Issue resolution when closing the period

Issue-1

Sometimes we cannot close the period and get message: “you cannot close financial year period 8 because it referenced open transaction”

It does mean there are open transactions in any module like, inventory transaction or ledger journal transaction or project transaction.

To verify the open transactions, we can take below actions:

1) Can check open inventory transactions from **Inventory-Periodic-Closing and adjustment -Close procedure - 1. check open quantities.**

2) Also, you can debug and verify the open transactions for that period from **FiscalCalendars** Class and the **checkNoExisistingUnpostedTransInJournals** method.

3) Also, uncheck the option “"set fiscal year Status to closed” in general ledger parameters and try to close the period.

You can use below job to retrieve open transaction for particular period:

**static** **void** dev\_OpenTransactionList(Args \_args)

{

LedgerJournalTrans ledgerJournalTrans;

LedgerJournalTable ledgerJournalTable;

InventJournalTrans inventJournalTrans;

InventJournalTable inventJournalTable;

ProjJournalTrans projJournalTrans;

ProjJournalTable projJournalTable;

ProjProposalJour projProposalJour;

ProjControlPeriodTable projControlPeriodTable;

ProjProposalTransUnion projProposalTransUnion;

**container** \_companyCnt = [**curext**()];

**date** \_startDate, \_endDate;

\_startDate = **1\8\2014**;

*//str2Date("01/08/2014", DateFormat::Auto);*

\_endDate = **31\8\2014**;

*//str2Date("31/08/2014", DateFormat::Auto);*

info("Ledger Transactions..................");

**if** (\_companyCnt)

{

**while** **select** **crossCompany** : \_companyCnt RecId, DataAreaId **from** ledgerJournalTrans

**where** ledgerJournalTrans.TransDate >= \_startDate &&

ledgerJournalTrans.TransDate <= \_endDate

**join** RecId, JournalNum, JournalType, JournalName **from** ledgerJournalTable

**where** ledgerJournalTable.JournalType != LedgerJournalType::Periodic &&

ledgerJournalTable.JournalType != LedgerJournalType::VendInvoicePool &&

ledgerJournalTable.JournalType != LedgerJournalType::VendInvoicePoolNotPosted &&

!ledgerJournalTable.Posted &&

ledgerJournalTable.JournalNum == ledgerJournalTrans.JournalNum

{

info(**strFmt**("Company:%1, Journal Type:%2, JournalName: %3 Journal Number:%4",

ledgerJournalTrans.dataAreaId, ledgerJournalTable.JournalType, ledgerJournalTable.JournalName, ledgerJournalTable.JournalNum));

}

}

info("Inventory Transactions..................");

**if** (\_companyCnt)

{

**while** **select** **crossCompany** : \_companyCnt RecId **from** inventJournalTrans

**where** inventJournalTrans.TransDate >= \_startDate &&

inventJournalTrans.TransDate <= \_endDate

**join** RecId, JournalId **from** inventJournalTable

**where** inventJournalTable.Posted == NoYes::No &&

inventJournalTable.JournalId == inventJournalTrans.JournalId

{

info(**strFmt**("Company:%1, ,Journal Number:%2", inventJournalTrans.dataAreaId, inventJournalTable.JournalId));

}

}

info("Project Journal ..................");

**if** (\_companyCnt)

{

**while** **select** **crossCompany** : \_companyCnt RecId, DataAreaId **from** projJournalTrans

**where** projJournalTrans.TransDate >= \_startDate &&

projJournalTrans.TransDate <= \_endDate

**join** RecId, JournalId, ProjId **from** projJournalTable

**where** !projJournalTable.Posted &&

projJournalTable.JournalId == projJournalTrans.JournalId

{

info(**strFmt**("Company:%1, ,Journal Number:%2, Project ID:%3", projJournalTrans.dataAreaId, projJournalTable.JournalId, projJournalTable.ProjId));

}

}

info("Project proposal Journal ..................");

