Ltd	145 St Peter's Green	Perth		CA10
CHE001	Chebam Restorations	60 Broad St	Chesham		HP5 3EF
CIT001	City and Country Group	Test change of address	Warley		BW2 2HD
CIT002	City and Country Group	Bentfield Place	Bentfield Road	Essex	CM24 8HL
COR001	Cornwell Builders	2 The Avenue	Stansted		CB6 2DS
COR002	Cornwell Builders	Covent Garden	Cambridge		L1 1QP
DAV001	Davies Engineering Ltd	4 Briar Hill	London		IP205PP
EBO001	Ebööe Planning Limited	34 Goodison Way	Suffolk		L1 4EZ
ELU003	Elsec Limited	Oxford	Liverpool		WV16 5FF
EMA001	EB Emails Limited	Email House	Shropshire		WV16 5RR
GRA001	Gray Builders Ltd	12 Kingston Road	Kingston		KT15 2LK
HIG001	Highway Paints	78 Anfield Road			L1
JON001	Jones Lintel Supplies Limited	87 Lowdown Road	Exeter		EX2
LL0001	Lloyds TSB	34 The Grove	Slough		SL1
POP001	Popes Builders Merchants Ltd	2 Great Avenue	Chichester		BH7 6BT
SMH001	Stevens, Farrell & Vaux Solicitors	13 Standard park	Weston-super-Mare		NE11
SPE001	Speedy Hire Centres (Western) Ltd	22 Mace road	Bristol		BS2 0TX
SUT001	Sutton Electricals Ltd	56 Old Trafford Way	Warrington		WA6 0DB
Z Test	Z Test	Spb			CR9 1JT

Printed using QA COINS V10.27.121201 by Nigel Longley at 11:35:23 on 14/05/13 (NLM1Exercise6)

Page 1

#### 10.1 Create the Function and Menu

Create the function that will be used to run the report

Field	Value
Function Code	Initials + M1 + Exercise No.e.g. NLM1Exercise6
Function Name	Description identifier.e.g. NLM1Exercise6
Function Type	Function
Function Module	PL – Purchase Ledger
Function Category	Report
Function Context	Leave as defaulted from the Function Description.
Function Program	wou005

The screenshot shows the 'System - Function Maintenance' page. A new function 'NLM1Exercise6' has been created and is highlighted in yellow. The 'Type' dropdown is set to 'Function'. The 'Module' dropdown is set to 'PL\_Purchase Ledger'. The 'Category' dropdown is set to 'REP\_Report'. Below the table, there are fields for 'Context' (set to 'NLM1Exercise6'), 'Program' (set to 'woud05'), 'Parameters', 'Notes', 'Parent' (set to 'NLM1Exercise6'), 'Access Type' (set to 'Read Only'), and 'Rule Type'.

## 10.2 Set up the Page

Set up the page section using Page Designer. Click Add to create the new Page and fill in the fields as follows (leave the others blank):

Field	Value
Page	Give the page section the same name as the function created.
Form Service Procedure	plfrep.p

The screenshot shows the 'System - Add' page for creating a new page section. The 'Page' field is set to 'NLM1Exercise6'. Under the 'Header' tab, the 'Header Program' dropdown is set to 'PopUp'. The 'Form Service Procedure' dropdown is set to 'plfrep.p'. Other fields like 'Post Update', 'Next Action', 'Header Query', and 'Link Advanced Filter' are left blank. The 'Record' button is visible at the top right.

Click Save to save the new Page. COINS returns to the summary for the page section being created.

## 10.3 Add the Page Section Forms

Select the Form tab create an Update form.

Click the Add Button and enter the following information to create the Form :

Field	Value
Form	Select Update
Field Type	Select Update
Linked To	Leave blank

Click the Save Button to create the form.

## 10.4 6.4 Add the Fields

In the Field Tab go to the FORM selector at the bottom of the page, select Update and click Apply Filter.

**NOTE :** You should always do this to ensure you are adding fields to the correct form - adding fields to an incorrect form will mean that they do not display.

Add the Field, leaving Field and Label blank :

In the Generate field, enter plfrep.avmSelectionGenerate. This is the standard selection generator for users.



Click the Save Button to create the field.

Test the report selection by running the function from the Menu.

Click the menu item for the report. COINS should display the selection criteria:

If you were to press the green arrow to generate the report it will not produce any output, because the report section has not yet been created. If you refer to the log file you will see an error message similar to the one below.

## 10.5 Create the report section

Set up the report section:

Click the Add button and fill in the details as follows:

Report Section	Give the report section the same name as the function created.
----------------	--

Header Tables	Enter the Header Table – using a configuration header (such as co_config or and module configuration table) allows the Page access to additional tables to the main Body Table.
Body Table	Enter the name of the database table being reporting on; in this case, ap_vendor.
Body Query	Enter the query. This will select which records are shown on the report. Enter the following:  FOR EACH ap_vendor WHERE kco = {kco}{avmSelect}>
Page Layout Class	Select %A4RLAND-A4 Report Landscape Select the font class to use. Since the user listing report doesn't have many fields on, use a font such as Arial 8pt (a larger font than Arial 6pt)

The query selects all supplier records (ap\_vendor) that are for the current logged in Company {kco}, and which match the selection criteria chosen when the report is run ({avmSelect}).

The query must include a WHERE statement, and {avmSelect} will be expanded to a query that begins AND; therefore to include the standard selection after a FOR EACH (for example if you were to leave out the kco selection), you must add WHERE TRUE.

For example:

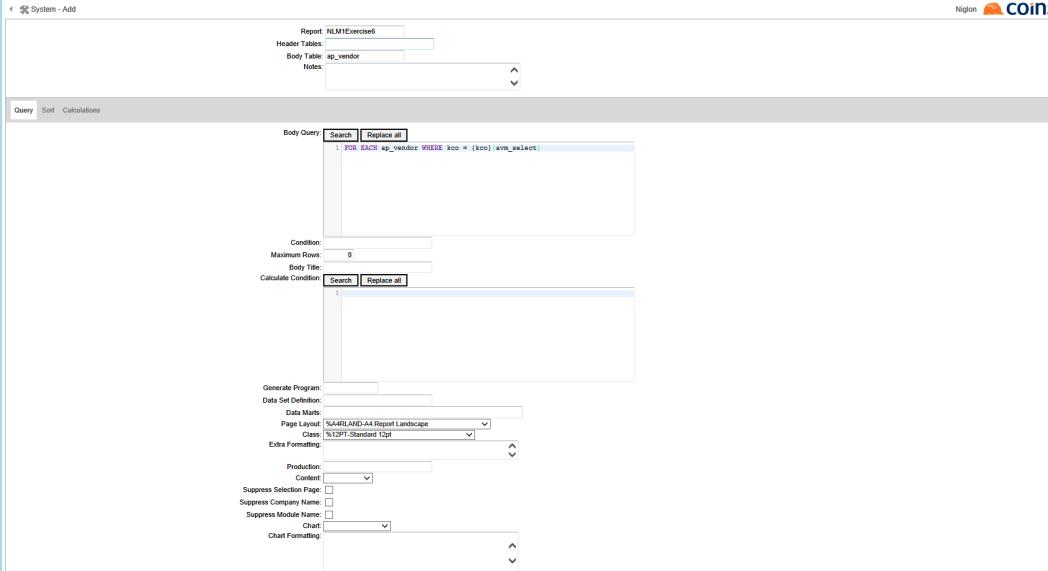
```
FOR EACH ap_vendor WHERE TRUE  
{avmSelect}
```



Running reports on transaction tables across companies by leaving out the kco selection may result in extremely large numbers of records being accessed which may severely impact both report and server performance.

