Purpose

This program Imports into Timekeeper tables from an external sql server database based on XML template files.
While the traditional import Import/Export utility uses ASCII source files, this application builds an XML input files from the source database and uses validity tests before the import takes place.

Why we developed it?

Workforce management (WFM) includes four major components: Payroll, Human resources management (HR), Time and Attendance and Access Control. Customers, all over the world, are seeking to integrate these applications.

The major issue in integration is the ability to specify a single entry point for all data that is required by all WFM applications. Examples of required data: Employee data: ID, Name etc., Organizational Data: Department names etc.

Most companies select one application as the main entry point and create interfaces to all other applications. The single entry point may vary from one company to another. Some companies put their emphasis on the Payroll application as the point of initializing data. Others utilize the HR application as the main entry point (HR basically maintains employee data follow up from recruitment to termination)

Lavie TimeTECH Ltd. has developed the **TK XML** to enable external applications to perform CRUD (create, update, read, delete) operations on several of the objects in TimeKeeper SQL.

The benefits of this approach:

A. The integration is based on a published XML Schema.
B. Any external application can update TK's database seamlessly by calling this .dll and providing as a parameter an XML populated with the required data.
C. TK's XML schema and .dll maintain integrity and validity of TK's DB.
D. Any future changes in TK's database will not require any change on behalf of the calling external application.
E. TK's database can be kept constantly synchronized with the data managed by other applications.
F. Since it is based on Microsoft's ActiveX technology, it can also be used through Windows Scripting to automate parts of TimeKeeper functionality.

Lavie TimeTECH Ltd. is encouraging the developers of Payroll, HR and Access Control applications to create a similar XML interface to their Database so the customer may select his entry point application.

Environment: TK-SQL

Required Program Files:

- Qdd_Sap.exe 27.05.2008
- Qddsap.dll 17.06.2008
- Tk_XML.dll 03.11.2008
- Lavlibsql.dll February 02, 2007
- Schema files
- Script:

```
CREATE proc DBO.LP_GETDAILY
    AS
   BEGIN
  select * from dbo.tk2xmlmap
  where s_tabletype = 20
END
```

- Must be at list one line in Table Laudit!

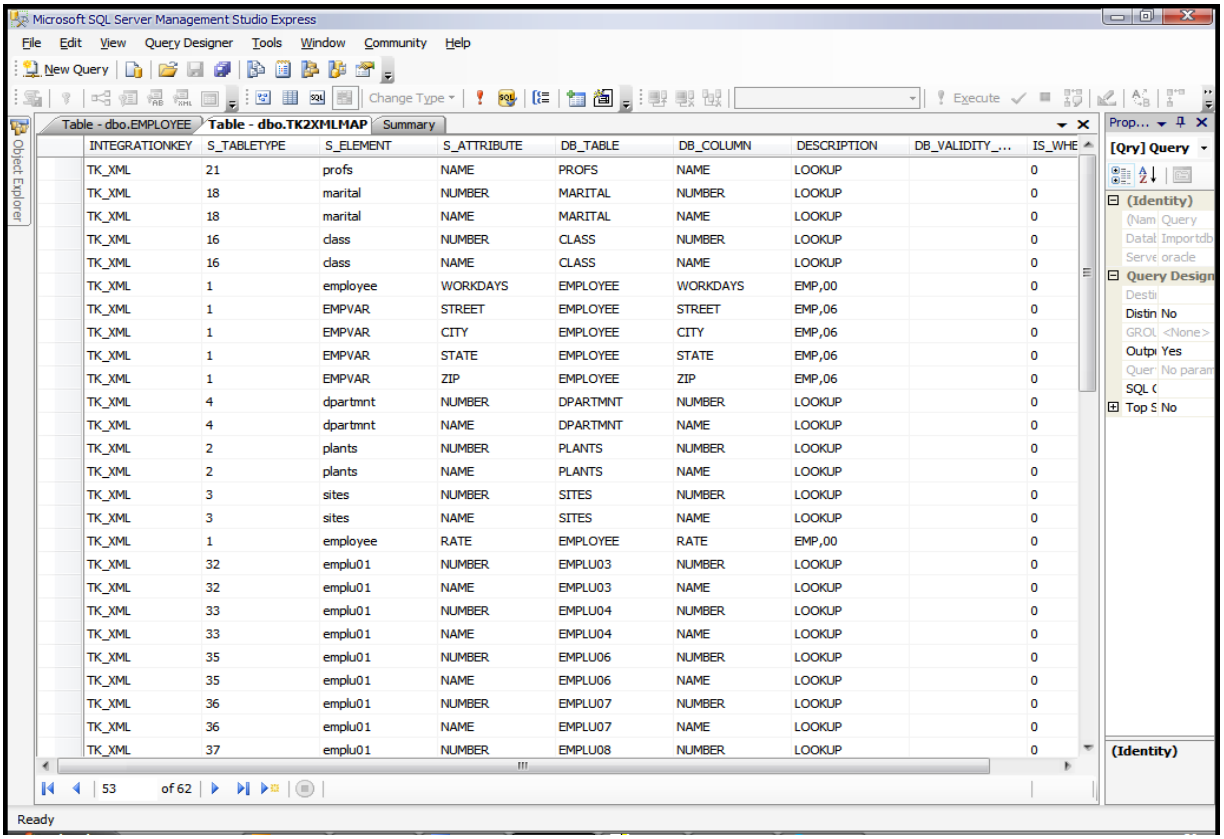
Setup Steps:

1. In Tksql ini add [QDD-SAP] section with server name, database name and an encryption string for the SQL_PARAM line (optional when SA blank is not used as SQL SERVER login information).

