How-to Guide **PUBLIC**

SAP Business One 8.82
SAP Business One 9.0 and SAP Business One 9.0, version for SAP HANA
SAP Business One 9.1 and SAP Business One 9.1, version for SAP HANA
Document Version: 1.0 – 2014-11-27

How to Manage UUID field in Existing Purchasing Documents (Electronic Accounting Statement Feature)

Mexico

Typographic Conventions

Type Style	Description
Example	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Textual cross-references to other documents.
Example	Emphasized words or expressions.
`EXAMPLE`	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
`Example`	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Example`	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`EXAMPLE`	Keys on the keyboard, for example, `F2` or `ENTER`.

Table of Contents

1 Introduction

To comply with the regulations regarding the Electronic Accounting Statement in Mexico, SAP Business One partners must manually create a UDF for the purpose of managing UUID information (for more information, see SAP Note 2048577). This document describes the process of managing the UUID field for invoice type documents on the purchasing side, which are created within the transition period (from July to December 2014). All such documents must be reported in the journal entry report (JE report). The UUID field appears on all invoices (in XML file structure) and consists of 36 characters provided by the Mexican tax authority (SAT).

For SDN information, refer to the section *SDN Information*.

2 SDN Information

Summary

Hereby provided Tool is designed for managing electronic document numbering information (UUID) provided by SAT Authority. Tool is relevant for electronic document communication in Mexico (herein referred to as 'Tool') is delivered within the SDN license structure and therefore no support is provided. This guide helps you run the application easily (no installation is needed).

This Tool is a freeware application provided by SAP SE to SAP® Business One customers in Mexico in order to provide automated way of managing UUID information from XML into existing documents created within the transitional period.

Tool shall provide automated way of associating UUID information from XML representing electronic form of document for documents existing in the database. During its run, temporary data are created to provide a mapping bridge between the documents and XML files in order to correctly update UUID information in the relevant existing documents in the database.

Feedback on Tool should be provided through SDN forum or through your official SAP Local Product expert.

Authors:

Pawel Kostorz, Martin Slavik, Martin Pavlinsky

Company:

SAP Slovensko, s.r.o.

Created on: July 10, 2014

Author Bio

SAP Business One Solution Management created this application to fulfill a business gap which emerged due to a recent legal requirement effective from January 2015.

i Note

Find the relevant how-to guide attached to this document.

You can download Tool here.

3 Updating the UUID Field in Existing Invoice Type Documents on the A/P Side

An overview of the process of updating the UUID field in existing purchasing documents is as follows:

1. The XML files representing electronic documents are generated by the SAP authority.

   Note

   - As a prerequisite, all XML files for relevant documents must exist in the database.
   - Each XML file represents one document.

2. Using a prepared query, the XML data are extracted and placed into a temporary table.

   Note

   - The temporary table identifies each relevant document existing in the database in order to update the UUID information.
   - Each table line represents one document.

3. A/P documents created during the transition period are identified and compared to the data in the temporary table as follows:
   - If the XML file matches the document created in the database, then the document type and document number are assigned to the relevant table line (the yellow column in the example below).
   - This identification and comparison between the XML files and the relevant documents in the database are based on unique parameters.
   - If more than one document in the database matches an XML file, then the user needs to manually identify which document is valid.
4. A bridge between the XML files and the documents in the database is created so the application can identify which UUID belongs to which document. Using the Data Transfer Workbench migration tool, users can then update the UUID information in each relevant document in the database using a batch run.

Process Visualization

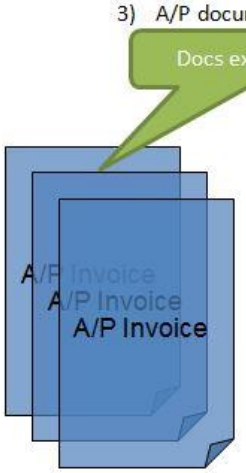
1) Prerequisite - Existing XML files available.

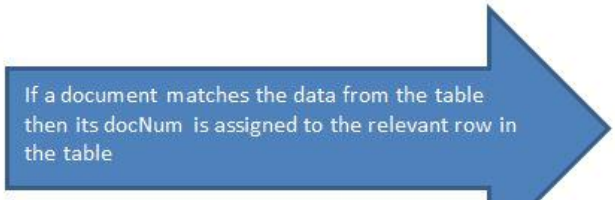
2) Data from the XML files are summarized into a temporary table.

Created table

RFC	DocTotal	Posting Date	UUID
MX1234	250 MXP	8.7.2014	UUID 1
MX5555	1300 MXP	9.8.2014	UUID 2
MX9999	2500 MXP	2.9.2014	UUID 3

3) A/P documents identification and their matching to the XML files.