The page layout determines the orientation (whether the report format is landscape or portrait), the margins, and the standard headers and footers.

For COINS reports, with a logo, a title, and a report footer, choose a report layout (with an R in the name: %A4RLAND or %A4RPORT).



The screenshot shows the 'Report' section of the OA Designer interface. The report is titled 'NLMEExercised'. The 'Body Query' field contains the following PL/SQL code:

```

Report NLMEExercised
Header Table:
Body Table: ap_vendor
Notes:

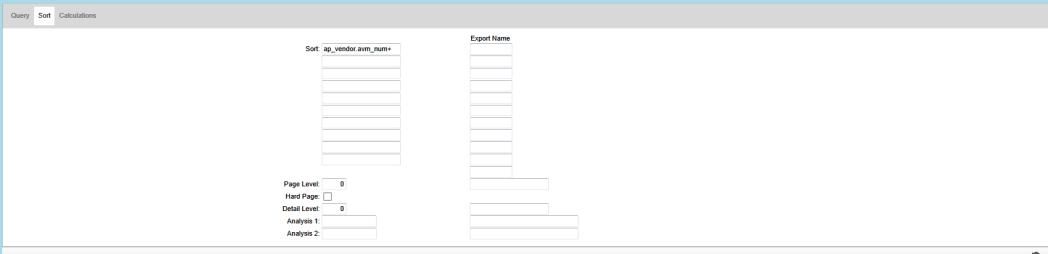
Body Query:
  Search Replace all
  i FOR EACH ap_vendor WHERE kco = (kco) avm_select;
Condition:
  MaxRows: 0
  Body Title: Calculate Condition
  Search Replace all

Generate Program:
Data Set Definition:
  Data Matrix:
    Page Layout: %480x144 Report Landscape
    Class: %12P1-Standard 12pt
  Extra Formating:
    Production:
      Content: 
      Suppress Selection Page: 
      Suppress Company Name: 
      Suppress Module Name: 
      Chart: 
      Chart Formating:

```

## 10.6 Sorting the Report

On the sort tab, enter ap\_vendor.avm\_num+ in the Sort field and Click Save .



The screenshot shows the 'Sort' tab of the OA Designer interface. The 'Sort' field contains 'ap\_vendor.avm\_num+'. The 'Export Name' field is empty. Below the fields, there are options for Page Level (0), Hard Page (unchecked), Detail Level (0), Analysis 1, and Analysis 2.

## 10.7 Create the Report Form

Select the Form tab create a Body form to use Body Fields.

Click the  Add Button and enter the following information to create the Form:

Field	Value
Form	Select Body
Field Type	Select Body
Linked To	Leave blank

Click the Save Button to create the form.

## 10.8 Add the Fields

In the Field Tab go to the FORM selector at the bottom of the page, select Body and click Apply Filter.



You should always apply the filter to ensure you are adding fields to the correct form - adding fields to an incorrect form will mean that they do not display.

For each of the fields to be shown on the report, fill in the following:

Field	Label	Width
avm_num	Supplier no.	15mm
avm_name	Supplier Name	60mm
avm_add_1	Address	40mm
avm_add_2		40mm
avm_add_3		40mm
avm_add_4		40mm
avm_pcode	Postcode	15mm

Click Save after entering each field.

The report is now ready to run.

## 10.9 Running the Report

1. Click the menu item for the report.
2. Enter selection criteria.

For example, to display only Suppliers that begin with D, enter D\* in the Matches field.

3. Click on the Output Options tab and select the appropriate options.
4. Click NEXT to run the report.
5. Go to Report Status (there is usually a Report Status option on the Reports menu in each module – or there is a shortcut button on the bottom of the page
6. The report should show on the list – look for the report title in the Description column. If the Status column shows Waiting or Generating, the report is not yet ready. Click Undo Button until the Status column shows Complete.
7. Click the report title link in the Description column.

COINS displays the report as a PDF file.

Purchase Ledger - Module 1 Exercise 6						COINS Construction	COINS
Supplier No.	Supplier Name	Address					Postcode
AAR001	Aardvark Decorating Services	9 Davey Lane	The Broadwalk	Carlisle			CA1 5FR
ABA001	Abacus Builders Merchants (Walsall) Ltd	Loreto, Main Road	Crawcrook	Ryton	Tyne & Wear	NE40 4TP	
ACD001	Acrylic Glass	18 Ryton Road	North Anston	Sheffield	South Yorkshire	S26 4LQ	
ABR002	Abbott Glass xxx	15 Ryton Road	North Anston	Sheffield	South Yorkshire	S26 4DL	
ABC002	A-B-C Ltd	14 Apsley Road	BRIDGTON				BH7 4OG
AGS001	Aggregate Supplies	33 St Andrews Road	Knowle	Bristol			BS30 6SJ
AHD001	A&H Demolition Services	212 Oxford Road	Wythenshawe	Manchester			M32 8FR
ARN001	Arnold Laver & Company Limited	Bramall Lane	SHEFFIELD				S2 4RJ
ASH001	Ashford Plant PLC	Wharfside	Ashford	Kent			TN23 1AA
ATS001	ATS Plumbing & Heating Supplies	35 Lord Lane	Fallsforth	Manchester			M35 3DN
BRU001	Bristol Builders Supplies Ltd	Unit 43	Orchard Court	Great Western Business Park	Bristol		BS37 5SY
BRU003	British Telecom	Thames Valley office	120 Riverside Way	Slough	Berkshire		SL2 7GF
BUD001	Bud & Bird Ltd	Parkinson Avenue	Southamptope	South Humbershire			DN16 7NA
C&S001	C&S Builders Merchants Ltd	97, Woodside Avenue	Chislehurst	Kent			BR7 6BT
CAR001	Carer Supplies	14 The Grove	Slough	Berkshire			SL1 7QQ
CLA002	T. Clarke	116-118 Walworth Road	London				SE17 0JV
COJ001	COINS Construction	12 The Grove	Slough				SL1 8DJ
COND001	Condron Concrete	Arden Road	Tullamore	Co Athlone			AB
COS001	Costello Engineering	Tallaght	Dublin 24				
DAI001	Dairygold Ltd	Mallow	Co Cork				
DUB001	Dubai Providers Ltd	Bathrom World co 01	Old Kilmainham	Dublin 8			SL1
EDP001	EDP Group	East Park Business Park	Fairview	Dublin 3			SL1
FAO001	Factored Supplier	150 High Street	Slough	Newcastle			SL1 7OD
FIN001	Financial Collections Ltd	PO Box 200	Gosforth				NE18 5RD
GAR001	Garnett Timber Supplies	Unit 2 54 Kimber Road	Wandsworth	London			SW18 4PP
god001	Godfrey Exports Limited			timdrake			DY13 0GH
GOO002	Goode Concrete	Unit 1 A	Naas Rd Ind Est	Naas Rd			D22
GRE001	Greenwood Trading	15-25 Cathedral Park	Guilford				GU12 7TH
GSD000	GSD Repairs	123 Chertsey Road	Chertsey	Surrey			KT16 0GA
HAN001	Handley Building Materials UK	Reading Road	Newbury	Berkshire			RG14 7TG
HEW001	Heon Buckley	123a Naas Road Ind Estate	Naas Road		Dublin		
HIL001	Hillson International	1 Main Street	Chipping Sodbury				D22
HOB002	Hobson Limeworks	1 Hill Street	Yate				BS23 6UL
HOW001	Aaron Howard Fencing	Long Lane Farm	Pad novità Road	Chelmsford	Essex		CM23 6GF
IWS001	Ibsworth Brick Limited	Nottingham Road	Ibsworth	Nottinghamshire			NG10 9PB
KEN001	Kenwood Tiles Limited	Kenwood House	Chester Road	Chelmsford			CM1 8GT
KEY001	Keyline Builders Merchants	Bolton Road	Radcliffe	Manchester			M27 6GH
NEW001	The New Company	The Road	Worcester	Worcestershire			DY14 8NF
OBD001	William O'Brien Plant	Athenry	Co Galway				
PAT001	Pat Collins	6 lillian Avenue					W3 9AW
PLU001	Plumb Centre	119 Ashton Road	Manchester				M11 1AA

