
BI Certification - Module One

| | |
|---------------|--------------------|
| Prepared by: | Documentation Team |
| Prepared for: | Learning Resources |
| Module: | BI |
| Date: | 29 June 2016 |
| Document Ref: | LMDSY0050 |
| Version: | 4.0 |

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



Contents

1 Database Overview

| | |
|--------------------------------------|----|
| 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 |

2 OA Query Language

| | |
|--------------------------|----|
| 2.1 FOR EACH | 19 |
| 2.2 WHERE | 20 |
| 2.3 Joining Tables | 22 |
| 2.3.1 EACH | 22 |
| 2.3.2 FIRST | 22 |
| 2.3.3 LAST | 22 |
| 2.3.4 OF | 22 |
| 2.4 Curly Braces | 24 |

3 OA and BI Utilities

| | |
|--|----|
| 3.1 Database Enquiry | 26 |
| 3.2 Query Editor | 28 |
| 3.2.1 Exporting from the Query Editor | 29 |
| 3.2.2 Saving Queries in Query Editor (10.27) | 29 |
| 3.3 Calculation Editor | 31 |

4 Functions and Sections

| | |
|---------------------------|----|
| 4.1 Function Naming | 33 |
|---------------------------|----|

5 Datasets - Overview

| | |
|--|----|
| 5.1 Using Data Sets in the Query Editor | 35 |
| 5.2 Using the Data Set in Queries | 36 |
| 5.2.1 In-Line Reports | 37 |
| 5.3 Granting Access to Data Sets for Other Users | 39 |
| 5.4 Rules for Keys | 40 |
| 5.5 Best Practice | 41 |
| 5.6 Cross Modular Reporting | 42 |
| 6.1 Report Pre-Processing | 43 |
| 6.1.1 Syuds.Calc | 44 |
| 6.1.2 Debug | 46 |
| 6.1.3 Delete | 47 |
| 6.1.4 Filter | 48 |
| 6.1.5 GroupQuery | 49 |

| | |
|--------------------------------|----|
| 6.1.6 Merge | 51 |
| 6.1.7 Syuds.Pivot | 53 |
| 6.1.8 Repeat | 59 |
| 6.1.9 Store | 60 |
| 6.1.10 Sum | 61 |
| 6.1.11 TableAlias | 62 |
| 6.1.12 Syuds.TimeSlice() | 63 |
| 6.1.13 Top | 65 |
| 6.1.14 Union | 66 |

7 Data Set Exercises

| | |
|--|-----------|
| 7.1 Exercise 1 - Creating the Dataset | 69 |
| 7.1.1 Adding the Fields | 70 |
| 7.1.2 Testing the dataset | 71 |
| 7.1.3 Create the Function and Menu | 72 |
| 7.1.4 Set up the Page | 73 |
| 7.1.5 Add the Page Section Forms | 73 |
| 7.1.6 Add the Fields | 73 |
| 7.1.7 Create the report section | 74 |
| 7.1.8 Sorting the Report | 75 |
| 7.1.9 Create the Report Form | 75 |
| 7.1.10 Add the Fields | 75 |
| 7.1.11 Running the Report | 76 |
| 7.2 Exercise 1a - Using multiple Data Sets | 78 |
| 7.2.1 Create the Function and Menu | 78 |
| 7.2.2 Creating the new Data Set | 78 |
| 7.2.3 Adding the Fields | 79 |
| 7.2.4 Create the Report | 79 |
| 7.2.5 Running the report | 81 |
| 7.3 Exercise 1b - Using multiple Data Sets to merge data | 82 |
| 7.3.1 Create the Function and Menu | 82 |
| 7.3.2 Adding the Fields | 82 |
| 7.3.3 Adding the Fields | 83 |
| 7.3.4 Running the report | 86 |
| 7.4 Exercise 1c – Using Datasets for Exporting Data | 87 |
| 7.4.1 Create the Function and Menu | 87 |
| 7.4.2 Create the Report Section | 87 |
| 7.4.3 Running the report | 88 |
| 7.5 Exercise 1d – Dynamically switching between multiple Data Sets (v10.26 onwards) | 90 |
| 7.5.1 Create the Function and Menu | 90 |
| 7.5.2 Build the datasets | 90 |
| 7.5.3 Create the page | 90 |
| 7.5.4 Create the Report | 91 |
| 7.5.5 Running the report | 92 |
| 7.6 Exercise 2 - Generating Pages from a Data Set | 93 |
| 7.6.1 Create a Browse Page from a Data Set | 93 |
| 7.6.2 Add New Functions for the Browse Page | 94 |
| 7.6.3 Create the Page | 95 |
| 7.6.4 Running the Function | 96 |

8 Page Design Exercises

| | |
|--|-----------|
| 8.1 Creating a Contract Browse Screen | 99 |
| 8.1.1 Create the Function | 99 |
| 8.1.2 Create the Page | 99 |
| 8.1.3 Create a Body Page Form | 100 |

| | |
|---|-------------|
| 8.1.4 Add the Fields | 100 |
| 8.1.5 Adding a Sort to a Column | .102 |
| 8.1.6 Linking a Field to another Enquiry | .102 |
| 8.1.7 Running The Function | .103 |
| 8.2 Enhancing the Contract Browse Screen | .104 |
| 8.2.1 Modifying the Page Section | .104 |
| 8.2.2 Adding Totals to the Page | .104 |
| 8.2.3 Adding an Excel Output Function to the Page | .106 |
| 8.3 Adding Sections to the Contract Browse Screen | .108 |
| 8.3.1 Adding a Body Span Section | .108 |
| 8.3.2 Adding a Body Detail Section | .109 |
| 8.3.3 Adding a Context Section | .111 |
| 8.4 Adding Filters to the Contract Browse Screen | .113 |
| 8.4.1 Adding a Page Browse Filter | .113 |
| 8.4.2 Using Populates with a Browse Filter | .113 |
| 8.5 Adding a Body Selector | .115 |
| 8.5.1 Adding a Body Selector Section | .115 |
| 8.5.2 Using the Body Selector value | .115 |
| 9 A Simple listing Report | |
| 9.1 Create the Function and Menu | .118 |
| 9.2 Set up the Page | .119 |
| 9.3 Add the Page Section Forms | .119 |
| 9.4 6.4 Add the Fields | .120 |
| 9.5 Create the report section | .121 |
| 9.6 Sorting the Report | .122 |
| 9.7 Create the Report Form | .122 |
| 9.8 Add the Fields | .123 |
| 9.9 Running the Report | .124 |
| 10 A Contract Report | |
| 10.1 Create the Function | .126 |
| 10.2 Create the Page | .127 |
| 10.3 Add the Page Section Forms | .127 |
| 10.4 Add the Page Fields | .127 |
| 10.5 Create the report section | .128 |
| 10.6 Sorting the Report | .129 |
| 10.7 Create the Report Forms | .130 |
| 10.8 Add the Fields | .131 |
| 10.9 Add Totals to the Report | .132 |
| 10.10 Run the report | .132 |
| 10.11 Summarising the Contract Report | .134 |
| 10.11.1 Copy the existing Example 7 Report | .134 |
| 10.11.2 Select the Sort Level | .134 |
| 10.11.3 Select the correct Header and Footer Forms | .134 |
| 10.11.4 Amend the fields on the report | .135 |
| 10.12 Calculated Fields | .136 |
| 10.12.1 Copy the existing Example 7 Report | .136 |
| 10.12.2 Add Calculated Fields | .136 |
| 10.13 Using Report Selection Criteria in the Report Query (1) | .139 |
| 10.13.1 Amend the Page Fields | .139 |
| 10.13.2 Modify the Report Query | .140 |
| 10.14 Using Report Selection Criteria in the Report Data (2) | .141 |
| 10.14.1 Create the Function | .141 |
| 10.14.2 Add the Page Fields | .141 |

| | |
|--|-----|
| 10.14.3 Modify the Report Fields | 142 |
|--|-----|

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

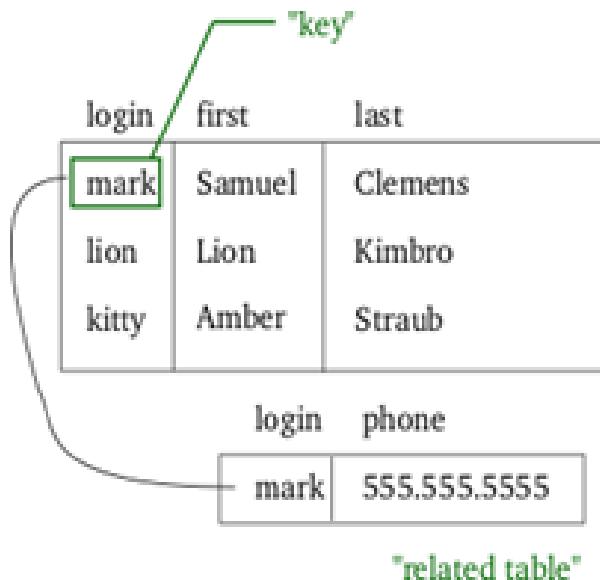
The model in most common use today is the relational model, which represents all information in the form of multiple related tables each consisting of rows and columns.

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 :

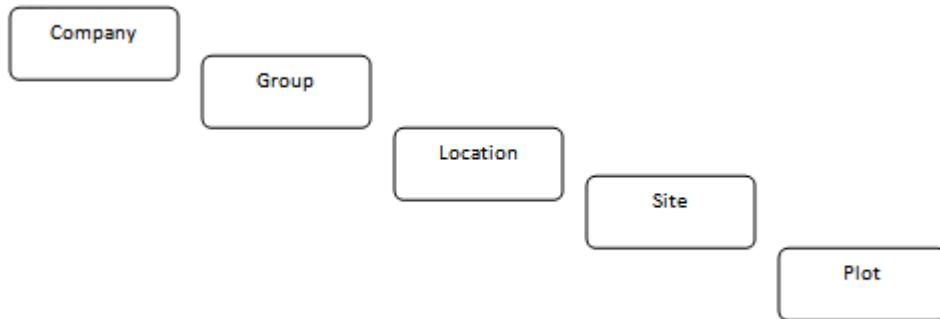
BP Professional Billing
CR Credit Control
DC/DM Document Control / Management
IB Insurance and Bonds

1.3 Summary and Detail Tables

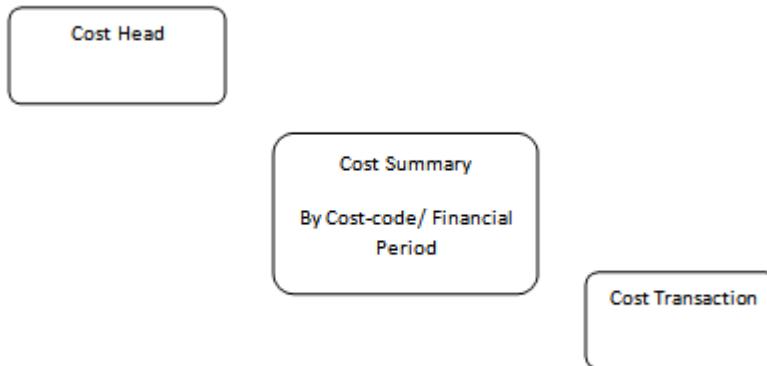
Coins has designed its database to match the processes of the industry and therefore the tables mirror the process components. Each of the modules will cover different process and within each module we can find different sub-processes. The organizational structure of the company is also related to the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

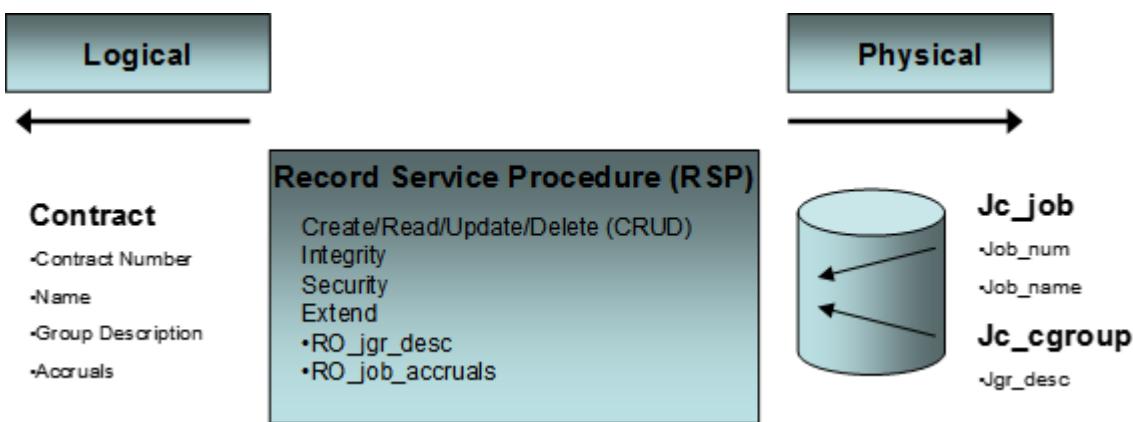
Every table in the coins database has a RSP (Record Service Procedure). Each of these RSP's provides the OA Reporting tool with the business logic required to extract the appropriate data from the database.

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.



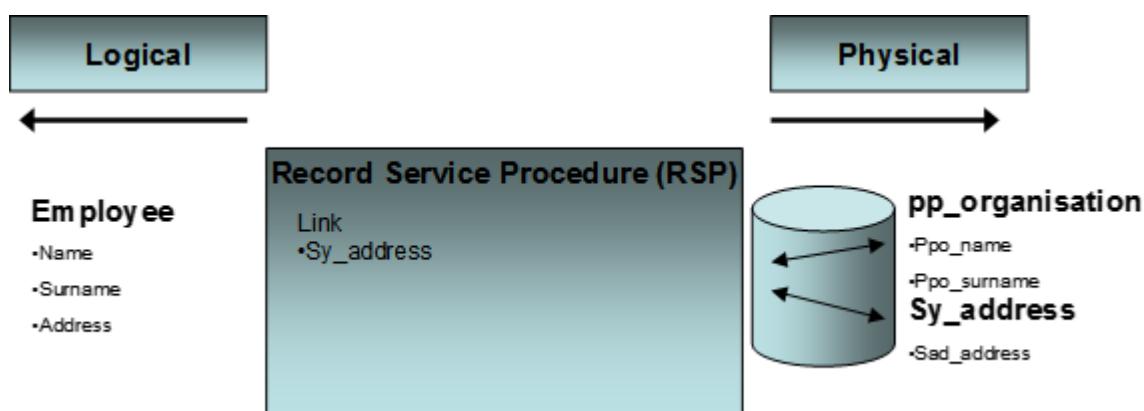
The RSP can also extend the database table to produce a logical view of data for the business logic to work with. A by-product of this is that we are to provide an XML field on every record in the database in which the user can configure their own extended fields for

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 into 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_subl_details | Supplier Tax Details | character x(60) | RO If the supplier provides subcontract labour (avm_subl=TRUE) then this is a composite description of the G/S details for the subcontractor. |
| RO_avm_Title_agt | Aging: Title Age | character x(10) | RO Text "Age" |
| RO_avm_Title_Amt | Aging: Title Amount | character x(10) | RO Text "Amount" |
| RO_avm_Title_Key | Aging: Title Key | character x(10) | RO Text "Key" |
| RO_avm_Title_QtryAmt | Aging: Title Query Amount | character x(12) | RO Text "Query Amount" |
| RO_avm_Total | Oustanding Total | decimal ->xxxx,>>>9.99 RO | Total outstanding invoice/payments for the supplier. Sum of the open invoices and payments in account currency. |
| RO_avm_unclarred | Uncleared To Pay | decimal ->xxxx,>>>9.99 RO | The total of invoices that are due to be paid and are held. |
| RO_avm_wfvar | Active Workflow Variable | character x(30) | RO |
| RO_BalD2[2] | Discount Lost | decimal ->xxxx,>>>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_BalD2[2] | Discount Taken | decimal ->xxxx,>>>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_BalPw[2] | Payments | decimal ->xxxx,>>>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_BalPw[2] | Purchases | decimal ->xxxx,>>>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_co_store | <class><parent> <StoreID> | decimal ->xxxx,>>>9.99 RO | Standard store options. See co_config.RW_co_store |
| RO_contractaccounts_auth | <ContractAccounts> </Parent><Offset> </Date> </ExcludeCostHgt> </ExcludeCostLgt> | decimal ->xxxx,>>>9.99 RO | The contract accounts for the supplier. The contract for which accounts are required =C4DO list of cost heads to exclude |
| RO_contractaccounts_auth | Contract Accounts by Contract | decimal ->xxxx,>>>9.99 RO | The contract accounts for the supplier. The contract for which accounts are required =C4DO list of cost heads to exclude |
| RO_contractcosts_auth | <ContractNumber> <ExclCostHgt> <ExclCostLgt> <PeriodOffset> <PeriodType> <Offset> </ExcludeCostHgt> </ExcludeCostLgt> | decimal ->xxxx,>>>9.99 RO | The contract costs for the supplier. The contract for which accounts are required =C4DO list of cost heads to exclude Period Type, Offset, Date as for co_config.RW_co_store |
| RO_currency | Reporting Currency | character x(4) | RO If RS_base 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_gln_1 | Turnover Q1 | decimal ->xxxx,>>>9.99 RO | Turnover at Q1 for the year held in variable RS_adv_year |
| RO_gln_2 | Turnover Q2 | decimal ->xxxx,>>>9.99 RO | Turnover in Q2 for the year held in variable RS_adv_year |
| RO_gln_3 | Turnover Q3 | decimal ->xxxx,>>>9.99 RO | Turnover in Q3 for the year held in variable RS_adv_year |
| RO_gln_4 | Turnover Q4 | decimal ->xxxx,>>>9.99 RO | Turnover in Q4 for the year held in variable RS_adv_year |
| RO_rcp_changed | Record Changed | character x(4) | RO The date and time of the most recent change to the record, and the user ID of the user who changed it. |
| RO_rcp_created | Record Created | character x(17) | RO The date and time that the record was created. |
| RO_sbmc_code | Subcontractor Account | character x(10) | RO If avm_sbmc = TRUE and the subcontractor [sc_main] record exists for this supplier then this shows the subcontractor account code otherwise blank. |
| RO_supplier | Group Supplier | character x(8) | RO The company the plant item is based from. |
| RO_wfwm_branch | Bank Branch (Workflow) | character x(30) | RO If variable RS_cons = YES then if the group account (avm_gracc) is non blank then the group account otherwise the supplier account code. |
| RO_wfstat_name | International Bank Name (Workflow) | character x(30) | RO |
| RW_avm_mn_desc | Notes | character | RW Any notes pertaining to the supplier. |
| RW_CvFilters | | | RW |
| <Contract><parent> | | | RW |

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>[| <FDates>[| 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.CRM_NUM=JC_JOB.RDM_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

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 | ID: | Type: | Field: | DB | ID |
|--------------|----------------------------------|-----|-------|--------|-------|-----|
| abi_dta | ABI Data dta file | | | | coins | abi |
| abi_dtc | ABI Data dtc file | | | | coins | abc |
| abi_dtd | ABI Data dtd file | | | | coins | abd |
| abi_fields | ABI field mapping | | | | coins | afd |
| 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_schtype | Scheme Types | | | | coins | ach |
| al_asset | Asset Repository | | | | coins | ais |
| 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_invdist | Invoice Distribution File | | | | coins | aid |
| ap_invhist | Invoice History | | | | coins | ahf |
| ap_invline | P/I, Invoice Lines | | | | coins | ail |
| ap_invoice | P/I, Invoice | | | | coins | ain |
| ap_invoicen | P/I, Open Invoice | | | | coins | aop |
| ap_inquiry | Invoice queries | | | | coins | aiq |
| ap_invdist | P/I, Intrastat Distribution Line | | | | coins | api |
| ap_paz | P/I, Payment Allocation | | | | coins | apy |
| ap_rcard | PCards | | | | coins | app |
| ap_scardline | PCard Transaction Lines | | | | coins | apd |
| ap_scardtran | PCard Transactions | | | | coins | apt |

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

| Table | Filter: Table: | Description: |
|-------|----------------|--------------|
|-------|----------------|--------------|

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: ap_invoice | | | | Number: 18 |
|-------------------|-------------------------------|-----------|---|---|
| Index | Primary | Unique | Field | Documentation |
| ain_key | | | kco+ avm_num+ ain_inv+ | kco+avm_num+ain_inv+ |
| ain_key1 | | | kco+ cob_num+ ain_inv+ | kco+cob_num+ain_inv+ |
| ain_key10 | | | kco+ col_line+ | |
| ain_key2 | | | kco+ ain_inv+ | kco+ain_inv+ |
| ain_key3 | | | kco+ avm_num+ ain_supref+ | kco+avm_num+ain_supref+ |
| ain_key4 | | | kco+ ain_supref+ ain_entry+ | |
| ain_key5 | | | kco+ avm_num+ ain_entry+ | |
| ain_key6 | | | kco+ avm_num+ ain_entry+ | |
| ain_key7 | | | kco+ ain_entry+ | |
| ain_key8 | | | kco+ avm_num+ ain_date+ | |
| ain_key9 | | | kco+ ain_date+ avm_num+ ain_inv+ | |
| Field | Label | Data Type | Format | Documentation |
| ain.altcur | Supplier Alternative Currency | logical | yes/no | RW Whether alternative currencies i.e. different to the account currency (sp_vendor.cur_code) are allowed to be entered on the account. |
| ain.amount | Gross Amount | decimal | >>,>>,>>9.99 | D8 The gross amount of the invoice. |
| ain.anal | Analysis (Contract or Dept) | character | x 8 | D8 The main contract or department to which the invoice is assigned. Depends on ain_entry. |
| ain.apacct | P/L Control | character | X 19 | D8 The creditor's control account posted to when the invoice was committed. |
| ain.atoftaxpay | ATO Reporting | character | x 4 | D8 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: <ul style="list-style-type: none">- Include in the ATO Taxable Payments report- Exclude from the ATO Taxable Payments report- Blank Default to the Supplier setting or if not set default to the Company workbench setting for this company (check PL/USECIS parameter). |

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.an-invp, invoice_an_inv AND ap_cinvline.anh-invp, invoice_anh_inv |
| 1 | * | ap_invlist | ap_invlist OF ap_invoice | ap_invlist,kco-ap, invoice,kco AND ap_invlist.an-invp, invoice_an_inv AND ap_invlist.anh-invp, invoice_anh_inv |
| 1 | * | ap_invline | ap_invline OF ap_invoice | ap_invline,kco-ap, invoice,kco AND ap_invline.an-invp, invoice_an_inv AND ap_invline.anh-invp, invoice_anh_inv |
| 1 | 1 | ap_invoopen | ap_invoopen OF ap_invoice | ap_invoopen,kco-ap, invoiceavm_num AND ap_invoopen.anh-invp, invoice_anh_inv AND ap_invoopen.avm-num-ap, invoiceavm_num AND ap_invoopen.an-invp, invoice_an_inv |
| 1 | 1 | ap_inquiry | ap_inquiry OF ap_invoice | ap_inquiry,kco-ap, invoice,kco AND ap_inquiry.an-invp, invoice_an_inv AND ap_inquiry.anh-invp, invoice_anh_inv |
| 1 | * | ap_invlist | ap_invlist OF ap_invoice | ap_invlist,kco-ap, invoice,kco AND ap_invlist.an-invp, invoice_an_inv AND ap_invlist.anh-invp, invoice_anh_inv |
| 1 | * | ap_validlist | ap_validlist OF ap_invoice | ap_validlist,kco-ap, invoice,kco AND ap_validlist.an-invp, invoice_an_inv AND ap_validlist.anh-invp, invoice_anh_inv |
| * | 1 | ap_vendor | ap_vendor OF ap_invoice | ap_vendor,kco-ap, invoice,kco AND ap_vendor.an-invp, invoice_an_inv AND ap_vendor.anh-invp, invoice_anh_inv |
| * | 1 | ap_vendsum | ap_vendsum OF ap_invoice | ap_vendsum,kco-ap, invoice,kco AND ap_vendsum.an-invp, invoice_an_inv AND ap_vendsum.anh-invp, invoice_anh_inv |
| * | 1 | ar_invoice | ar_invoice OF ap_invoice | ar_invoice,kco-ap, invoice,kco AND ar_invoice.cob-line-ap, invoice_cob_line OR ar_invoice.cob-lineh-ap, invoice_cob_line |
| * | 1 | cb_tdet | cb_tdet OF ap_invoice | cb_tdet,kco-ap, invoice,kco AND cb_tdet.line-ap, invoice_cob_line |
| * | 1 | cb_topen | cb_topen OF ap_invoice | cb_topen,kco-ap, invoice,kco AND cb_topen.cob-num-ap, invoice_cob_num AND cb_topen.cob-lineh-ap, invoice_cob_line |
| * | 1 | co_currency | co_currency OF ap_invoice | co_currency,kco-ap, invoice,kco AND co_currency.cur_code-ap, invoice_cur_code |
| * | 1 | co_vat | co_vat OF ap_invoice | co_vat,kco-ap, invoice,kco AND co_vat.vat_code-ap, invoice_vat_code |
| * | 1 | cs_certificate | cs_certificate OF ap_invoice | cs_certificate,kco-ap, invoice,kco AND cs_certificate.col_num-ap, invoice_col_num AND cs_certificate.cob-line-ap, invoice_cob_line |
| * | 1 | fa_invoice | fa_invoice OF ap_invoice | fa_invoice,kco-ap, invoice,kco AND fa_invoice.lineh-ap, invoice_cob_line |
| * | 1 | hs_devsum | hs_devsum OF ap_invoice | hs_devsum,kco-ap, invoice,kco AND hs_devsum.job-num-ap, invoice_job_num AND hs_devsum.job_phase-ap, invoice_job_phase |
| * | 1 | jc_job | jc_job OF ap_invoice | JC_JOB,kco-ap, invoice,kco AND JC_JOB.job-num-ap, invoice_job_num |
| * | 1 | jc_jobsum | jc_jobsum OF ap_invoice | JC_JOBSUM,kco-ap, invoice,kco AND JC_JOB_SUM.job_num-ap, invoice_job_num |
| * | 1 | jc_phase | jc_phase OF ap_invoice | JC_PHASE,kco-ap, invoice,kco AND JC_PHASE.job_num-ap, invoice_job_phase |
| * | 1 | jc_phssum | jc_phssum OF ap_invoice | JC_PHSSUM,kco-ap, invoice,kco AND JC_PHSSUM.job_num-ap, invoice_job_num |

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.

Stored: Load Save Delete

Search Replace all

Query:

Fields:

Data Set:
Condition:
Maximum Rows: 10

Extra Parameters:

Run Export Create Dataset

To use the query editor simply enter the query, and (optionally) any fields required - space separated - and select 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.