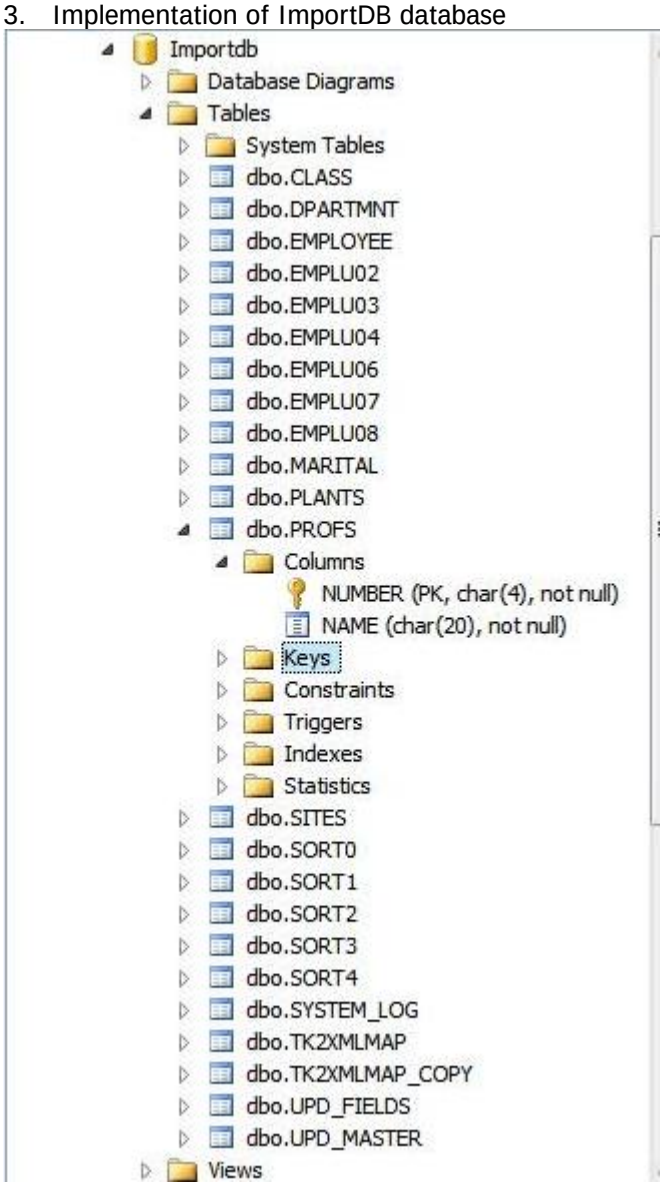
2. On the source database a map table with the name Tk2XMLMAP must exist. This table indicates the required links between the imported database and Timekeeper's database and tables.
   The structure of this table is as follows:

Field Name	Type	PK	Case Sensitive	Description
INTEGRATIONKEY	Char 20	Yes	No	Constant tk_xml
S_TABLETYPE	Int	Yes		Schema identifier taken from Schema's list
S_ELEMENT	Sysname		Yes	TimeKeeper's table name
S_ATTRIBUTE	Sysname		Yes	Timekeeper's field name
DB_TABLE	Sysname		No	ImportDB's table name
DB_COLUMN	Sysname		No	ImportDB's field name
DESCRIPTION	NVARCHAR 1024		No	Element type can have the following values: **Lookup** for lookup tables, **TC** for TimeCosting tables, **EMP,nn** for employees groups and **History** for records
DB_VALIDITY_DATE_COL	Sysname		No	1 for history support
IS_WHERE_COL	Tinyint			1 for Foreign key type field

An implementation of the map named TK2XMLMAP should look like the attached screen shot:
Values imported into lookup tables must have a line for Number and a line for Name. (each one of the 2 lines per value indicates different information).


Microsoft SQL Server Management Studio Express
File Edit View Query Designer Tools Window Community Help
New Query
Change Type
Execute
Table - dbo.EMPLOYEE
Table - dbo.TK2XMLMAP
Summary
Object Explorer
INTEGRATIONKEY S_TABLETYPE S_ELEMENT S_ATTRIBUTE DB_TABLE DB_COLUMN DESCRIPTION DB_VALIDITY_... IS_WHE
TK_XML 21 profs NAME PROFS NAME LOOKUP 0
TK_XML 18 marital NUMBER MARITAL NUMBER LOOKUP 0
TK_XML 18 marital NAME MARITAL NAME LOOKUP 0
TK_XML 16 class NUMBER CLASS NUMBER LOOKUP 0
TK_XML 16 class NAME CLASS NAME LOOKUP 0
TK_XML 1 employee WORKDAYS EMPLOYEE WORKDAYS EMP,00 0
TK_XML 1 EMPVAR STREET EMPLOYEE STREET EMP,06 0
TK_XML 1 EMPVAR CITY EMPLOYEE CITY EMP,06 0
TK_XML 1 EMPVAR STATE EMPLOYEE STATE EMP,06 0
TK_XML 1 EMPVAR ZIP EMPLOYEE ZIP EMP,06 0
TK_XML 4 dpartmnt NUMBER DPARTMNT NUMBER LOOKUP 0
TK_XML 4 dpartmnt NAME DPARTMNT NAME LOOKUP 0
TK_XML 2 plants NUMBER PLANTS NUMBER LOOKUP 0
TK_XML 2 plants NAME PLANTS NAME LOOKUP 0
TK_XML 3 sites NUMBER SITES NUMBER LOOKUP 0
TK_XML 3 sites NAME SITES NAME LOOKUP 0
TK_XML 1 employee RATE EMPLOYEE RATE EMP,00 0
TK_XML 32 emplu01 NUMBER EMPLU03 NUMBER LOOKUP 0
TK_XML 32 emplu01 NAME EMPLU03 NAME LOOKUP 0
TK_XML 33 emplu01 NUMBER EMPLU04 NUMBER LOOKUP 0
TK_XML 33 emplu01 NAME EMPLU04 NAME LOOKUP 0
TK_XML 35 emplu01 NUMBER EMPLU06 NUMBER LOOKUP 0
TK_XML 35 emplu01 NAME EMPLU06 NAME LOOKUP 0
TK_XML 36 emplu01 NUMBER EMPLU07 NUMBER LOOKUP 0
TK_XML 36 emplu01 NAME EMPLU07 NAME LOOKUP 0
TK_XML 37 emplu01 NUMBER EMPLU08 NUMBER LOOKUP 0
53 of 62
Prop...
[Qry] Query
(Identity)
(Identity)
Ready