Printed using COINS V10.23.110520+ by Nigel Longley at 11:49:58 on 02/05/11 (NLM/Exercise)

Page 1

The excel and XML files can be opened by selecting the relevant icon.

Report Status	Saved Reports	Scheduler	Report Runner	Archived Reports	Status	Queue	Size	User	
<input type="checkbox"/>		30/04/13	16:00:51	1 PL	intTestexcell8	Complete	GENERAL	17,139	NIGLON

The log file  can be used to debug the report if you experience any errors with the report.

## Discovery BI0004

### 11 Creating a Contract Listing Report

This contract report includes totals, sorting, header and footer forms, and a calculation.

		Code	Name	Location	Costs to Date	Revenue to Date	Profit
<b>Group: Build - Commercial Building</b>							
Type: CompTen - Competitively Tendered							
1006	Slough Housing Estate		West London		26.88	0.00	-26.88
Type: CompTen - Competitively Tendered					<b>26.88</b>	<b>0.00</b>	<b>-26.88</b>
<b>Group: Build - Commercial Building</b>							
					<b>26.88</b>	<b>0.00</b>	<b>-26.88</b>
<b>Group: Housing - LA &amp; Housing Assoc Housing</b>							
Type: SchRate - Schedule of Rates MTC							
1001	Carlton Road - Gateshead		North East (Leeds & Gateshead)		58,044.83	67,952.25	9,907.42
Type: SchRate - Schedule of Rates MTC					<b>58,044.83</b>	<b>67,952.25</b>	<b>9,907.42</b>
Type: SW - Small Works							
1002	Xheltered Housing Cambridge		East Anglia (Peterborough)		0.00	0.00	0.00
Type: SW - Small Works					<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
<b>Group: Housing - LA &amp; Housing Assoc Housing</b>							
					<b>58,044.83</b>	<b>67,952.25</b>	<b>9,907.42</b>
<b>Group: LA - Local Authority general</b>							
Type: CompTen - Competitively Tendered							
1000	Burnham Sports Centre		Underpinning NE		117,600.81	84,897.69	-32,703.12
1003	Blenheim Place		Midlands (Bromsgrove & Derby)		146,307.63	400.00	-145,907.63
1004	Melchester Rovers F C Site		Southern Buying Region		0.00	600.00	600.00
Type: CompTen - Competitively Tendered					<b>263,908.44</b>	<b>85,897.69</b>	<b>-178,010.75</b>
<b>Group: LA - Local Authority general</b>							
					<b>263,908.44</b>	<b>85,897.69</b>	<b>-178,010.75</b>

#### 11.1 Create the Function

Set up the function to run the report.

Field	Value
Function Code	Initials + M1 + Exercise No.e.g. NLM1Exercise7
Function Name	Description identifier.e.g. NLM1Exercise7
Function Type	Function
Function Module	Contract Status
Function Category	Report
Function Context	Leave as defaulted from the Function Description.
Function Program	wou005

Set up a second function with the same code but suffixed with a "T", this function will be used for additional selection criteria for the report.

For example:

Function Name = NLM1Exercise7  
Tab Function = NLM1Exercise7T

	Function ▾	Description ▾	Type	Module ▾	Category
□	↳ ↴ nlm1exercise7	nlm1exercise7	Function	JC-Contract Status	REP-Report
□	↳ ↴ nlm1exercise7T	nlm1exercise7T	Function	JC-Contract Status	REP-Report

## 11.2 Create the Page

Set up the page section to prompt for the selection criteria. The Form Service Procedure for Contract Selection is jcfrrep.p.

The screenshot shows the 'Header' tab of the Page creation interface. It includes fields for Header Tables, Body Table, Notes, Header Program, Popup, Form Service Procedure (set to jcfrrep.p), Post Update, Next Action (with Forward selected), Header Query, Summary, Include Script, Send Data, Pre Get, Link Named Filter, and Link Advanced Filter.

## 11.3 Add the Page Section Forms

Select the Form tab and add an Update form to use Update Fields.

The screenshot shows the 'Forms' tab of the Oracle Application Designer interface. A single row is listed under the 'Forms' column, labeled 'Update' with a 'Field Type' of 'Update'. The 'Linked To' column is empty. The bottom of the screen shows standard toolbar icons and a search bar.

Click the Save Button to create the form.

## 11.4 Add the Page Fields

In the Field Tab go to the FORM selector at the bottom of the page, select Update and click Apply Filter.



You should always do this to ensure you are adding fields to the correct form - adding fields to an incorrect form will mean that they do not display.

The FSP (form service procedure) already has a function to build the standard selection criteria.

Add a new field, and in the Tab field, enter the function code of the second function (i.e. the one suffixed with a 'T') created in Stage 1 above. This puts the standard contract selection criteria on a different named tab from Main.

**NOTE:** Once you have started to use Tab names you cannot leave any further Tab fields blank.

In the Generate field, enter jcfrrep.jobSelectionGenerate.