Stored: Load Save Delete

Search Replace all

Query:

Fields:

Data Set:
Condition:
Maximum Rows: 10

Extra Parameters:

Run Export Create Dataset

| kco | avm_num | avm_name | avm_addr_1 | avm_addr_2 | avm_addr_3 | avm_addr_4 | avm_pcode | avm_phone | avm_contact | avm_disctype | avm_discday | avm_disc% | avm_dueltype | avm_duedays | avm_hold | avm_fax | avm_faxnum | avm_sclab | avm_shname | avm_schname | avm |
|-----|---------|----------------------------------|----------------------------|------------------------|----------------|------------|-----------|--|-------------|--------------|-------------|-----------|--------------|-------------|----------|--|------------|-----------|-------------------------|----------------------------------|-----|
| 10 | AB0001 | Abbey Glass | 42 Bramall Lane | | Sheffield | | S25 4DL |  01642 887766 | | 1 | 30 | 0.00 | 1 | 30 | N |  01642 887767 | | N | Abbey | Abbey Glass | 0.0 |
| 10 | ABSO004 | Absolute Invoice Finance Limited | ST JAMES HOUSE | 7 CHARLOTTE STREET | MANCHESTER | | M1 4DZ | | | 1 | 30 | 0.00 | 1 | 30 | N | | | N | ABSO | Absolute Invoice Finance Limited | 0.0 |
| 10 | AGG001 | Aggregate Supplies | 22 Ashton Gates | | Bristol | | BS30 5SJ |  01675 556644 | | 1 | 30 | 0.00 | 1 | 30 | N |  01675 556643 | | N | Aggregate | Aggregate Supplies | 0.0 |
| 10 | AGR005 | Aggregate Industries UK Ltd | BARDON HILL | COALVILLE | LEICESTERSHIRE | | LE67 1TL |  01530 611956 | | 1 | 30 | 0.00 | 1 | 30 | N |  01530 615180 | | N | Aggregate | Aggregate Industries UK Ltd | 0.0 |
| 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 | 0.0 |
| 10 | B&Q001 | B&Q plc | Coypool Rd, | Plymouth | Plymouth | Devon | PL7 4SS | | | 1 | 30 | 0.00 | 1 | 30 | N | | | N | B&Q Ltd | B&Q plc | 0.0 |
| 10 | BER002 | Berts Bricks and Supplies | The High Street | | Surrey | | GU1 5BH | | | 1 | 30 | 0.00 | 1 | 30 | N | | | N | Berts Bricks & Supplies | Berts Bricks & Supplies | 0.0 |
| 10 | BER003 | Berties Brickets Limited | Slough Trading Estate | 107 - 113 Farnham Road | Slough | Berkshire | SL1 4UN | | | 1 | 15 | 10.00 | 1 | 30 | N | | | N | Berties Brickets | Berties Brickets Limited | 0.0 |

| 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 |
| 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 an 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 valid 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 & 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' &jccSelect= and jc_costcode.jcc_cc = 'YYYY' (where XXXX is a valid contract number and YYYY is a valid Costcode).</p> |

3.2.1 Exporting from the Query Editor

To export the query and associated results to excel simply click on the export button. 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.

3.2.2 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.



Stored:

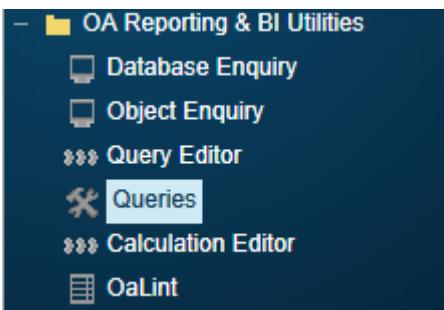
Once you have set up your query and selected the required fields, in the Stored fields, specify a name for the query and a description to further identify it.

| | | | | | | |
|---------|-------------|---|-----------------------|------|------|--------|
| Stored: | ap_vendor 1 | x | Basic ap_vendor Query | Load | Save | Delete |
|---------|-------------|---|-----------------------|------|------|--------|

Select the Save button.

The Query will now be stored and can be retrieved at any time by running Query Editor, specifying the Query name and selecting the 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

◀ System | Queries

niglon COINS

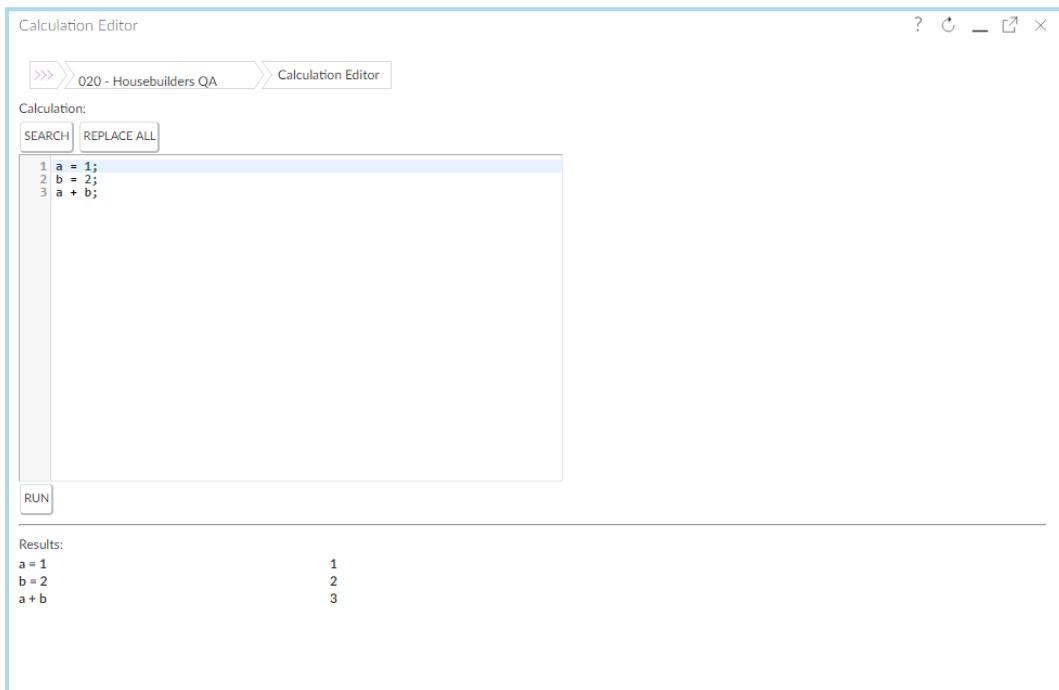
| | Query ▾ | Description |
|---|-------------|----------------------------|
| □ | ap_vendor1 | Basic ap_vendor Query |
| □ | njl_ic_job1 | Simple jl_ic_job selection |
| □ | njl_test_2 | Sample Save Query |

Search:

This screen will allow the creation, deletion and maintenance of the stored queries. Note that to see the results of the queries, you will need to return to Query Editor and Load the appropriate query.

3.3 Calculation Editor

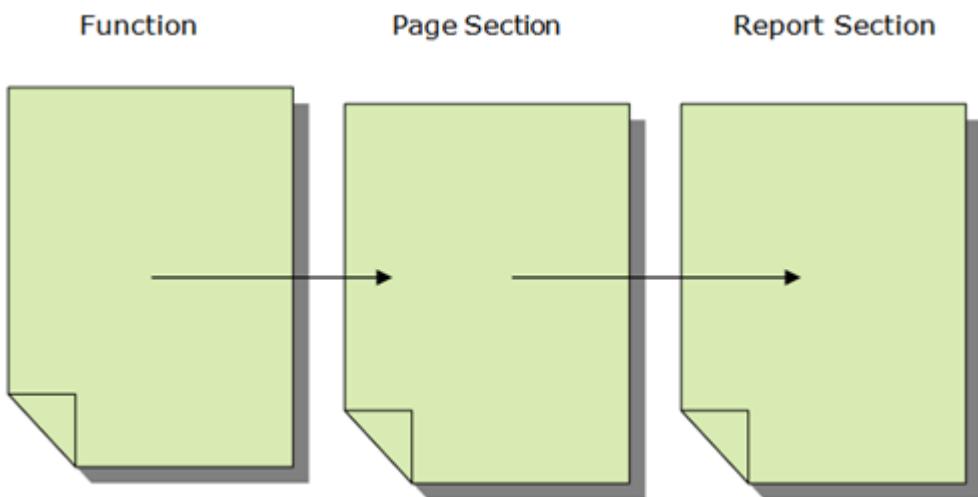
The Calculation Editor allows you to try out calculations using the coins business logic calculation methods (see accompanying documentation). This function can be useful to test calculations before being used in reports and enquiries.



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 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.

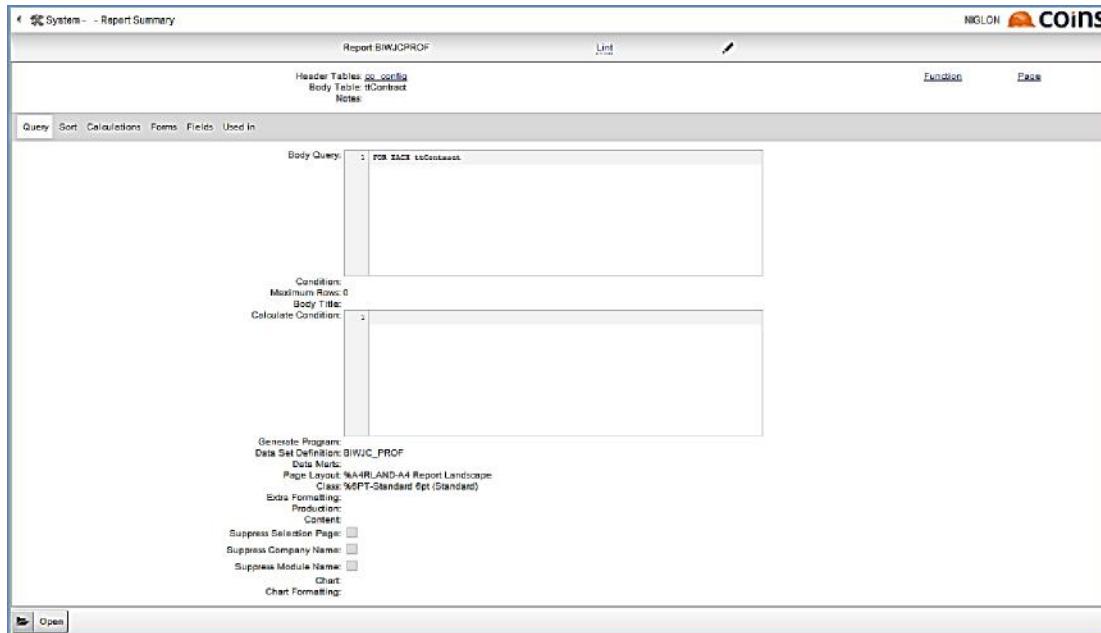
If the Dataset query takes a parameter replacement field (i.e. {RS_year}) then you can set the parameters by adding them to the URL on the Query Editor Page. (i.e. &RS_year=2009)

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.

5.2 Using the Data Set in Queries

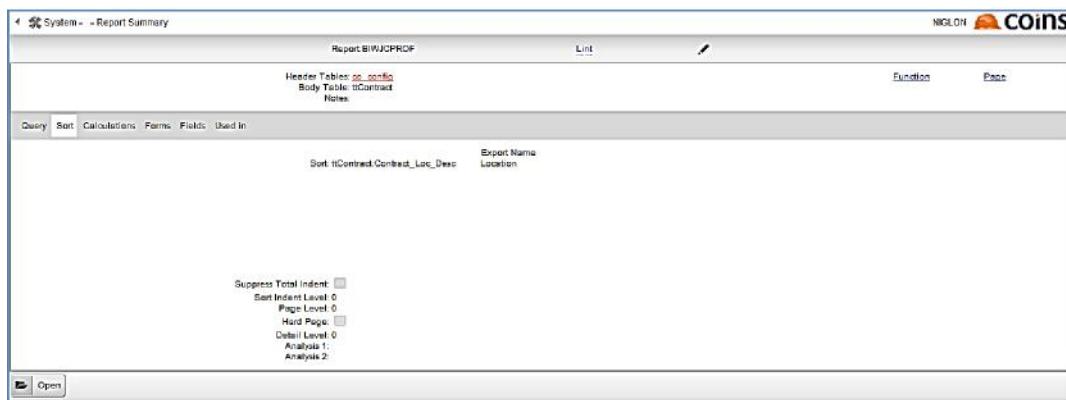
The Data Set can now be used in Queries in Report Designer. Using our example dataset, the body table should be ttContract, the body query should be - FOR EACH ttContract and the Data Set Definition would be BIWJC_PROF.



The screenshot shows the Report Designer interface for a report titled "Report BIWJC_PROF". In the "Body Query" field, the expression "FOR EACH ttContract" is entered. The "Data Set Definition" dropdown is set to "BIWJC_PROF". Other settings visible include "Page Layout: %A4R1LAND-A4 Report Landscape" and "Body Table: ttContract". The "Sort" tab is selected in the toolbar.

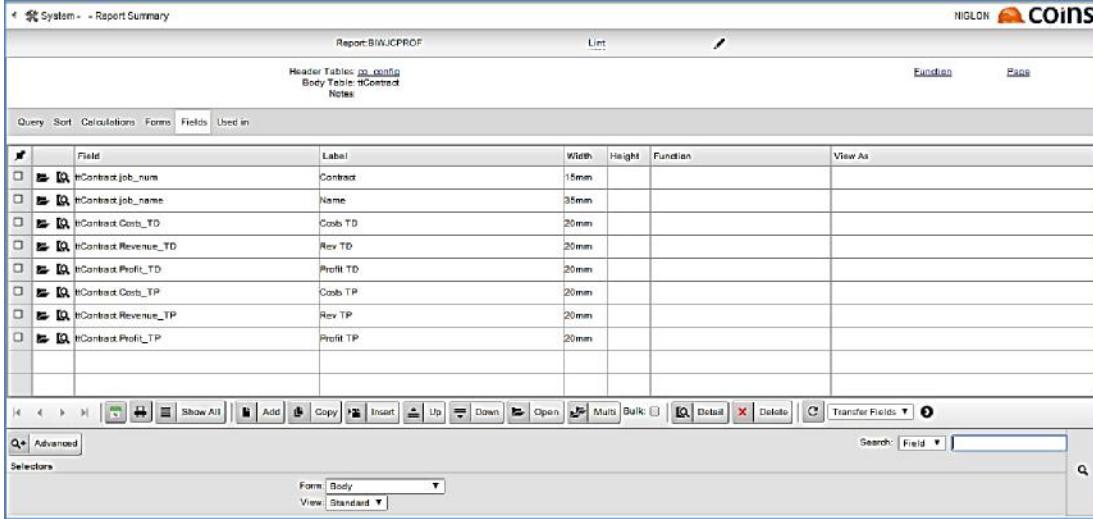
Note that you can use more than one dataset in a query. To do this enter each dataset name in the Data Set Definition field separated by a comma. You will need to reference each ttTablename as appropriate in the body query.

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)



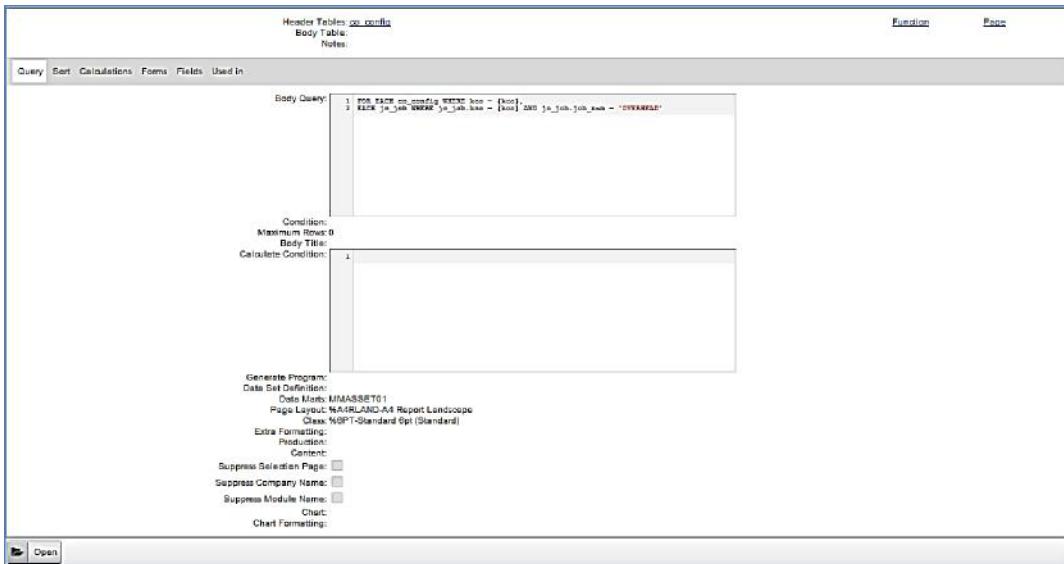
The screenshot shows the Report Designer interface for a report titled "Report BIWJC_PROF". In the "Sort" field, the expression "ttContract.Contract_Loc_Desc" is entered, and the "Export Name" is set to "Location". The "Data Set Definition" dropdown is set to "BIWJC_PROF". The "Sort" tab is selected in the toolbar.

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.

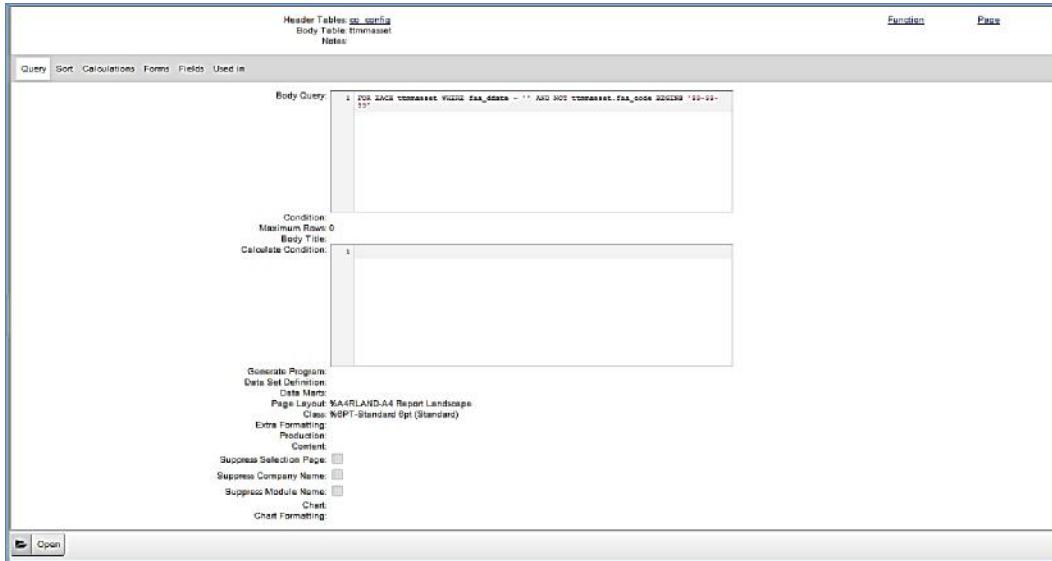


5.2.1 In-Line Reports

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



Then for each inline report, only use the ttTablelename 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.

5.3 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.

5.4 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.

5.5 Best Practice

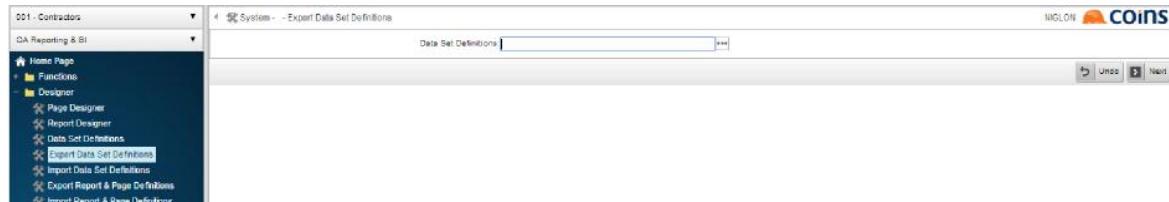
- » Only create fields in a DataSet that are going to be used in the equivalent report.
- » It is better to 'Source' a field rather than to 'Calculate' a field for efficiency.
- » It is recommended that any calculations that can be carried out on the report/page should be done at that stage.
- » For 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

5.6 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 DataType (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

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

Calc is new and again on the back of report builder 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 name, data type, extent, label, format, and calc string

Calc exec 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 (which is how I use it for calc fields in report builder)

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

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.calcrec');
method('syuds.sum','Contract','','Location','jcl_loc','jcl_desc','');
;
```

Dataset Fields:

| Definition | Fields | | | | | | |
|------------|----------------|-----|----------------------|-----------|------------|--------|----------------------------|
| | Field | Key | Extent | Label | Data Type | Format | Source |
| | jc_job_num | | Contract | Character | x(6) | | jc_job.job_num |
| | jc_job_name | | Name | Character | x(30) | | jc_job.job_name |
| | jccl_loc | | Location | Character | x(8) | | jc_job.jcl_loc |
| | jcl_desc | | Location Description | Character | x(30) | | jc_job.jcl_desc |
| | jgr_group | | Group | Character | x(8) | | jc_job.jgr_group |
| | jgr_desc | | Group Description | Character | x(30) | | jc_job.jgr_desc |
| | jc_job_costsTD | | Costs TD | Decimal | >>>,>>9.99 | | jc_job.RO_contractcosts^TD |
| | jc_job_costsTY | | Costs TY | Decimal | >>>,>>9.99 | | jc_job.RO_contractcosts^TY |
| | jc_job_costsTP | | Costs TP | Decimal | >>>,>>9.99 | | jc_job.RO_contractcosts^TP |

Output: ttContract

◀ 20 - Housebuilders QA - System - Query Editor

Stored:

| | | | | |
|--------|-------------|------|------|--------|
| Search | Replace all | Load | Save | Delete |
|--------|-------------|------|------|--------|

```
1 FOR EACH ttContract
```

Query:

Fields:

Data Set:
Condition:
Timeout(Seconds):
Maximum Rows:
Extra Parameters:

| |
|--------------------|
| Timjob |
| 50 |
| 100 |
| &RS_job_num_3=1000 |

Run Export Create Dataset

| job_num | job_name | jcl_loc | jcl_desc | jgr_group | jgr_desc | job_costsID | job_costsTY | job_costsTP |
|---------|-----------|---------|--------------|-----------|-------------|-------------|-------------|-------------|
| 1000 | Milfields | WARKS | Warwickshire | 00 | Head Office | 71124.4 | 4273 | 200 |

Output: ttCalc

◀ 20 - Housebuilders QA - System - Query Editor

Stored:

| | | | | |
|--------|-------------|------|------|--------|
| Search | Replace all | Load | Save | Delete |
|--------|-------------|------|------|--------|

```
1 FOR EACH ttCalc
```

Query:

Fields:

Data Set:
Condition:
Timeout(Seconds):
Maximum Rows:
Extra Parameters:

| |
|--------------------|
| Timjob |
| 10 |
| 100 |
| &RS_job_num_3=1000 |

Run Export Create Dataset

| ttContractRowid | lastYTD | lastPTD |
|--------------------|---------|---------|
| 0x0000000000123100 | 75397.4 | 71324.4 |

6.1.2 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.

```
Method('syuds.debug','CostRev','','',0);
```

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).

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

6.1.3 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 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 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 YTD','');
method('syuds.calcrec');
method('syuds.sum','Contract','','Location','jcl_loc','jcl_desc','');
method('syuds.group','Contract','','Group1','jcl_loc','job_costsSTD');
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 |
| | jc_job_num | | Contract | Character | x(8) | | jc_job.job_num |
| | jc_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 |
| | jc_group | | Group | Character | x(8) | | jc_job.jc_group |
| | jc_desc | | Group Description | Character | x(30) | | jc_job.jc_desc |
| | jc_costsTD | | Costs TD | Decimal | >>>,>>9.99 | | jc_job.RO_contractcosts*TD |
| | jc_costsTY | | Costs TY | Decimal | >>>,>>9.99 | | jc_job.RO_contractcosts*TY |
| | jc_costsTP | | Costs TP | Decimal | >>>,>>9.99 | | jc_job.RO_contractcosts*TP |

Buttons at the bottom: Show All, Add, Copy, Insert, Up, Down, Open, Concurrent, Multi, Bulk, Detail, Delete, C.

Q Advanced

Output:

20 - Housebuilders QA - System - Query Editor

Stored: Load Save Delete

Search Replace all

```
1 FOR EACH ttGroup1
```

Query:

Fields:

Data Set: Timjob

Condition: 10

Timeout(Seconds): 100

Maximum Rows: &RS_job_num_3=1000

Extra Parameters:

Run Export Create Dataset

| level | kcj_loc | count | job_costsTD | job_costsTD_max | job_costsTD_min | job_costsTD_avg |
|-------|---------|-------|-------------|-----------------|-----------------|-----------------|
| 0 | 1 | 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: Load Save Delete

Search Replace all

```
1 FOR EACH ttGroup2
```

Query:

Fields:

Data Set: Timjob

Condition: 10

Timeout(Seconds): 100

Maximum Rows: &RS_job_num_3=1000

Extra Parameters:

Run Export Create Dataset

| level | kcj_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 | 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 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.

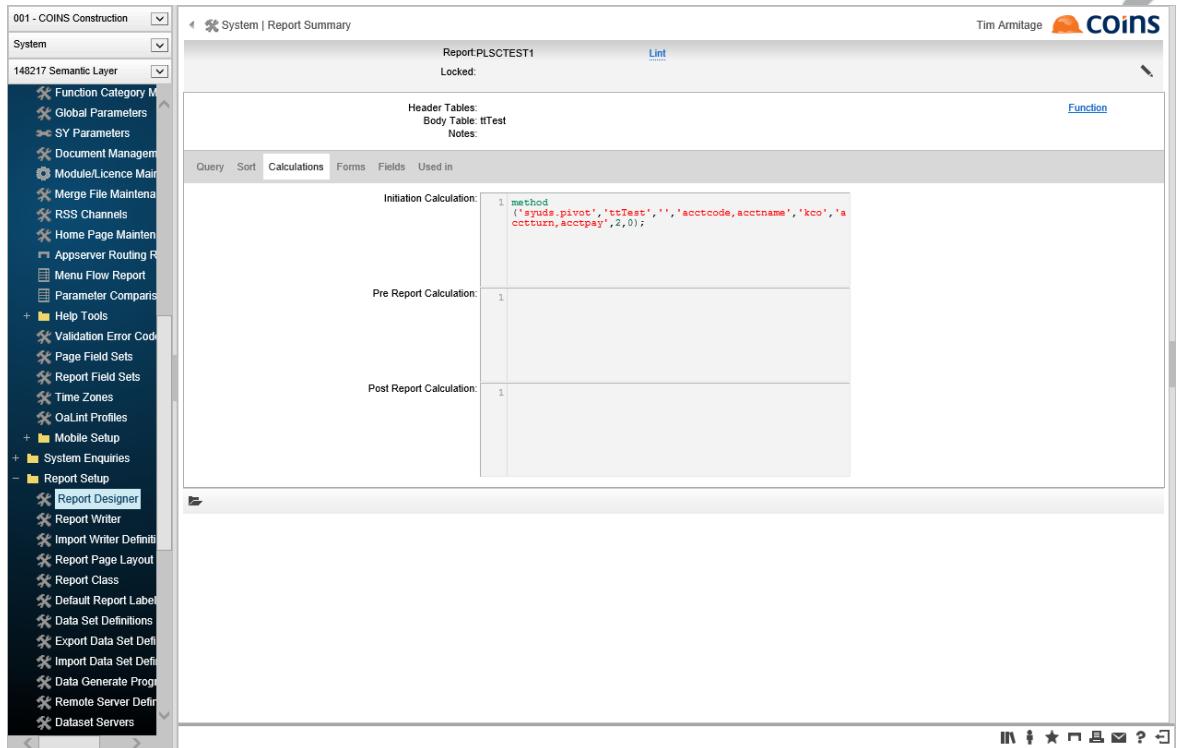
Syuds.Pivot take the following form:

```
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 | Accturn | 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 |

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

Table: ttTestRow

| Acct-num | Acct-name | iCoun-t_1 | iCoun-t_2 | iCoun-t_3 | iCoun-t_4 | Acct-turn_1 | Acct-turn_2 | Acct-turn_3 | Acct-turn_4 | Acct-pay_1 | Acct-pay_2 | Acct-pay_3 | Acct-pay_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.

