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.

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.