**Layout**

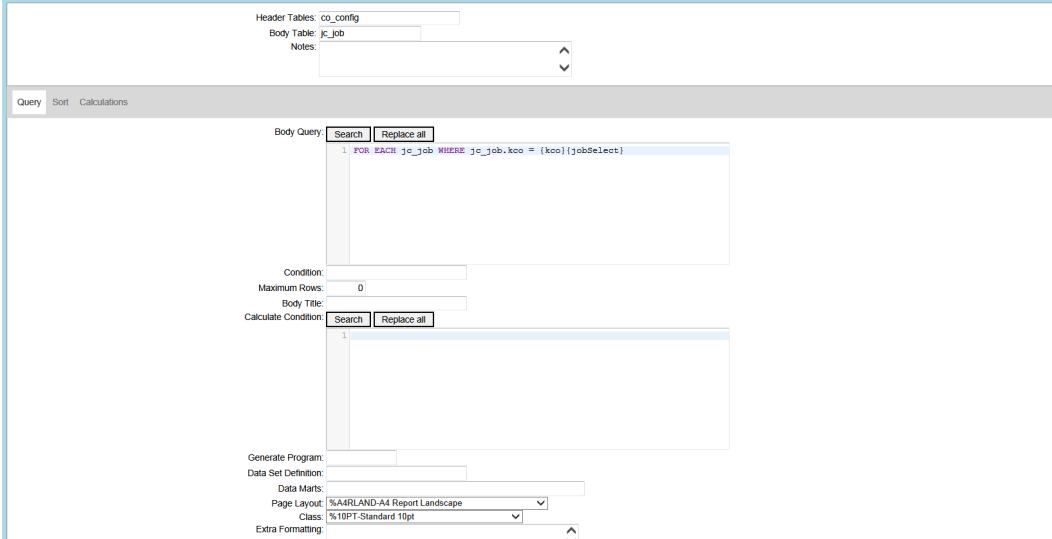
Label Column Spans:	<input type="text" value="0"/>
Column Spans:	<input type="text" value="0"/> Row Spans: <input type="text" value="0"/>
Mandatory:	<input type="checkbox"/>
Alignment:	<input checked="" type="radio"/> Default <input type="radio"/> Left <input type="radio"/> Centre <input type="radio"/> Right
Format:	<input type="text"/>
Class:	<input type="text"/>
Label Class:	<input type="text"/>
Build:	<input type="text"/>
Generate:	<input type="text" value="jcfrrep.jobSelectionGenerate"/>

## 11.5 Create the report section

[Go to Report Designer](#)

Click the Add button  and fill in the details as follows:

Field	Value
Report Section	Give the report section the same name as the function created.
Header Tables	Enter the Header Table – using a configuration header (such as co_config) allows the Page access to additional tables to the main Body Table.
Body Table	Enter the name of the database table being reported on; in this case, jc_job.
Body Query	<p>Enter the query. This will select which records are shown on the report. Enter the following:</p> <pre><b>FOR EACH jc_job</b> <b>WHERE jc_job.kco = {kco}{jobSelect}</b></pre> <p>This selects each contract in the current company that matches the selection criteria chosen when the report is run.</p>
Page Layout	%A4RLAND-A4 Report Landscape
Class	8pt Standard



## 11.6 Sorting the Report

The report will sort and subtotal by Contract Group, then by Type, then by Contract Code, so on the sort tab, enter the following fields in the Sort fields:

jgr\_group+  
jty\_type+  
job\_num+

A + after the fieldname means sort in ascending order; a - means sort in descending order.

Note: By using a - you can break the index so be aware you may slow down your report.

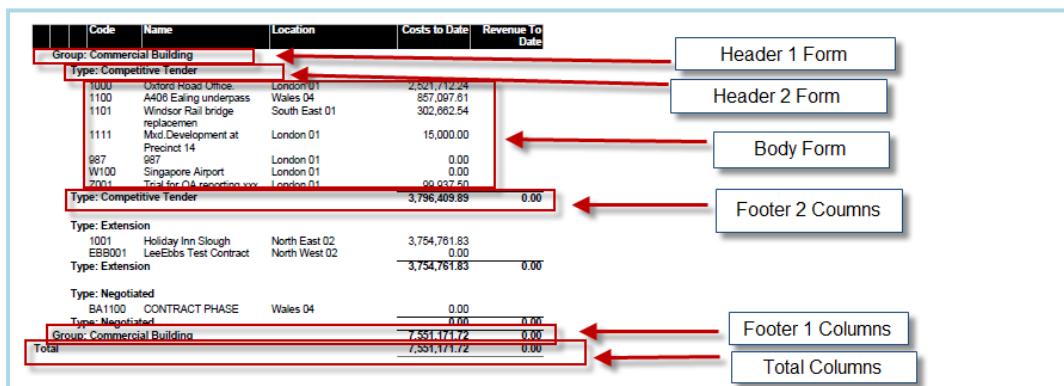
The screenshot shows the COINS OA Designer interface. At the top, there are tabs for 'Query', 'Sort', and 'Calculations'. Below these are sections for 'Sort' (jgr\_group, jty\_type, job\_num), 'Export Name', and 'Page Level' (0). There are also fields for 'Hard Page' (checkbox), 'Detail Level' (0), 'Analysis 1', and 'Analysis 2'.

Click Save

## 11.7 Create the Report Forms

Set up the forms for the report.

The report will have subtotals for contract type and contract group (shown in Footer Column forms), as well as totals for the whole report (shown in a Total Columns form). Also, we can show information about each new group and type at the beginning of each section, using Header Forms.



Set up the following forms (select Forms from the drop-down menu):

Body	This will show the details for each record in the body query; in this case, one per contract.
Footer 1 Columns	This will show subtotals for the first field being sorted by; in this case, contract group (jgr_group).
Footer 2 Columns	This will show subtotals for the second field being sorted by; in this case, contract type (jty_type).
Header 1 Form	This will show a header for the first field being sorted by.
Header 2 Form	This will show a header for the second field being sorted by.
Total Columns	This will show grand totals for the whole report.

## 11.8 Add the Fields

In the Field Tab go to the FORM selector at the bottom of the page, select the Body form type and click Apply Filter.

**NOTE:** You should always do this to ensure you are adding fields to the correct form - adding fields to an incorrect form will mean that they do not display.

For each of the fields to be shown on the report, fill in the following:



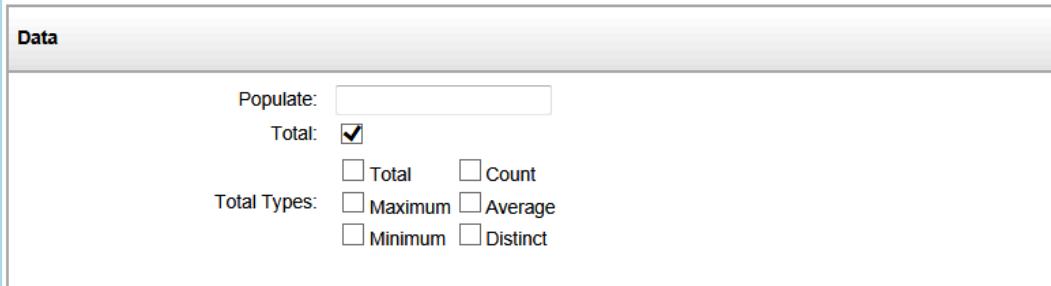
For Report column widths you must specify the unit of measure - we will use mm).