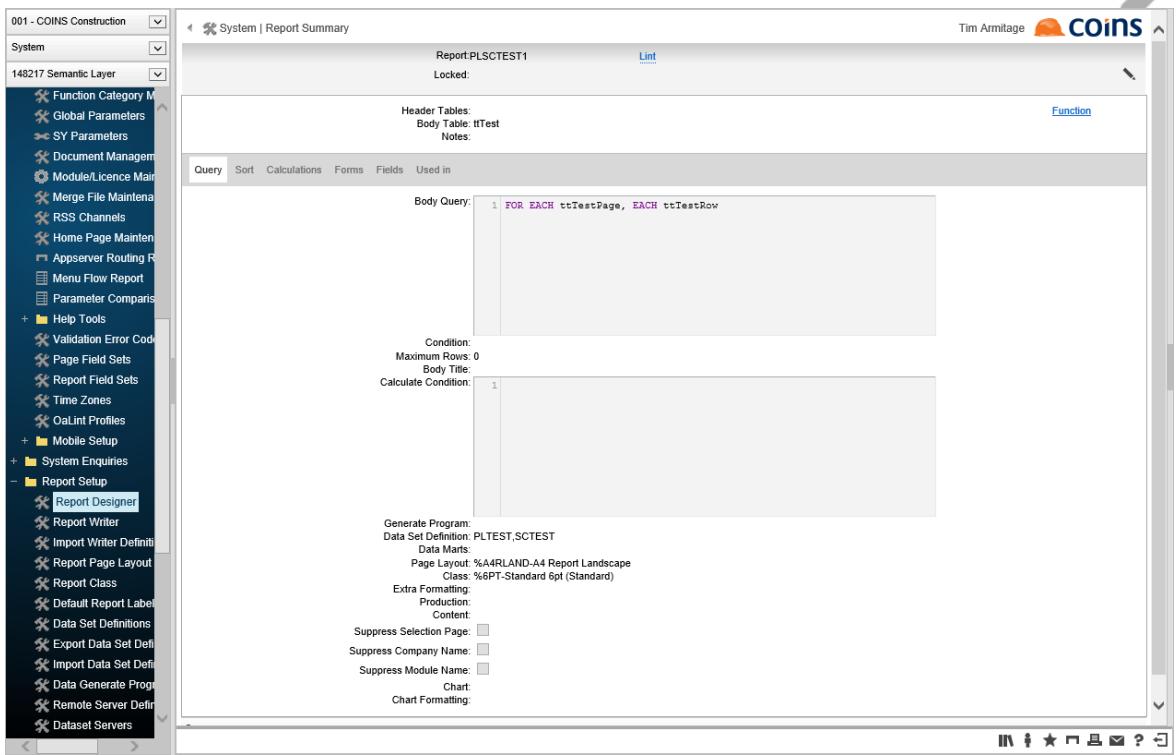
The records with SumType=TOT would not be processed because they were excluded with the query condition.

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.

Report Design

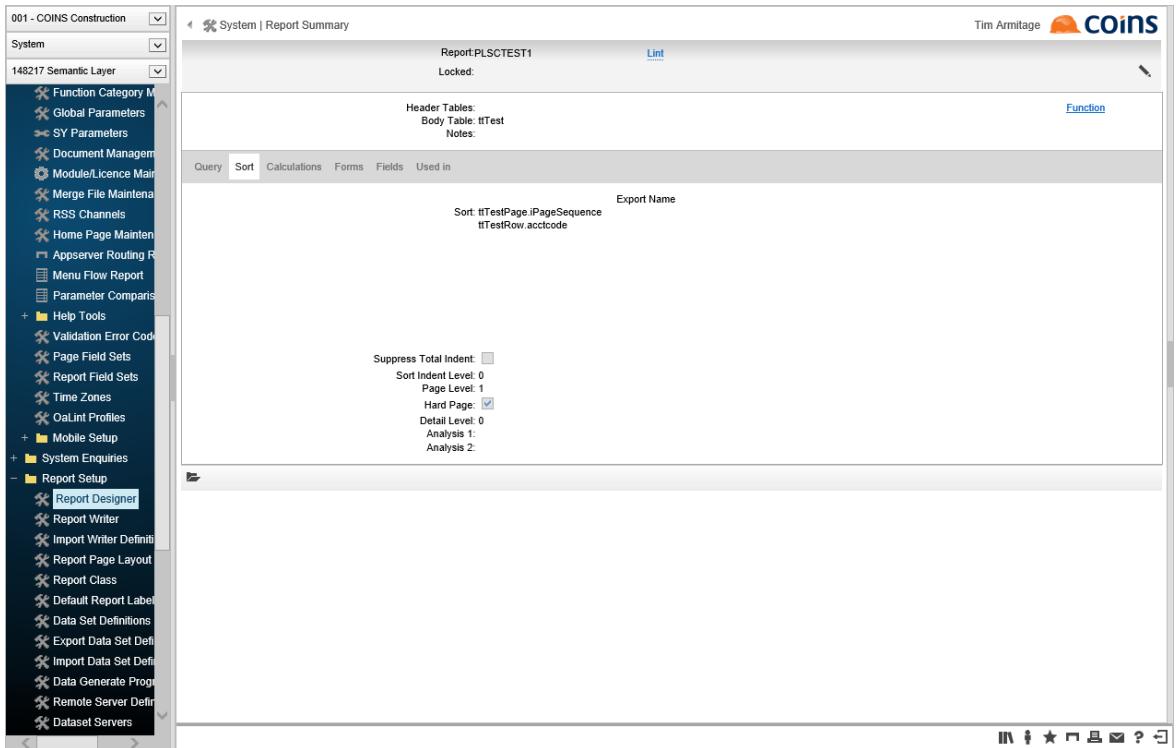
The temp tables (ttTestRow and ttTestPage) produced are designed to be used in OA designer and there are supporting methods and techniques to allow a matrix report to be built.

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 screenshot shows the COINS Report Designer interface. On the left is a navigation tree with categories like System, Function Category Manager, Global Parameters, etc. The main window displays a report summary for 'Report:PLSCTEST1'. The 'Sort' tab is selected in the toolbar. The report details include a Body Query with the code: `1 FOR EACH ttTestPage, EACH ttTestRow`. Below this are sections for Condition, Maximum Rows: 0, Body Title: '1', Calculate Condition: '1', and various export options like PLTEST, SCTest, Data Marts, and Page Layout. The bottom of the window shows standard toolbar icons.

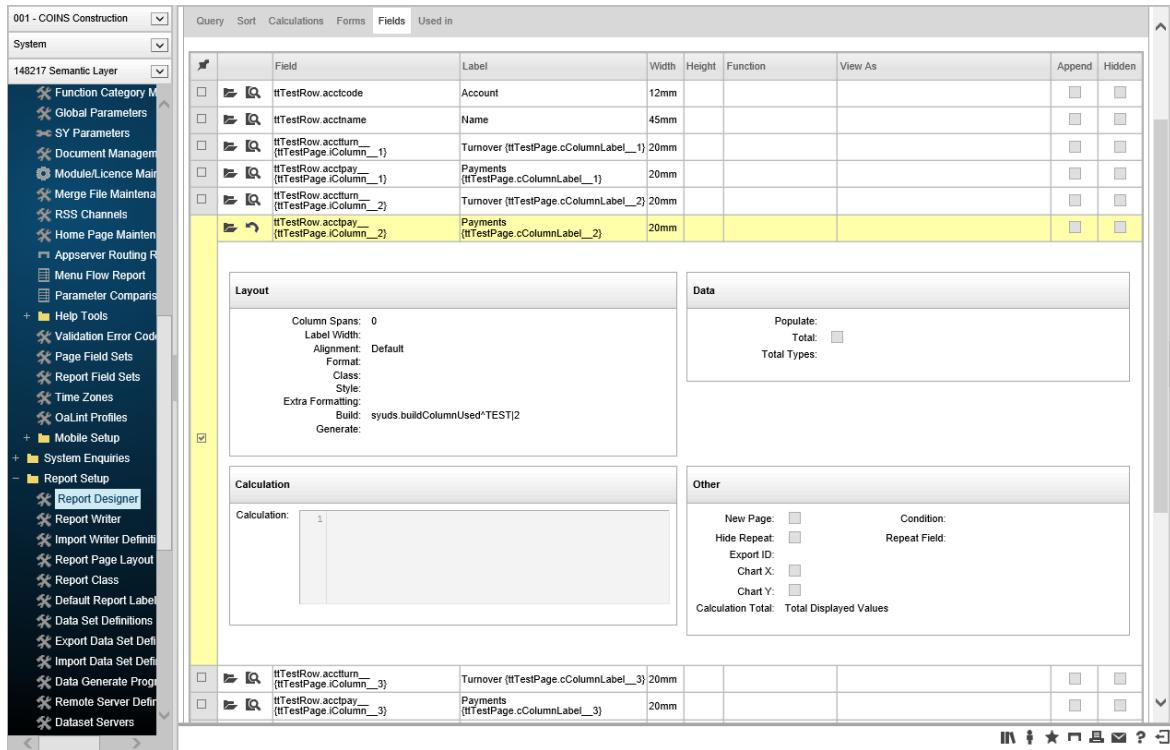
The sort order should include the page sequence at the top to allow the pages to come out in the correct sequence.



This screenshot is similar to the previous one but highlights the 'Sort' tab in the toolbar. In the report summary window, under the 'Sort' tab, it shows 'Export Name' as 'ttTestPage.iPageSequence' and 'ttTestRow.acctcode'. Below this, there are settings for Suppress Total Indent, Sort Indent Level: 0, Page Level: 1, Hard Page: checked, Detail Level: 0, Analysis 1, and Analysis 2. The rest of the interface is identical to the first screenshot.

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.cColumnName_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.



The screenshot shows the COINS Report Designer application. On the left is a navigation tree with categories like System, Help Tools, Validation Error Codes, etc. The main area has tabs for Query, Sort, Calculations, Forms, Fields, and Used in. The Fields tab is active, displaying a table of fields:

| | Field | Label | Width | Height | Function | View As | Append | Hidden |
|--------------------------|--|-------------------------------------|-------|--------|----------|---------|-------------------------------------|--------------------------|
| <input type="checkbox"/> | <input type="checkbox"/> ttTestRow.acccode | Account | 12mm | | | | <input type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> | <input type="checkbox"/> ttTestRow.acctname | Name | 45mm | | | | <input type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> | <input type="checkbox"/> ttTestRow.acctnum {ttTestPage.iColumn_1} | Turnover {ttTestPage.cColumnName_1} | 20mm | | | | <input type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> | <input type="checkbox"/> ttTestRow.accpay {ttTestPage.iColumn_1} | Payments {ttTestPage.cColumnName_1} | 20mm | | | | <input type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> | <input type="checkbox"/> ttTestRow.acctnum {ttTestPage.iColumn_2} | Turnover {ttTestPage.cColumnName_2} | 20mm | | | | <input type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> | <input type="checkbox"/> ttTestRow.accpay {ttTestPage.iColumn_2} | Payments {ttTestPage.cColumnName_2} | 20mm | | | | <input checked="" type="checkbox"/> | <input type="checkbox"/> |

Below the table are several configuration panels:

- Layout:** Column Spans: 0, Label Width: Default, Alignment: Default, Format: , Class: , Style: , Extra Formatting: Build: syuds.buildColumnUsed"TEST|2", Generate:
- Data:** Populate: , Total: , Total Types:
- Calculation:** Calculation: 1
- Other:** New Page: , Condition: , Hide Repeat: , Repeat Field: , Export ID: , Chart X: , Chart Y: , Calculation Total: Total Displayed Values

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.

Sample Output

With a total field

Purchase Ledger - PLSC Combined1
COINS Construction


| Account | Name | Turnover 1 | Payments 1) | Turnover 2 | Payments 2) | Total Turnover | Total Payments |
|---------|---------------------------------|------------|-------------|------------|-------------|----------------|----------------|
| A1 | Abba Sealants Ltd | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1 | Ashmore Contract Servicesas | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A10 | A & E Holdings Ltd | 1,900.00 | 57.75 | 0.00 | 0.00 | 1,900.00 | 57.75 |
| A1000 | Allsport Builders MerchantsX | 14200.00 | 10329.10 | 0.00 | 0.00 | 14200.00 | 10329.10 |
| A1001 | Anderson & Sub | 1,050.00 | 0.00 | 0.00 | 0.00 | 1,050.00 | 0.00 |
| A1002 | ADM Contracts & Plant Hire | 14,200.00 | 16,000.00 | 0.00 | 0.00 | 15,850.00 | 16,000.00 |
| A1003 | Abba Sealants | 5782.50 | 0.00 | 0.00 | 0.00 | 9,762.50 | 0.00 |
| A1002 | ASD Andrews Brown Cast Steel | 112,040.00 | 0.00 | 0.00 | 0.00 | 112,040.00 | 0.00 |
| A1002 | Invoicing A1002 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1003 | A & J Beverage | 4,110.00 | 0.00 | 0.00 | 0.00 | 4,011.00 | 0.00 |
| A1003 | Authorised receipt A1003 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1004 | Alban Roofing Ltd*2 | 0.00 | 2,176.68 | 0.00 | 0.00 | 0.00 | 2,176.68 |
| A1004 | ARC Quarry Products Ltd | 280.00 | 0.00 | 0.00 | 0.00 | 280.00 | 0.00 |
| A1005 | Arc East Scotland Oil Supply | 565.00 | 100.00 | 0.00 | 0.00 | 565.00 | 100.00 |
| A1007 | Carter Inc | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1008 | Carter Inc | 1,219.00 | 1,780.00 | 0.00 | 0.00 | 1,219.00 | 1,780.00 |
| A1008 | Carter Inc 2 | 0.00 | 680.00 | 0.00 | 0.00 | 0.00 | 680.00 |
| A101 | A & E Contracts (S) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A11 | A & J Beverage | -100.00 | 0.00 | 0.00 | 0.00 | -100.00 | 0.00 |
| A11 | Andrews Ltd | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A110 | Albion Contracts & Eng Serv Ltd | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A110 | ATS Scotland Tyre Service | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1101 | Albion Contracts & Eng Serv Ltd | 255.00 | 100.00 | 0.00 | 0.00 | 255.00 | 100.00 |
| A111 | A Proctor (Insulation) Ltd | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1178 | ApeX Industrial Ltd | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1181 | Join Contracting | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A12 | A G S Scotland | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1234 | Abba Fencing Ltd. | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A12346 | ASD Andrews Brown Cast Steel | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1235 | Abba Sealants Ltd | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1326 | Admiral Rentals & Co. Ltd | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1347 | Abraflex Plastic Ltd | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1355 | F L Wabco Co A | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1419 | Albion Contracts Ltd | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1476 | Agents Decoking Co Ltd | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1500 | ACS Environmental Ltd | 0.00 | 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 | Total Payments |
|---------|---------------------------------|------------|-------------|------------|-------------|----------------|----------------|
| A1 | Abba Sealants Ltd | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1 | Ashmore Contract Servicesas | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A10 | A & E Holdings Ltd | 1,900.00 | 57.75 | 0.00 | 0.00 | 1,900.00 | 57.75 |
| A1000 | Allsport Builders MerchantsX | 14200.00 | 10329.10 | 0.00 | 0.00 | 14200.00 | 10329.10 |
| A1001 | Anderson & Sub | 1,050.00 | 0.00 | 0.00 | 0.00 | 1,050.00 | 0.00 |
| A1002 | ADM Contracts & Plant Hire | 15,850.00 | 16,000.00 | 0.00 | 0.00 | 15,850.00 | 16,000.00 |
| A1003 | Abba Sealants | 5782.50 | 0.00 | 0.00 | 0.00 | 9,762.50 | 0.00 |
| A1002 | ASD Andrews Brown Cast Steel | 112,040.00 | 0.00 | 0.00 | 0.00 | 112,040.00 | 0.00 |
| A1002 | Invoicing A1002 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1003 | A1116 | 4,110.00 | 0.00 | 0.00 | 0.00 | 4,011.00 | 0.00 |
| A1003 | Authorised receipt A1003 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1004 | Alban Roofing Ltd*2 | 0.00 | 2,176.68 | 0.00 | 0.00 | 0.00 | 2,176.68 |
| A1004 | ARC Quarry Products Ltd | 280.00 | 0.00 | 0.00 | 0.00 | 280.00 | 0.00 |
| A1005 | Arc East Scotland Oil Supply | 565.00 | 100.00 | 0.00 | 0.00 | 565.00 | 100.00 |
| A1007 | Carter Inc | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1008 | Carter Inc 2 | 1,219.00 | 1,780.00 | 0.00 | 0.00 | 1,219.00 | 1,780.00 |
| A101 | A & E Contracts (S) | 0.00 | 680.00 | 0.00 | 0.00 | 0.00 | 680.00 |
| A11 | A & J Beverage | -100.00 | 0.00 | 0.00 | 0.00 | -100.00 | 0.00 |
| A110 | Albion Contracts & Eng Serv Ltd | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A110 | ATS Scotland Tyre Service | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A111 | A Proctor (Insulation) Ltd | 255.00 | 100.00 | 0.00 | 0.00 | 255.00 | 100.00 |
| A1178 | ApeX Industrial Ltd | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1181 | Join Contracting | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A12 | A G S Scotland | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1234 | Abba Fencing Ltd. | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1235 | Abba Sealants Ltd | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1326 | Admiral Rentals & Co. Ltd | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1347 | Abraflex Plastic Ltd | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1355 | F L Wabco Co A | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1419 | Albion Contracts Ltd | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1476 | Agents Decoking Co Ltd | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1500 | ACS Environmental Ltd | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

Purchase Ledger - PLSC Combined1
COINS Construction


| Account | Name | Turnover Total | Payments (S) | Total Turnover | Total Payments |
|---------|---------------------------------|----------------|--------------|----------------|----------------|
| A1 | Abba Sealants Ltd | 0.00 | 0.00 | 0.00 | 0.00 |
| A1 | Ashmore Contract Servicesas | 0.00 | 0.00 | 0.00 | 0.00 |
| A10 | A & E Holdings Ltd | 1,900.00 | 57.75 | 0.00 | 0.00 |
| A1000 | Allsport Builders MerchantsX | 14,200.00 | 10,329.10 | 0.00 | 0.00 |
| A1001 | Anderson & Sub | 1,050.00 | 0.00 | 0.00 | 0.00 |
| A1002 | ADM Contracts & Plant Hire | 14,200.00 | 16,000.00 | 0.00 | 0.00 |
| A1003 | Abba Sealants | 5,782.50 | 0.00 | 0.00 | 0.00 |
| A1002 | ASD Andrews Brown Cast Steel | 112,040.00 | 0.00 | 0.00 | 0.00 |
| A1002 | Invoicing A1002 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1003 | A1116 | 4,110.00 | 0.00 | 0.00 | 0.00 |
| A1003 | Authorised receipt A1003 | 0.00 | 0.00 | 0.00 | 0.00 |
| A1004 | Alban Roofing Ltd*2 | 0.00 | 2,176.68 | 0.00 | 0.00 |
| A1004 | ARC Quarry Products Ltd | 280.00 | 0.00 | 0.00 | 0.00 |
| A1005 | Arc East Scotland Oil Supply | 565.00 | 100.00 | 0.00 | 0.00 |
| A1007 | Carter Inc | 0.00 | 0.00 | 0.00 | 0.00 |
| A1008 | Carter Inc 2 | 1,219.00 | 1,780.00 | 0.00 | 0.00 |
| A101 | A & E Contracts (S) | 0.00 | 680.00 | 0.00 | 0.00 |
| A11 | A & J Beverage | -100.00 | 0.00 | 0.00 | 0.00 |
| A110 | Albion Contracts & Eng Serv Ltd | 0.00 | 0.00 | 0.00 | 0.00 |
| A110 | ATS Scotland Tyre Service | 0.00 | 0.00 | 0.00 | 0.00 |
| A111 | A Proctor (Insulation) Ltd | 255.00 | 100.00 | 0.00 | 0.00 |
| A1178 | ApeX Industrial Ltd | 0.00 | 0.00 | 0.00 | 0.00 |
| A1181 | Join Contracting | 0.00 | 0.00 | 0.00 | 0.00 |
| A12 | A G S Scotland | 0.00 | 0.00 | 0.00 | 0.00 |
| A1234 | Abba Fencing Ltd. | 0.00 | 0.00 | 0.00 | 0.00 |
| A1235 | Abba Sealants Ltd | 0.00 | 0.00 | 0.00 | 0.00 |
| A1326 | Admiral Rentals & Co. Ltd | 0.00 | 0.00 | 0.00 | 0.00 |
| A1347 | Abraflex Plastic Ltd | 0.00 | 0.00 | 0.00 | 0.00 |
| A1355 | F L Wabco Co A | 0.00 | 0.00 | 0.00 | 0.00 |
| A1419 | Albion Contracts Ltd | 0.00 | 0.00 | 0.00 | 0.00 |
| A1476 | Agents Decoking Co Ltd | 0.00 | 0.00 | 0.00 | 0.00 |
| A1500 | ACS Environmental 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 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.