3. Implementation of ImportDB database

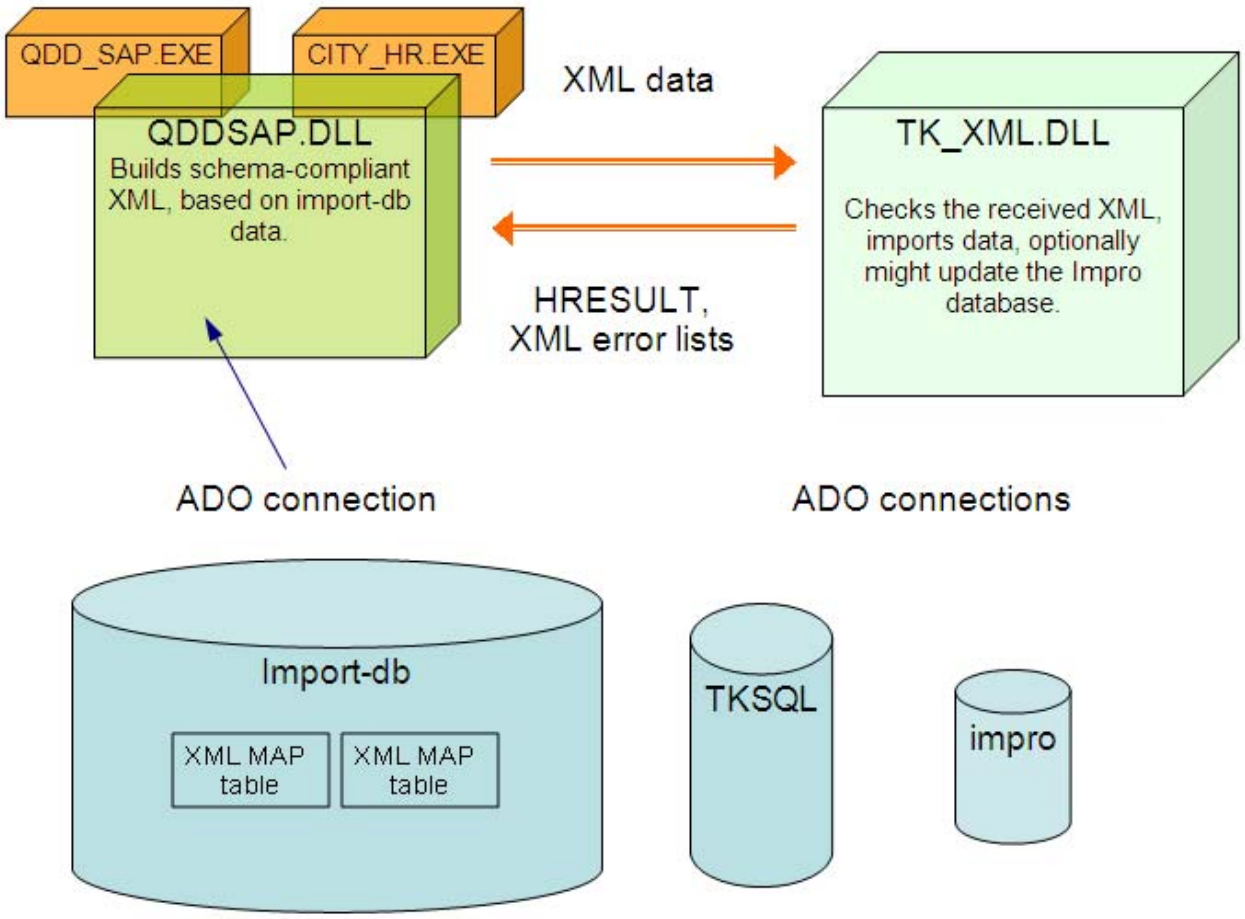


4. Schema's list

TimeKeeper Table	Value
Employees	1
Plants	2
Sites	3
Departments	4
Sort0	5
Sort1	6
Sort2	7
Sort3	8
Sort4	9
Contract	10
Schdgrp	11
Schedule	12
TableQ	13
Dndint	14
Rate	15
Class	16
Ranks	17
Martial	18
Absgroup	19
Daily	20
Profs	21
TableH	22
TableGZ	23
PrdTypes	24
OccupTypes	25
Order	26
Batch	27
Part	28
Proc	29
EmployeeChgKey	1001

5.

How does it work

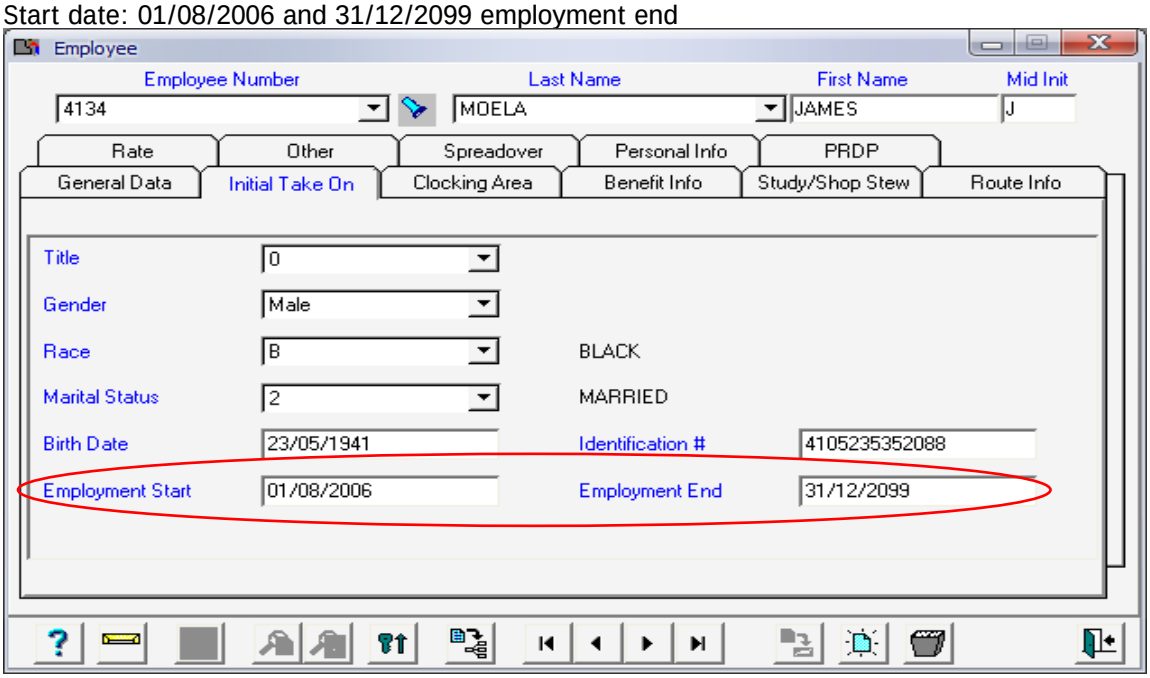


6. Discharging an employee using QDDSAP:

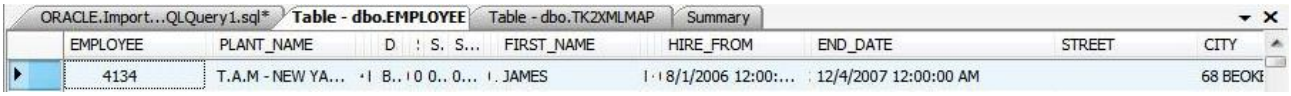
One of the main key advantages of using the QDDSAP utility is its date handeling features.
I.e. sending a termaination date to the hire_till field will update the employee's hire_till field and also changes the employee status to 99. We can also update the employee's badge validity dates on Timekeeper as well as on Impro databases.

Implemantation of discharging:
Current Status: the employee has status 0 on his status field.