Field	Label	Width
job_num	Code	15mm
job_name	Name	35mm
jcl_desc	Location	35mm
RO_ContractCosts^TD	Costs to Date	20mm
RO_ContractRevenue^TD	Revenue to Date	20mm

The RO Fields will display the costs and Revenue to date for the default date when the report is run.

## 11.9 Add Totals to the Report

To show the subtotal (for group and type) and total values on the report, tick the Total field for the columns to be totalled: in this case, the Costs to Date and Revenue to Date.



The screenshot shows the 'Data' tab of the COINS OA Designer interface. It includes fields for 'Populate:' (a dropdown menu), 'Total:' (a checked checkbox), and 'Total Types:' (checkboxes for Total, Count, Maximum, Average, Minimum, and Distinct).

It is not necessary to define any fields for the Header Form, Footer Column or Total forms. COINS will automatically display the labels, and lines up the total and subtotal values up with the columns in the body. However, to override the standard layout (to show the total columns in different places, or to put more information in the headers), this is possible by specifying the fields for the relevant form.

The side labels that COINS shows on the headers and footers are set up using Default Report Labels. Labels for most of the common fields have already been set up, so it is not necessary to add anything.

But if the report sorts by a field that does not already have a default label, the report will show Sort n, and it will be necessary to set up a default label for the field.

## 11.10 Run the report

Run your report for Exercise 7 and check you get output similar to:

**Purchase Ledger - Module 1 Exercise 7**  
COINS Construction



	Code	Name	Location	Costs to Date	Revenue To Date
Group: Commercial Building					
Type: Competitive Tender					
1000	Oxford Road Office	London 01		2,521,712.24	
1100	A406 Ealing underpass	Wales 04		857,007.61	
1101	Windsor Rail bridge replacement	South East 01		302,662.54	
1111	Mxd Development at	London 01		15,000.00	
	Precinct 14				
887	987	London 01		0.00	
W100	Singapore Airport	London 01		0.00	
Z001	Tral for OA reporting xxx	London 01		99,937.50	
				<u>3,756,409.89</u>	<u>0.00</u>
Type: Competitive Tender					
Type: Extension					
1001	Holiday Inn Slough	North East 02		3,754,761.83	
EBB001	LeeEbs Test Contract	North West 02		0.00	
				<u>3,754,761.83</u>	<u>0.00</u>
Type: Extension					
Type: Negotiated					
BA100	CONTRACT PHASE	Wales 04		0.00	
Type: Negotiated					
Group: Commercial Building					
Total				<u>7,551,171.72</u>	<u>0.00</u>
				<u>7,551,171.72</u>	<u>0.00</u>

Printed using COINS V10.23.1 (0552) by Nigel Longley at 12:35:56 on 03/09/11 (NLM\Exercise7)

Page 1

# Discovery BI0005

## 11.1 Summarising Reports

### 11.1.1 Copy the Report from BI0004

So that both a detailed and summary report can be run, first copy each of the Function, Page and Report Sections of the example created in Discovery BI0004 and call them xxBI0005 (where xx = your initials) and add the new function to the menu.



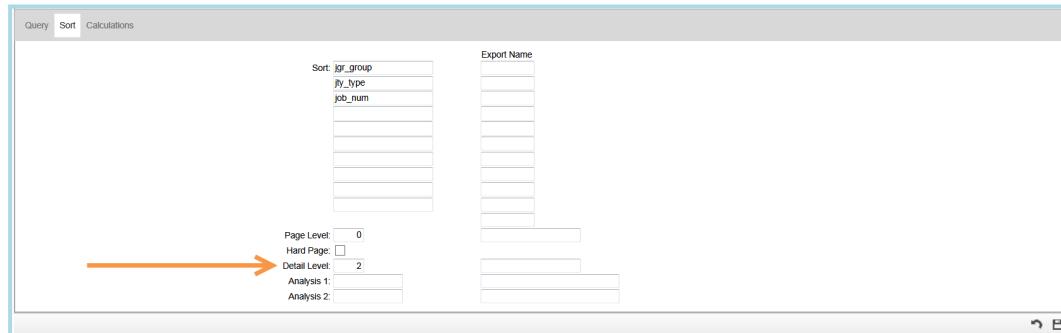
You will need to amend the field on the Update form to use the Exercise 8 Tab Function not the Exercise 7 Tab function. Be aware of changes like this when you copy functions/pages/reports unless you specifically want to reuse functions.

### 11.1.1.1 Select the Sort Level

The report will sort and subtotal by Contract Group, then by Type, then by Contract Code :

jgr\_group+  
jty\_type+  
job\_num+

To summarise the data at Contract Type enter 2 into the Detail field on the Sort Tab.



### 11.1.1.1.1 Select the correct Header and Footer Forms

When selecting a report to summarise it is not necessary (nor is it permitted) to have headers and footers at any sort level on or below the summary level.

In this exercise, headers and footers are only required at Sort Level 1 so delete the headers and footers at level 2

Run the report to return data summarised at Contract Type.

## Purchase Ledger - Module 1 Exercise 8

COINS Construction

	Code	Name	Location	Costs to Date	Revenue To Date
Group:					
z004	test auto number sequence	Test		218.45	0.00
Group:				218.45	0.00
Group: Commercial Building					
2001	Trial for OA reporting xxx	London 01		3,798,409.89	0.00
EEB8001	LeeEbbis Test Contract	North West 02		3,754,761.83	0.00
BA1100	CONTRACT PHASE	Wales 04		0.00	0.00
Group: Commercial Building				7,551,171.72	0.00
Group: Civil Engineering					
1200	Bluewater shopping M & E	Overseas 00		21,881,437.65	0.00
TIM999	Tim D Test	London 01		0.00	0.00
1103	Heathrow T4 hydrant fuel insta	Scotland 02		387,821.36	0.00
1002	Terminal 5 retail fitout	North West 02		6,888,047.81	0.00
Group: Civil Engineering				29,157,306.82	0.00
Group: Commercial Fitout					
1102	Eton River Thames bank rebuilding	Scotland 02		555,629.23	0.00
T2	Terminal 2	North West 02		0.00	0.00
6000	FM Contract	South West 04		-400.00	0.00
Group: Commercial Fitout				555,229.23	0.00
Group: Facilities Management					
6110	Oxford CC Maintenance	South East 01		244.00	0.00
Group: Facilities Management				244.00	0.00
Group: Miscellaneous					
T1	T1	Test		0.00	0.00
FSS0	FSS Overheads	South West 04		0.00	0.00
6200	A465 Vale of Neath Bridge	South West 04	Hirwaun	395.00	0.00
Group: Miscellaneous				395.00	0.00
Group: Rail					

Printed using COINS V10.23.110502+ by Nigel Longley at 12:59:47 on 02/08/11 (NLM IT Services)

Page 1

### **11.11 Amend the fields on the report**

Due to the fact that the report is now summarised it is necessary to ensure that the fields you are displaying reflect the type of report you are producing

For example : Job Number and Job Name are no longer relevant to display as multiple contracts may be used to make up the detail line

Therefore you may want to delete or hide these fields from the report .... Also you may like to add the jty\_desc field to the report so that you can see that the Type Description as this is now the detail level.