6.1.11 TableAlias

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

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

1st parameter is the source dataset table and the 2nd 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 1st parameter is the source temp-table. The 2nd parameter is the condition applied to the query, the 3rd parameter is the sort order, the 4th parameter is the number of records returned , the 5th 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');
```

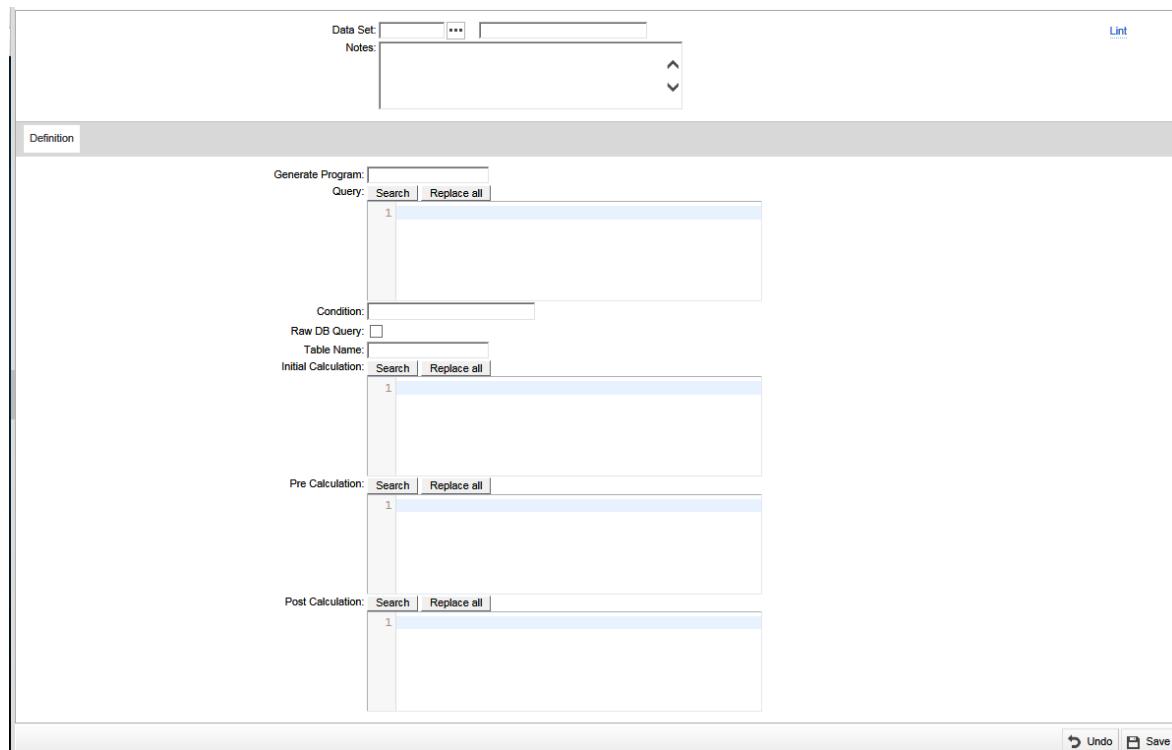
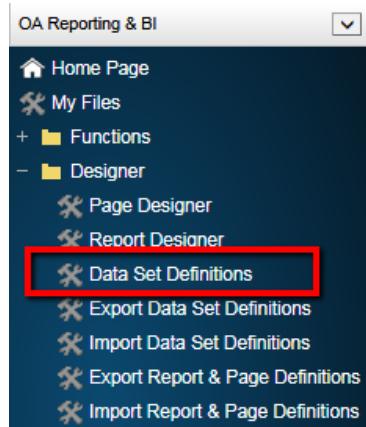
ttCost and ttRev would be combined with fields kco,job_num,dCosts,dRev in to dataset CostRev. This is equivalent to the first example above.

7 Data Set Exercises

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

7.1 Exercise 1 - Creating the Dataset

In Designer, navigate to Data Set Definitions.

A screenshot of the Data Set Definitions configuration screen. The screen has a header with 'Data Set' and 'Notes' fields, and a 'Lint' button. Below the header is a 'Definition' tab. The configuration area contains several sections: 'Generate Program' (Query: Search | Replace all, value '1'), 'Condition' (Raw DB Query: , Table Name: , Initial Calculation: Search | Replace all, value '1'), 'Pre Calculation' (Search | Replace all, value '1'), and 'Post Calculation' (Search | Replace all, value '1'). At the bottom right are 'Undo' and 'Save' buttons.

Click 

Enter a Data Set name and description for the data set. For this exercise use your initials followed by M2DS1 (e.g. xxM2DS1)

| | | | |
|-----------|--|-----|-----------------------|
| Data Set: | NLM2DS1 | ... | M2 Dataset Exercise 1 |
| Notes: | <div style="height: 100px; border: 1px solid #ccc; padding: 5px;"> <p> </p> </div> | | |

The Notes section is there to allow you to document the purposes of the dataset so that you can refer to this in the future.

In the Query box, enter:

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

{jobSelect} is being specified here as we will be creating a page to pass a standard contract selection later on.

| | |
|-------------------|---|
| Generate Program: | |
| Query: | <pre>1 FOR EACH jc_job WHERE jc_job.kco = {kco} {jobSelect}</pre> |

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

| | |
|---------------|---------------------------------|
| Raw DB Query: | <input type="button" value=""/> |
| Table Name: | Contract |

7.1.1 Adding the Fields

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

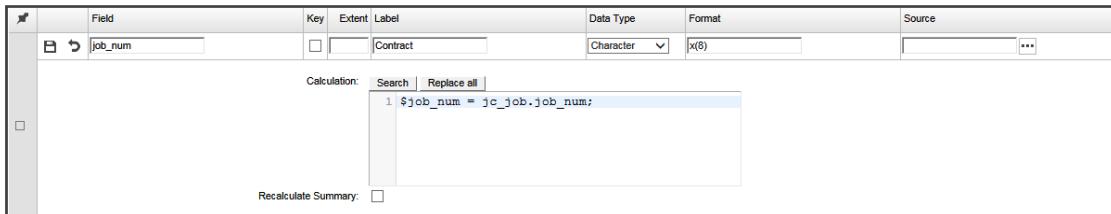
| Field | Label | Data Type | Format | Source |
|-------|---------|-----------|--------------|------------|
| kco | Company | Integer | ->>>,>>>,>>9 | jc_job.kco |

| # | Field | Key | Extent | Label | Data Type | Format | Source |
|---|-------|-----|--------|---------|-----------|--------------|------------|
| 1 | kco | | | Company | Integer | ->>>,>>>,>>9 | jc_job.kco |

For the next field we will use an example of assigning a table field to a string variable.

| Field | Label | Data Type | Format | Source |
|---------|----------|-----------|--------|----------------|
| job_num | Contract | Character | x(8) | jc_job.job_num |

in the Calculation box enter : \$job_num = jc_job.job_num;



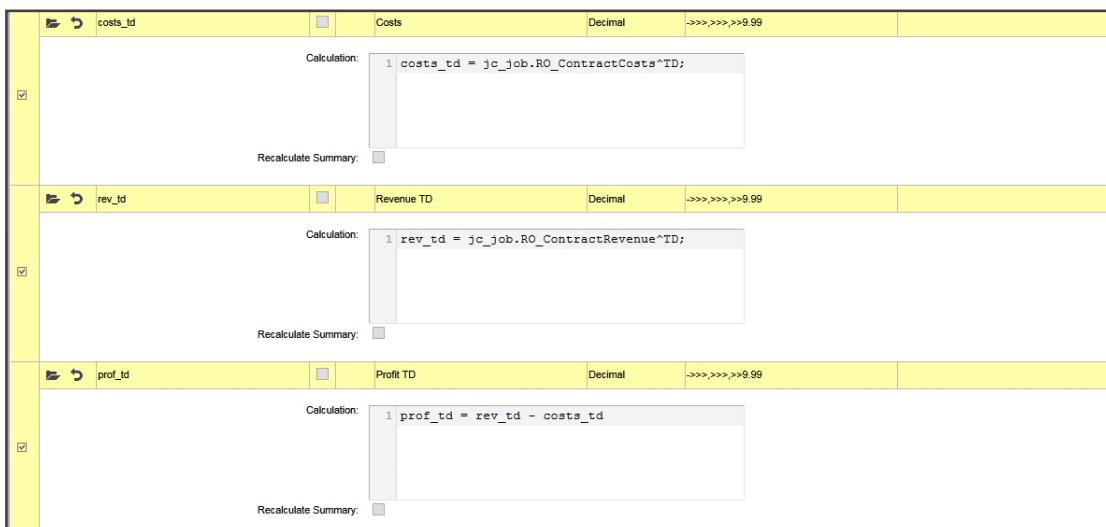
The screenshot shows a dataset configuration window. In the 'Field' column, 'job_num' is selected. In the 'Calculation' section, the formula '\$job_num = jc_job.job_num;' is entered. The 'Source' column is empty.

The third field is another direct source field:

| Field | Label | Data Type | Format | Source |
|----------|-------|-----------|--------|-----------------|
| job_name | Name | Character | x(8) | jc_job.job_name |

The final three fields are examples of assigning numeric variables and creating a calculation:

| Field | Label | Data Type | Format | Source |
|----------|--|-----------|-------------|--------|
| costs_td | Costs | Decimal | ->>>,>>,>>9 | |
| | Calculation:costs_td = jc_job.RO_ContractCosts^TD; | | | |
| rev_td | Revenue | Decimal | ->>>,>>,>>9 | |
| | Calculation:rev_td = jc_job.RO_ContractRevenue^TD; | | | |
| prof_td | Profit | Decimal | ->>,>>,>>9 | |
| | Calculation:prof_td = jc_job.RO_ContractRevenue^TD - jc_job.RO_ContractCosts^TD; | | | |



The screenshot shows a dataset configuration window with three rows. Row 1: Field 'costs_td' (Label 'Costs', Data Type 'Decimal', Format '->>>,>>,>>9'), Calculation: '1 costs_td = jc_job.RO_ContractCosts^TD;'. Row 2: Field 'rev_td' (Label 'Revenue TD', Data Type 'Decimal', Format '->>>,>>,>>9.99'), Calculation: '1 rev_td = jc_job.RO_ContractRevenue^TD;'. Row 3: Field 'prof_td' (Label 'Profit TD', Data Type 'Decimal', Format '->>,>>,>>9.99'), Calculation: '1 prof_td = rev_td - costs_td'.

7.1.2 Testing the dataset

It is possible to use a dataset in the Query Editor to check the data being returned.

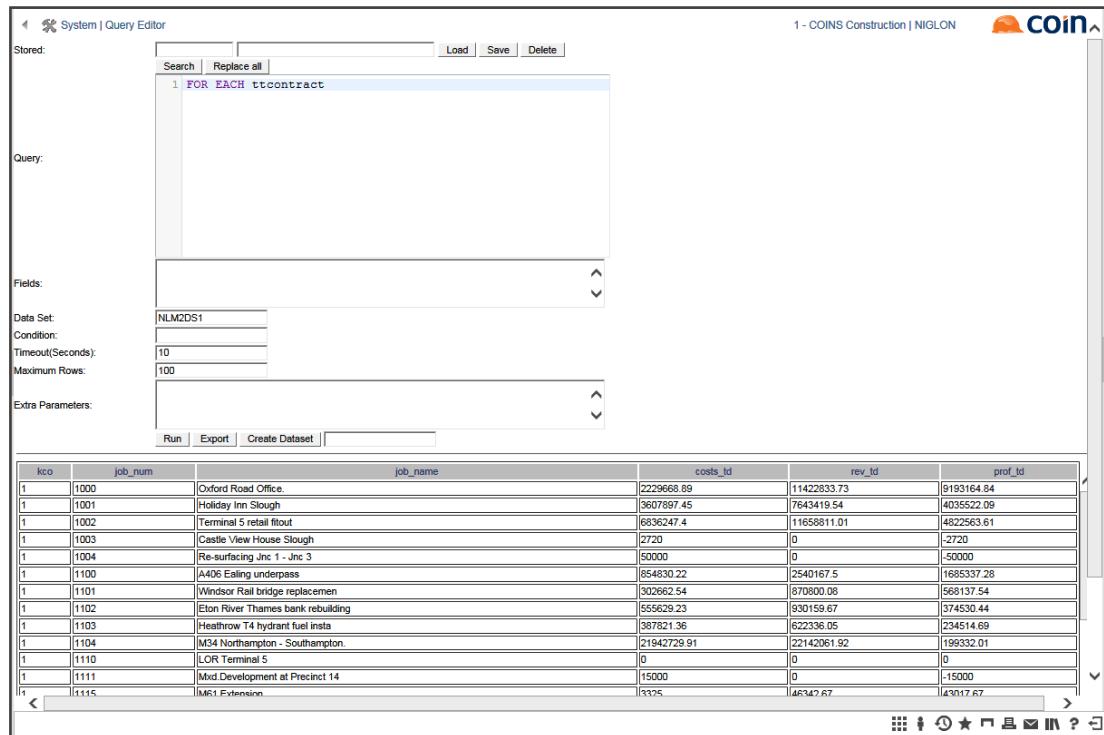
For our dataset, open up query editor and enter the following query:

FOR EACH ttcontract

Dataset tables are known as Temporary tables (as they are only exist at the time of the query). For this reason, any Dataset table must be prefixed by tt when used in a query.

Enter the name of the data set in the Data Set box – for our example we enter xxM2DS1 (replace xx with your initials).

Set the number of rows to 10 and click Run.



The screenshot shows the COINS System | Query Editor interface. The 'Query' pane contains the command:

```
1 FOR EACH ttcontract
```

The 'Fields' pane is empty. The 'Data Set' field is set to 'NLM2DS1'. The 'Condition' field is empty. The 'Timeout(Seconds)' field is set to '10'. The 'Maximum Rows' field is set to '100'. The 'Run' button is highlighted.

The results pane displays 15 rows of data:

| kco | job_num | job_name | costs_id | rev_id | prof_id |
|-----|---------|-----------------------------------|-------------|-------------|------------|
| 1 | 1000 | Oxford Road Office. | 2229668.89 | 11422833.73 | 9193164.84 |
| 1 | 1001 | Holiday Inn Slough | 3607697.45 | 7643419.54 | 4035522.09 |
| 1 | 1002 | Terminal 5 retail fitout | 6836247.4 | 11658811.01 | 4822563.61 |
| 1 | 1003 | Castle View House Slough | 2720 | 0 | -2720 |
| 1 | 1004 | Re-surfacing Inc 1 - Jnc 3 | 50000 | 0 | -50000 |
| 1 | 1100 | A406 Ealing underpass | 854830.22 | 2540167.5 | 1685337.28 |
| 1 | 1101 | Windsor Rail bridge replacement | 302662.54 | 870800.08 | 568137.54 |
| 1 | 1102 | Eton River Thames bank rebuilding | 555629.23 | 930159.67 | 374530.44 |
| 1 | 1103 | Heathrow T4 hydrant fuel insta | 387821.36 | 622336.05 | 234514.69 |
| 1 | 1104 | M34 Northampton - Southampton. | 21942729.91 | 22142061.92 | 199332.01 |
| 1 | 1110 | LOR Terminal 5 | 0 | 0 | 0 |
| 1 | 1111 | Mxd Development at Precinct 14 | 15000 | 0 | -15000 |
| 1 | 1115 | Mk1 Extension | 13375 | 14547.67 | 143017.67 |

7.1.3 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. NLM2Exercise1

Function Name -Description identifier.e.g. NLM2EX1

Function Module - Contract Status

Function Category - Report

Function Context – Leave as defaulted from the Function Description.

Function Program - wou005

| | Function ^ | Description ^ | Type | Module ^ | Category |
|--|---------------|------------------|----------|--------------------|------------|
| | NLM2Exercise1 | NL M2 Exercise 1 | Function | JC-Contract Status | REP-Report |
| Context: NL M2 Exercise 1 Program: wou005 Parameters: nolinfo=Y Notes: Parent: Access Type: Role Type: - | | | | | |

7.1.4 Set up the Page



Set up the page section using 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 Save to save the new Page. COINS returns to the summary for the page section being created.

7.1.5 Add the Page Section Forms

Select the Form tab create an Update form.

Click and enter the following information to create the Form :

| | |
|------------|---------------|
| Form | Select Update |
| Field Type | Select Update |
| Linked To | Leave blank |

Click the Save to create the form.

7.1.6 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.

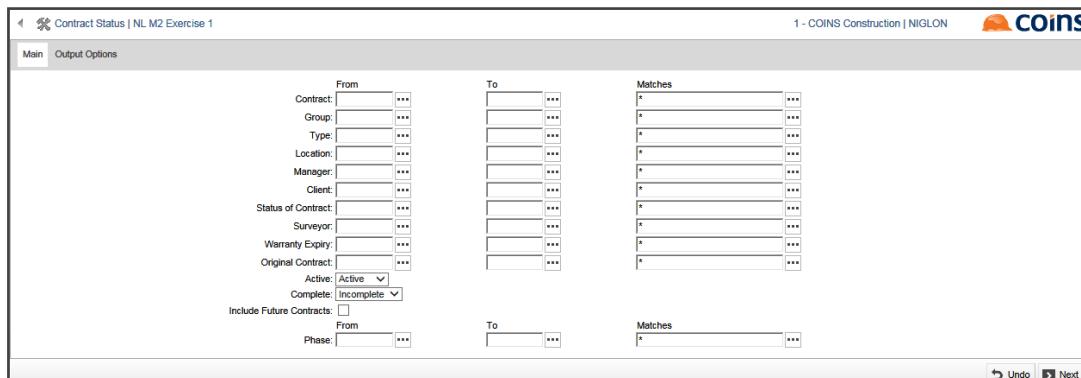
Add the Field, leaving Field and Label blank :

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

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.



7.1.7 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 our temporary table from the dataset ttcontract

Body Query Enter the query. This will select which records are shown on the report. Enter the following:

FOR EACH ttcontract

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.

DataSetSpecify 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 .

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).

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).

7.1.8 Sorting the Report

On the sort tab, enter ttcontract.job_num+ in the Sort field and Click Save  . COINS returns to the summary for the report section being created.

7.1.9 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:

FormSelect Body

Field TypeSelect Body

Linked ToLeave blank

Click the Save Button  to create the form.

7.1.10 Add the Fields

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

NOTE: 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:

FieldThe name of the field.

LabelThe label to appear in the column heading.

WidthThe width of the column on the report.

FieldLabelWidth

job_numContract15mm

job_nameContract Name60mm

costs_tdCosts TD20mm

rev_tdRevenue TD20mm

prof_tdProfit TD20mm

Click Save  after entering each field.

The report is now ready to run.

7.1.11 Running the Report

Click the menu item for the report.

Enter selection criteria.

Click on the Output Options tab and select the appropriate options.

Click NEXT to run the report.

Example Output:

| Contract | Contract Name | Costs ID | revenue ID | Profit ID |
|----------|-----------------------------------|---------------|---------------|--------------|
| 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 | 6,888,047.81 | 11,633,811.01 | 4,745,763.20 |
| 1100 | A406 Ealing underpass | 857,097.61 | 2,540,227.06 | 1,683,129.45 |
| 1101 | Windsor Rail bridge replacement | 302,662.54 | 870,800.08 | 568,137.54 |
| 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 | 5,061.33 | 1,289,733.19 | 1,284,671.86 |
| 1401 | NTL Cabling installation | 926.00 | 67,879.00 | 66,953.00 |
| 1966 | FM - Wembley National Stadium | 200.00 | 128.00 | -72.00 |
| 500 | test05 | 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 | LeeEbs Test Contract | 0.00 | 0.00 | 0.00 |
| euro | Euro Contract | 68.45 | 0.00 | -68.45 |
| FSS0 | 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 |
| TIM999 | Tim D Test | 0.00 | 0.00 | 0.00 |
| W100 | Singapore Airport | 0.00 | 0.00 | 0.00 |
| Z001 | Test for OA reporting xxx | 99,937.50 | 0.00 | -99,937.50 |
| z004 | test auto number sequence | 150.00 | 0.00 | -150.00 |

7.2 Exercise 1a - Using multiple Data Sets

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

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.

Function Code -Initials + M2 + Exercise No.e.g. NLM2Exercise1a

Function Name -Description identifier.E.g. NLM2EX1a

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 Parametersstn_code=nlm2exercise1

7.2.2 Creating the 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 M2DS2.

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

Data Set: NLM2DS2 Data Set Example - Multiple
Notes: P/L Open Invoice

Definition Fields

Generate Program:
Query:
1 FOR EACH jc_job WHERE jc_job.kco = {kco}
2 EACH ap_invoice OF jc_job

Condition:
Raw DB Query:
Table Name: plinvoice

Initial Calculation: 1

Pre Calculation: 1

Post Calculation: 1

 Open  Next

7.2.3 Adding the Fields

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

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.

| | 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"/> |   plgross | | <input type="checkbox"/> | PL Gross | Decimal | >>>,>>>,>>9.99 | ap_invoice.rin_amount |

The third field is another direct source field:

FieldLabelData TypeFormatSource

plgrossPL GrossDecimal->>>,>>>,>>9.99ap_invoice.rin_amount

7.2.4 Create the Report

Copy the report definition for xxM2exercise1 and rename it xxM2exercise1a.

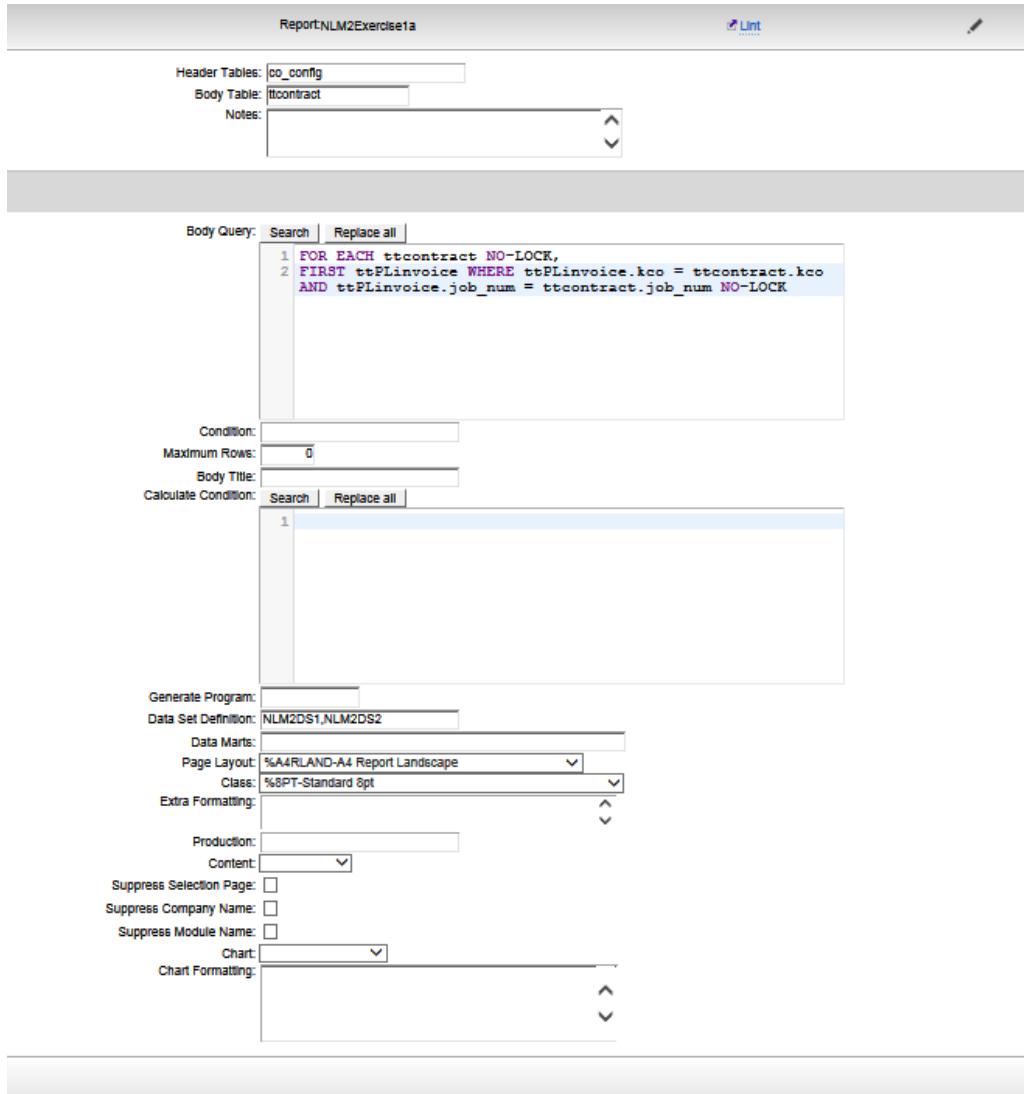
For this new report, modify the query to read our two Data Set tables:

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

Note that 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.

NLM2DS1,NLM2DS2



The screenshot shows the configuration interface for a report named "Report:NLM2Exercise1a". The "Header Tables" field contains "kco_config" and the "Body Table" field contains "ttcontract". The "Notes" field is empty. In the "Body Query" section, the following SQL code is entered:

```
1 FOR EACH ttcontract NO-LOCK,
2 FIRST ttPLinvoice WHERE ttPLinvoice.kco = ttcontract.kco
AND ttPLinvoice.job_num = ttcontract.job_num NO-LOCK
```

The "Condition" field is empty. The "Maximum Rows" field is set to 0. The "Body Title" field is empty. In the "Calculate Condition" section, there is a single row labeled "1". The "Generate Program" field is empty. The "Data Set Definition" field contains "NLM2DS1,NLM2DS2". The "Data Marts" field is empty. The "Page Layout" field is set to "%A4RLAND-A4 Report Landscape". The "Class" field is set to "%8PT-Standard 8pt". The "Extra Formatting" field is empty. The "Production" field is empty. The "Content" field is empty. Under "Suppress Selection Page", "Suppress Company Name", and "Suppress Module Name", there are three checkboxes, all of which are unchecked. The "Chart" field is empty. The "Chart Formatting" field is empty.

On the Body Field add an additional column for plgross



7.2.5 Running the report

Run the function `xxM2Exercise1a`. You should get output similar to the following:

JC Contract Status - NL M2 Exercise 1a
COINS Construction



| Contract | Contract Name | Costs ID | Revenue ID | Profit ID | PL Gross |
|----------|-------------------------------------|---------------|---------------|--------------|--------------|
| 1000 | Oxford Rail Office. | 2,521,712,24 | 10,801,715,46 | 8,280,003,21 | 208,445,19 |
| 1001 | Holiday Inn Slough | 5,418,730,39 | 7,634,354,54 | 2,224,889,15 | 2,043,416,71 |
| 1002 | Terminal 5 retail fitout | 6,888,047,81 | 11,633,811,01 | 4,745,763,81 | 6,699,56 |
| 1100 | A406 Ealing underpass | 857,097,61 | 2,540,227,06 | 1,683,129,45 | 9,688,32 |
| 1101 | Windsor Rail bridge replacement | 302,622,54 | 870,800,00 | 568,137,54 | 14,728,60 |
| 1102 | Eton River Thames bridge rebuilding | 555,620,23 | 930,159,67 | 374,530,44 | 185,72 |
| 1103 | Heathrow T4 hydraulik fuel insta | 387,821,36 | 622,336,05 | 234,514,69 | 203,75 |
| 1104 | M34 Northampton - Southampton. | 21,870,051,32 | 21,408,916,57 | -46,113,745 | 46,419,50 |
| 1115 | M6 1 Extension | 6,325,00 | 46,342,67 | 40,017,67 | 5,942,50 |
| 1200 | Bluewater shopping M & E | 5,061,33 | 1,289,733,19 | 1,284,671,86 | 11,975,00 |
| 1966 | FM - Wembley National Stadium | 200,00 | 128,00 | -72,00 | 235,00 |
| 501 | ed | 0,00 | 0,00 | 0,00 | 1,050,00 |
| 6000 | FM Contract | -400,00 | 80,00 | 480,00 | 517,50 |
| 6100 | FM Property Maintenance | 44,00 | 4,848,00 | 4,804,00 | 51,70 |
| 6200 | A465 Vale of Neath Bridge Hirwaun | 395,00 | 6,775,60 | 6,380,60 | 464,13 |
| euro | Euro Contract | 68,45 | 0,00 | -68,45 | 68,45 |
| 2001 | Trial for OA reporting xxx | 99,937,50 | 0,00 | -99,937,50 | 11,456,25 |

7.3 Exercise 1b - Using multiple Data Sets to merge data

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 -Initials + M2 + Exercise No.e.g. NLM2Exercise1b

Function Name -Description identifier.E.g. NLM2EX1b

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`

1.4.2 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 M2DS3.

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



7.3.2 Adding the Fields

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

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

slgrossSLGrossDecimal->>>,>>>,>>9.99ar_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 SLCSinvoice here too.

| Data Set NLM2DS4 Data Set Example - Merge Notes: Contract Sales Transactions | |
|---|--------|
| Definition | Fields |
| <p>Generate Program:</p> <p>Query:</p> <pre>1 FOR EACH jc_job WHERE jc_job.kco = {kco} {jobSelect}, 2 EACH cs_certificate OF jc_job</pre> <p>Condition:</p> <p>Raw DB Query: <input type="checkbox"/></p> <p>Table Name: SLCInvoice</p> | |

7.3.3 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 | Extent | Label | Data Type | Format | Source |
|---|------------|-------------------------------------|--------|----------|-----------|----------------|---------------------------------|
| □ | ic kco | <input checked="" type="checkbox"/> | | Company | Integer | >>>,>>>,>>9 | jc_job.kco |
| □ | ic job_num | <input checked="" type="checkbox"/> | | Contract | Character | x(8) | jc_job.job_num |
| □ | ic 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

Report.NLM2Exercise1b

Header Tables: *co_config*
Body Table: *je_job*
Notes: DATASET MULTIPLE & MERGE

Query Sort Calculations

Body Query: Search | Replace all
`1 FOR EACH je_job WHERE kco = (kco){jobSelect},
2 EACH ttSLCSinvoice WHERE ttSLCSinvoice.kco = je_job.kco
AND ttSLCSinvoice.job_num = je_job.job_num`

Condition: Maximum Rows: 6
Body Title: Calculate Condition: Search | Replace all
`1`

Generate Program:
Data Set Definition: NLM2DS3,NLM2DS4
Data Matrix:
Page Layout: KAPRLAND-A4 Report Landscape
Case: %SFT-Standard Opt
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

Report.NLM2Exercise1b

Header Tables: *co_config*
Body Table: *je_job*
Notes: DATASET MULTIPLE & MERGE

Query Sort Calculations Forms Fields Used in

| | Field | Label | Width | Height | Function | View As | Append | Hidden |
|---|---|---------------|-------|--------|----------|---------|--------------------------|--------------------------|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> <input type="checkbox"/> <i>je_job.job_num</i> | Contract | 20mm | | | | <input type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> <input type="checkbox"/> <i>je_job.job_name</i> | Contract Name | 60mm | | | | <input type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> <input type="checkbox"/> <i>RO_ContractCosts^TD</i> | Costs TD | 20mm | | | | <input type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> <input type="checkbox"/> <i>RO_ContractRevenue^TD</i> | Revenue TD | 20mm | | | | <input type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> <input type="checkbox"/> <i>Profit TD</i> | Profit TD | 20mm | | | | <input type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> <input type="checkbox"/> <i>ttSLCSinvoice.SLGross</i> | Sales Balance | 20mm | | | | <input type="checkbox"/> | <input type="checkbox"/> |
| <hr/> | | | | | | | | |
| <input type="button" value="Show All"/> <input type="button" value="Add"/> <input type="button" value="Copy"/> <input type="button" value="Insert"/> <input type="button" value="Up"/> <input type="button" value="Down"/> <input type="button" value="Open"/> <input type="button" value="Multi"/> <input type="button" value="Bulk:"/> <input type="button" value="Detail"/> <input type="button" value="Delete"/> <input type="button" value="Transfer Fields"/> <input type="button" value=""/> | | | | | | | | |
| <input type="button" value="Advanced"/> <input type="text" value="Search: Field"/> <input type="button" value=""/> Selectors <input type="button" value="Form: Body"/> <input type="button" value="View: Grouped"/> | | | | | | | | |

7.3.4 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 |

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.

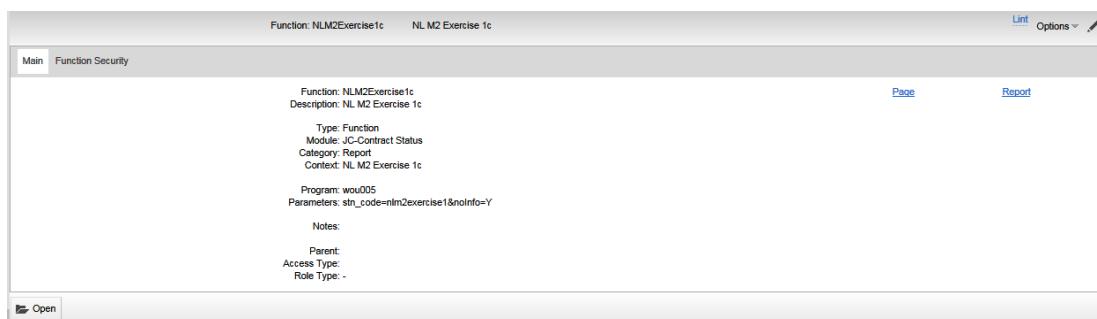
7.4.1 Create the Function and Menu

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 stn_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

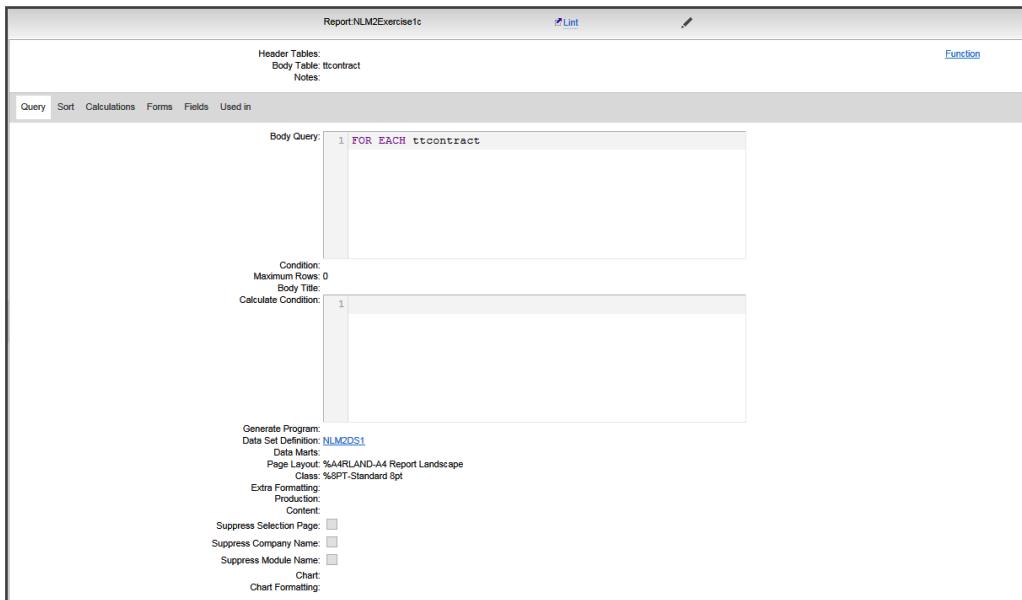
Body Table Enter the name of the database table being reporting on; in this case our temporary table from the Exercise 1 dataset ttcontract

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).



The screenshot shows the Report.NLM2Exercise1c interface. In the Body Query section, there is a code editor containing the following SQL-like query:

```
1 FOR EACH ttcontract
```

Below the query, there are several configuration sections:

- Condition:** Maximum Rows: 0
- Body Title:** Calculate Condition: 1
- Generate Program:** Data Set Definition: NLM2DS1
- Data Marts:** Page Layout: %A4RLAND-A4 Report Landscape
Class: %\$PT-Standard 8pt
- Extra Font:** Production, Content
- Suppress Selection Page:**
- Suppress Company Name:**
- Suppress Module Name:**
- Chart:** Chart Formatting:

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 C | 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 Th | 3085 | 0 | -3085 |
| 23 | ttContract | 10 | 20015 | Eastside Locks | 1355 | 0 | -1355 |

7.5 Exercise 1d – Dynamically switching between multiple Data Sets (v10.26 onwards)

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

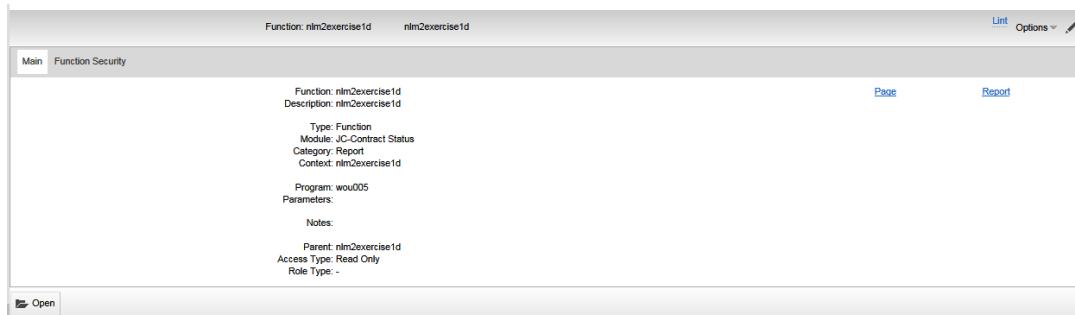
Function Category - Report

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

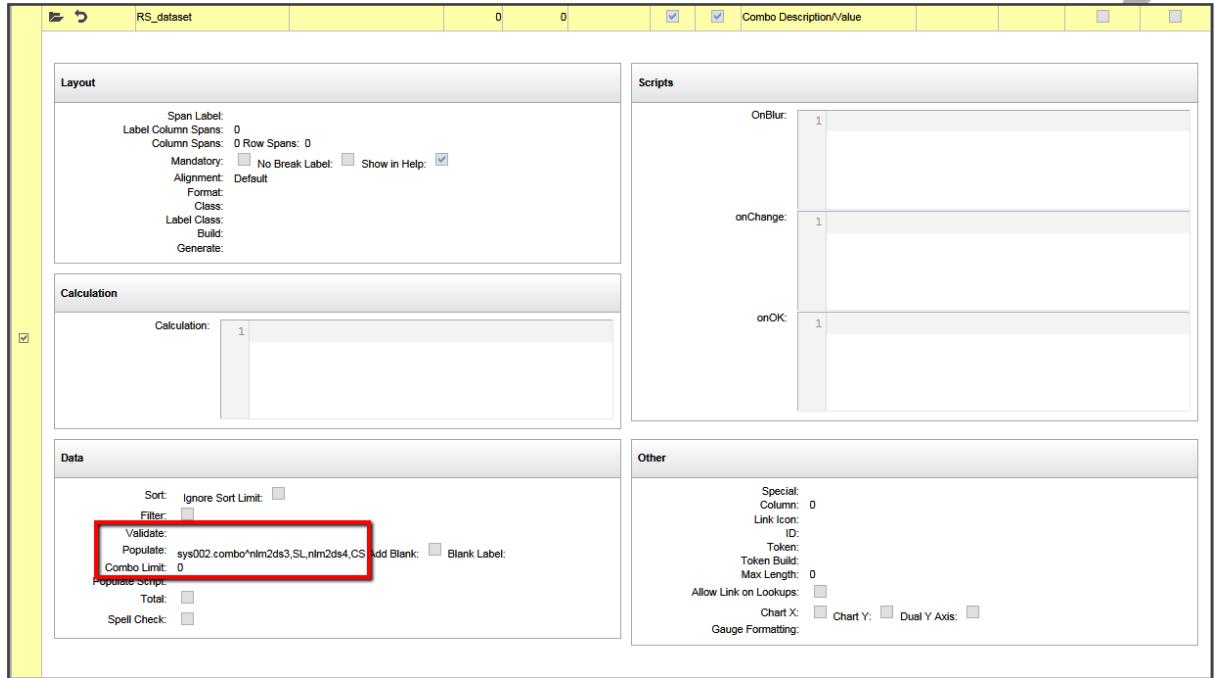
For this example, we will use the datasets nlm2ds3 and nlm2ds4 that we created in the previous exercise so there is no need to build new ones.

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



The screenshot shows the RS_dataset configuration interface. The 'Data' tab is selected, displaying various settings for a dropdown selection field. A red box highlights the 'Populate' field, which contains the value: sys002.combo*nlm2ds3,SL,nlm2ds4,CS. This configuration will create a dropdown menu where users can select either 'SL' or 'CS', and the corresponding dataset name ('nlm2ds3' or 'nlm2ds4') will be returned.

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

Header Tables: [co_config](#)
Body Table: [ic_job](#)
Notes: DATASET MULTIPLE & MERGE

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:
Data Set Definition: (RS_Dataset)
Data Marts:
Page Layout: %A4LANDA4 Report Landscape
Class: %BPT-Standard 8pt
Extra Formating:
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

Contract Status | nim2exercise1d

Main Output Options

| From | To | Matches |
|---|------------------------------------|------------------------------------|
| Contract: <input type="text"/> | <input type="button" value="..."/> | <input type="button" value="..."/> |
| Group: <input type="text"/> | <input type="button" value="..."/> | <input type="button" value="..."/> |
| Type: <input type="text"/> | <input type="button" value="..."/> | <input type="button" value="..."/> |
| Location: <input type="text"/> | <input type="button" value="..."/> | <input type="button" value="..."/> |
| Manager: <input type="text"/> | <input type="button" value="..."/> | <input type="button" value="..."/> |
| Client: <input type="text"/> | <input type="button" value="..."/> | <input type="button" value="..."/> |
| Status of Contract: <input type="text"/> | <input type="button" value="..."/> | <input type="button" value="..."/> |
| Surveyor: <input type="text"/> | <input type="button" value="..."/> | <input type="button" value="..."/> |
| Warranty Expiry: <input type="text"/> | <input type="button" value="..."/> | <input type="button" value="..."/> |
| Original Contract: <input type="text"/> | <input type="button" value="..."/> | <input type="button" value="..."/> |
| Active: <input type="button" value="Active"/> | | |
| Complete: <input type="button" value="Incomplete"/> | | |
| Include Future Contracts: <input type="checkbox"/> | | |
| <input type="button" value="Select"/> | | |

1 - COINS Construction | NIGLON 

Undo Next

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

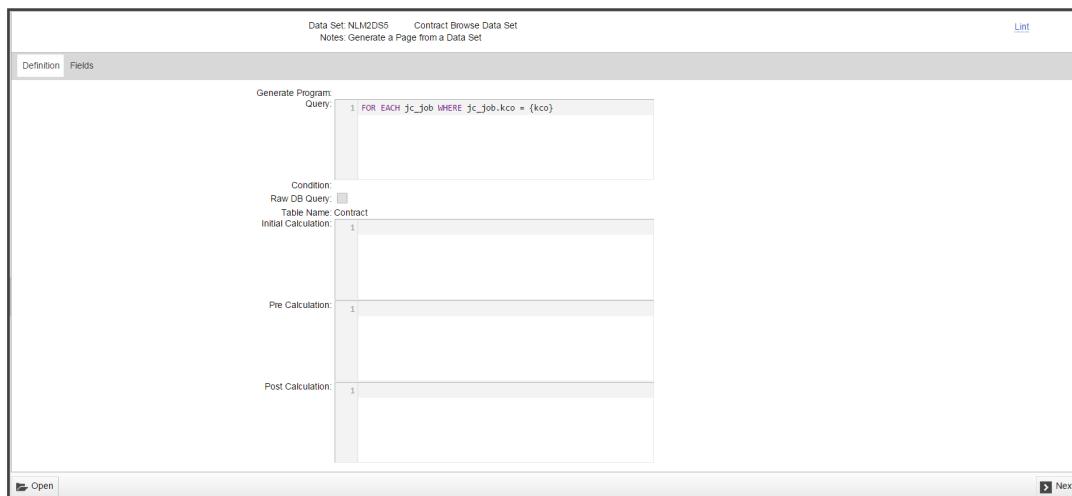
7.6 Exercise 2 - Generating Pages from a Data Set

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 |

| Field | Label | Data Type | Format | Source |
|-----------|---------|-----------|----------------|------------------------------|
| smp_dec_1 | Costs | Decimal | >>>,>>>,>>9.99 | jc_job.RO_contractcosts^TD |
| smp_dec_2 | Revenue | Decimal | >>>,>>>,>>9.99 | jc_job.RO_ContractRevenue^TD |

Data Set: NLM2DS5 Contract Browse Data Set
Notes: Generate a Page from a Data Set

| Definition | Fields | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--|--------|-----------------|-----------|----------------|------------------------------|--------|--------|----------|--|--|----------|-----------|------|----------------|-----------|--|--|------|-----------|------|-----------------|-----------|--|--|----------|-----------|------|----------------|-----------|--|--|---------------|------|----------|--------------------|-----------|--|--|--------|-----------|------|-------------------|-----------|--|--|-------|---------|----------------|----------------------------|-----------|--|--|-----------------|---------|----------------|------------------------------|
| | <table border="1"> <thead> <tr> <th>Field</th> <th>Key</th> <th>Extent</th> <th>Label</th> <th>Data Type</th> <th>Format</th> <th>Source</th> </tr> </thead> <tbody> <tr> <td>smp_key1</td> <td></td> <td></td> <td>Contract</td> <td>Character</td> <td>x(8)</td> <td>jc_job.job_num</td> </tr> <tr> <td>smp_cha_1</td> <td></td> <td></td> <td>Name</td> <td>Character</td> <td>x(8)</td> <td>jc_job.job_name</td> </tr> <tr> <td>smp_cha_2</td> <td></td> <td></td> <td>Location</td> <td>Character</td> <td>x(8)</td> <td>jc_job.l1_desc</td> </tr> <tr> <td>smp_dat_1</td> <td></td> <td></td> <td>Contract Date</td> <td>Date</td> <td>99/99/99</td> <td>jc_job.job_condate</td> </tr> <tr> <td>smp_cha_3</td> <td></td> <td></td> <td>Status</td> <td>Character</td> <td>x(8)</td> <td>jc_job.job_active</td> </tr> <tr> <td>smp_dec_1</td> <td></td> <td></td> <td>Costs</td> <td>Decimal</td> <td>>>>,>>>,>>9.99</td> <td>jc_job.RO_contractCosts^TD</td> </tr> <tr> <td>smp_dec_2</td> <td></td> <td></td> <td>Revenue To Date</td> <td>Decimal</td> <td>>>>,>>>,>>9.99</td> <td>jc_job.RO_ContractRevenue^TD</td> </tr> </tbody> </table> | Field | Key | Extent | 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.l1_desc | smp_dat_1 | | | Contract 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 To Date | Decimal | >>>,>>>,>>9.99 | jc_job.RO_ContractRevenue^TD |
| Field | Key | Extent | 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.l1_desc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| smp_dat_1 | | | Contract 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 To Date | Decimal | >>>,>>>,>>9.99 | jc_job.RO_ContractRevenue^TD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="button" value="Show All"/> <input type="button" value="Add"/> <input type="button" value="Copy"/> <input type="button" value="Insert"/> <input type="button" value="Up"/> <input type="button" value="Down"/> <input type="button" value="Open"/> <input type="button" value="Concurrent"/> <input type="button" value="Multi"/> <input type="button" value="Bulk:"/> <input type="button" value="Detail"/> <input type="button" value="Delete"/> <input type="button" value="C"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="checkbox"/> Advanced | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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

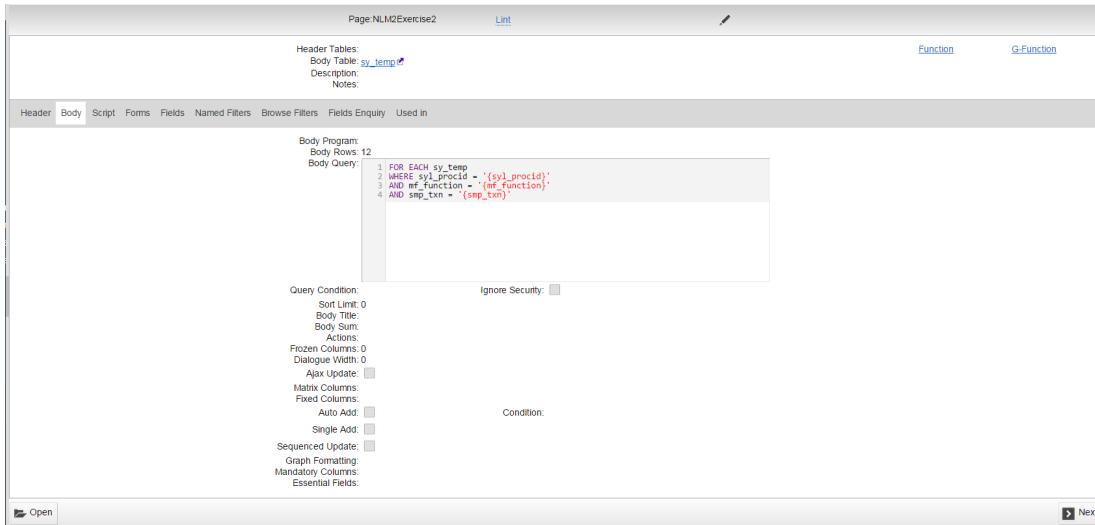
| Function | Description | Type | Module | Category |
|----------------|---------------------------|----------|--------------------|------------|
| NLM2Exercise2 | NL M2 Exercise 2 | Function | JC-Contract Status | REP-Report |
| NLM2Exercise2G | NL M2 Exercise 2 Generate | Function | JC-Contract Status | 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 Table jc_job

Body Query FOR EACH sy_temp WHERE syl_procid = '{syl_procid}' AND mf_function = '{mf_function}' AND smp_txn = '{smp_txn}'



