

ORACLE INTEGRATION FOR ORACLE 12

TROUBLESHOOTING GUIDE



Table of Contents

OVERVIEW: Where to locate XML\Logging by Application 3

OVERVIEW: Controlling XML, Integration & Adapter Logging 3

OVERVIEW: Tables Used For Troubleshooting 4

Setting Profile Options 5

Obtain Integration\Adapter Data and XML 6

Provide Log Extracts and XML From SABRIX_LOG Table 6

FND Logging 8

OVERVIEW: WHERE TO LOCATE XML/LOGGING BY APPLICATION

AR/OM Manual Transactions

1. Sabrix_log Table: Adapter logging and XML
2. Fnd_log_messages Table: eBTax and Sabrix Integration Logging

AR Autoinvoice

1. Concurrent log: AI logging ON
2. Sabrix_log Table: Adapter logging and XML
3. Fnd_log_messages: eBTax and Sabrix Integration Logging

AP Invoices

1. Concurrent log
2. Sabrix_log Table: Adapter logging and XML (Concurrent Output on error when parameter set)
3. Fnd_log_messages: eBTax and Sabrix Integration Logging (For 5.4.x Online Tax Calculations)

PO Manual Transactions

1. Sabrix_log Table: Adapter logging and XML
2. Fnd_log_messages: eBTax and Sabrix Integration Logging

Note: In order to get complete Integration and eBTax logging both the **integration logging** and **FND logging** needs to be enabled.

OVERVIEW: CONTROLLING XML, INTEGRATION & ADAPTER LOGGING

Integration and Adapter Log Message

1. **AR/OM/PO**
 - a. Sabrix Profiles: Integration logging
 - b. FND Debug Profiles: eBTax logging
 - c. Connection Options Table: adapter and xml logging
2. **AP**
 - a. Concurrent Request Parameters: Integration logging
 - b. Sabrix Print Structure Profile: enables invoice data to be logged
 - c. Connection Options Table: adapter xml logging

OVERVIEW: TABLES USED FOR TROUBLESHOOTING

INTERFACE TABLES:

Support may require from the below tables if logging can't be provided.

1. Input Interface Tables (ERP to Determination)
 - a. Sabrix_invoice
 - b. Sabrix_line
 - c. Sabrix_registration
2. Output Interface Tables (Determination to ERP)
 - a. Sabrix_invoice_out
 - b. Sabrix_line_out
 - c. Sabrix_line_tax
 - d. Sabrix_invoice_tax
 - e. Sabrix_message
 - i. Stores any information messages sent by Sabrix
 - ii. Amount of messaging save depends on the Message Threshold Preference Company Option in Determination.
 - iii. May be helpful if debugging is off
 - f. Sabrix_license_out

LOG TABLE:

1. SABRIX_LOG
 - a. Integration log messages
 - b. Adapter log messages
 - c. INPUT and OUTPUT XML

CONFIGURATION TABLES

1. SABRIX_CONNECTIONS
 - a. Connection_Type (Oracle12 O2C or P2P)
 - b. Version of the Oracle R12 Integration
 - c. URL used to call Determination XML Invoice Servlet for Tax Calculation

Row #	CONNECTION_ID	CONNECTION_NAME	CONNECTION_TYPE	CONNECTION_VERSION	CONNECTION_SERVER_BASE_URL	SABRIX_CALC_URL
1	1	IntegrationO2C	ORACLE12 O2C	5.3.0.0		http://psorion.sabrix.com:5130/sabrix/xmlinvoice
2	2	IntegrationP2P	ORACLE12 P2P	5.3.0.0		http://psorion.sabrix.com:5130/sabrix/xmlinvoice

2. SABRIX_CONNECTION_OPTIONS
 - a. DEBUG_LEVEL: Control adapter and XML logging:
 - 0 to log everything
 - 5 to log the Input & Output XML from Determination
 - 10 to log nothing
 - b. TAX_PROCESS_WORKER: used in submitting the concurrent request worker
 - c. BATCH_SIZE: for AP Integration to control the number of invoices submitted with each concurrent request worker.

