

# Fusion SaaS R12: Implementing Data Security for Oracle Fusion Financials in Custom BI Publisher Reports

For customer's using the new R12 security model, this document details how to implement data security for Custom BI Publisher reports in Fusion SaaS Release 12 for the following products:

- Payables and Receivables
- General Ledger
- Fixed Assets

## Payables and Receivables

For Payables and Receivables data can be secured using Multi-Organization Access Control (MOAC).

Data security on the database tables is controlled through Fine Grained Access Control. Synonyms are created and Row Level Security is applied to these synonyms based upon the roles assigned to the user.

For example, a synonym AP\_INVOICES will be created for table AP\_INVOICES\_ALL. Row Level Security will then be applied to AP\_INVOICES using the MO\_GLOBAL procedures.

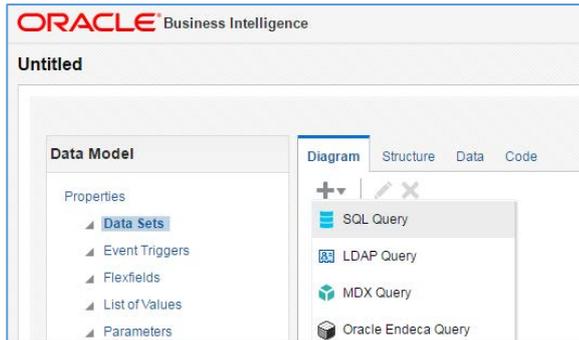
A predicate restricting the rows accessed to only the Business Units authorized to the User's roles then will be applied each time the secured synonym is referenced by a SQL statement.

The following privileges can be used to secure reports for the following products:

Product	Privilege
Payables	AP_MANAGE_PAYABLES_INVOICE_DATA
Receivables	AR_VIEW_RECEIVABLES_ACTIVITIES_DATA

## Implementing MOAC security on custom reports in SaaS

1. Navigate to the BI Publisher Server and create a new data model
2. In the new data model, create a SQL Query data set:



For the new data set, provide values for the required fields:

<b>Name</b>	Provide a name of your choice, e.g. Security
<b>Data Source</b>	Leave the data source as the default
<b>Type of SQL</b>	Procedure Call
<b>Row Tag Name</b>	Leave blank

3. Enter a SQL Query for the procedure Call data set

Copy the following into the SQL Query section. Substitute the privilege passed to the MO\_GLOBAL.Init procedure for the one required by your report. See the section *Overview of MOAC in Fusion* for the list of privileges.

```

DECLARE
    type refcursor is REF CURSOR;
    xdo_cursor refcursor;

BEGIN
    MO_GLOBAL.Init('AP_MANAGE_PAYABLES_INVOICE_DATA');

    OPEN :xdo_cursor FOR
    SELECT SYSDATE RUN_DATE
    from dual;

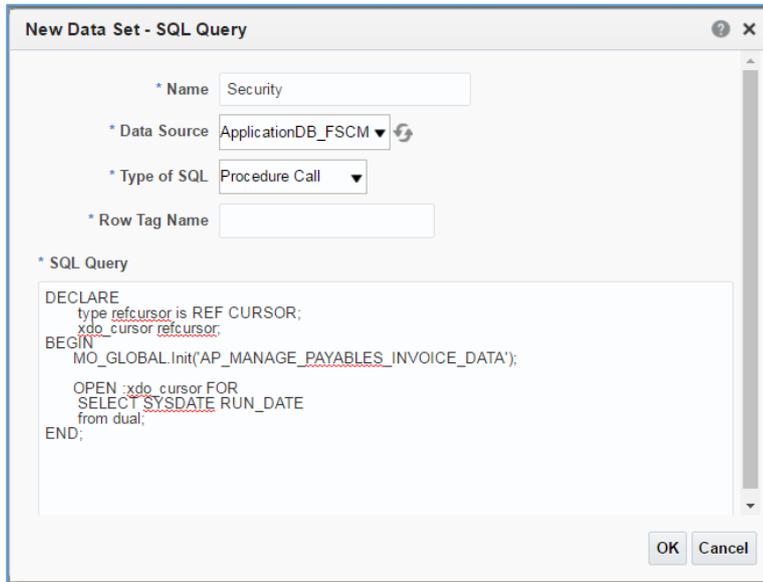
END;
```

---

*Note: Data sets of type Procedure Call use REF CURSORS which must return some data. Therefore, once the MO\_GLOBAL.init procedure has been executed, the cursor returns the report run date as the data for the data set.*

---

For example:



4. Create a data set for the main report query

Create an additional data set of type **Standard SQL**. This will contain the main report query which will be restricted by the row level data security applied by the first data set.