**if** (\_companyCnt)

{

**while** **select** **crossCompany** : \_companyCnt RecId, DataAreaId, ProposalId **from** projProposalJour

**where** projProposalJour.InvoiceDate >= \_startDate &&

projProposalJour.InvoiceDate <= \_endDate &&

(projProposalJour.LineProperty == ProjLinePropertyCode::Open ||

projProposalJour.LineProperty == ProjLinePropertyCode::Approved)

{

**select** **firstonly** ProjId **from** projProposalTransUnion **where** projProposalTransUnion.ProposalId == projProposalJour.ProposalId;

info(**strFmt**("Company:%1, ,Project Id:%2", projProposalJour.dataAreaId, projProposalTransUnion.ProjId));

}

}

info("Project control period ..................");

**if** (\_companyCnt)

{

**while** **select** **crossCompany** : \_companyCnt RecId, DataAreaId, ProjId **from** projControlPeriodTable

**where** projControlPeriodTable.PeriodTo >= \_startDate &&

projControlPeriodTable.PeriodTo <= \_endDate &&

projControlPeriodTable.PeriodStatus == ProjControlPeriodStatus::Created

{

info(**strFmt**("Company:%1, Project Id:%2", projControlPeriodTable.dataAreaId, projControlPeriodTable.ProjId));

}

}

}

Below is the reference link to process the steps

<https://docs.microsoft.com/en-us/dynamics365/unified-operations/financials/general-ledger/year-end-close>

Issue-2

Sometimes we cannot close the period and get message: “Cannot close period Period 11 when there are source documents remaining to be journalised with an accounting date within the period”

This error message is due the open accounting distribution entry in that period. I got below link with the resolution to delete entry from the table “AccountingDistribution” with accountingEvent=0

<https://community.dynamics.com/ax/f/33/t/113881>

Job to info open AccountingDistribution

**static** **void** dev\_SFTYClosingLedgerDocumentsIssue(Args \_args)

{

   AccountingDistribution  accountingDistribution, objaccountingDistribution;

   FiscalCalendar          fiscalCalendar;

   FiscalCalendarPeriod    fiscalCalendarPeriod;

   FiscalCalendarPeriod    \_currentPeriod;

   Ledger                  ledger;

**int** i;

purchline objpurchline;

SubledgerJournalAccountEntryDistribution SLJAED;

SubledgerJournalAccountEntry SLJAE;

SubledgerJournalEntry SLJE;

GeneralJournalAccountEntry GJAE;

GeneralJournalEntry GJE;

   ;

i=**0**;

**select** \* **from** \_currentPeriod **where** \_currentPeriod.RecId==**5637145379**;

*//select \* from \_currentPeriod where \_currentPeriod.RecId== 5637146896;*

**if** (\_currentPeriod)

   {

**while** **select** \* **from** accountingDistribution

**where** accountingDistribution.AccountingDate >= \_currentPeriod.StartDate &&

               accountingDistribution.AccountingDate <= \_currentPeriod.EndDate &&

               accountingDistribution.AccountingEvent == **0**

**exists** **join** ledger

**where** ledger.PrimaryForLegalEntity == accountingDistribution.AccountingLegalEntity

**exists** **join** fiscalCalendar

**where** fiscalCalendar.RecId == ledger.FiscalCalendar

**exists** **join** fiscalCalendarPeriod

**where** fiscalCalendarPeriod.FiscalCalendar == fiscalCalendar.RecId &&

                   fiscalCalendarPeriod.RecId == \_currentPeriod.RecId

       {

*//if (accountingDistribution)*

*//{*

*//info(queryValue(accountingDistribution.RecId),queryValue());*

info(**strFmt**("%1, %2, %3, %4",accountingDistribution.RecId,accountingDistribution.ParentDistribution,accountingDistribution.SourceDocumentHeader,accountingDistribution.SourceDocumentLine));

i++;