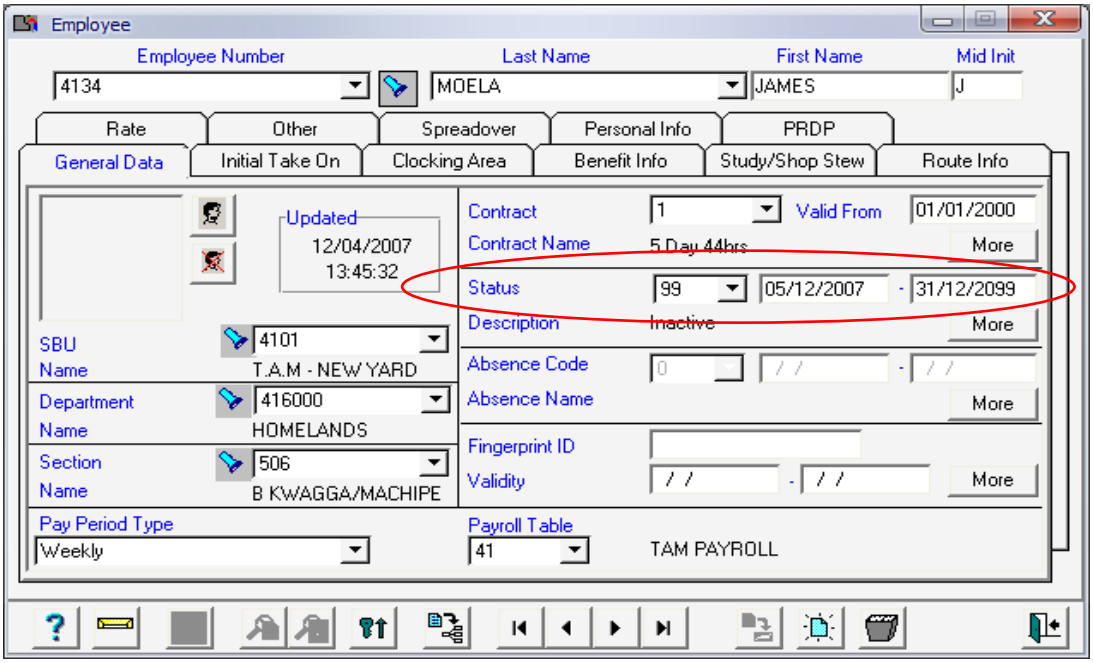
Start date: 01/08/2006 and 31/12/2099 employment end

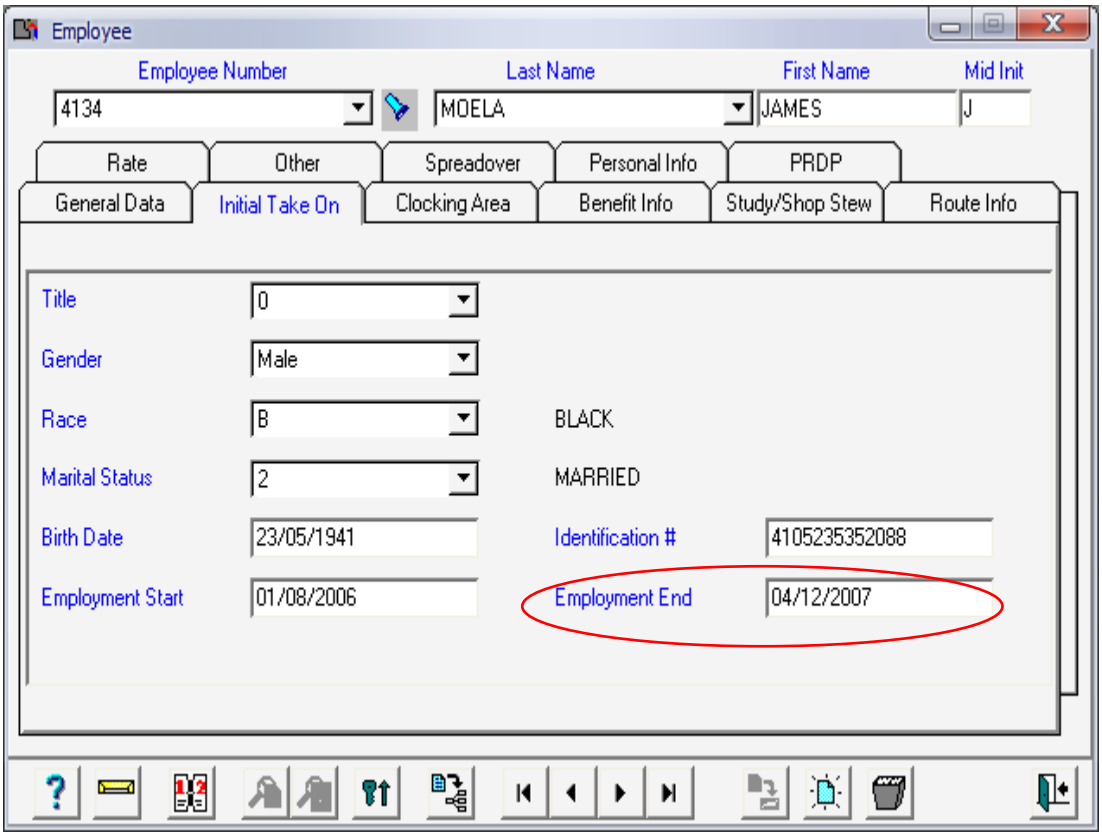


On the ImportDB, the end_date is set to 12/04/2007



After Running the program: the employee's status changed to 99 with termination start date that is greater by 1 day from employee's end date.

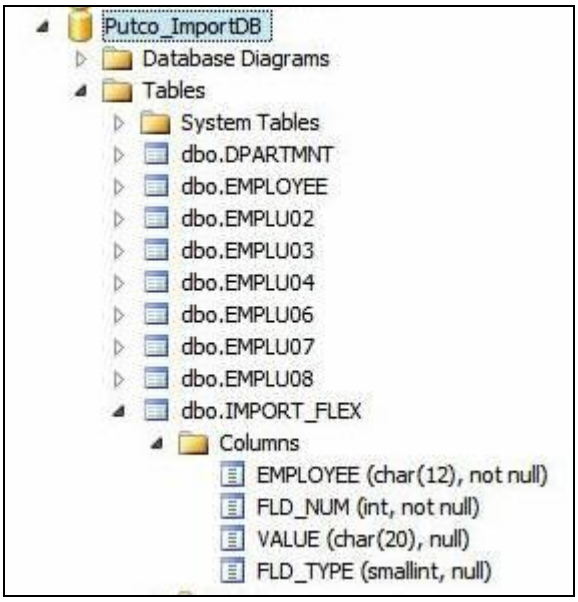

Employee
Employee Number
4134
Last Name
MOELA
First Name
JAMES
Mid Init
J
Rate
Other
Spreadover
Personal Info
PRDP
General Data
Initial Take On
Clocking Area
Benefit Info
Study/Shop Stew
Route Info
Updated
12/04/2007
13:45:32
Contract
1
Valid From
01/01/2000
Contract Name
5 Day 44hrs
More
Status
99
05/12/2007
31/12/2099
Description
Inactive
More
SBU
4101
Name
T.A.M - NEW YARD
Absence Code
Department
416000
Name
HOMELANDS
Absence Name
More
Section
506
Name
B KWAGGA/MACHIPE
Fingerprint ID
Validity
More
Pay Period Type
Weekly
Payroll Table
41
TAM PAYROLL