For this data set, provide values for the required fields:

<b>Name</b>	Provide a name of your choice, e.g. Main
<b>Data Source</b>	Leave the data source as the default
<b>Type of SQL</b>	Standard SQL

5. Enter a SQL Query for the Standard SQL data set

Enter the query to return the data that you wish displayed in your report.

---

*Note: It is important to reference the MOAC enabled synonyms as opposed to the underlying tables. The underlying tables will return all rows whilst the MOAC enabled synonyms will only return the rows to which the user has access.*

*For example, reference AP\_INVOICES, which is a MOAC enabled synonym, instead of AP\_INVOICES\_ALL, which will return all rows regardless of access.*

*Please see Appended I for a list of the available MOAC enabled synonyms.*

---

For example:

**Edit Data Set - Main**

\* Name: Main

\* Data Source: ApplicationDB\_FSCM

\* Type of SQL: Standard SQL

\* SQL Query: `SELECT invoice_num, invoice_amount, invoice_date FROM AP_INVOICES`

Buttons: Query Builder, Generate Explain Plan, OK, Cancel

## Examples

Please find some examples for securing AP and AR reports below.

### 1. Accounts Payable

Create the Procedure Call data set as follows:

```
DECLARE
  type refcursor is REF CURSOR;
  xdo_cursor refcursor;
BEGIN
  MO_GLOBAL.Init('AP_MANAGE_PAYABLES_INVOICE_DATA');

  OPEN :xdo_cursor FOR
  SELECT SYSDATE RUN_DATE
  from dual;
END;
```

Next, create the additional data set with the main report query. Here's an example query to fetch invoice number, invoice amount and invoice date:

```
SELECT invoice_num,
       invoice_amount,
       invoice_date
FROM AP_INVOICES
```

### 2. Accounts Receivable

Create the Procedure Call data set as follows:

```

DECLARE
    type refcursor is REF CURSOR;
    xdo_cursor refcursor;
BEGIN
    MO_GLOBAL.Init('AR_VIEW_RECEIVABLES_ACTIVITIES_DATA');

    OPEN :xdo_cursor FOR
    SELECT SYSDATE RUN_DATE
    from dual;
END;

```

---

*Note: As this is an AR report, we are passing the 'AR\_VIEW\_RECEIVABLES\_ACTIVITIES\_DATA' privilege to the MO\_GLOBAL.Init procedure.*

---

Next, create the additional data set with the main report query. Here's an example query to fetch transaction number, transaction date and a flag to determine completion.

```

SELECT trx_number,
       trx_date,
       complete_flag
FROM RA_CUSTOMER_TRX

```

## General Ledger

Data security for General Ledger is managed using:

- Data access sets
- Segment value security rules

### Data Access Sets

You assign users to the appropriate security context, in this case data access set, for job roles using the Manage Data Access for Users page.

The Data Access Sets assigned to a user via the Manage Data Access for Users UI are recorded in the table FUN\_USER\_ROLE\_DATA\_ASGNMNTS.

This table can be used in BI Publisher data models to secure access to the appropriate data access sets for each user.

Data Access Sets can secure Full Ledgers or can secure individual Balancing Segments. Therefore, restricting a report purely by ledger could have security implications as a user may not have access to the full ledger, only certain Balancing Segments. Therefore, the code needs to ensure that only segments to which the user has access are reported. For example:

```

select l.name,
       &BALANCING_SEGMENT_VALUE balancing_segment
from gl_ledgers l,
     gl_code_combinations cc
where l.chart_of_accounts_id = cc.chart_of_accounts_id
and (l.ledger_id, &BALANCING_SEGMENT_VALUE) in
     (select asa.ledger_id,
          decode(a.security_segment_code,
                'F', &BALANCING_SEGMENT_VALUE,
                asa.segment_value)
     from gl_access_sets a,
          gl_access_set_assignments asa
     where a.access_set_id = :P_DAS_ID
          and asa.access_set_id = a.access_set_id
     )

```

P\_DAS\_ID is a parameter based upon a list of values which will only return the data access sets assigned to the user, for example:

```
select distinct a.name, a.access_set_id
from gl_access_sets a,
     fusion.fun_user_role_data_asgnmnts urda
where urda.user_guid = FND_GLOBAL.USER_GUID
and a.access_set_id = urda.access_set_id
```

&BALANCING\_SEGMENT\_VALUE is defined as a flexfield lexical in the data model and will return the balancing segment value:

*Lexical Name	Flexfield Type	Lexical Type	Application Short Name	Flexfield Code	R
BALANCING_SEGM	Key Flexfield	Select	GL	GL#	

**BALANCING\_SEGMENT\_VALUE: Type: Select**

Enable Multiple Structure Instances

Code Combination Table Alias: cc

Structure Instance Number: :P\_STRUCT\_NUM

Segments: GL\_BALANCING

Show Parent Segments

Output Type: Value

When defining the flexfield lexical, it is necessary to supply a Structure Instance number which is the Chart of Accounts Id. In the example above, :P\_STRUCT\_NUM is used which is a parameter based upon the following List of Values:

```
select l.chart_of_accounts_id
from gl_access_sets a,
     fusion.fun_user_role_data_asgnmnts urda,
where urda.user_guid = FND_GLOBAL.USER_GUID
and a.access_set_id = urda.access_set_id
and a.access_set_id = :P_DAS_ID
```

### Segment Value Security

Set up segment value security rules on value sets to control access to parent or detail segment values for chart of accounts segments, also called flexfield segments. Segment value security rules restrict data entry, online inquiry, and reporting.

To enable Segment value security in BI Publisher, it is necessary to create a flexfield lexical as follows:

*Lexical Name	Flexfield Type	Lexical Type	Application Short Name	Flexfield Code	Reorder
FLEX_SECURE	Key Flexfield	Select	GL	GL#	▲▼

**FLEX\_SECURE: Type: Select**

Enable Multiple Structure Instances

Code Combination Table Alias

Structure Instance Number

Segments

Show Parent Segments

Output Type

Then the following clause should be added to the report query:

```
AND &FLEX_SECURE = 'N'
```

---

*Note: The return value of &FLEX\_SECURE would be 'Y' if the account is secured, or 'N' if it's unsecured.*

---

## Fixed Assets

In Oracle Fusion Assets, you can secure access to assets to perform transactions and view their information by asset book. The permission must be explicitly granted to each user.

You assign users to the appropriate security context, in this case asset book, for job roles using the Manage Data Access for Users page.

The asset books assigned to a user via the Manage Data Access for Users UI are recorded in the table FUN\_USER\_ROLE\_DATA\_ASGNMNTS.

This table can be used in BI Publisher data models to secure access to the appropriate asset books for each user.

The following code could be used to report on the asset books:

```
select book_type_code, book_type_name, book_class
from fa_book_controls bc
where bc.book_control_id in (select da.book_id
                             from fun_user_role_data_asgnmnts da,
                             fa_book_controls bc
                             where bc.book_control_id = da.book_id
                             and da.user_guid = FND_GLOBAL.USER_GUID
                             and role_name in
                                 ('ORA_FA_ASSET_ACCOUNTING_MANAGER_JOB'))
```

And if required, a parameter could be added which only allows assets books to be selected if assigned to the user. The parameter would be based upon a list of values, for example:

```
select bc.book_type_code, da.book_id
from fun_user_role_data_asgnmnts da,
     fa_book_controls bc
where bc.book_control_id = da.book_id
and da.user_guid = FND_GLOBAL.USER_GUID
and role_name in ('ORA_FA_ASSET_ACCOUNTING_MANAGER_JOB')
```

And the report query would be amended to reference the parameters, for example:

```
select book_type_code,book_type_name, book_class
from fa_book_controls bc
where bc.book_control_id = :P_BOOK_ID
```