*/\*select \* from SLJAED*

*where SLJAED.AccountingDistribution == accountingDistribution.RecId*

*join SLJAE*

*where SLJAE.RecId == SLJAED.SubledgerJournalAccountEntry*

*join SLJE*

*where SLJE.RecId == SLJAE.SubledgerJournalEntry*

*join GJAE*

*where GJAE.RecId == SLJAE.GeneralJournalAccountEntry*

*join GJE*

*where GJE.RecId == GJAE.GeneralJournalEntry*

*&& GJE.SubledgerVoucherDataAreaId == curext();*

*info(strfmt("%1, %2",GJE.JournalNumber,GJE.JournalCategory));\*/*

*/\* select \* from objaccountingDistribution where objaccountingDistribution.RecId == accountingDistribution.RecId;*

*select \* from objpurchline where objpurchline.SourceDocumentLine == objaccountingDistribution.SourceDocumentLine;*

*if(objpurchline.RecId != 0)*

*{*

*info(objpurchline.PurchId);*

*}\*/*

*//\_fiscalCalendars.addWarning(strFmt("@SYS344293",\_currentPeriod.Name));*

*//}*

       }

info(**strFmt**("%1",i));

   }

}

Job to delete Open AccountingDistribution

**static** **void** dev\_LKLGrootboekPeriodeAfsluitingWD(Args \_args)

{

  AccountingDistribution  accountingDistribution;

  FiscalCalendar          fiscalCalendar;

  FiscalCalendarPeriod    fiscalCalendarPeriod;

  FiscalCalendarPeriod    \_currentPeriod;

  Ledger                  ledger;

  counter teller =**0**;

  counter tellervoor =**0**;

  counter tellerna=**0**;

  ;

**while** **select** accountingDistribution

   {

       tellervoor++;

   }

**select** \* **from** \_currentPeriod **where** \_currentPeriod.RecId==**5637145379**; *//Replace  RecID by FiscalCalendarPeriod recID of affected period*

**if** (\_currentPeriod)

  {

**ttsBegin**;

**while** **select** **forUpdate**  RecId **from** accountingDistribution

**where** accountingDistribution.AccountingDate >= \_currentPeriod.StartDate &&

              accountingDistribution.AccountingDate <= \_currentPeriod.EndDate &&

              accountingDistribution.AccountingEvent == **0**

**exists** **join** ledger

**where** ledger.PrimaryForLegalEntity == accountingDistribution.AccountingLegalEntity

**exists** **join** fiscalCalendar

**where** fiscalCalendar.RecId == ledger.FiscalCalendar

**exists** **join** fiscalCalendarPeriod

**where** fiscalCalendarPeriod.FiscalCalendar == fiscalCalendar.RecId &&

                  fiscalCalendarPeriod.RecId == \_currentPeriod.RecId

      {

*//Enable the row below to automatically delete the record*

         AccountingDistribution.doDelete();

          teller++;

      }

**ttscommit**;

**while** **select** accountingDistribution

       {

       tellerna++;

       }

      info(**strFmt**("Amount of records before the problem record(s) were deleted from table Accountingdistribution: %1",tellervoor));

      info(**strFmt**("Deleted: %1",teller));

      info(**strFmt**("Amount of records after the problem record(s) were deleted in table Accountingdistribution: %1",tellerna));

**if**(tellervoor-teller==tellerna)

      {

      info("Check: OK!");

      }

**else**

      {

      info(**strFmt**("Check: Error! Before %1 - deleted %2 is not %3 (result after deletion)",tellervoor,teller,tellerna));

      }

  }

}

I run the job above on dev environment to delete the open AccountingDistribution and after that I was successfully able to close the period.

But, further there was a question - Other effects after deleting the open AccountingDistribution.

To verify, I have started to review how Accounting Distribution in AX is working.

“Accounting Distribution can be done in the Accounts Payable module are are used to define how an amount will be accounted for, such as how the expense, asset, tax, or charges will be encumbered on the purchase order or accounted for on the product receipt or vendor invoice.”