# Discovery BI0006

## 11.12 Calculated Fields

### 11.12 Copy the Report created in BI0004

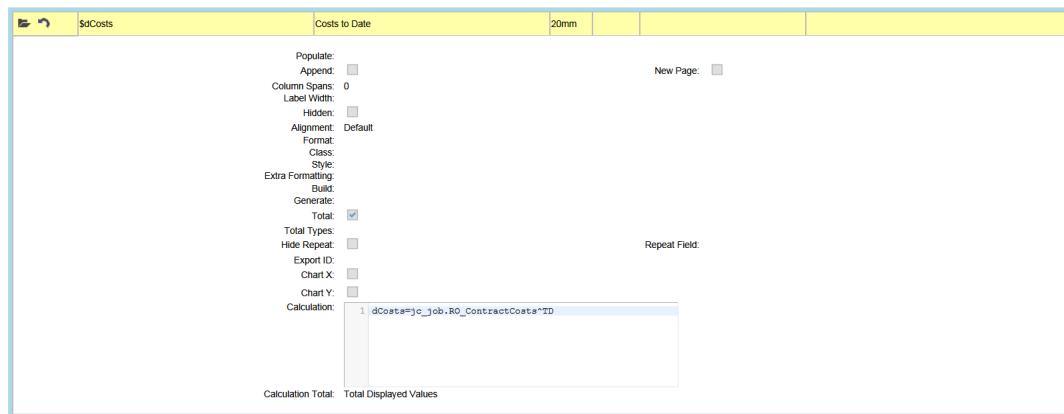
Copy each of the Function, Page and Report Sections of the example created in Discovery BI0004 and call them xxBI0006 (where xx = your initials) and add the new function to the menu.

### 11.12 Add Calculated Fields

To calculate a value from fields on the report, create variables within the report to hold the values of the fields, then create a new calculation field to hold (and display) the result.

In the Field field of the Costs to Date line, enter a \$variable name to accumulate the totals for this field... and assign the value of the RO field in the calculation box.

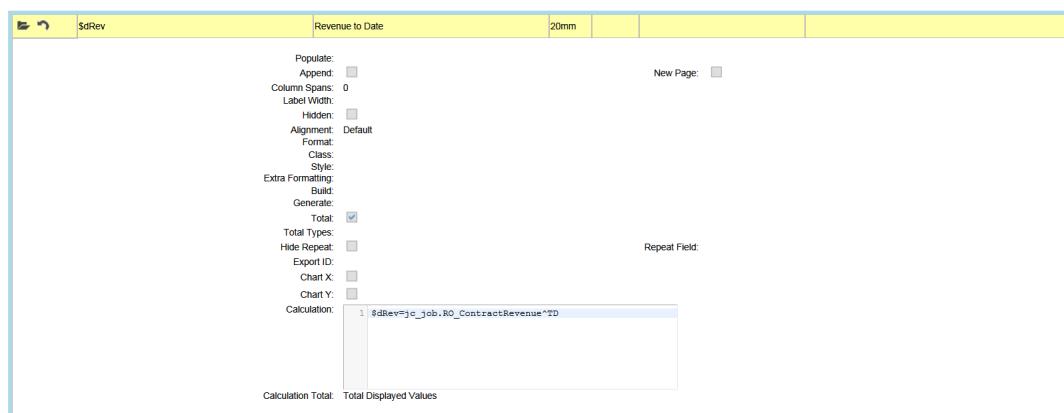
Field	Label	Width	Calculation
\$dCosts	Costs to Date	20mm	dCosts = jc_job.RO_ContractCosts^TD



The screenshot shows the OA Designer interface for creating a calculated field. The field is named '\$dCosts' with a label 'Costs to Date' and a width of '20mm'. The 'Calculation' field contains the formula 'dCosts = jc\_job.RO\_ContractCosts^TD'. The 'Total' checkbox is checked, indicating that a total will be calculated for this field. Other options like 'Append', 'Column Spans', 'Label Width', 'Hidden', 'Alignment', 'Format', 'Class', 'Style', 'Extra Formatting', 'Build', 'Generate', 'Total Types', 'Hide Repeat', 'Export ID', 'Chart X', 'Chart Y', and 'Repeat Field' are visible but not selected.

Repeat this procedure for the Revenue To Date.

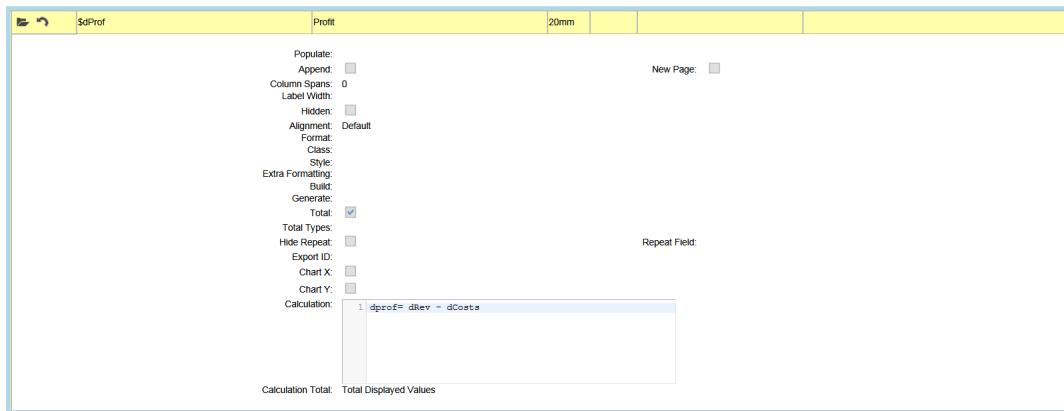
Field	Label	Width	Calculation
\$dRev	Revenue to Date	20mm	dRev = jc_job.RO_ContractRevenue^TD



The screenshot shows the OA Designer interface for creating a calculated field. The field is named '\$dRev' with a label 'Revenue to Date' and a width of '20mm'. The 'Calculation' field contains the formula 'dRev = jc\_job.RO\_ContractRevenue^TD'. The 'Total' checkbox is checked, indicating that a total will be calculated for this field. Other options like 'Append', 'Column Spans', 'Label Width', 'Hidden', 'Alignment', 'Format', 'Class', 'Style', 'Extra Formatting', 'Build', 'Generate', 'Total Types', 'Hide Repeat', 'Export ID', 'Chart X', 'Chart Y', and 'Repeat Field' are visible but not selected.