Employee
Employee Number
4134
Last Name
MOELA
First Name
JAMES
Mid Init
J
Rate
Other
Spreadover
Personal Info
PRDP
General Data
Initial Take On
Clocking Area
Benefit Info
Study/Shop Stew
Route Info
Title
0
Gender
Male
Race
B
BLACK
Marital Status
2
MARRIED
Birth Date
23/05/1941
Identification #
4105235352088
Employment Start
01/08/2006
Employment End
04/12/2007

7. Importing into Emplu tables:

On the Emplu tables each employee might have several records so importing can be done using the following procedure:

(1) On the Importdb database add a new table called Import_Flex. The table should have the attached structure.

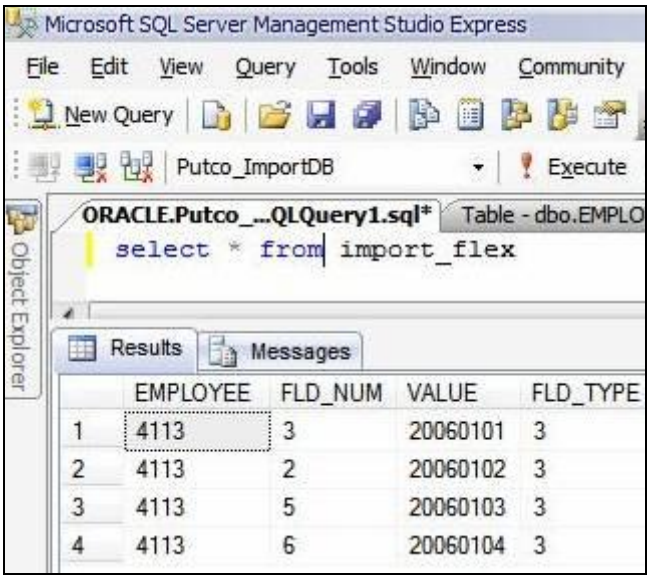


(2) On the map named tk2xmlmap add the following lines, **note that this table is case sensitive**.

Results | Messages

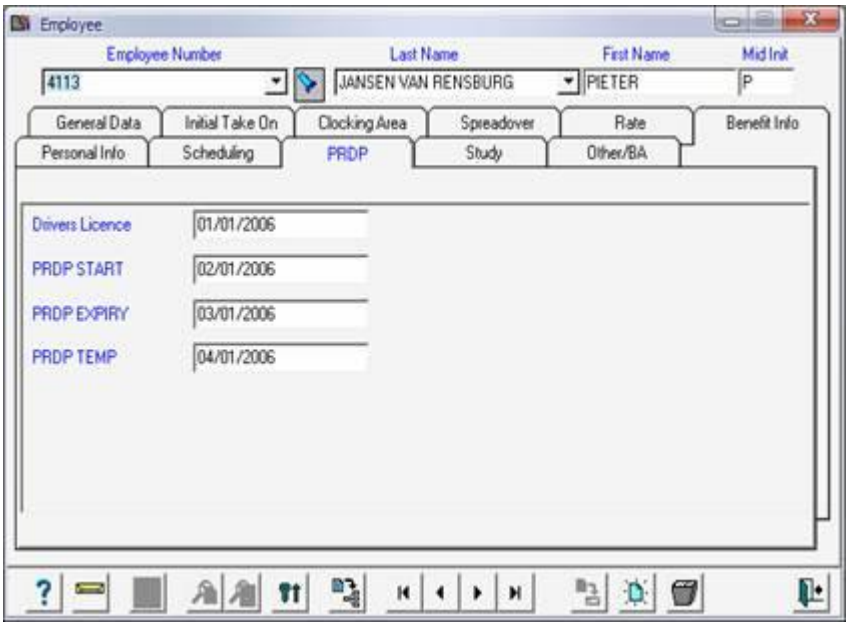
	INTEGRATIONKEY	S_TABLETYPE	S_ELEMENT	S_ATTRIBU...	DB_TABLE	DB_COLUMN	DESCRIPTION	DB_VALIDITY_DATE_COL	IS_WHERE_COL
1	Tk_XML	1	empflex	STAM_SHEM	IMPORT_FLEX	EMPLOYEE	EMP,14		1
2	TK_XML	1	empflex	FLD_NUM	IMPORT_FLEX	Fld_Num	Emp,14		0
3	TK_XML	1	empflex	VALUE	IMPORT_FLEX	Value	Emp,14		0
4	TK_XML	1	empflex	FLD_TYPE	IMPORT_FLEX	Fld_Type	Emp,14		0

(3) A sample data in Import_flex table



	EMPLOYEE	FLD_NUM	VALUE	FLD_TYPE
1	4113	3	20060101	3
2	4113	2	20060102	3
3	4113	5	20060103	3
4	4113	6	20060104	3

(4) Getting Result:



Note that the data for the employee must reside on the Import_master and on the Import_flex. So importing only from import_flex is not possible.