## Appendix I – MOAC enabled Synonyms

AP_1096_DATA	AR_CONS_INV_TRX_LINES	JG_FSCL_DOC_LINE_DTLS
AP_1099_TAPE_DATA	AR_DEFERRED_LINES	JG_FSCL_DOC_RELATIONS
AP_AWT_BUCKETS	AR_DISTRIBUTIONS	JG_FSCL_HDRS_ATRB_EXT
AP_AWT_GROUP_TAXES	AR_DISTRIBUTION_SETS	JG_FSCL_HDR_DTLS_ATRB_EXT
AP_AWT_TAX_RATES	AR_DISTRIBUTION_SET_LINES	JG_FSCL_LINES_ATRB_EXT
AP_AWT_TEMP_DISTRIBUTIONS	AR_INTEREST_BATCHES	JG_FSCL_LN_DTLS_ATRB_EXT
AP_BATCHES	AR_INTEREST_HEADERS	JG_FSCL_TAX_LINES
AP_CHECKS	AR_INTEREST_LINES	JG_SYSTEM_OPTIONS
AP_DISTRIBUTION_SETS	AR_INTERFACE_CONTS	JL_BR_AP_BANK_RETURNS
AP_DISTRIBUTION_SET_LINES	AR_INTERIM_CASH_RECEIPTS	JL_BR_AP_COLLECTION_DOCS
AP_HISTORY_CHECKS	AR_INTERIM_CASH_RECEIPT_LINES	JL_BR_AP_COLL_DOCS_DET
AP_HISTORY_INVOICES	AR_JOURNAL_INTERIM	JL_BR_AP_CONSOLID_INVOICES
AP_HISTORY_INV_PAYMENTS	AR_LINE_CONTS	JL_BR_AP_INT_COLLECT
AP_HOLDS	AR_MISC_CASH_DISTRIBUTIONS	JL_BR_AP_INT_COLLECT_EXT
AP_INVOICES	AR_PAYMENTS_INTERFACE	JL_BR_AR_BANK_RETURNS
AP_INVOICE_DISTRIBUTIONS	AR_PAYMENT_SCHEDULES	JL_BR_AR_BORDEROS
AP_INVOICE_KEY_IND	AR_RATE_ADJUSTMENTS	JL_BR_AR_COLLECTION_DOCS
AP_INVOICE_LINES	AR_RECEIPT_METHOD_ACCOUNTS	JL_BR_AR_COLL_BATCHES
AP_INVOICE_PAYMENTS	AR_RECEIVABLES_TRX	JL_BR_AR_COLL_DOCS_DET
AP_INV_APRVL_HIST	AR_RECEIVABLE_APPLICATIONS	JL_BR_AR_COLL_DOCS_DET_WHT
AP_INV_SELECTION_CRITERIA	AR_REC_TRX_LE_DETAILS	JL_BR_AR_COMP_INV
AP_PAYMENT_HISTORY	AR_REF_ACCOUNTS	JL_BR_AR_DISTRIBUTNS
AP_PAYMENT_KEY_IND	AR_REVENUE_ADJUSTMENTS	JL_BR_AR_OCCURRENCE_DOCS
AP_PAYMENT_SCHEDULES	AR_STATEMENTS_HISTORY	JL_BR_AR_REC_MET_ACCTS_DTL
AP_PREPAY_HISTORY	AR_STATEMENT_CYCLE_DATES	JL_BR_AR_REC_MET_ACCTS_EXT
AP_REPORTING_ENTITIES	AR_SYSTEM_PARAMETERS	JL_BR_AR_REMIT_BORDEROS
AP_REPORTING_ENTITY_LINES	AR_TRANSACTION_HISTORY	JL_BR_AR_REMIT_COLL_BAT
AP_SELECTED_INVOICES	AR_TRANSMISSIONS	JL_BR_AR_RET_INTERFACE
AP_SELF_ASSESSED_TAX_DIST	CE_BANK_ACCT_USES_OU	JL_BR_AR_RET_INTERFACE_EXT
AP_SYSTEM_PARAMETERS	FINANCIALS_SYSTEM_PARAMETERS	JL_BR_AR_SELECT_ACCOUNTS
AP_TAX_CODES	JA_CN_MISC_PARAM	JL_BR_AR_SELECT_CONTROLS
AP_UNSELECTED_INVOICES	JA_CN_TRX_HEADERS	RA_ACCOUNT_DEFAULTS
AP_VENDOR_KEY_IND	JA_CN_TRX_LINES	RA_BATCHES
AR_ADJUSTMENTS	JA_CN_VAT_HEADERS	RA_CUSTOMER_TRX
AR_BATCHES	JA_CN_VAT_LINES	RA_CUSTOMER_TRX_LINES
AR_BATCH_SOURCES	JG_DOC_SEQ_DERIVATIONS_F	RA_CUST_TRX_LINE_GL_DIST
AR_CASH_RECEIPTS	JG_FSCL_ATRB_INT_ERRORS_GT	RA_CUST_TRX_LINE_SALESREPS
AR_CASH_RECEIPT_HISTORY	JG_FSCL_DOC_ASSET_CNTRLS_F	RA_INTERFACE_DISTRIBUTIONS
AR_CASH_RECOS	JG_FSCL_DOC_ASSET_XFER_INT	RA_INTERFACE_ERRORS
AR_CASH_RECO_LINES	JG_FSCL_DOC_GEN_CNTRLS_F	RA_INTERFACE_LINES
AR_CASH_REMIT_REFS	JG_FSCL_DOC_GEN_ERRORS	RA_INTERFACE_SALESCREDITS
AR_CONS_INV	JG_FSCL_DOC_HDRS	ZX_ID_TCC_MAPPING
AR_CONS_INV_TRX	JG_FSCL_DOC_LINES	