Add a new line for Profit. This does not need a field name, since it is not actually a database field however we are going to assign a variable as we are going to use this field in a further calculation. Fill in the fields as follows:

Field	Label	Width	Calculation
\$dProf	Profit	20mm	<p>dprof=dRev - dCosts</p> <p>This subtracts the value in the variable dCosts from the value in the variable dRev and displays the result</p> <p>NOTE: Leave a space either side of the minus sign.</p>

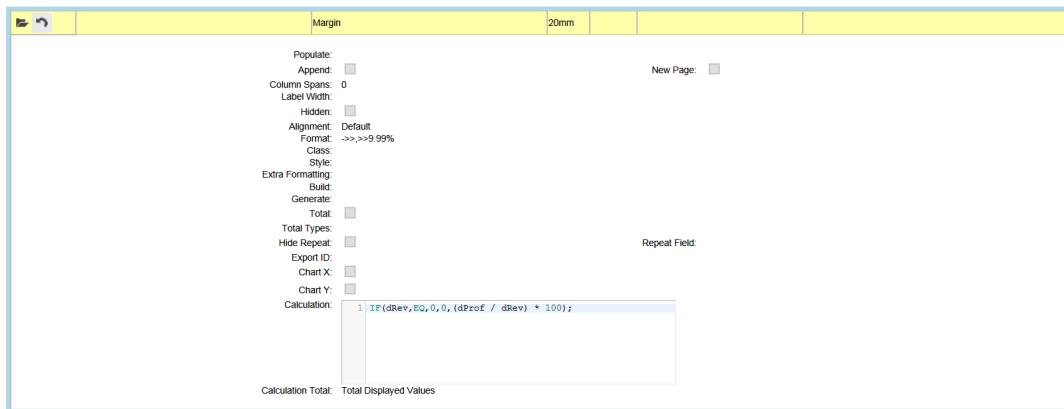


Add a new line for Profit Margin. Fill in the fields as follows:

Field	Label	Width	Calculation
Must be blank as percentage should calculated at time of calculation not accumulate	Profit Margin	20mm	<p>IF(dRev, EQ, 0, 0, (dProf / dRev) * 100);</p> <p>This calculation will calculate the Margin as a percentage figure. It will also allow for the case if drev is zero and force the answer to be zero. (As dividing by zero is not possible.)</p>

Set the Format to >>,>>9.99%

Because this is not an actual database field, it is necessary to specify a display format. (Fields in the database have a pre-defined format. Always allow for Neg figures)



Run your report and check that you have output similar to:

**JC Contract Status - Module 1 Exercise 9**  
COINS Construction



	Code	Name	Location	Costs to Date	Revenue To Date	Profit	Margin
Group:							
Type:							
euro	Euro Contract	Test		68.45	0.00	-68.45	0.00%
P001	Discontinued contract	Test		0.00	0.00	0.00	0.00%
z004	test auto number sequence	Test		150.00	0.00	-150.00	0.00%
				<b>218.45</b>	<b>0.00</b>	<b>-218.45</b>	
				<b>218.45</b>	<b>0.00</b>	<b>-218.45</b>	
Group: Commercial Building							
Type: Competitive Tender							
1000	Oxford Road Office.	London 01		2,521,712.24	10,803,715.45	8,292,003.21	76.66%
1100	A408 Ealing underpass	Wales 04		857,097.61	2,540,227.08	1,683,129.45	66.29%
1101	Windsor Rail bridge	South East 01		302,662.54	870,800.08	568,137.54	65.24%
	replacement						
1111	New Development at Preconct 14	London 01		15,000.00	0.00	-15,000.00	0.00%
987		London 01		0.00	0.00	0.00	0.00%
W100	Singapore Airport	London 01		0.00	0.00	0.00	0.00%
Z001	Trial for OA reporting xx	London 01		99,937.50	0.00	-99,937.50	0.00%
				<b>3,796,409.89</b>	<b>14,214,742.59</b>	<b>10,418,332.70</b>	
Type: Competitive Tender							
1001	Holiday Inn Slough	North East 02		3,754,761.83	7,643,419.54	3,888,657.71	50.88%
EBB001	LeeEdds Test Contract	North West 02		0.00	0.00	0.00	0.00%
				<b>3,754,761.83</b>	<b>7,643,419.54</b>	<b>3,888,657.71</b>	
Type: Extension							
1104	M34 Northampton - Southampton.	Overseas 00		21,870,051.32	21,408,916.57	-461,134.75	-2.15%
1115	M01 Extension	Overseas 00		6,325.00	46,342.67	40,017.67	86.35%
1200	Bluewater shopping M & E	Overseas 00		5,051.33	1,289,733.19	1,284,671.86	99.61%

Printed Using COINS V10.25.11 (0502) by Nigel Longley at 13:20:57 on 03/09/11 (N.M.L@Events&S)

Page 1

# Discovery BI0007

## 11.13 Using Report Selection Criteria in the Report Query

This exercise modifies the existing report to use further selection criteria within the Report Data using RS\_ fields. Please copy the Function, Page and Report from BI0004 and call them xxBI0007.

## 11.13 Amend the Page Fields

This exercise will replace the standard selection criteria with some User Defined Report Selection fields.

In the Fields Tab go to the FORM selector at the bottom of the page, select Update and click Apply Filter.

**NOTE:** You should always do this to ensure you are adding fields to the correct form - adding fields to an incorrect form will mean that they do not display.

Delete the field which contains the standard jobSelectionGenerate command.

Fields that begin RS\_ are for report selections. They are defined in the FSPs. They typically have three elements: From, To and Matches; these are identified by adding double-underscore plus a number: 1, 2 or 3.

Add in the following fields : (Don't forget the Double Underscore prior to the last number)

Field	Label	Append
RS_job_num_1	Contract From	
RS_job_num_2	To <input checked="" type="checkbox"/>	
RS_job_fore_3	Manager <input checked="" type="checkbox"/>	

Against this last field also include the Populate for Contract Managers (job-rsp.job\_fore) using the View-As and Populate fields as below :

### View As      **Combo Description/Value**

The View As field determines how the field is displayed. In this case, it is a combo that shows the period code and the period end date.

Populate job-rsp.job\_fore

Combos, and other fields that open with data already there, need to be populated from somewhere. This specifies how coins gets the information to show in the combo. In this case, the procedure to populate job\_fore is defined in job-rsp.p. You can add a blank record with a label of ALL.