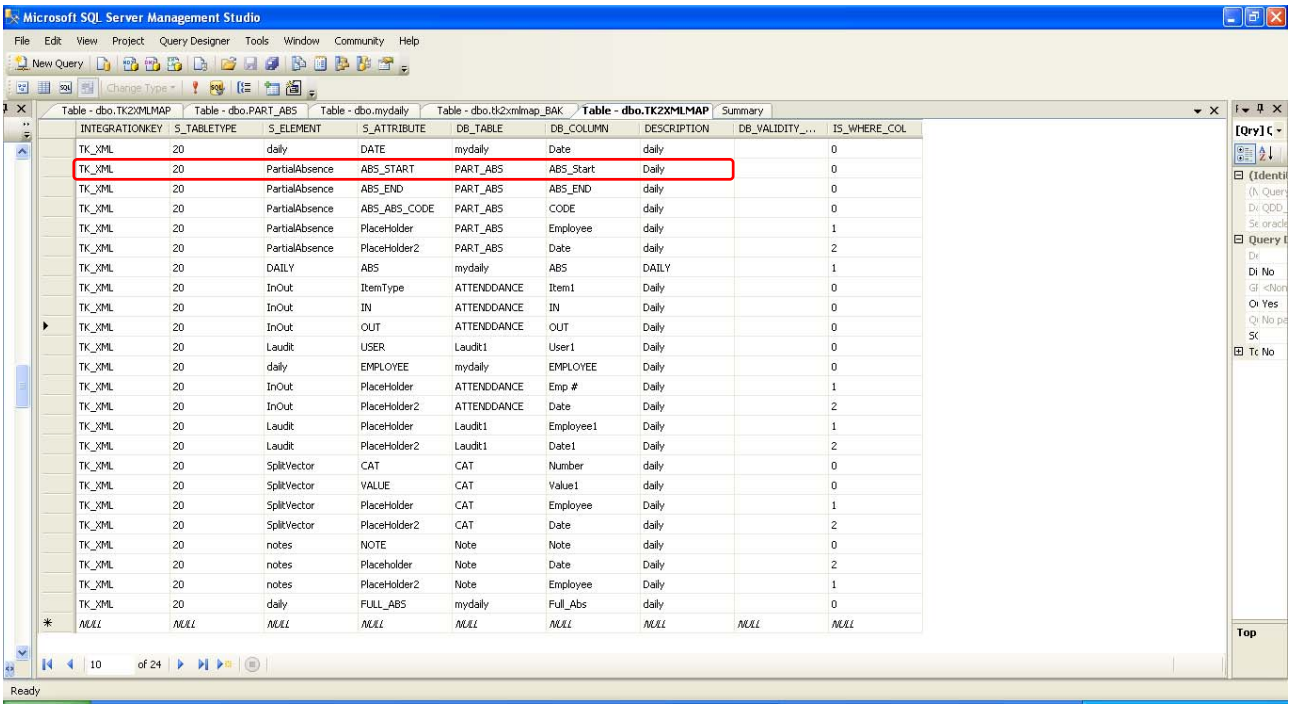
Import into Daily:

Daily information will be imported only after the Employee information has been imported (this is because daily information can only be imported for existing employees). If for some reason a record cannot be imported (e.g. the employee does not exist) this event will appear in the log file as an error.

In order to import the daily, every element that should be imported into TK needs to have its own table.

In this example you can see that in order to import the start time of the partial absence you need to define that the value from the ABS_Start DB_COLUMN inside of the PART_ABS DB_TABLE will be imported.

Full absence should be defined in the daily table while there should be a deferent table to partial absences.



This is the list of fields that will be imported.

Full Absence

#	Field	Element	Attribute
1	Employee	daily	EMPLOYEE
2	Date	daily	DATE
3	Absence Code	Daily	FULL_ABS
4		Commands	GENERATE

Note1: In the scenario that there is no day created, the "generate" command should be set to yes.

These fields should be added to the TK2XMLMAP table as below:

SQL-SERVER\IN...dbo.TK2XMLMAP | Summary

	INTEGRATIONKEY	S_TABLETYPE	S_ELEMENT	S_ATTRIBUTE	DB_TABLE	DB_COLUMN	DESCRIPTION	DB_VALIDITY_...	IS_WHERE_COL
	TK_XML	20	daily	DATE	IMPORT_DAILY	REC_DATE	DAILY		0
	TK_XML	20	daily	EMPLOYEE	IMPORT_DAILY	EMPLOYEE	DAILY		0
	TK_XML	20	daily	FULL_ABS	IMPORT_DAILY	FULL_ABS	DAILY		0
	TK_XML	20	Commands	GENERATE	IMPORT_DAILY	GENERATE	DAILY		0

79 of 80

Partial Absence

#	Field	Element	Attribute
1	Employee	daily	EMPLOYEE
2	Date	daily	DATE
3	Absence Code	PartialAbsence	ABS_ABS_CODE
4	Start Time	PartialAbsence	ABS_START
5	End Time	PartialAbsence	ABS_END
6		Commands	GENERATE

Note1: In the scenario that there is no day created, the "generate" command should be set to yes.

These fields should be added to the TK2XMLMAP table as below:

Table - dbo.TK2XMLMAP | Table - dbo.PART_ABS | Table - dbo.mydaily | Table - dbo.tk2xmlmap_BAK | **Table - dbo.TK2XMLMAP** | Summary

	INTEGRATIONKEY	S_TABLETYPE	S_ELEMENT	S_ATTRIBUTE	DB_TABLE	DB_COLUMN	DESCRIPTION	DB_VALIDITY_...	IS_WHERE_COL
	TK_XML	20	PartialAbsence	ABS_START	PART_ABS	ABS_Start	Daily		0
	TK_XML	20	PartialAbsence	ABS_END	PART_ABS	ABS_END	daily		0
	TK_XML	20	PartialAbsence	ABS_ABS_CODE	PART_ABS	CODE	daily		0
	TK_XML	20	PartialAbsence	PlaceHolder	PART_ABS	Employee	daily		1
	TK_XML	20	PartialAbsence	PlaceHolder2	PART_ABS	Date	daily		2

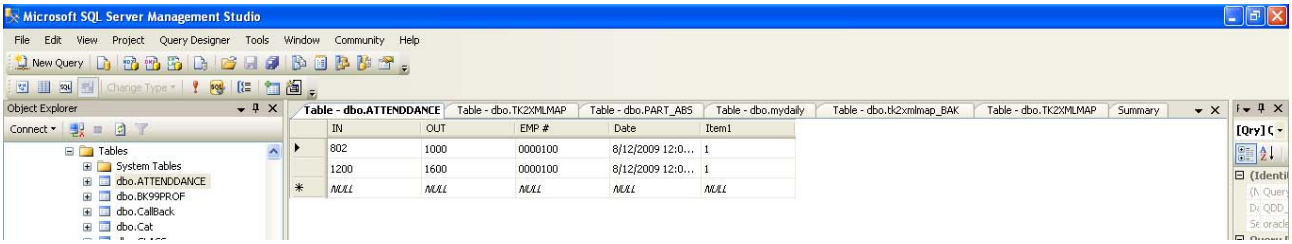
Example of Part_Abs table:

Table - dbo.TK2XMLMAP | **Table - dbo.PART_ABS** | Table - dbo.mydaily | Table - dbo.tk