After configuring for accounting distribution, I have generated and posted the purchase order and have reviewed effect in AccountingDistribution Table at each step i.e. confirming PO, product receipt, invoice. I observed that initially after reviewing distributed amount from PO lines, “AccountingEvent” updates with “0” value. This field later on updates with the RecId at each step confirmation, receipt, invoice. So, we can say that AccountinDistribution with AccountingEvent=0 means that is the open transaction in AX.

Thereafter, I searched for the link to track the transaction from the front end. I used below query:

select SQLDICTIONARY.name,SOURCEDOCUMENTLINE.recid,SOURCEDOCUMENTLINE.SOURCERELATIONTYPE,accountingdate, ACCOUNTINGDISTRIBUTION.TRANSACTIONCURRENCYAMOUNT,

SOURCEDOCUMENTLINE.RECID as [SOURCEDOCUMENTLINE],

(case SQLDICTIONARY.name

when 'PURCHLINE' then (select count(\*) from PURCHLINE where PURCHLINE.SOURCEDOCUMENTLINE=SOURCEDOCUMENTLINE.RECID)

when 'TRVEXPTRANS' then (select count(\*) from TRVEXPTRANS where TRVEXPTRANS.SOURCEDOCUMENTLINE=SOURCEDOCUMENTLINE.RECID)

when 'TSTIMESHEETTRANS' then (select count(\*) from TSTIMESHEETTRANS where TSTIMESHEETTRANS.SOURCEDOCUMENTLINE=SOURCEDOCUMENTLINE.RECID)

when 'VENDINVOICEINFOLINE' then (select count(\*) from VENDINVOICEINFOLINE where VENDINVOICEINFOLINE.SOURCEDOCUMENTLINE=SOURCEDOCUMENTLINE.RECID)

else ''

end) as IsExist,

(case SQLDICTIONARY.name

when 'PURCHLINE' then (select PURCHID from PURCHLINE where PURCHLINE.SOURCEDOCUMENTLINE=SOURCEDOCUMENTLINE.RECID)

else ''

end) as [Purchase order number],

(case SQLDICTIONARY.name

when 'TRVEXPTRANS' then (select EXPNUMBER from TRVEXPTRANS where TRVEXPTRANS.SOURCEDOCUMENTLINE=SOURCEDOCUMENTLINE.RECID)

else ''

end) as [Travel Expense number],

(case SQLDICTIONARY.name

when 'TSTIMESHEETTRANS' then (select timesheetnbr+' - '+transid from TSTIMESHEETTRANS where TSTIMESHEETTRANS.SOURCEDOCUMENTLINE=SOURCEDOCUMENTLINE.RECID)

else ''

end) as [Timesheet Number - transaction id],

(case SQLDICTIONARY.name

when 'VENDINVOICEINFOLINE' then (select ORIGPURCHID from VENDINVOICEINFOLINE where VENDINVOICEINFOLINE.SOURCEDOCUMENTLINE=SOURCEDOCUMENTLINE.RECID)

else ''

end) as [Purchase order number]

from ACCOUNTINGDISTRIBUTION

join SOURCEDOCUMENTLINE on ACCOUNTINGDISTRIBUTION.SOURCEDOCUMENTLINE=SOURCEDOCUMENTLINE.recid

join SQLDICTIONARY

on SOURCEDOCUMENTLINE.SOURCERELATIONTYPE=SQLDICTIONARY.TABLEID

and SQLDICTIONARY.FIELDID=0

where ACCOUNTINGDISTRIBUTION.ACCOUNTINGDATE >= '2014-11-01' AND ACCOUNTINGDISTRIBUTION.ACCOUNTINGDATE <= '2014-11-30'

and ACCOUNTINGEVENT=0

order by SQLDICTIONARY.name, ACCOUNTINGDISTRIBUTION.ACCOUNTINGDATE