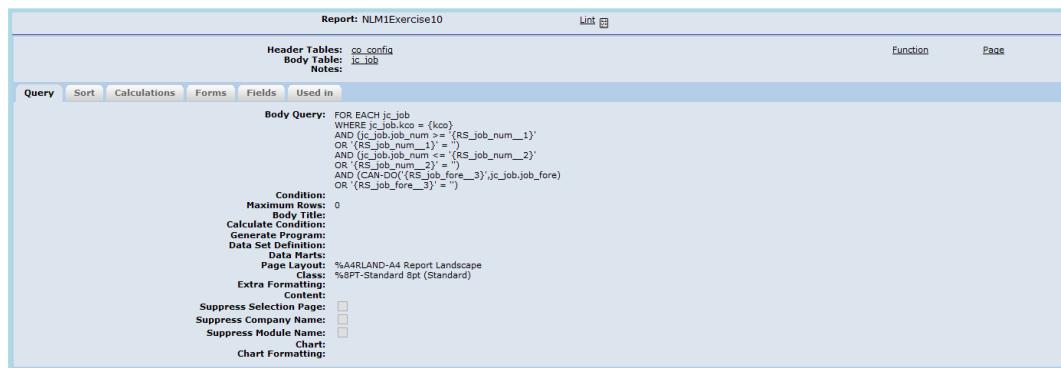
RS_job_fore_3	Manager	0	0	...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Combo Description/Value	▼	
<p>Layout: <input type="text"/></p> <p>Validate: <input type="text"/></p> <p>Mandatory: <input type="checkbox"/></p> <p>Populate: job-rsp.Job_fore Add Blank: <input checked="" type="checkbox"/> Blank Label: ALL</p> <p>Combo Limit: 0</p> <p>Populate Script: <input type="text"/></p> <p>Append: <input checked="" type="checkbox"/></p> <p>Column Spans: 0</p> <p>Row Spans: 0</p> <p>No Break Label: <input type="checkbox"/></p> <p>Label Column Spans: 0</p> <p>Column: 0</p> <p>Hidden: <input type="checkbox"/> Show in Help: <input checked="" type="checkbox"/></p>									

## 11.13 Modify the Report Query

Instead of using the standard selection criteria in the Report Query {jobSelect}, it is now necessary to pass the RS\_ fields used in the Page Section to the report query to ensure that the data returned to the report is within the selection criteria specified by the User.

Amend the report query to reflect the RS\_ fields as follows :

```
FOR EACH jc_job
WHERE jc_job.kco = {kco}
AND (jc_job.job_num >= '{RS_job_num_1}'
OR '{RS_job_num_1}' = "")
AND (jc_job.job_num <= '{RS_job_num_2}'
OR '{RS_job_num_2}' = "")
AND (CAN-DO('{RS_job_fore_3}',jc_job.job_fore)
OR '{RS_job_fore_3}' = "")
```



Re-run the report using the user defined selection criteria.

**NOTE:** If using the combo and populate then you can default the Manager to blank in the Function Parameter by adding RS\_job\_fore\_3= This will remove the <select> from the combo box and will stop the need to only run the report for a specific Manager. - If you add the parameter to the function then you MUST refresh the Menu for this action to take place

## 11.14 Using Report Selection Criteria to Pass Values to RO Fields

This exercise modifies the existing report from Exercise 10 to use further selection criteria within the Report Data using RS\_ fields.

### 11.14 Create the Function

Set up a new function to create an additional tab for Period Selection.

Example:

Function Name	xxBI0007	Main Report Function
Tab Function	xxBI0007T	Holding the Contract Selection
Additional Tab	xxBI0007	T1Holding the Financial Date Selection

		NLM1Exercise10	Module 1 Exercise 10	Function	JC-JC Contract Status	REP-Report
		NLM1Exercise10T	Module 1 Exercise 10 Tab	Function	JC-JC Contract Status	REP-Report
		NLM1Exercise10T1	Module 1 Exercise 10 Tab 1	Function	JC-JC Contract Status	REP-Report

## 11.14 Add the Page Fields

The report needs to prompt for the period to calculate costs up to, as well as allowing contract selection. The Update form already exists so add the fields to allow for Period Selection.

In the Field Tab go to the Form selector at the bottom of the page, select Update and click Apply Filter.

**NOTE:** You should always do this to ensure you are adding fields to the correct form - adding fields to an incorrect form will mean that they do not display.

To create a date selection range for the report fill in the fields as follows:

Field	RS_glp_fdate__2
Label	To Financial Period
View As	Combo Description/Value

The View As field determines how the field is displayed. In this case, it is a combo that shows the period code and the period end date.

Tab	xxBI0007T1
Populate	glp-rsp.glp_fdate

Combos, and other fields that open with data already there, need to be populated from somewhere. This specifies how COINS gets the information to show in the combo. In this case, the procedure to populate glp\_fdate is defined in glp-rsp.p.



## 11.14 Modify the Report Fields

In the Field Tab go to the Form selector at the bottom of the page, select the Body form type and click Apply Filter.

**NOTE :** You should always do this to ensure you are adding fields to the correct form - adding fields to an incorrect form will mean that they do not display.

Modify the following fields as follows:

Field	Label
RO_ContractCosts^TD 0 {RS_glp_fdate__2}	Costs to Date

This will display the costs to date for the date entered when the report is run. The field RO\_ContractCosts takes three parameters:

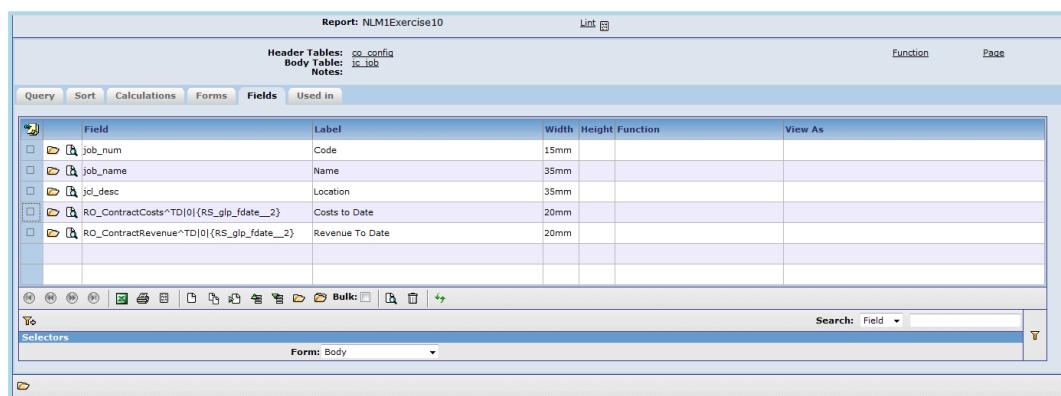
- the type of cost (TD = to date; TP = this period; TY = this year; TO = total)

- the offset from the period selected. For example: 0 = the period selected (no offset); -1 = the period before the one selected; !12-0 = period twelve in the year of the period selected; !1203 = period 12 in 2003.

- the period to report on. This is the value selected on the report selection.

### **RO\_ContractRevenue^TD|0|{RS\_glp\_fdate\_\_2} Revenue to Date**

This works in the same way as RO\_ContractCosts.



The screenshot shows the COINS OA Designer interface with the 'Fields' tab selected. The report title is 'NLM1Exercise10'. The 'Header Tables' are listed as 'ox\_config' and 'ic\_102'. The 'Body Table' is 'ic\_102'. The 'Notes' section is empty. The 'Fields' table lists the following fields:

Field	Label	Width	Height	Function	View As
job_num	Code	15mm			
job_name	Name	35mm			
job_desc	Location	35mm			
RO_ContractCosts^TD 0 {RS_glp_fdate__2}	Costs to Date	20mm			
RO_ContractRevenue^TD 0 {RS_glp_fdate__2}	Revenue To Date	20mm			

Below the table are various toolbar icons and a search bar labeled 'Search: Field'. The 'Selectors' section shows 'Form: Body'.