```

FOR EACH sy_temp
WHERE syl_procid = '{syl_procid}'
AND mf_function = '{mf_function}'
AND smp_txn = '{smp_txn}'

```

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

Data Set Exercises



Contract Status | NL M2 Exercise 2

1 - COINS Construction | NIGLON

COINS

| 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 | 870,600.08 | 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 | M54 Northampton - Southampton | Overseas 00 | 01/01/05 | 21,942,729.91 | 22,442,061.92 | 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 | 3,925.00 | 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 |

Regenerate

Search: Contract ▾

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 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.

8.1 Creating a Contract Browse Screen

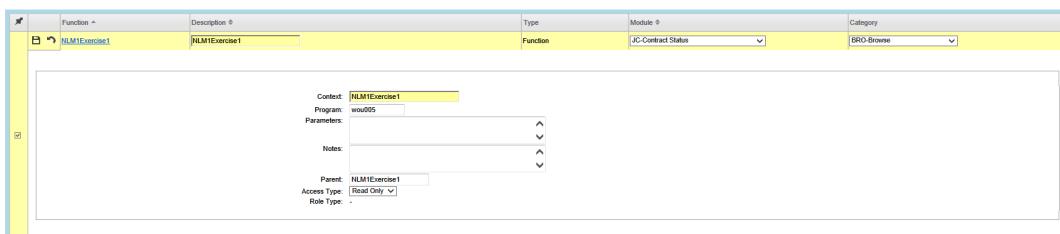
The first exercise is to create a simple contract enquiry screen. The screen will select contracts and return basic information from the jc_job record.

| Contract Status - NLM1Exercise1 | | | | | | | Niton COINS |
|---------------------------------|-------|----------------------|-------------|----------------|-------------------|--------|-------------|
| Contract Number | Phase | Name | Group | Type | Location | Status | |
| 0001 | N | Kingsley Meadows | Head Office | House Building | Oxfordshire | A | |
| 0002 | N | The Bullring | Head Office | House Building | Vale of Glamorgan | A | |
| 0003 | N | Uttwiler 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 | Millfields | Head Office | House Building | Warwickshire | A | |
| 1001 | N | Contractors Contract | Head Office | Commercial | Bristol | A | |
| 1010 | N | Bovis Test | Head Office | Commercial | Aberdeen City | A | |
| 1011 | N | New Contract | Head Office | Commercial | Aberdeen City | A | |
| 1012 | N | Draft Cert 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 | |
| 22788 | N | Test PDR | Head Office | Commercial | Aberdeen City | A | |
| 22789 | N | Test PDR | Head Office | Commercial | Aberdeen City | A | |

8.1.1 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 |



The screenshot shows the 'Function' creation dialog. The 'Function' field contains 'NLM1Exercise1'. The 'Module' dropdown is set to 'JC-Contract Status'. The 'Category' dropdown is set to 'BRO-Browse'. The 'Program' field contains 'wou005'. Below the main fields, there are sections for 'Context' (set to 'NLM1Exercise1'), 'Program' (set to 'wou005'), 'Parameters' (empty), 'Notes' (empty), and 'Reset' (set to 'NLM1Exercise1'). At the bottom, 'Access Type' is set to 'Read Only' and 'Role Type' is empty.

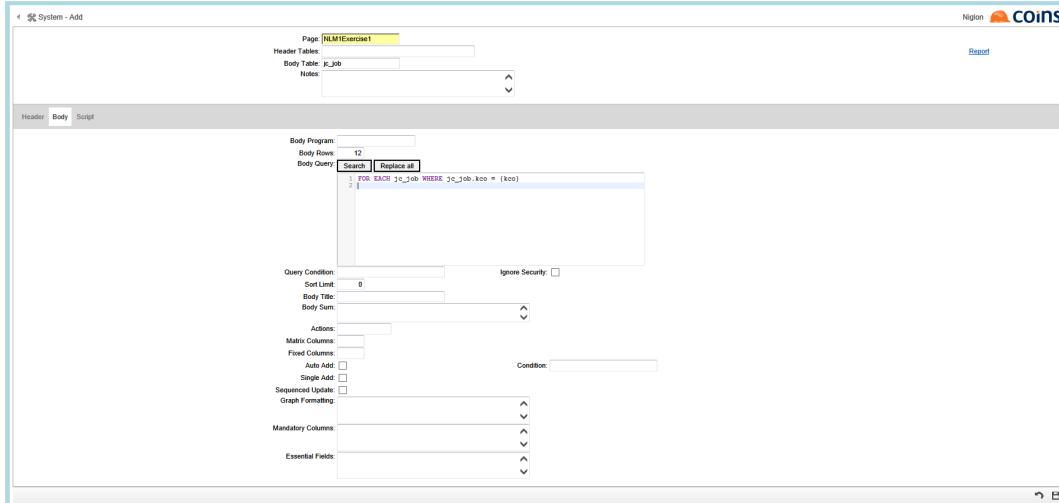
8.1.2 Create the Page

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

Enter the information into the Page Summary as follows:

| Field | Value |
|--------------|--|
| Header Table | jc_config,co_config |
| Body Table | jc_job |
| Body Rows | 12 |
| Body Query | FOR EACH jc_job WHERE jc_job.kco = {kco} |

 Click  Save to complete the Page Summary.



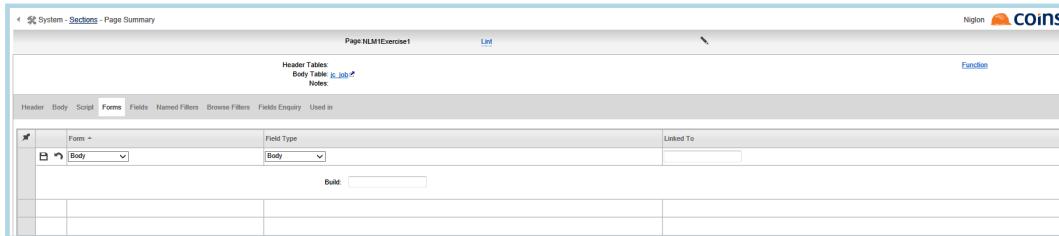
8.1.3 Create a Body Page Form

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

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

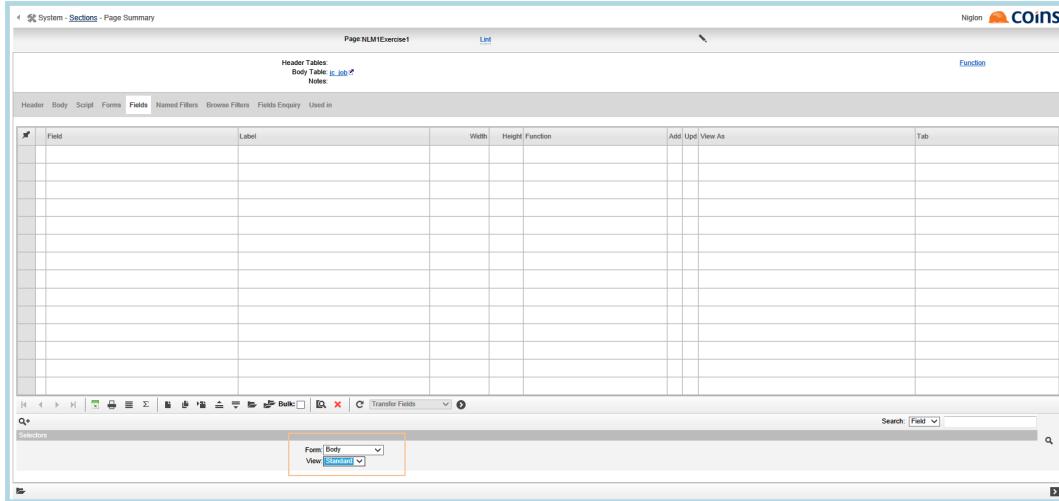
| Field | Value |
|------------|-------------|
| FormSelect | Body |
| Field Type | Body |
| Linked To | Leave blank |

Click the Save Button  to create the form.



8.1.4 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.

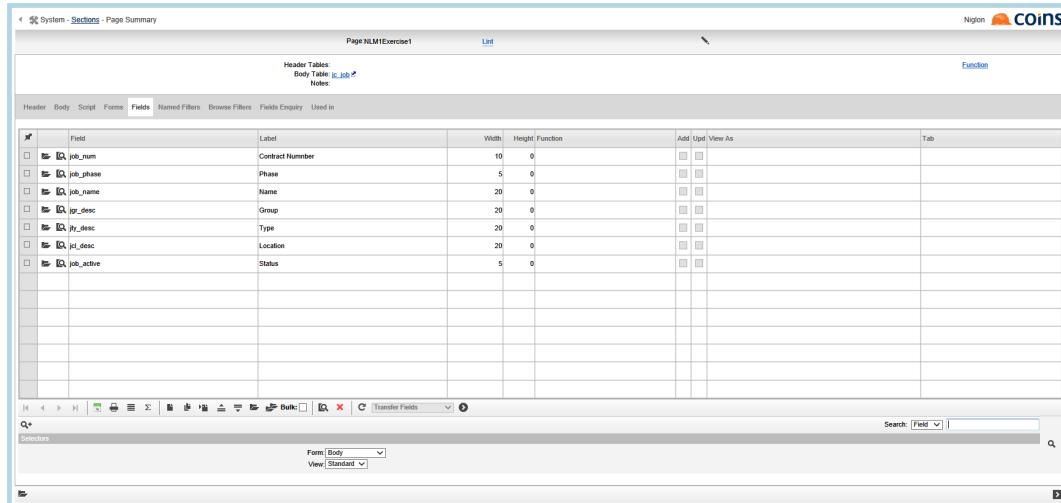


You should always do this to ensure you are adding fields to the correct form – adding fields to an incorrect form will mean that they do not display.

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 |
| 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.



Click Save after entering each field.

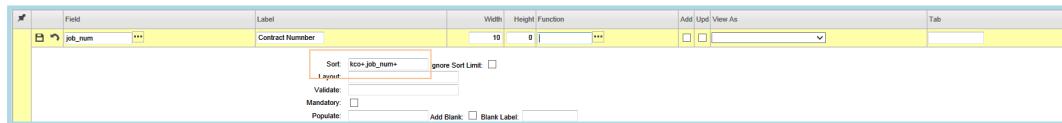
8.1.5 Adding a Sort to a Column

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+



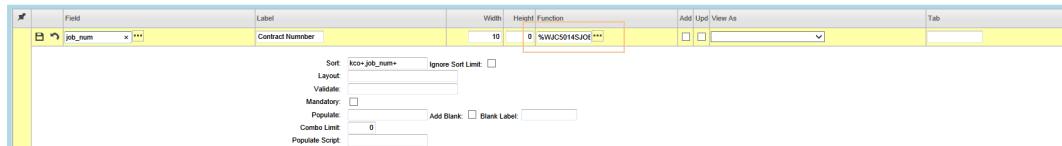
This will also provide a simple search filter to the Enquiry screen.

8.1.6 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:

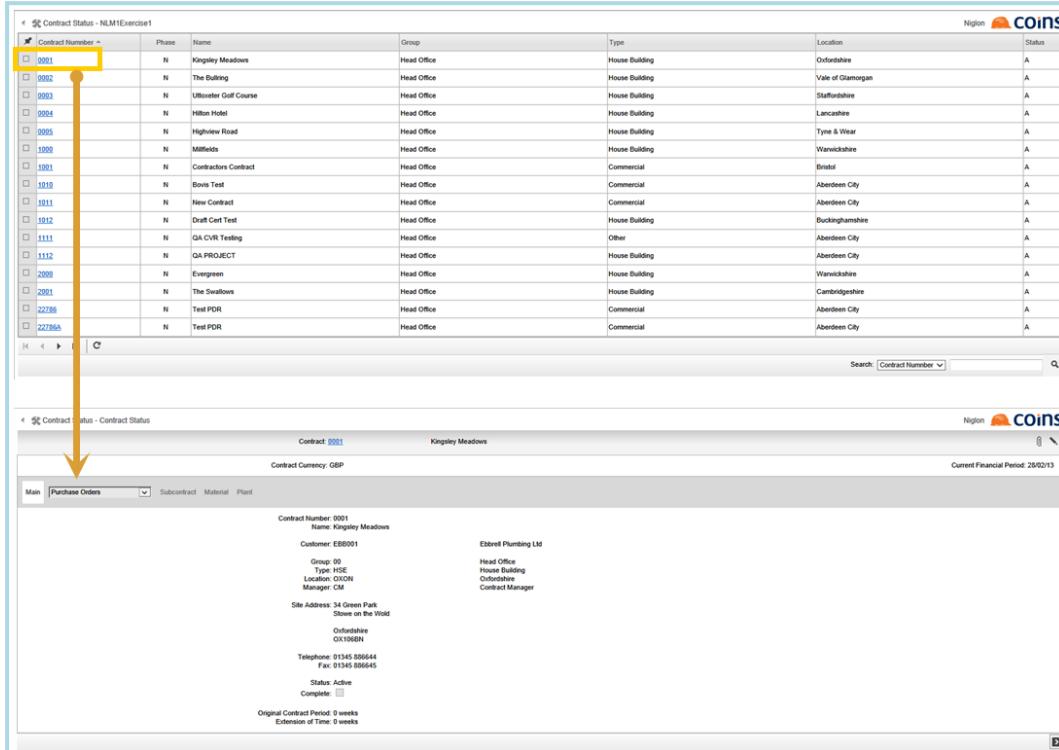
On the Function field enter **%WJC5014SJ0B** (this is a standard coins enquiry).



Save the field.

8.1.7 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 Bullring | 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 |
| 1000 | N | Miffields | Head Office | House Building | Warwickshire | A |
| 1001 | N | Contractors Contract | Head Office | Commercial | Bristol | A |
| 1010 | N | Bovis Test | Head Office | Commercial | Aberdeen City | A |
| 1011 | N | New Contract | Head Office | Commercial | Aberdeen City | A |
| 1012 | N | Draft Cert 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 |
| 22236 | N | Test PDR | Head Office | Commercial | Aberdeen City | A |
| 22236A | N | Test PDR | Head Office | Commercial | Aberdeen City | A |

| Contract Status - Contract Status | |
|--|------------------|
| Contract: 0001 | Kingsley Meadows |
| Contract Currency: GBP | |
| Main Purchase Orders Subcontract Material Plant | |
| Contract Number: 0001 Name: Kingsley Meadows Customer: EBB8001 Group: 00 Type: HSE Location: OXON Manager: CM Site Address: 34 Green Park Offsite Address: Kingsley Meadow OX10BN Telephone: 01345 886444 Fax: 01345 886445 Status: Active Complete: <input checked="" type="checkbox"/> <small>Original Contract Period: 8 weeks Extension of Time: 0 weeks</small> | |

8.2 Enhancing the Contract Browse Screen

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

8.2.1 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

| Data | |
|--|-------------------------------------|
| Populate: | <input type="text"/> |
| Total: | <input checked="" type="checkbox"/> |
| <input type="checkbox"/> Total <input type="checkbox"/> Count <input type="checkbox"/> Maximum <input type="checkbox"/> Average <input type="checkbox"/> Minimum <input type="checkbox"/> Distinct | |

| 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.

8.2.2 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 |

| Header | Body | Script | Forms | Fields | Named Filters | Browse Filters | Fields Enquiry | Used in |
|-------------------------------------|--------|--------|-------|--------|---------------|----------------|----------------|---------|
| | | | | | | | | |
| <input checked="" type="checkbox"/> | Form - | | | | | | | |
| <input type="checkbox"/> | Body | | | | | | | |
| <input type="checkbox"/> | 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.

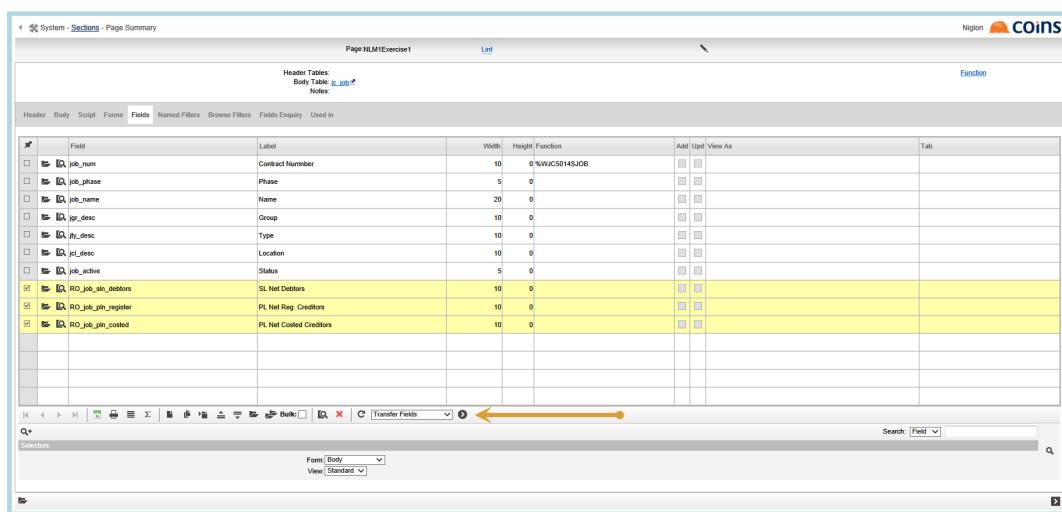
In the Fields selector at the bottom of the page, select Body and click Apply Filter.

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

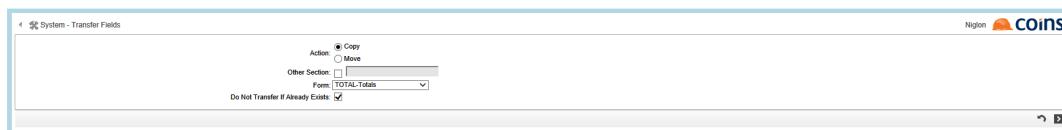
| Fields |
|---------------------|
| RO_job_sln_debtors |
| RO_job_pln_register |
| RO_job_pln_costed |

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 |
|----|---------------------|-------------------------|-------|--------|--------------|--------------------------|--------------------------|--------------------------|-----|
| 1 | RO_job_num | Contract Number | 10 | 0 | 0%WJC50HSJOB | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| 2 | RO_job_phase | Phase | 5 | 0 | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| 3 | RO_job_name | Name | 20 | 0 | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| 4 | RO_job_desc | Group | 10 | 0 | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| 5 | RO_pt_desc | Type | 10 | 0 | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| 6 | RO_pc_desc | Location | 10 | 0 | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| 7 | RO_pt_active | Status | 5 | 0 | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| 8 | RO_job_sln_debtors | SL Net Debtors | 10 | 0 | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| 9 | RO_job_pln_register | PL Net Reg Creditors | 10 | 0 | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| 10 | RO_job_pln_costed | PL Net Costed Creditors | 10 | 0 | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

Select Copy, and specify the Totals Form as the destination



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 | 15,980.00 | 17,545.00 |
| 0003 | N | Ullswater 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 | Warrickshire | 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 | Bova 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 Cet 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 |
| 2009 | N | Evergreen | Head Office | House Building | Warrickshire | A | 0.00 | 0.00 | 0.00 |
| 2001 | N | The Swallows | Head Office | House Building | Cambridgeshire | A | 0.00 | 0.00 | 0.00 |
| 22286 | N | Test PDR | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 0.00 |
| 22286A | N | Test PDR | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 0.00 |

Σ 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 | 15,980.00 | 17,545.00 |
| 0003 | N | Ullswater 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 |
| 1000 | N | Millets | Head Office | House Building | Warrickshire | 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 | Bova 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 Cet 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 |
| 2009 | N | Evergreen | Head Office | House Building | Warrickshire | A | 0.00 | 0.00 | 0.00 |
| 2001 | N | The Swallows | Head Office | House Building | Cambridgeshire | A | 0.00 | 0.00 | 0.00 |
| 22286 | N | Test PDR | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 0.00 |
| 22286A | N | Test PDR | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 0.00 |

8.2.3 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 |
| NLM1Exercise1X | M1 Exercise1 - Contract Listing | Function | JC-Contract Status | BRO-Browse |

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 Status - NLM1Exercise1 | | | | | | | | | Niglon COINS |
|---------------------------------|-------|-----------------------|-------------|----------------|-------------------|--------|----------------|-----------------------|-------------------------|
| 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 | 15,980.00 | 17,548.00 |
| 0003 | N | Ullswater 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 | Hightown Road | Head Office | House Building | Tyne & Wear | A | 0.00 | 750.00 | 3,700.00 |
| 1000 | N | Matfield | 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 Cert 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 |
| 22798 | N | Test PDR | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 0.00 |
| 22798A | N | Test PDR | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 0.00 |

8.3 Adding Sections to the Contract Browse Screen

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

8.3.1 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.

| Header | Body | Script | Forms | Fields | Named Filters | Browse Filters | Fields Enquiry | Used In |
|--------|------|--------|--|------------|---------------|----------------|----------------|-----------|
| | | | Form ▾ | | | | | |
| | | | | Field Type | | | | Linked To |
| | | | <input type="checkbox"/>  Body | Body | | | | |
| | | | <input type="checkbox"/>  Body Span | Body Span | | | | |
| | | | <input 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 |

| Layout |
|---|
| <p>Label Column Spans: <input type="text" value="0"/></p> <p>Column Spans: <input type="text" value="7"/> Row Spans: <input type="text" value="0"/></p> <p>Mandatory: <input type="checkbox"/> No Break Label: <input type="checkbox"/> Show in Help: <input checked="" type="checkbox"/></p> <p>Alignment: <input checked="" type="radio"/> Default <input type="radio"/> Left <input type="radio"/> Centre <input type="radio"/> Right</p> <p>Format: <input type="text"/></p> <p>Class: <input type="text"/></p> <p>Label Class: <input type="text"/></p> <p>Build: <input type="text"/></p> <p>Generate: <input type="text"/></p> |

Click Save  after adding each field.

Run the function:

| Contract Status - NI M1Exercise1 | | | | | | | | | |
|-------------------------------------|-----------------|------------------------|-------------|----------------|-------------------|----------|----------|-------------------|-------------------|
| Contract Details | | | | | | | | | |
| | Contract Number | Name | Type | Address | City | Postcode | Value | Value (Excl. VAT) | Value (Incl. VAT) |
| <input checked="" type="checkbox"/> | 0001 | N Kingsley Meadows | Head Office | House Building | Oxfordshire | A | 1,002.38 | 6,700.00 | 3,652.25 |
| <input type="checkbox"/> | 0002 | N The Buring | Head Office | House Building | Vale of Glamorgan | A | 690.00 | 15,900.00 | 17,548.00 |
| <input type="checkbox"/> | 0003 | N Ultzeler Golf Course | Head Office | House Building | Staffordshire | A | 0.00 | 0.00 | 1,092.50 |
| <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 Highview Road | Head Office | House Building | Tyne & Wear | A | 0.00 | 750.00 | 3,700.00 |
| <input type="checkbox"/> | 1009 | N Millfield | Head Office | House Building | Warwickshire | A | 0.00 | 0.00 | 3,512.00 |
| <input type="checkbox"/> | 1091 | N Contractors Contract | Head Office | Commercial | Bristol | A | 543.62 | 100.00 | 956.30 |
| <input type="checkbox"/> | 1019 | N Bovis Test | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 406.00 |
| <input type="checkbox"/> | 1011 | N New Contract | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 0.00 |
| <input type="checkbox"/> | 1912 | N Draft Cert 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"/> | 22786 | N Test PDR | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 0.00 |
| <input type="checkbox"/> | 22786A | N Test PDR | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 0.00 |

8.3.2 Adding a Body Detail Section

Edit the page section by clicking on the Open button .

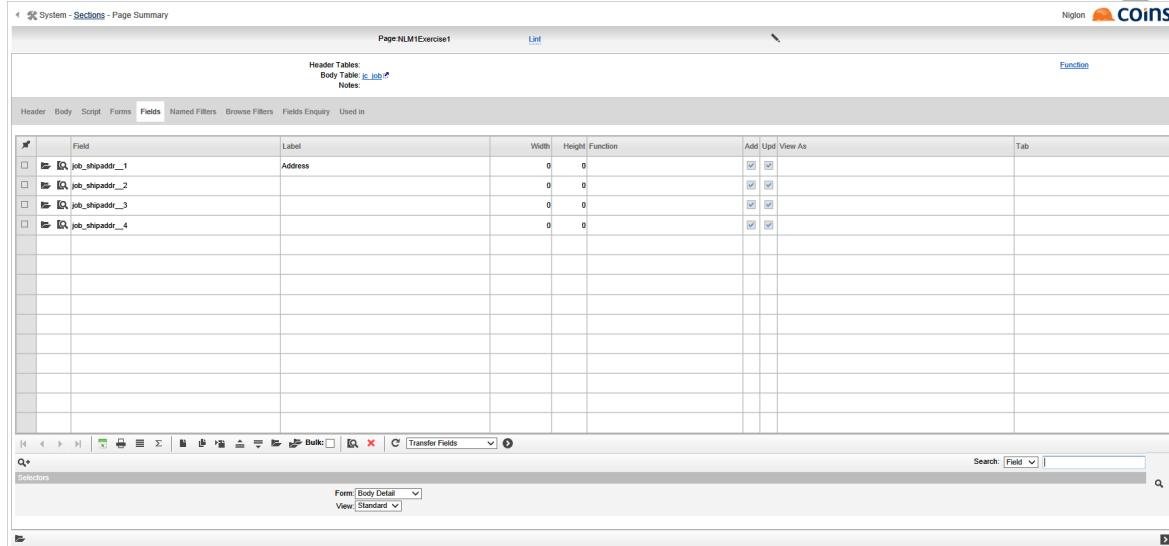
Select the Form tab create a Body Detail form to use Body Detail Fields.

| Header | Body | Script | Forms | Fields | Named Filters | Browse Filters | Fields Enquiry | Used in |
|-------------------------------------|---|--------|-------|--------|---------------|----------------|----------------|---------|
| <input checked="" type="checkbox"/> | Form ^ | | | | | | | |
| <input type="checkbox"/> |  Body | | | | | | | |
| <input type="checkbox"/> |  Body Detail | | | | | | | |
| <input type="checkbox"/> |  Body Span | | | | | | | |
| <input type="checkbox"/> |  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 :

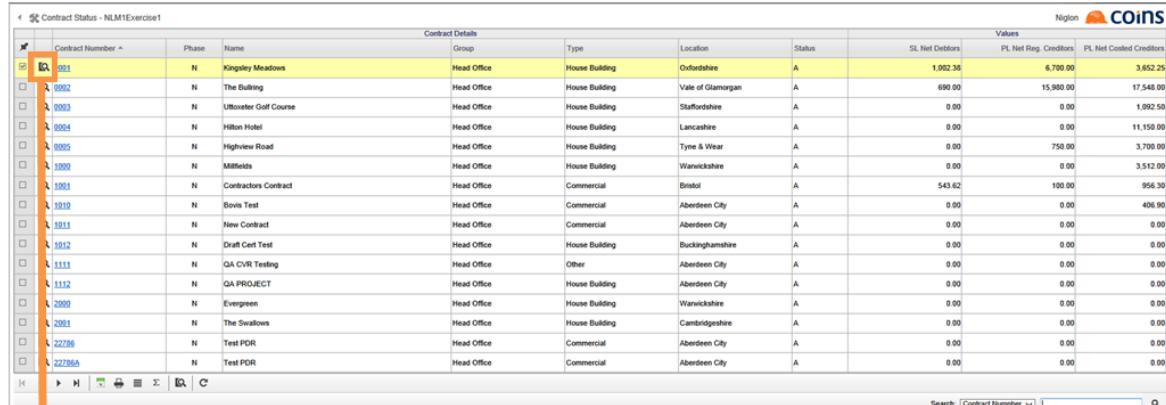
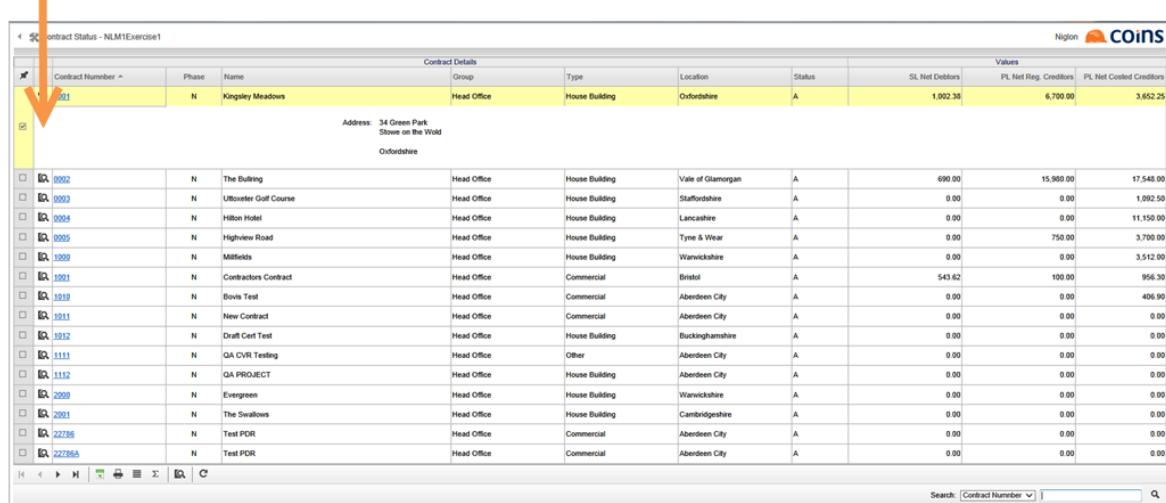
| Field | Label |
|----------------|---------|
| job_shipaddr_1 | Address |
| job_shipaddr_2 | (Blank) |
| job_shipaddr_3 | (Blank) |
| job_shipaddr_4 | (Blank) |



| # | Field | Label | Width | Height | Function | Add | Upd | View As | Tab |
|---|----------------|---------|-------|--------|----------|-------------------------------------|-------------------------------------|---------|-----|
| 1 | job_shipaddr_1 | Address | 0 | 0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | |
| 2 | job_shipaddr_2 | | 0 | 0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | |
| 3 | job_shipaddr_3 | | 0 | 0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | |
| 4 | job_shipaddr_4 | | 0 | 0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | |

Click Save  after adding each field.

Now run the function.

| Contract Details | | | | | | | | Values | | |
|-------------------------------------|-----------------|-------|-----------------------|-------------|----------------|-------------------|--------|----------------|-----------------------|-------------------------|
| | Contract Number | Phase | Name | Group | Type | Location | Status | SL Net Debtors | PL Net Reg. Creditors | PL Net Costed Creditors |
| <input checked="" type="checkbox"/> | 0001 | N | Kingsley Meadows | Head Office | House Building | Oxfordshire | A | 1,002.38 | 6,700.00 | 3,652.25 |
| <input type="checkbox"/> | 0002 | N | The Bulding | Head Office | House Building | Vale of Glamorgan | A | 690.00 | 15,980.00 | 17,548.00 |
| <input type="checkbox"/> | 0003 | N | Ullswater Golf Course | Head Office | House Building | Staffordshire | A | 0.00 | 0.00 | 1,992.50 |
| <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 | Highview Road | Head Office | House Building | Tyne & Wear | A | 0.00 | 750.00 | 3,700.00 |
| <input type="checkbox"/> | 0006 | N | Milfields | Head Office | House Building | Warwickshire | A | 0.00 | 0.00 | 3,512.00 |
| <input type="checkbox"/> | 0007 | N | Contractors Contract | Head Office | Commercial | Bristol | A | 543.62 | 100.00 | 956.30 |
| <input type="checkbox"/> | 0010 | N | Bovis Test | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 406.90 |
| <input type="checkbox"/> | 0011 | N | New Contract | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 0.00 |
| <input type="checkbox"/> | 0012 | N | Draft Cert 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"/> | 22786 | N | Test PDR | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 0.00 |
| <input type="checkbox"/> | 22786A | N | Test PDR | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 0.00 |

Search: Contract Number 

| Contract Details | | | | | | | | Values | | |
|--|-----------------|-------|-----------------------|-------------|----------------|-------------------|--------|----------------|-----------------------|-------------------------|
| | Contract Number | Phase | Name | Group | Type | Location | Status | SL Net Debtors | PL Net Reg. Creditors | PL Net Costed Creditors |
| <input checked="" type="checkbox"/> | 0001 | N | Kingsley Meadows | Head Office | House Building | Oxfordshire | A | 1,002.38 | 6,700.00 | 3,652.25 |
| Address: 34 Green Park Stow on the Wold Oxfordshire | | | | | | | | | | |
| | | | | | | | | | | |
| <input type="checkbox"/> | 0002 | N | The Bulding | Head Office | House Building | Vale of Glamorgan | A | 690.00 | 15,980.00 | 17,548.00 |
| <input type="checkbox"/> | 0003 | N | Ullswater Golf Course | Head Office | House Building | Staffordshire | A | 0.00 | 0.00 | 1,992.50 |
| <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 | Highview Road | Head Office | House Building | Tyne & Wear | A | 0.00 | 750.00 | 3,700.00 |
| <input type="checkbox"/> | 0006 | N | Milfields | Head Office | House Building | Warwickshire | A | 0.00 | 0.00 | 3,512.00 |
| <input type="checkbox"/> | 0007 | N | Contractors Contract | Head Office | Commercial | Bristol | A | 543.62 | 100.00 | 956.30 |
| <input type="checkbox"/> | 0010 | N | Bovis Test | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 406.90 |
| <input type="checkbox"/> | 0011 | N | New Contract | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 0.00 |
| <input type="checkbox"/> | 0012 | N | Draft Cert 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"/> | 22786 | N | Test PDR | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 0.00 |
| <input type="checkbox"/> | 22786A | N | Test PDR | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 0.00 |

Search: Contract Number 

8.3.3 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.

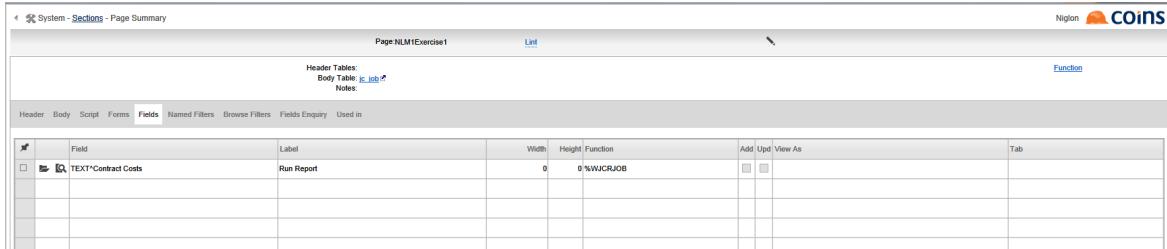
| Header | Body | Script | Forms | Fields | Named Filters | Browse Filters | Fields Enquiry | Used in |
|--------|--------------------------|---|-------------|------------|---------------|----------------|----------------|---------|
| | | | Form | Field Type | | | Linked To | |
| | <input type="checkbox"/> |  Body | Body | | | | | |
| | <input type="checkbox"/> |  Body Detail | Body Detail | | | | | |
| | <input type="checkbox"/> |  Body Span | Body Span | | | | | |
| | <input type="checkbox"/> |  Context | Context | | | | | |
| | <input type="checkbox"/> |  Totals | Totals | | | | | |

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:

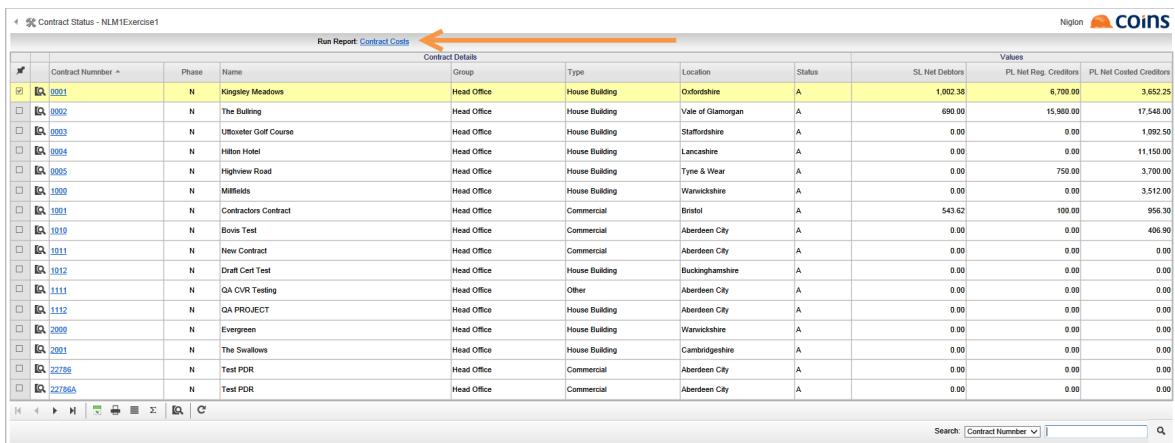
| Field | Label | Function |
|----------------------|------------|----------|
| Text^ Contract Costs | Run Report | WJCRJOB |

Click Save  after adding the field.



| Field | Label | Width | Height | Function | Add | Upd | View As | Tab |
|---|------------|-------|--------|----------|-----|-----|---------|-----|
| <input checked="" type="checkbox"/> TEXT*Contract Costs | Run Report | 0 | 0 | %WUCRJOB | | | | |

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



| Contract Details | | | | | | | | Values | | |
|------------------|-------|-----------------------|-------------|----------------|-------------------|--------|----------------|-----------------------|-------------------------|--|
| 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 Bulring | Head Office | House Building | Vale of Glamorgan | A | 690.00 | 15,980.00 | 17,548.00 | |
| 0003 | N | Uttorster 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 | |
| 1000 | N | Miffields | 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 Cert 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 | |
| 2226 | N | Test PDF | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 0.00 | |
| 2226A | N | Test PDF | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 0.00 | |

8.4 Adding Filters to the Contract Browse Screen

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

8.4.1 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 |

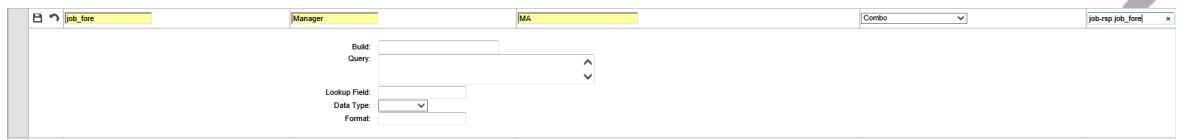
Click Save  after adding each field.

8.4.2 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_foreManagerMACombo 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.



8.5 Adding a Body Selector

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

8.5.1 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 Enquiry | Used In |
|--------------------------|---|--------|--------|---------------|---------------|----------------|----------------|-----------|
| | | | Form ^ | | | | | |
| <input type="checkbox"/> |    Body | | | Field Type | | | | Linked To |
| <input type="checkbox"/> |    Body Detail | | | Body | | | | |
| <input type="checkbox"/> |    Body Selector | | | Body Detail | | | | |
| <input type="checkbox"/> |    Body Span | | | Body Selector | | | | |
| <input type="checkbox"/> |    Context | | | Body Span | | | | |
| <input type="checkbox"/> |    Totals | | | Context | | | | |
| | | | | 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: <input type="text"/> Populate: <input type="text"/> glp-rsp.glp_fdate 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.

8.5.2 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.

| Fields | | | | | | | | | | | | |
|--------------------------|-------------------------------------|-----------------|-------|--------|---------------|-------------------------------------|-------------------------------------|---------|-----|--------|--------|--------|
| | Field | Label | Width | Height | Function | Add | Upd | View As | Tab | Layout | Append | Hidden |
| <input type="checkbox"/> | RO_job_num | Contract Number | 5 | 0 | 0%WJC5014SJ0B | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | | |
| <input type="checkbox"/> | RO_job_phase | Phase | 5 | 0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | | |
| <input type="checkbox"/> | RO_job_name | Name | 15 | 0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | | |
| <input type="checkbox"/> | RO_gr_desc | Group | 10 | 0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | | |
| <input type="checkbox"/> | RO_ty_desc | Type | 10 | 0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | | |
| <input type="checkbox"/> | RO_lc_desc | Location | 10 | 0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | | |
| <input type="checkbox"/> | RO_job_active | Status | 5 | 0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | | |
| <input type="checkbox"/> | RO_job_st_debtors | SL Debtors | 10 | 0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | | |
| <input type="checkbox"/> | RO_job_pln_register | PL Registered | 10 | 0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | | |
| <input type="checkbox"/> | RO_job_pln_costed | PL Costed | 10 | 0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | | |
| <input type="checkbox"/> | RO_ContractCosts^TD 0 {EndPeriod} | Costs to Date | 10 | 0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | | |
| <input 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.

| Run Report: Contract Costs | | | | | | | | | | | | |
|----------------------------|-------|------------------------|-------------|----------------|-------------------|--------|----------------|-----------------------|-------------------------|----------|----------|--------|
| Contract Details | | | | | | | | | | | | Values |
| 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 Bulring | Head Office | House Building | Vale of Glamorgan | A | 690.00 | 15,980.00 | 17,548.00 | 0.00 | 0.00 | |
| 0003 | N | Uttlesford 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 | Hightown Road | Head Office | House Building | Tyne & Wear | A | 0.00 | 750.00 | 3,700.00 | 0.00 | 0.00 | |
| 1000 | N | Milfields | Head Office | House Building | Warwickshire | 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 Cart 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 | |
| 2009 | N | Evergreen | Head Office | House Building | Warwickshire | 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 | |
| 22786 | N | Test PDR | Head Office | Commercial | Aberdeen City | A | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | |
| 22786A | N | Test PDR | 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.

9 A Simple listing Report

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 |
| ABB001 | Abbey Glass | 42 Bramall Lane | | Sheffield | S25 4DL |
| AGG001 | Aggregate Supplies | 22 Ashton Gates | | Bristol | BS30 5SJ |
| BOD001 | BOD Plumbing & Heating Supplies | 12 High Row | | Darlington | DL37QQ |
| BR001 | British Gas | 14 Jesmond road | | Newcastle | NE11 |
| CHA001 | Chase Flooring Ltd | 145 The Green | | Penneth | CA10 |
| CHI001 | Cheshire Restorations | 60 Broad St | Chesham | | HP5 3EF |
| CTT001 | City and County Group | Test change of address | Warley | | BW2 2HD |
| CTT002 | City and County Group | Bentfield Place | Bentfield Road | Stansted | CB64 8HL |
| COR001 | Cornwall Builders | 2 The Avenue | Smith Street | Cambridge | CB6 2DS |
| COR002 | Cornwell Builders | Covent Garden | London | | L1 1OP |
| DAV001 | Davies Engineers Ltd | 4 Briar Hill | Woolpit | Suffolk | IP309SD |
| EBB001 | Ebbrell Plumbing Ltd | 34 Goodison Way | | Liverpool | L19 4EZ |
| ELU003 | Ellbee Limited | Oldbury | | | WV16 5FF |
| EMA001 | EB Emails Limited | Email House | Bridgnorth | Shropshire | WV16 5RR |
| GRA001 | Gray Builders Ltd | 12 Kingston Road | Kingston | Surrey | KT15 2LK |
| HIG001 | Highway Paints | 78 Anfield Road | | Liverpool | L1 |
| JON001 | Jones Lintel Supplies Limited | 87 Lowdown Road | | Exeter | EX2 |
| LLO001 | Lloyds TSB | 34 The Grove | | Slough | SL1 |
| POP001 | Popes Builders Merchants Ltd | 2 Gren Avenue | | Chislehurst | BR7 6BT |
| SMI001 | Smith Ebbrell & Vaux Solicitors | 1 St James park | | Newcastle | NE11 |
| SPE001 | Speedy Hire Centres (Western) Ltd | 22 Mace road | | Bristol | BS2 0TX |
| SUT001 | Sutton Electrics Ltd | 56 Old Trafford Way | | Warrington | WA6 0DB |
| Z_Test | Z_Test | Spb | | | CR9 1JT |

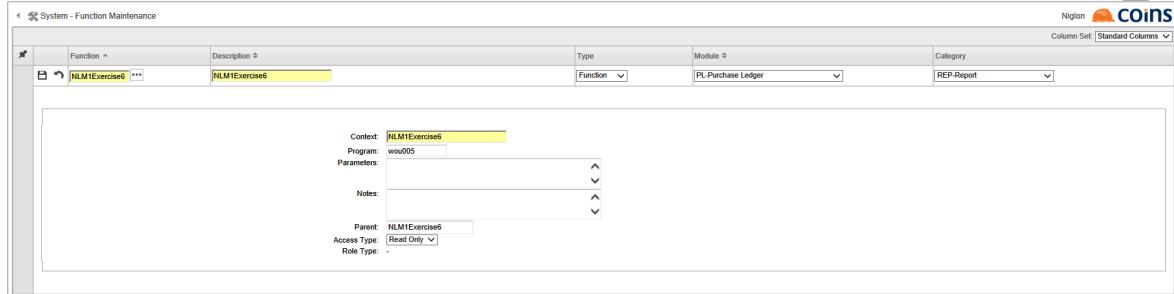
Printed using QA COINS V10.27.121201 by Nigel Longley at 11:35:23 on 14/05/13 (NLM1Exercise6)

Page 1

9.1 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 |

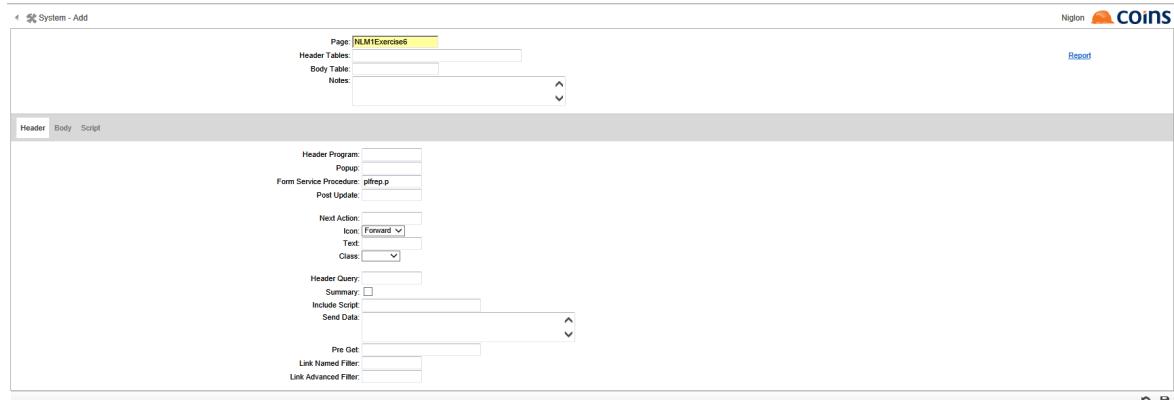


The screenshot shows the 'System - Function Maintenance' interface. The 'Function' dropdown is set to 'NLM1Exercise6'. The 'Description' field contains 'NLM1Exercise6'. The 'Type' dropdown is set to 'Function'. The 'Module' dropdown is set to 'PL-Purchase Ledger'. The 'Category' dropdown is set to 'REP-Report'. The 'Column Set' dropdown is set to 'Standard Columns'.

9.2 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' interface for creating a new page section. The 'Page' dropdown is set to 'NLM1Exercise6'. The 'Header Table' dropdown is empty. The 'Body Table' dropdown is empty. The 'Notes' field is empty. The 'Header Program' dropdown is empty. The 'PopUp' dropdown is empty. The 'Form Service Procedure' dropdown is set to 'plfrep.p'. The 'Post Update' dropdown is empty. The 'Next Action' section is empty. The 'Header Query' section is empty. The 'Pre Get' dropdown is empty. The 'Link Named Filter' dropdown is empty. The 'Link Advanced Filter' dropdown is empty.

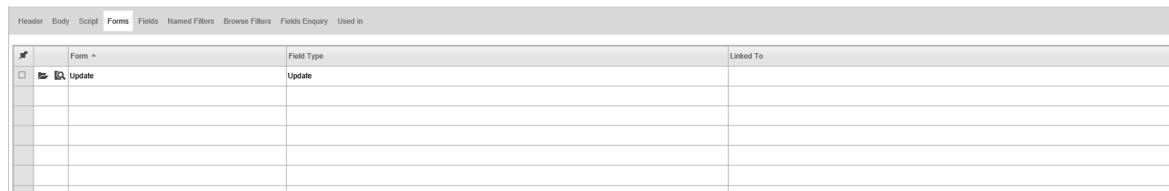
Click Save  to save the new Page. COINS returns to the summary for the page section being created.

9.3 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.

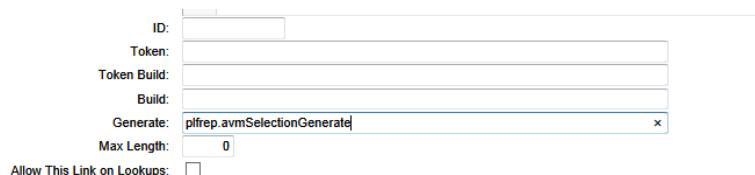
9.4 Add the Fields

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.