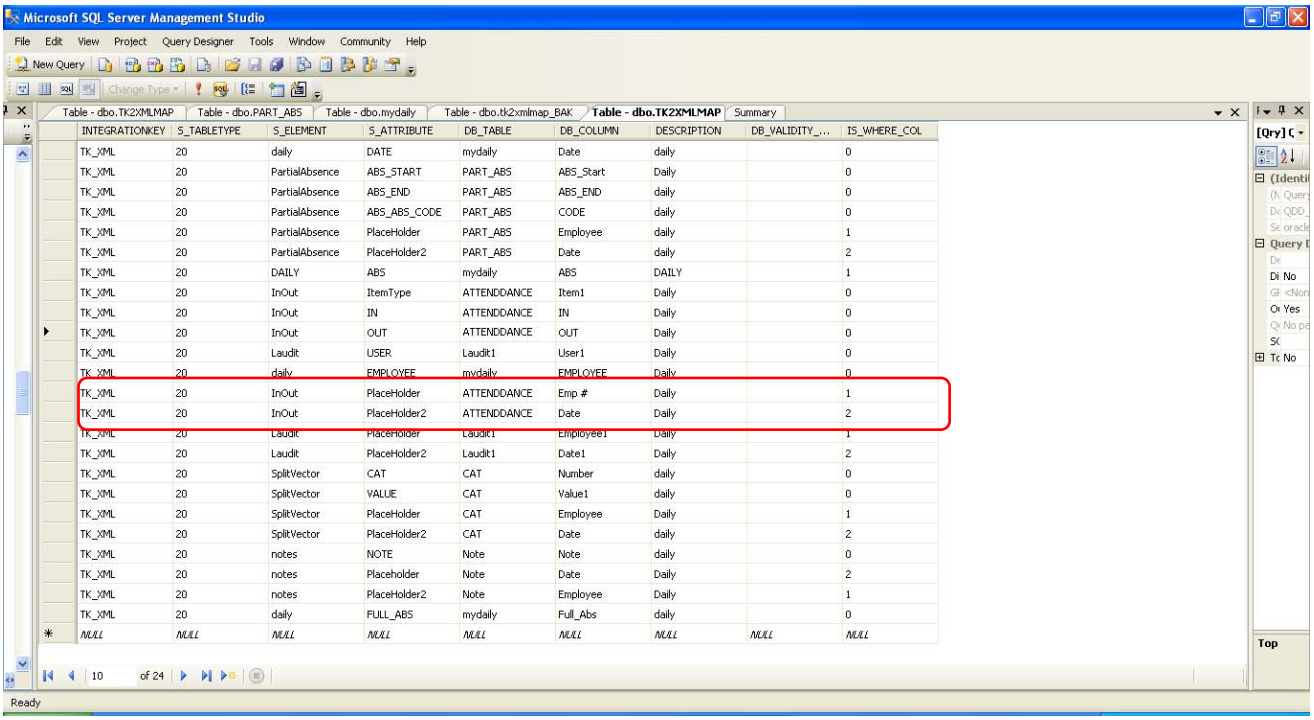
	ABS_Start	ABS_End	Employee	Date	CODE
	800	1600	0000100	8/13/2008 12:0...	14
*	NULL	NULL	NULL	NULL	NULL

Note2: When a partial absence record is generated it must include all fields (i.e. ABS_START, ABS_END, ABS_ABS_CODE). Meaning all values can either contain a value or a null. Having a null in one of these fields is impossible.

The IN and OUT should be also in a different table (e.g. ATTENDANCE table):



In order that the system will know which punches are belonging to each employee for example, it is necessary to open two lines with the S_ATRIBUTE “PlaceHolder” and “PlaceHolder2”.
The “PlaceHolder” will refer to the employee and the “PlaceHolder2” will refer to the date.
In the IS_WHERE_COL column it is necessary to use the 1 and 2 values.
1 refer to the employee and 2 refer to the date.



TSTTK_XML.exe

The TK_XML interface only imports data into TimeKeeper, but using the TSTTK_XML.exe program some fine tuning can be achieved. There are two ways to change data coming into TimeKeeper:

1. Assigning a procedure that runs after the employee import and before the commit.
2. Assigning a DLL that runs functions before and after the employee import.

Environment: TKSQL

Required program files:

TK_XML.dll 05/14/2006 (or later)
tkxmlGate.exe 05/09/2006 (or later)
TSTTK_XML.exe 03/19/2006 (or later)
XML Schemas

The functions exposed

The interface is described in detail in the document "The TKXMLGate Interface". There are seven functions, whose names are self-describing:

- ***GetSchemaFullPath*** gets the path for the schema used by the **TKXMLGate** to verify objects inserted into it, and to build the objects exported by it.
- ***InsertObjectAsXML*** inserts the object described by the XML passed to it into the TimeKeeper database.
- ***GetObject*** exports an object, using the relevant schema.
- ***GetLookupTable*** gets a list of codes and names in tables which correspond to the attributes of an employee, thus enabling the user, for instance, to pick a contract number based on the contract's description.
- ***ChangeKey*** exposes the functionality in TimeKeeper to change an employee's number.
- ***DeleteObject*** (not implemented for employees) allows third-party software to synchronize lookup tables, if they are managed by it: e.g., the list of departments in the calling software and in TimeKeeper can thus be managed at a single point.
- ***TruncateTable*** deletes all records in a table. Its main functionality is to allow initial synchronization of data when installing the system.

How to use

A sample application, ***TSTTK_XML.EXE***, schemas, and sample XML files are included. You can type parameters (or drop XML files) in the upper right quarter of the screen, and test their functionality. You will need a TKSQL database and Internet Explorer 6.0 (at least) on your development machine. You'll see the results in the lower right window of the screen.

C programmers can see, by clicking the 'details' button, the function declaration.

A JavaScript file is also included, to show the COM automation features.

Sample files

The 'samples' folder holds sub-folders with five exercises, which can give you the gist of the use of the **TKXMLGate.**