Identified document

RFC	DocTotal	Posting Date	UUID	DocNum
MX1234	250 MXP	8.7.2014	UUID 1	327
MX5555	1300 MXP	9.8.2014	UUID 2	389
MX9999	2500 MXP	2.9.2014	UUID 3	531

4) Updating A/P documents with an appropriate UUID based on successful matching (through a DTW template)

4 Updating UUID in Existing Documents

To update the UUID field in existing relevant documents, proceed as follows:

1. Open SQL Enterprise Manager and run the following query:

```
BEGIN TRANSACTION
SET QUOTED_IDENTIFIER ON
SET ARITHABORT ON
SET NUMERIC_ROUNDABORT OFF
SET CONCAT_NULL_YIELDS_NULL ON
SET ANSI_NULLS ON
SET ANSI_PADDING ON
SET ANSI_WARNINGS ON
COMMIT
BEGIN TRANSACTION
GO
CREATE TABLE dbo.TTST
    (
    EntryNo int NOT NULL,
    CardName nvarchar(50) NULL,
    RFC nvarchar(50) NULL,
    DocTotal numeric(19, 6) NULL,
    DocDate nvarchar(50) NULL,
    UUID nvarchar(50) NULL
    )  ON [PRIMARY]
GO
ALTER TABLE dbo.TTST SET (LOCK_ESCALATION = TABLE)
GO
COMMIT
```

> **Note**
>
> This query creates a temporary TTST table which you can delete after you complete the entire procedure.

2. From the *Start* menu, choose *Run* and enter "**cmd**".
3. Open the path where `XMLRead.exe` is located and then enter the following parameters:

```
FILE;[path where the xml files are stored] SERVER; [name and alias of the server]
DATABASE; [name of the database where TTST table from Step 1 was created]
```

Note

"FILE", "SERVER, and "DATABASE" are fixed string expressions.

Example

A user may enter the following parameters:

```
XMLRead.exe FILE;C:\temp\ SERVER;BTSN60264814A\b1 DATABASE;PKO_MX_TEST
```

4. Press ENTER.

   The console reports that the process has been finished successfully.

5. Go to SQL Enterprise Manager and run the query below to select all invoices that match at least one record from the XML files:

```
select
    DocEntry,
    DocNum,
    T1.CardName as xmlCardName,
    T0.Cardname,
    convert(varchar(10),cast(replace(T1.DocDate, 'T',' ') as datetime),10) as
xmlDate,
    convert(varchar(10),T0.DocDate,10) as DocDate,
    T1.DocTotal as xmlDocTotal,
    T0.DocTotal,
    T1.RFC,
    T0.LicTradNum,
    T1.UUID

from
    OPCH T0 right outer join
    TTST T1 on T0.CardName = T1.CardName and T0.LicTradNum = T1.RFC and
convert(varchar(10),cast(replace(T1.DocDate, 'T',' ') as datetime),10) =
convert(varchar(10),T0.DocDate,10) and T1.DocTotal = T0.DocTotal

union all
select
    DocEntry,
    DocNum,
    T1.CardName as xmlCardName,
    T0.Cardname,
    convert(varchar(10),cast(replace(T1.DocDate, 'T',' ') as datetime),10) as
xmlDate,
    convert(varchar(10),T0.DocDate,10) as DocDate,
    T1.DocTotal as xmlDocTotal,
```

```
    T0.DocTotal,
    T1.RFC,
    T0.LicTradNum,
    T1.UUID

from
          ORPC T0 right outer join
    TTST T1 on T0.CardName = T1.CardName and T0.LicTradNum = T1.RFC and
convert(varchar(10),cast(replace(T1.DocDate, 'T',' ') as datetime),10) =
convert(varchar(10),T0.DocDate,10) and T1.DocTotal = T0.DocTotal
```

6. The query below helps you create an XML document for the DTW import:

```
select

'18' as ObjType,
T0.DocEntry,
T1.UUID

from
    OPCH T0 right outer join
    TTST T1 on T0.CardName = T1.CardName and T0.LicTradNum = T1.RFC and
convert(varchar(10),cast(replace(T1.DocDate, 'T',' ') as datetime),10) =
convert(varchar(10),T0.DocDate,10) and T1.DocTotal = T0.DocTotal
where
    T0.CardName = T1.CardName
    and T0.LicTradNum = T1.RFC
    and T1.DocTotal = T0.DocTotal
    and convert(varchar(10),cast(replace(T1.DocDate, 'T',' ') as
datetime),10) = convert(varchar(10),T0.DocDate,10)
union all

select

'19' as ObjType,
T0.DocEntry,
T1.UUID

from
    ORPC T0 right outer join
    TTST T1 on T0.CardName = T1.CardName and T0.LicTradNum = T1.RFC and
convert(varchar(10),cast(replace(T1.DocDate, 'T',' ') as datetime),10) =
convert(varchar(10),T0.DocDate,10) and  T1.DocTotal = T0.DocTotal
```

```
        where
            T0.CardName = T1.CardName
            and T0.LicTradNum = T1.RFC
            and T1.DocTotal = T0.DocTotal
    and convert(varchar(10),cast(replace(T1.DocDate, 'T',' ') as datetime),10) =
convert(varchar(10),T0.DocDate,10)
```

7. Copy the results into Microsoft Excel and update the *UDF_UUID* field in existing A/P invoices either manually or through the Data Transfer Workbench. For more information, refer to the documentation related to DTW.

www.sap.com/contactsap

Material Number