```

Start Time:14/05/2013 11:23:55.201+01:00
Database User :5
Report Process:28821
Report does not exist: nlmlexercise6

Display detail info?
End Time:14/05/2013 11:23:55.207+01:00
Timings: Startup- ? Generate- ? Prepare- ? Production- ? PDF- ? Completion- ? Total- 6

```

9.5 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 | Select %A4RLAND-A4 Report Landscape |
| Class | 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:

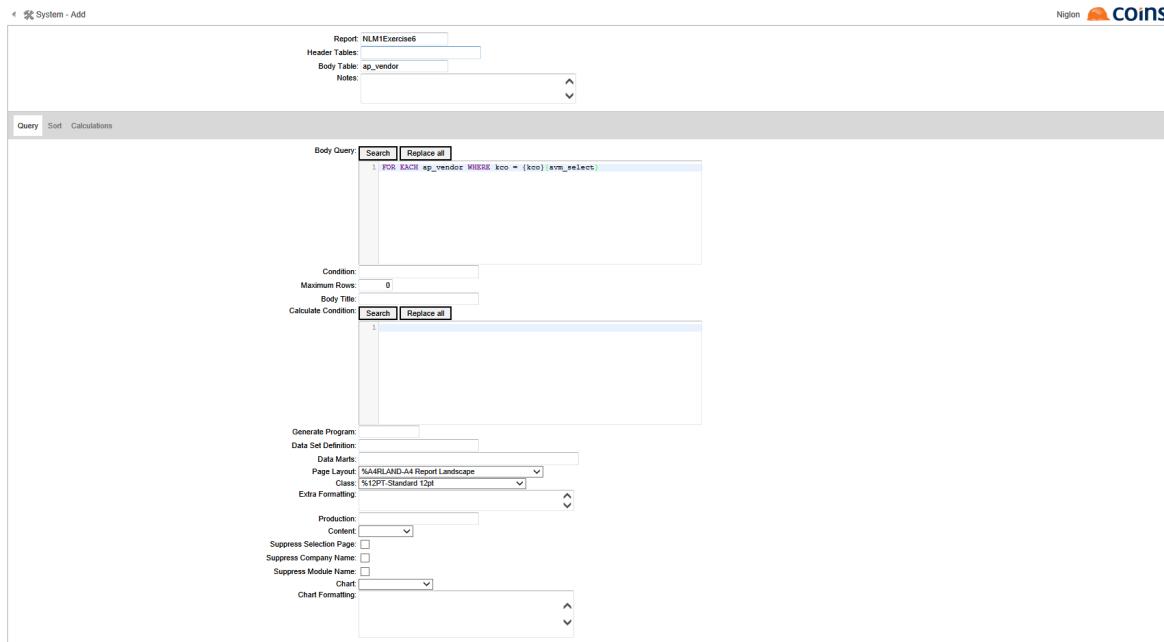
```
FOR EACH ap_vendor WHERE TRUE
{avmSelect}
```



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).



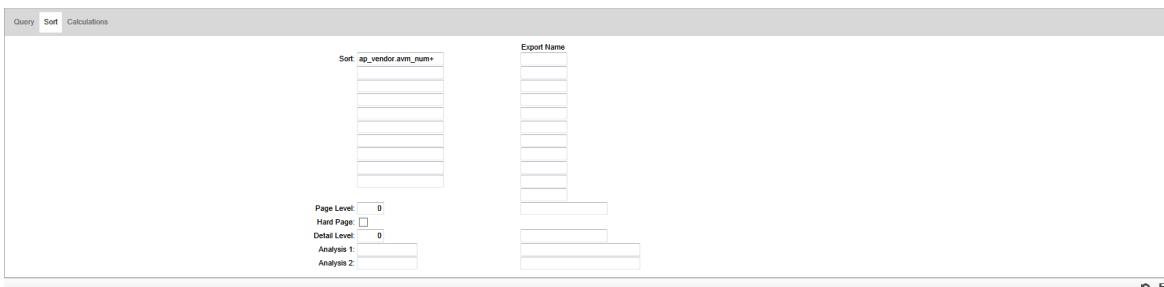
The screenshot shows the 'System - Add' interface for creating a new report named 'NLM1Exercises6'. The 'Query' tab is selected, displaying a SQL query:

```
1 FOR EACH ap_vendor WHERE kco = (kco) avm_select;
```

Below the query, there are fields for 'Condition', 'Maximum Rows', and 'Body Title'. Under 'Calculate Condition', there is a search bar. The 'Page Layout' is set to '%A4RLAND-A4 Report Landscape' and 'Class' to '%12P1-Standard 12pt'. The 'Sort Definition' section contains a 'Sort' field with 'ap_vendor.avm_num+' and an 'Export Name' field.

9.6 Sorting the Report

On the sort tab, enter ap_vendor.avm_num+ in the Sort field and Click Save  . COINS returns to the summary for the report section being created.



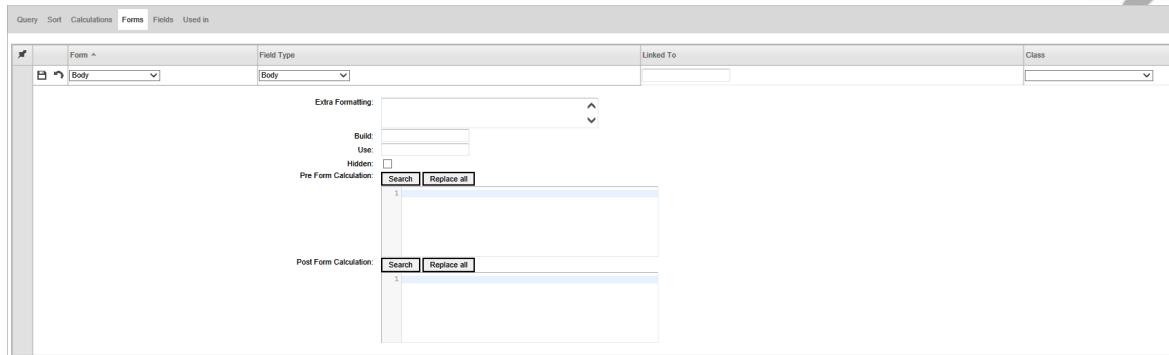
The screenshot shows the 'Sort' tab of the report configuration. The 'Sort' field contains 'ap_vendor.avm_num+'. The 'Page Level' is set to 0, and 'Hard Page' and 'Detail Level' are both checked. There are also fields for 'Analysis 1' and 'Analysis 2'.

9.7 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.

9.8 Add the Fields

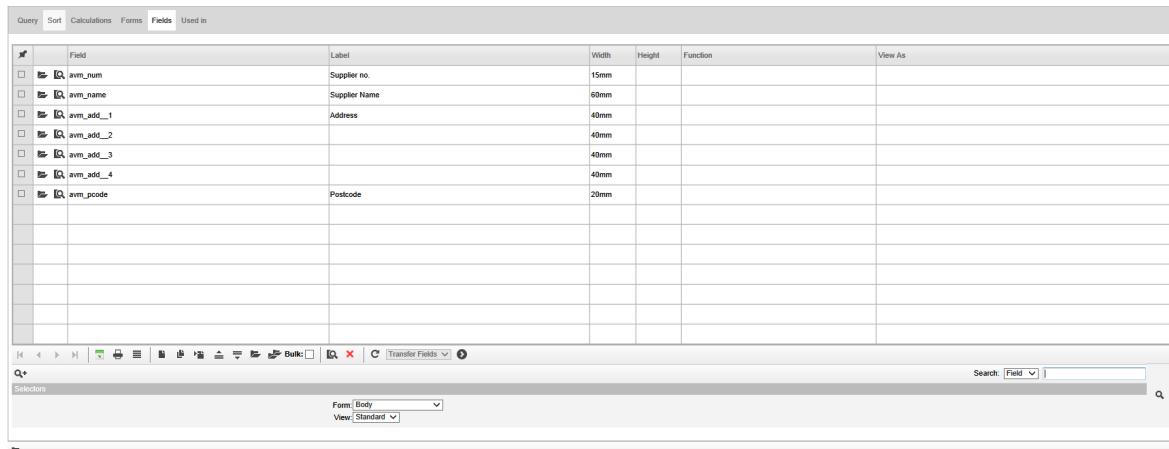
In the Field Tab go to the FORM selector at the bottom of the page, select Body and click Apply Filter.



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.

9.9 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 short-cut 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



| 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 | NE04 4TP | |
| ABB001 | Abbey Glass | 15 Ryton Road | North Anston | Sheffield | S25 4DL | |
| ABB002 | Abbey Glass xxx | 15 Ryton Road | North Anston | Sheffield | S25 4DL | |
| ABC002 | A-B-C Ltd | 14 Apsley Road | BRIGHTON | | BH7 4QG | |
| AG0001 | Aggregate Supplies | 33 St Andrews Road | Knowle | Bristol | BS30 6SJ | |
| AHD001 | A&H Demolition Services | 212 Oxford Road | Wythenshawe | Manchester | M32 6FR | |
| ARN001 | Arnold Laver & Company Limited | Bramall Lane | SHEFFIELD | | S24RJ | |
| ASH001 | Ashford Plant PLC | Wharfside | Ashford | Kent | TN23 1AA | |
| ATS001 | ATS Plumbing & Heating Supplies | 35 Lord Lane | Falsworth | Manchester | M35 3DN | |
| BR001 | Bristol Builders Supplies Ltd | Unit 43 | Orchard Court | Great Western Business Park | Bristol | BS37 5GW |
| BR1003 | British Telecom | Thames Valley office | 120 Riverside Way | Slough | Berkshire | SL2 7GF |
| BUD001 | Budd & Bird Ltd | Parkinson Avenue | Scunthorpe | South Humberside | DN15 7NA | |
| C&S001 | C&S Builders Merchants Ltd | 97, Woodside Avenue | Chislehurst | Kent | BR7 6BT | |
| CAR001 | Carter Supplies | 14 The Grove | Slough | Berkshire | SL1 1QQ | |
| CLA002 | T. Clarke | 116-118 Walworth Road | London | | SE17 1JY | |
| COI001 | COINS Construction | 12 The Grove | Slough | | SL1 8DJ | |
| CON001 | Condron Concrete | Arden Road | Tullamore | Co Athlone | AB | |
| COS001 | Costello Engineering | Tallaght | Dublin 24 | | | |
| DAI001 | Dairygold Ltd | Mallow | Co Cork | | | |
| DUB001 | Dublin Providers Ltd | Bathroom World co 01 | Old Kilmainham | Dublin 8 | SL1 | |
| EIR001 | Eircorn | East Port Business Park | Fairview | Dublin 3 | SL1 | |
| FAC001 | Factored Supplier | 150 High Street | Slough | Berkshire | SL1 7CD | |
| FIN001 | Financial Collections Ltd | PO Box 200 | Gosforth | Newcastle | NE18 5RD | |
| GAR001 | Garratt Timber Supplies | Unit 2 54 Kimber Road | Wandsworth | London | SW18 4PP | |
| god001 | Goffrey Exports Limited | | | timdrake | DY13 9GH | |
| GO0002 | Goode Concrete | Unit 1 A | Naas Rd Ind Est | Naas Rd | D22 | |
| GRE001 | Greenwood Trading | 15-25 Cathedral Park | Guildford | Surrey | GU12 7TH | |
| GSD001 | GSD Repairs | 123 Chertsey Road | Chertsey | Surrey | KT16 0GA | |
| HAN001 | Handley Building Materials UK | Reading Road | Newbury | Berkshire | RG14 7IG | |
| HEI001 | Heiton Buckley | 123a Naas Road Ind Estate | Naas Road | | D22 | |
| HOB001 | Hobson International | 1 Main Street | Chipping Sodbury | | BS23 6JL | |
| HOB002 | Hobson Limited | 1 High Street | Yate | | BS23 6JL | |
| HOW001 | Aaron Howard Fencing | Long Lane Farm | Padworth Road | Chelmsford | CM2 3GF | |
| IBS001 | Ibsworth Brick Limited | Ibsworth Nottingham Road | Ibsworth | Nottinghamshire | NG10 8PB | |
| 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 | |
| O'B001 | William O'Brien Plant | Atherry | Co Galway | | W3 9AW | |
| PAT001 | Pat Collins | 6 Illian Avenue | | | M11 1AA | |
| PLU001 | Plumb Centre | 119 Ashton Road | Manchester | | | |

Printed using COINS V10.23.110802+ by Nigel Longley at 11:43:58 on 03/09/11 (NL/M1Exercise6)

Page 1

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

| Report Status | Saved Reports | Schedulers | Report Runner | Archived Reports | Status | Queue | Size | User | | |
|-------------------------------------|----------------|----------------|---------------|---------------------------|----------|---------|---------------|---|---|--|
| <input checked="" type="checkbox"/> | Date: 30/04/13 | Time: 16:00:51 | Co: 1 PL | Description: nlmExercise6 | Complete | GENERAL | 17,139 NIGLON |   |   | |
| <input type="checkbox"/> | | | | | | | | | | |

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

10 A Contract Report

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

| Contract Listing | | | | | | |
|---|-----------------------------|------|--------------------------------|-------------------|------------------|--------------------|
| | | | Location | Costs to Date | Revenue to Date | Profit |
| Group: Build - Commercial Building | Code | Name | | | | |
| Type: CompTen - Competitively Tendered | | | | | | |
| 1006 | Slough Housing Estate | | West London | 26.88 | 0.00 | -26.88 |
| Type: CompTen - Competitively Tendered | | | | 26.88 | 0.00 | -26.88 |
| Group: Build - Commercial Building | | | | 26.88 | 0.00 | -26.88 |
| Group: Housing - LA & Housing Assoc Housing | | | | | | |
| 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 | | | | 58,044.83 | 67,952.25 | 9,907.42 |
| Type: SW - Small Works | | | | | | |
| 1002 | Xheltered Housing Cambridge | | East Anglia (Peterborough) | 0.00 | 0.00 | 0.00 |
| Type: SW - Small Works | | | | 0.00 | 0.00 | 0.00 |
| Group: Housing - LA & Housing Assoc Housing | | | | 58,044.83 | 67,952.25 | 9,907.42 |
| Group: LA - Local Authority general | | | | | | |
| 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 | | | | 263,908.44 | 85,897.69 | -178,010.75 |
| Group: LA - Local Authority general | | | | 263,908.44 | 85,897.69 | -178,010.75 |

10.1 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.

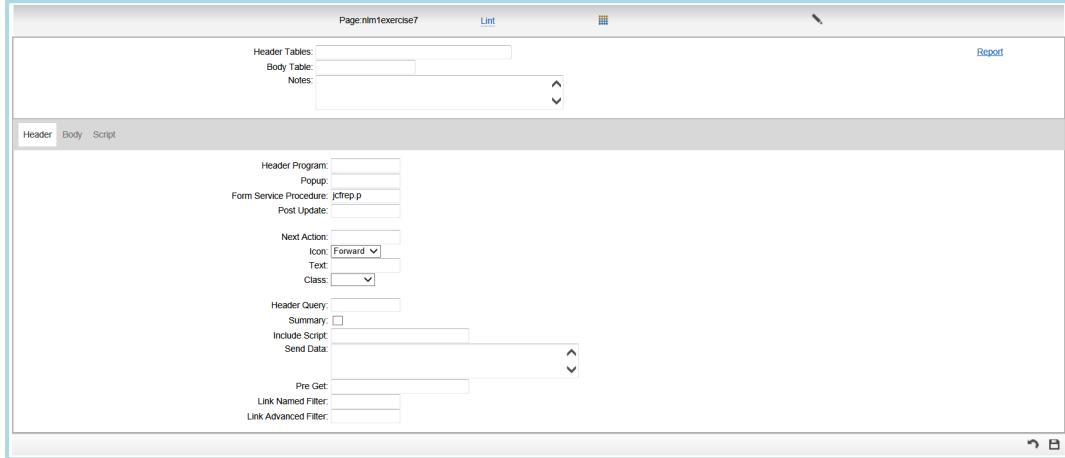
For example:

Function Name = NLM1Exercise7
 Tab Function = NLM1Exercise7T

| | Function | Description | Type | Module | Category |
|--------------------------|---|---------------|----------|--------------------|------------|
| <input type="checkbox"/> |   nlmExercise7 | nlmExercise7 | Function | JC-Contract Status | REP-Report |
| <input type="checkbox"/> |   nlmExercise7T | nlmExercise7T | Function | JC-Contract Status | REP-Report |

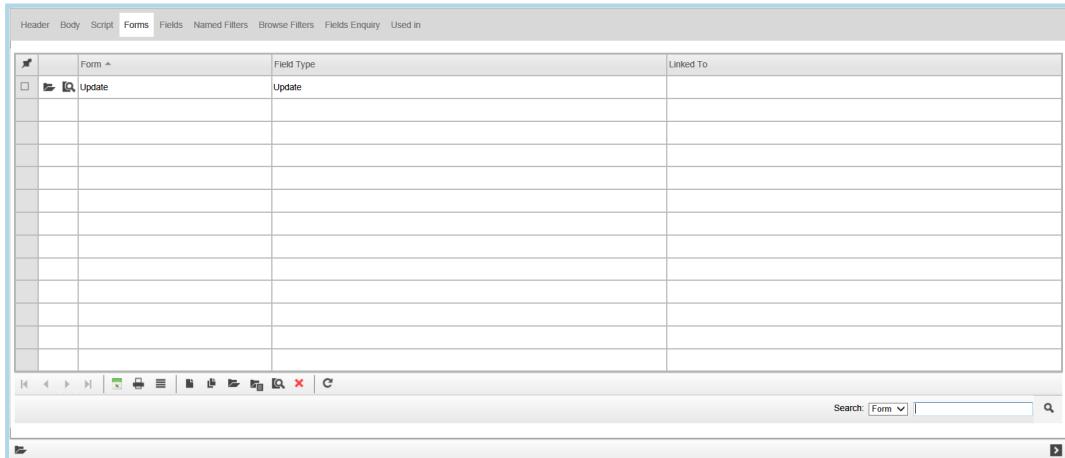
10.2 Create the Page

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



10.3 Add the Page Section Forms

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



Click the Save Button to create the form.

10.4 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 jcfrp.jobSelectionGenerate.

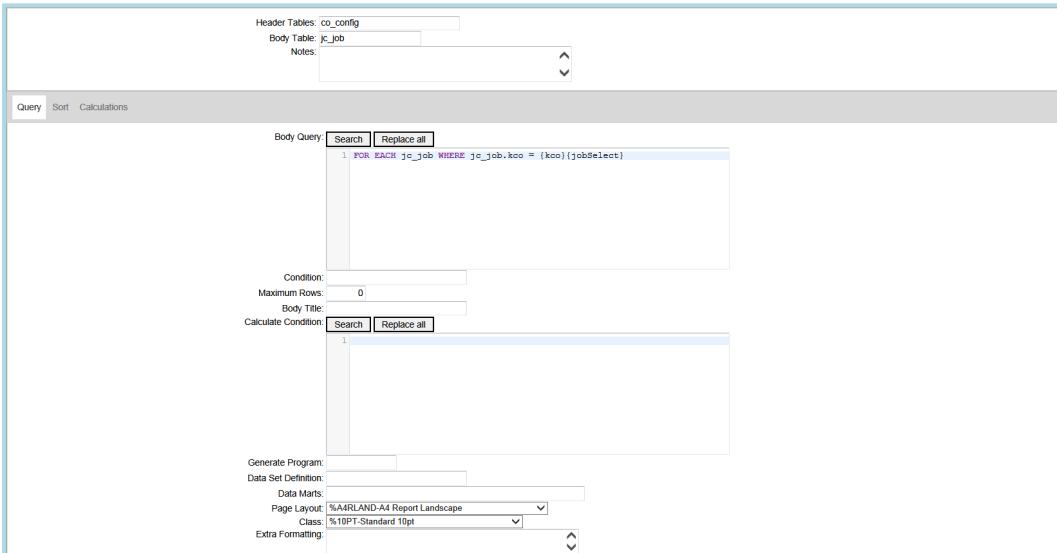
| | |
|---------------------|--|
| Label Column Spans: | <input type="text" value="0"/> |
| Column Spans: | <input type="text" value="0"/> Row Spans: <input type="text" value="0"/> |
| Mandatory: | <input type="checkbox"/> No Break Label: <input type="checkbox"/> Show in Help: <input checked="" type="checkbox"/> |
| 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="jcfrp.jobSelectionGenerate"/> |

10.5 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>FOR EACH jc_job WHERE jc_job.kco = {kco}{jobSelect}</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 |



The screenshot shows a software interface for report configuration. At the top, there are fields for 'Header Tables' (co_config), 'Body Table' (jc_job), and 'Notes'. Below this is a toolbar with 'Query', 'Sort', and 'Calculations' buttons. The main area is titled 'Body Query' and contains a code editor with the following SQL-like query:

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

Below the code editor are sections for 'Condition', 'Maximum Rows' (set to 0), 'Body Title', and 'Calculate Condition'. Further down are sections for 'Generate Program', 'Data Set Definition', 'Data Marts', 'Page Layout' (set to '%A4RLAND-A4 Report Landscape'), 'Class' (set to '%10PT-Standard 10pt'), and 'Extra Formatting'.

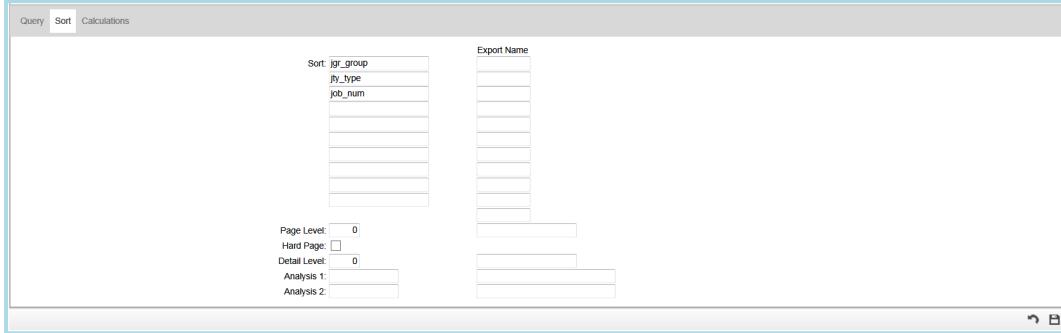
10.6 Sorting the Report

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.



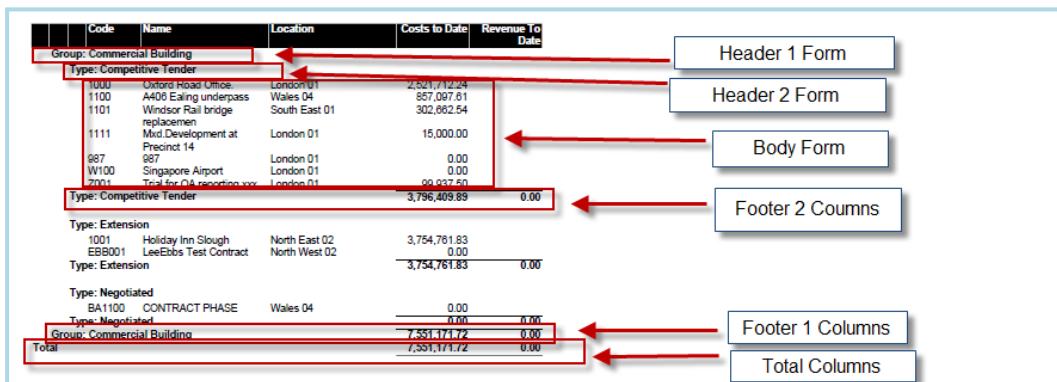
The screenshot shows a software interface for report configuration. At the top, there are tabs for 'Query', 'Sort', and 'Calculations'. Below these are sections for 'Sort' (with fields for 'jgr_group', 'jty_type', and 'job_num') and 'Export Name' (with a list of columns). There are also sections for 'Page Level' (with '0' selected), 'Hard Page' (unchecked), 'Detail Level' (with '0' selected), 'Analysis 1', and 'Analysis 2'. At the bottom right of the interface is a small window with a close button.

Click Save

10.7 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. |

10.8 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:

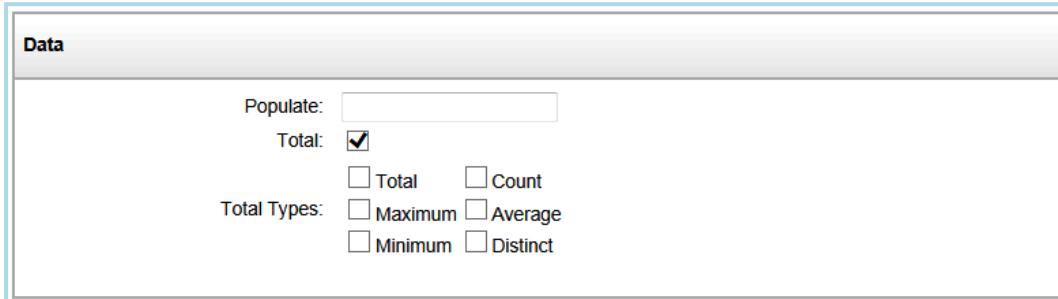


| Field | Label | Width |
|-----------------------|-----------------|-------|
| job_num | Code | 15mm |
| job_name | Name | 35mm |
| jcl_desc | Location | 35mm |
| RO_ContractCosts^TD | Costs to Date | 20mm |
| RO_ContractRevenue^TD | Revenue to Date | 20mm |

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

10.9 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.



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.

10.10 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 | Orford Head Office | London 01 | | 2,521,712.24 | |
| 1100 | A404 Ealing underpass | Wales 04 | | 657,067.61 | |
| 1101 | Windsor Rail bridge replacement | South East 01 | | 302,662.54 | |
| 1111 | Mxd Development at Precinct 14 | London 01 | | 15,000.00 | |
| 987 | 987 | London 01 | | 0.00 | |
| W100 | Singapore Airport | London 01 | | 0.00 | |
| 2001 | Thal for OA reporting xxx | London 01 | | 99,037.50 | |
| | | | | 3,796,409.89 | 0.00 |
| Type: Competitive Tender | | | | | |
| 1001 | Holiday Inn Slough | North East 02 | | 3,754,761.83 | |
| EBB001 | LeeEdds Test Contract | North West 02 | | 0.00 | |
| | | | | 3,754,761.83 | 0.00 |
| Type: Extension | | | | | |
| 1001 | Holiday Inn Slough | Wales 04 | | 0.00 | |
| EBB001 | LeeEdds Test Contract | | | 0.00 | |
| | | | | 0.00 | 0.00 |
| Type: Negotiated | | | | | |
| BA100 | CONTRACT PHASE | | | 0.00 | |
| | | | | 0.00 | 0.00 |
| Type: Negotiated | | | | | |
| | | | | 7,551,171.72 | 0.00 |
| Group: Commercial Building | | | | | |
| | | | | 7,551,171.72 | 0.00 |
| | | | | Total | 7,551,171.72 |

Printed using COINS V10.23.110502+ by Nigel Longley at 12:35:56 on 02/09/11 (NLM1\COINS7)

Page 1

10.11 Summarising the Contract Report

10.11.1 Copy the existing Example 7 Report

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 Exercise 7 and call them xxM1Exercise8 (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.

10.11.2 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.

10.11.3 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



| | 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 | | | | | |
| Z001 | Trial for OA reporting xxx | London 01 | | 3,796,409.89 | 0.00 |
| EBB001 | LeeEbs 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 | Blueswater shopping M & E | Overseas 00 | | 21,881,437.65 | 0.00 |
| TIM999 | Tim D Test | London 01 | | 0.00 | 0.00 |
| 1103 | Heathrow 5 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 | T222 | North West 02 | | 0.00 | 0.00 |
| 6000 | FM Contract | South West 04 | | 400.00 | 0.00 |
| Group: Commercial Fitout | | | | 555,629.23 | 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 |
| FSSO | FSS Overheads | South West 04 | | 0.00 | 0.00 |
| 6200 | A465 Vale of Neath Bridge | South West 04 | | 395.00 | 0.00 |
| Hinwa | | | | | |
| Group: Miscellaneous | | | | 395.00 | 0.00 |

—

10.11.4 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.

10.12 Calculated Fields

10.12.1 Copy the existing Example 7 Report

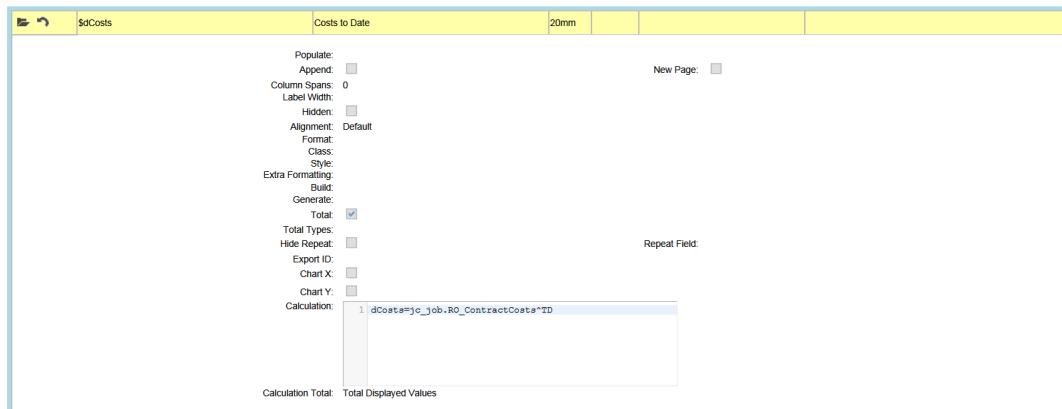
Copy each of the Function, Page and Report Sections of the example created in Exercise 7 and call them xxM1Exercise9 (where xx = your initials) and add the new function to the menu.

10.12.2 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.

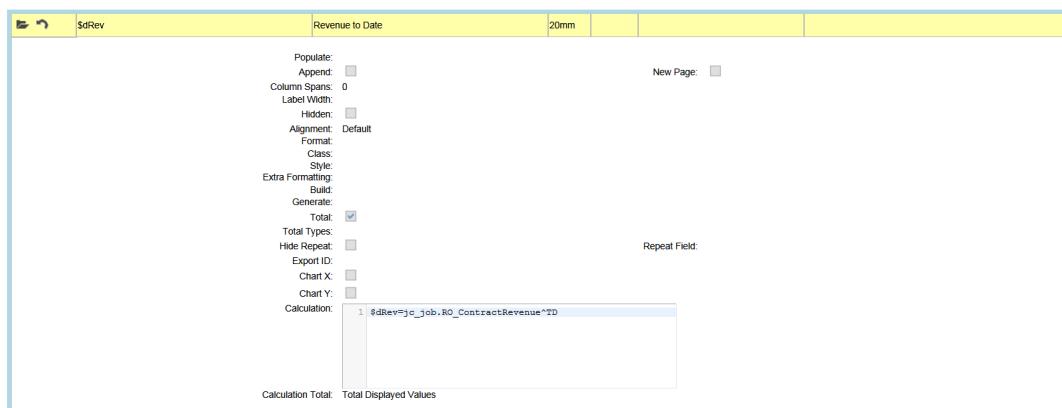
In the Field field of the Costs to Date line, enter a \$variable name to accumulate the totals for this field... and assign the value of the RO field in the calculation box.

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



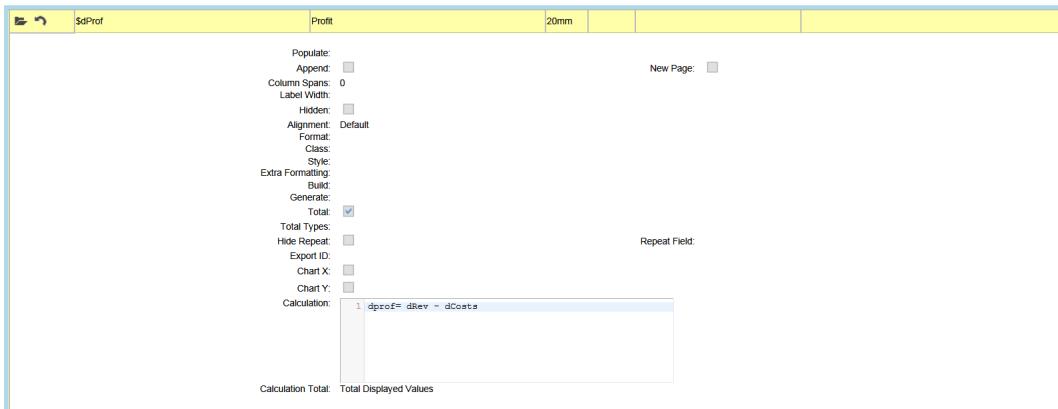
Repeat this procedure for the Revenue To Date.

| Field | Label | Width | Calculation |
|--------|-----------------|-------|-----------------------------------|
| \$dRev | Revenue to Date | 20mm | dRev=jc_job.RO_ContractRevenue^TD |



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> |

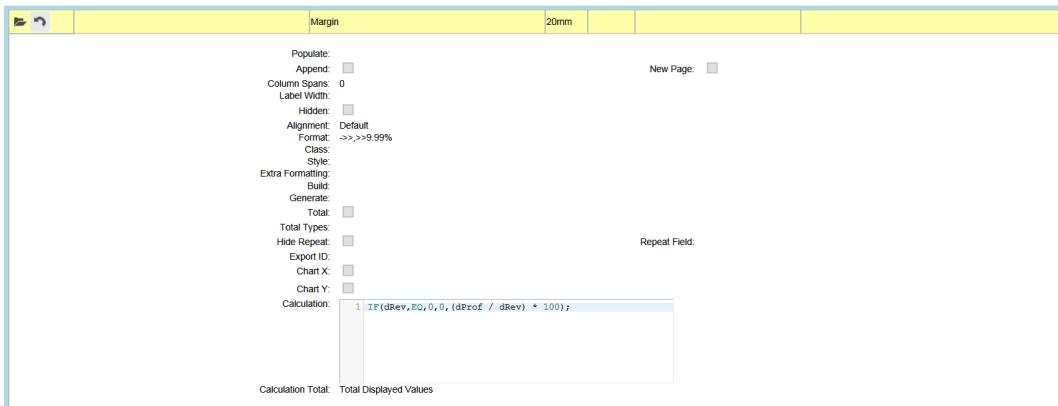


Add a new line for Profit Margin. Fill in the fields as follows:

| 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% |
| | | | | 218.45 | 0.00 | -218.45 | |
| | | | | 218.45 | 0.00 | -218.45 | |
| Group: | | | | | | | |
| Group: Commercial Building | | | | | | | |
| Type: Competitive Tender | | | | | | | |
| 1000 | Oxford Road Office. | London 01 | | 2,521,712.24 | 10,803,715.45 | 8,282,003.21 | 76.66% |
| 1100 | A406 Ealing underpass | Wales 04 | | 857,097.81 | 2,540,227.06 | 1,683,129.45 | 66.23% |
| 1101 | Windsor Rail bridge | South East 01 | | 302,662.54 | 870,800.08 | 568,137.54 | 65.24% |
| 1111 | Mxd Development at | London 01 | | 15,000.00 | 0.00 | -15,000.00 | 0.00% |
| | Princet 14 | | | 0.00 | 0.00 | 0.00 | 0.00% |
| 087 | 087 | 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 xxx | London 01 | | 09,937.50 | 0.00 | -09,937.50 | 0.00% |
| | | | | 3,795,409.89 | 14,214,742.59 | 10,418,332.70 | |
| Type: Extension | | | | | | | |
| 1001 | Holiday Inn Slough | North East 02 | | 3,754,761.83 | 7,643,419.54 | 3,888,657.71 | 50.88% |
| EBB001 | LeeEbs Test Contract | North West 02 | | 0.00 | 0.00 | 0.00 | 0.00% |
| | | | | 3,754,761.83 | 7,643,419.54 | 3,888,657.71 | |
| Type: Negotiated | | | | | | | |
| BA100 | CONTRACT PHASE | Wales 04 | | 0.00 | 0.00 | 0.00 | 0.00% |
| | | | | 0.00 | 0.00 | 0.00 | |
| Group: Commercial Building | | | | | | | |
| Group: Civil Engineering | | | | | | | |
| Type: Competitive Tender | | | | | | | |
| 1104 | M34 Northampton - | Overseas 00 | | 21,870,051.32 | 21,408,916.57 | -461,134.75 | -2.15% |
| | Southampton. | | | | | | |
| 1115 | M81 Extension | Overseas 00 | | 6,325.00 | 46,342.67 | 40,017.67 | 86.35% |
| 1200 | Bluewater shopping M & E | Overseas 00 | | 5,061.33 | 1,289,733.19 | 1,284,671.86 | 99.61% |

Printed using COINS V10.23.110920 by Nigel Longley at 13:25:57 on 02/09/11 (NLM) [Exceeded]

Page 1

10.13 Using Report Selection Criteria in the Report Query (1)

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 Exercise 7 and call them xxM1Exercise10.

10.13.1 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)

FieldLabelAppend

RS_job_num__1Contract From
RS_job_num__2To
RS_job_fore__3Manager

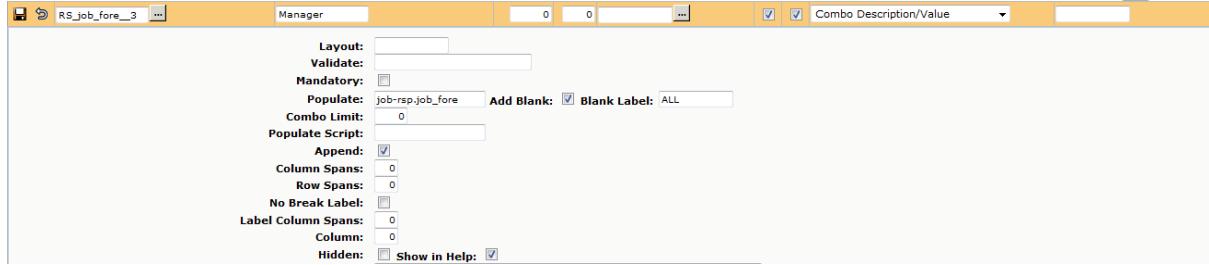
Against this last field also include the Populate for Contract Managers (job-rsp.job_fore) using the View-As and Populate fields as below :

View AsCombo 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.

Populatejob-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.



10.13.2 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}' = '')
```

The screenshot shows the 'Report' configuration screen for 'NLM1Exercise10'. The 'Query' tab is selected. The 'Body Query' section contains the modified report query:

```
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}' = '')
```

Below the query, there are various configuration options:

- Condition:** Maximum Rows: 0
- Body Title:**
- Calculate Condition:**
- Generate Program:**
- Data Set Definition:**
- Data Layout:**
- Page Layout:** %A4RLAND-A4 Report Landscape
- Extra Formatting:**
- Content:**
- Suppress Selection Page:** [checkbox]
- Suppress Company Name:** [checkbox]
- Suppress Module Name:** [checkbox]
- Chart:**
- Chart Formatting:**

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

10.14 Using Report Selection Criteria in the Report Data (2)

This exercise modifies the existing report from Exercise 10 to use further selection criteria within the Report Data using RS_ fields.

10.14.1 Create the Function

Set up a new function to create an additional tab for Period Selection.

Example:

Function Name - xxM1Exercise10Main Report Function

Tab Function - xxM1Exercise10THolding the Contract Selection

Additional Tab -xxM1Exercise10T1Holding the Financial Date Selection

| | | | | | | | |
|--------------------------|-------------------------------------|-------------------------------------|------------------|----------------------------|----------|-----------------------|------------|
| <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | NLM1Exercise10 | Module 1 Exercise 10 | Function | JC-JC Contract Status | REP-Report |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | NLM1Exercise10T | Module 1 Exercise 10 Tab | Function | JC-JC Contract Status | REP-Report |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | NLM1Exercise10T1 | Module 1 Exercise 10 Tab 1 | Function | JC-JC Contract Status | REP-Report |

10.14.2 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:

FieldRS_glp_fdate_2

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. In this case, we only need the "To" field.

LabelTo Financial Period

View AsCombo 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.

TabxxM1Exercise10T1

Populateglp-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.

| | | | | | | |
|---|-----------------|---------------------|---|---|---|------------------|
| | RS_glp_fdate__2 | To Financial Period | 0 | 0 | <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Combo Description/Value | NLM1Exercise10T1 |
| Special: Layout: Validate: Mandatory: <input type="checkbox"/> Populate: glp-rsp.glp_fdate Add Blank: <input type="checkbox"/> Blank Label: Combo Limit: 0 | | | | | | |

10.14.3 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:

FieldLabel

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.

| Report: NLM1Exercise10 | | | | | | |
|--|-------|---|-----------------|--------|----------|----------|
| Header Tables: co_config Body Table: ic_job Notes: | | | | | | Function |
| Query Sort Calculations Forms Fields Used in | | | | | | Page |
| | Field | Label | Width | Height | Function | View As |
| <input type="checkbox"/> | | job_num | Code | 15mm | | |
| <input type="checkbox"/> | | job_name | Name | 35mm | | |
| <input type="checkbox"/> | | jd_desc | Location | 35mm | | |
| <input checked="" type="checkbox"/> | | RO_ContractCosts^TD 0 {RS_glp_fdate__2} | Costs to Date | 20mm | | |
| <input type="checkbox"/> | | RO_ContractRevenue^TD 0 {RS_glp_fdate__2} | Revenue To Date | 20mm | | |
| | | | | | | |
| To: <input type="text"/> Search: Field <input type="button" value="Search"/> | | | | | | |
| Selectors <input type="button" value="New"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/> | | | | | | |
| Form: Body | | | | | | |