SETTING PROFILE OPTIONS

Profile Options can be set at the:

- 1. Organization
- 2. Application
- 3. Responsibility
- 4. User Level

The below describes the Profile Options for **eBTax: Sabrix System Print Structure** and for **eBTax: Sabrix Debug Level**

Note: If you need more information on Profile Options then please reference the **Oracle R12 Integration User Guide: CONFIGURING SABRIX INTEGRATION PROFILE OPTIONS.**

[Oracle 12 v5.4.0.0 Documentation](#)
[Oracle 12 v5.3.0.0 Documentation](#)

eBTax: Sabrix System Print Structure	This profile option is used in conjunction with eBTax: Sabrix Debug Level. • To print data structures set the value to <i>ON</i>. • <i>OFF</i> and any other value prevents the structures from printing to the sabrix_log table. Note: For O2C and P2P transactions the input transaction view values can be used for debugging. Set to <i>OFF</i> in a production environment in conjunction with the profile option eBTax: Sabrix Debug Level set to <i>OFF</i>.
eBTax: Sabrix Debug Level	Set to one of the following for SABRIX_LOG table: • <i>ALL</i> to write all debug messages. • <i>XML</i> to write only XML messages. Note: In order to generate the Sabrix XML Report or Sabrix Log Report, the profile option must be set to either <i>ALL</i> or <i>XML</i> prior to executing a transaction. • <i>OFF</i> to not write messages. • <i>USER</i> to write all debug messages. • <i>WARNING</i> to write non-severe warning messages. • <i>TIMING</i> to write procedure entry and exit messages for AP transactions in the SABRIX_LOG table.

OBTAIN INTEGRATION\ADAPTER DATA AND XML

Update Configuration Table

Update **sabrix_connection_options** Table

1. Update sabrix_connection_options and set the DEBUG_LEVEL = 0

```
UPDATE SABRIX_CONNECTION_OPTIONS
SET OPTION_VALUE = '0'
WHERE OPTION_NAME = 'DEBUG_LEVEL'
```

Set Profile Options

1. Set eBTax: Sabrix Debug Level = ALL
2. Set eBTax: Sabrix System Print Structure = ON
3. Set FND: Debug Profile Options
 - a. FND: Debug Log Enabled: YES
 - b. FND: Debug Log Filename: NULL
 - c. FND: Debug Log Level: STATEMENT
 - d. FND: Debug Log Module: %

PROVIDE LOG EXTRACTS AND XML FROM SABRIX_LOG TABLE

Using a database visualization tool (i.e. TOAD), run the below queries to obtain the data necessary for troubleshooting:

```
SELECT      LOG_ID,
            LOG_DATE,
            LOCATION,
            MESSAGE
FROM        SABRIX_LOG
ORDER BY    LOG_ID
```

1. Determine the BATCH_ID for the transaction that you are troubleshooting by querying the SABRIX_INVOICE table.
2. Query the SABRIX_LOG table for the specific batch_id from step 1 to obtain the XML Input\Output sent to Determination for tax calculation.

```
SELECT      EXTENDED_MESSAGE
FROM        SABRIX_LOG
WHERE       LOCATION LIKE '%XML%'
AND         BATCH_ID = -- 'INSERT_BATCH_ID'
ORDER BY    LOG_ID
```

***If you don't have access to a database visualization tool and need to run the queries using SQL*Plus then follow the below process.

1. SET NEWPAGE 0
 - a. The number of blank lines between the top of each page and the top title. 0 = a form feed between pages.
2. SET SPACE 0
 - a. The number of spaces between columns in output (1-10)
3. SET LINESIZE 32000
 - a. Width of a line (before wrapping to the next line)
4. SET PAGESIZE 0
 - a. The height of the page - number of lines. 0 will suppress all headings, page breaks, titles
5. SET ECHO OFF
 - a. Display commands as they are executed
6. SET HEADING OFF
 - a. Print column headings
7. SET COLSEP ;
 - a. The text to be printed between selected columns normally a space.
8. SET LONG 60000
 - a. Set the maximum width (in chars) for displaying and copying LONG values.
9. SPOOL full_path\file_name.CSV
10. RUN QUERY
11. SPOOL OFF

FND LOGGING

1. Truncate the fnd_log_messages table (if possible)

```
TRUNCATE TABLE APPLSYS.FND_LOG_MESSAGES
```

2. If Truncate is not possible then determine max log_sequence before running transaction being debugged.

```
SELECT MAX(LOG_SEQUENCE)
FROM APPLSYS.FND_LOG_MESSAGES
```

3. How to extract fnd_log_messages to a file.

- a. With truncate:

```
SELECT      LOG_SEQUENCE,
            MODULE,
            MESSAGE_TEXT
FROM        APPLSYS.FND_LOG_MESSAGES
ORDER BY    LOG_SEQUENCE
```

- b. Without truncate:

```
SELECT      LOG_SEQUENCE,
            MODULE,
            MESSAGE_TEXT
FROM        APPLSYS.FND_LOG_MESSAGES
WHERE       LOG_SEQUENCE > -- MAX LOG SEQUENCE BEFORE RUNNING TRANSACTION
ORDER BY    LOG_SEQUENCE
```