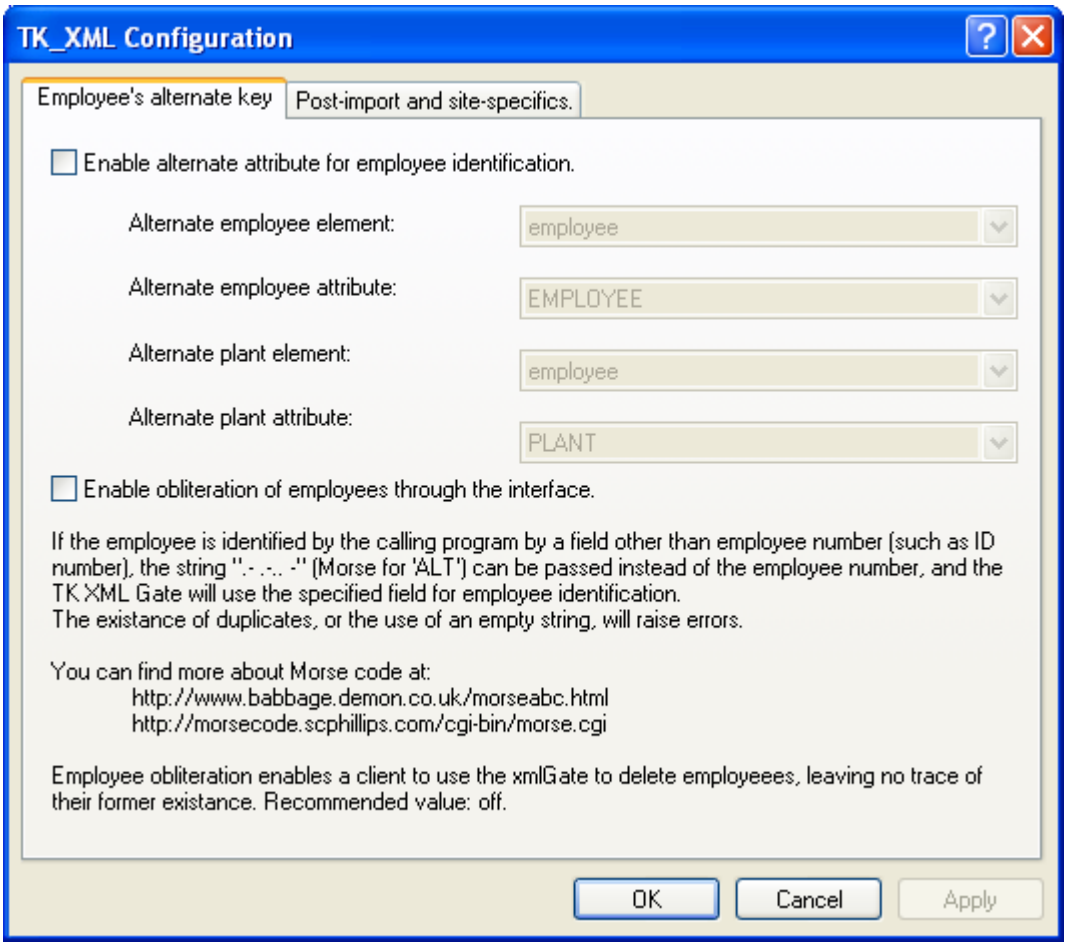
Note: *You will have to update the schemas referenced in the sample files to match those returned by the function* ***GetSchemaFullPath().***

- The folder **samples\exercise_1** holds an XML file which shows how to insert a record in table SORT0.
- The folder **samples\exercise_2** holds the following files:
    1. ***Get_plants.xml*** shows the details of three plants.
    2. ***Update_plants.xml*** inserts three new plants, and changes some of the details in the existing ones.
    3. ***Delete_plants.xml*** can be used to delete the three plants just inserted.
    4. ***Reset_plants.xml*** can be used to return the table to its initial state.
- The folder **samples\exercise_3** shows the results of function ***GetObject***, used with an employee. The employee's number should match one of the employees in your database.
- The folder **samples\exercise_4** shows how to change an employee's key, and how errors are reported.
- The folder **samples\exercise_5** shows how to create an employee, and update several of the employee's properties. A SQL file, which shows the tests to the modified data, is also enclosed.

When clicking on the “configuration” button, the TK_XML Configuration screen will be opened.

Employee’s alternate key tab:

The “Enable alternate attribute for employee identification” enable to identify employee with other attribute.
This is useful also when there are few employees with the same name.

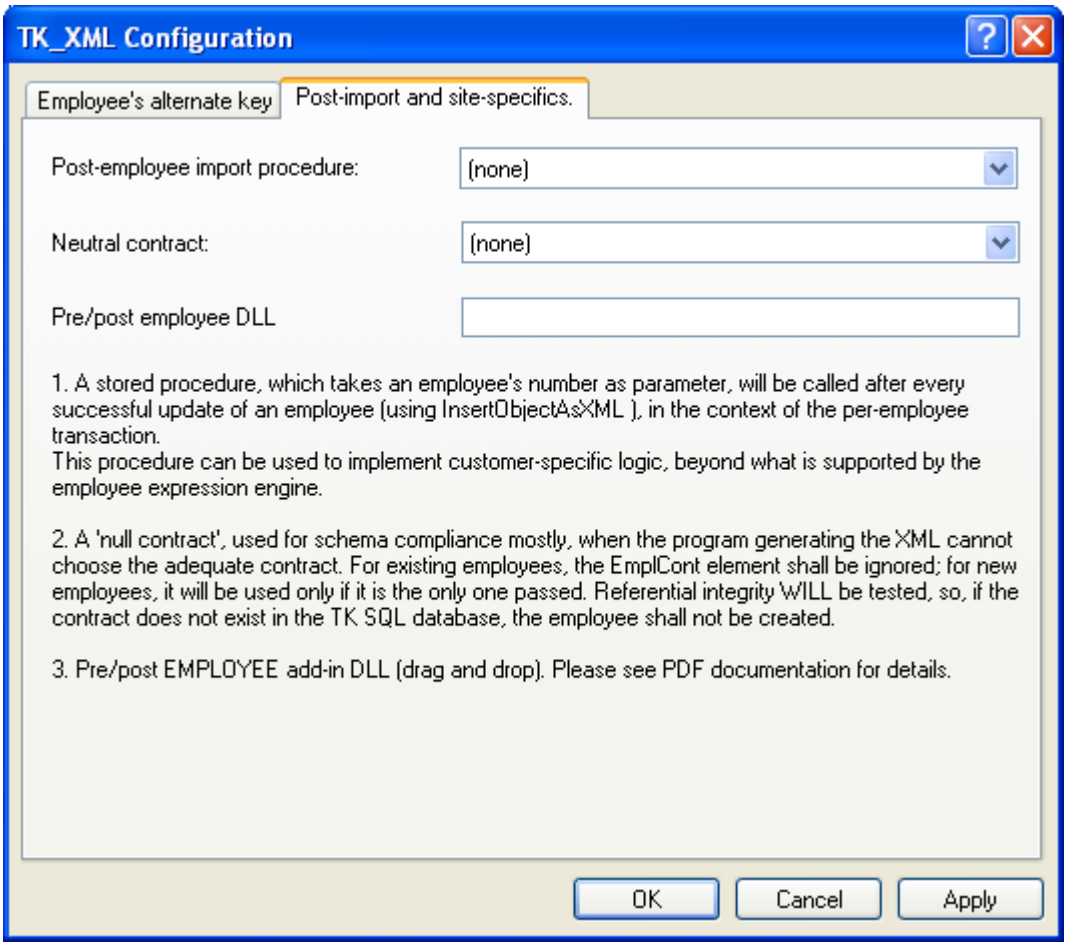


Post import and site specifics tab:

Post employee import procedure: when using QDD_SAP you can use define which procedure will run when the data will be imported. The programmer will write the procedure.

Neutral contract: when importing employee, a contract is a mandatory field. If there is no contract in the imported data, a default contract can be imported to the system in order to avoid from future problems.

Pre/post employee DLL: it is possible to use a dll that will do some actions before or after the import. For example: when importing ID number and there are some employees that have already one, it is possible to save the existing ID number with the dll before the import and after importing the new ID number, replace it back to the old one.



For more information how to use the TKXML gate interface please refer to the TKXML Gate Interface.PDF file.