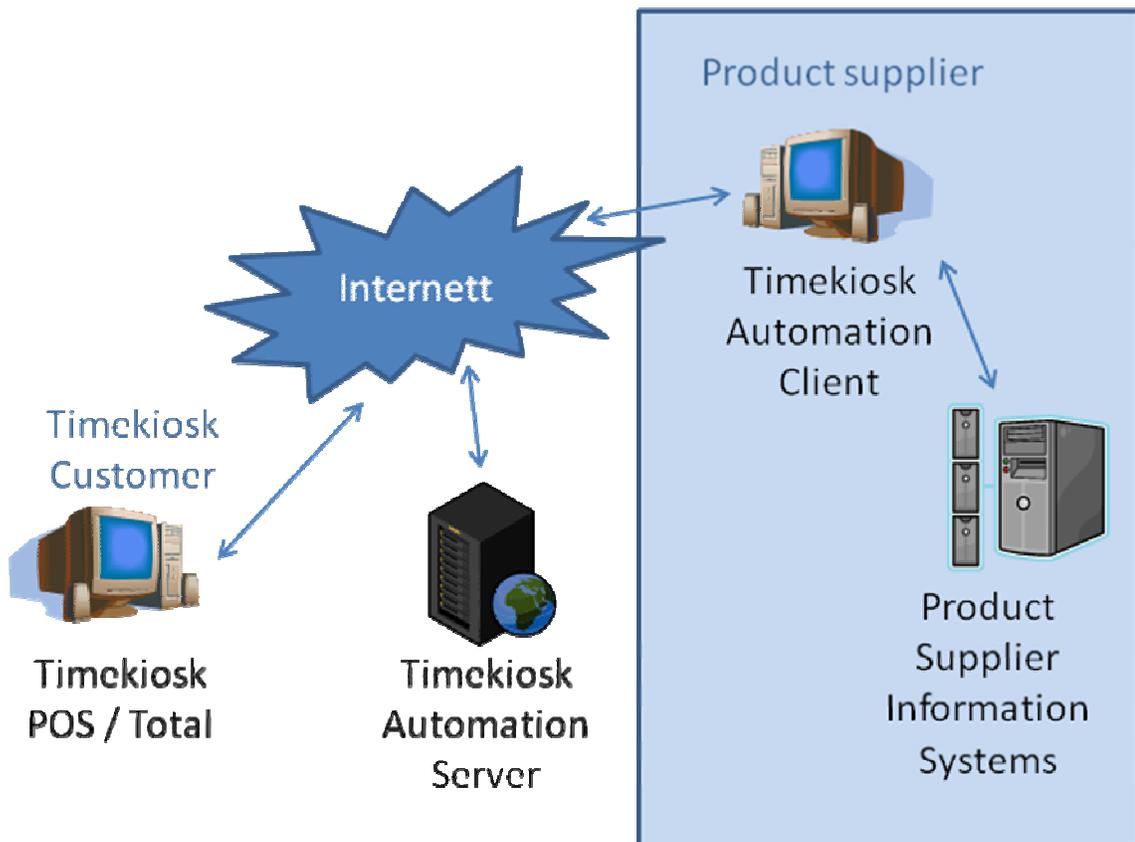


Timekiosk Automation – Automation Client

Date: 31.10.2008
Version: 0.02a



Introduction.

The intention of the Timekiosk Automation System is to allow Timekiosk customers (aka Providers) and Suppliers of products to perform data exchange operations with the Timekiosk central databases. Suppliers may as a result retrieve and update Provider product and stock information on behalf of one or more Providers.

The architecture is based on a central Web Service host and a local Automation Client designed to support easy data exchange operations with Business tools like Microsoft Excel™ and legacy systems like SAP™ or Visma™ Business System.

Numerous interfaces and data exchange operation modes are support.

All data transport between the Timekiosk Automation Client and the Automation Server is performed using standard Web Services and should in most cases not influence on any existing firewall security measures and infrastructure. Data is transported using a simple encryption method, but stronger encryption may be added as desired.

The Automation Client is a simple .Net "install-by-copy" set of files and libraries using open standard XML configuration files and command line arguments to control operations. Data is exchanged with local parties using text files with TAB separated fields. Other methods may be added as desired.

Security is maintained by having the central system controlling what operations are allowed and with whom these may be performed in every detail. A separate AccessCode needs to be configured in the Automation Client to gain access to any operation.

Changelog.

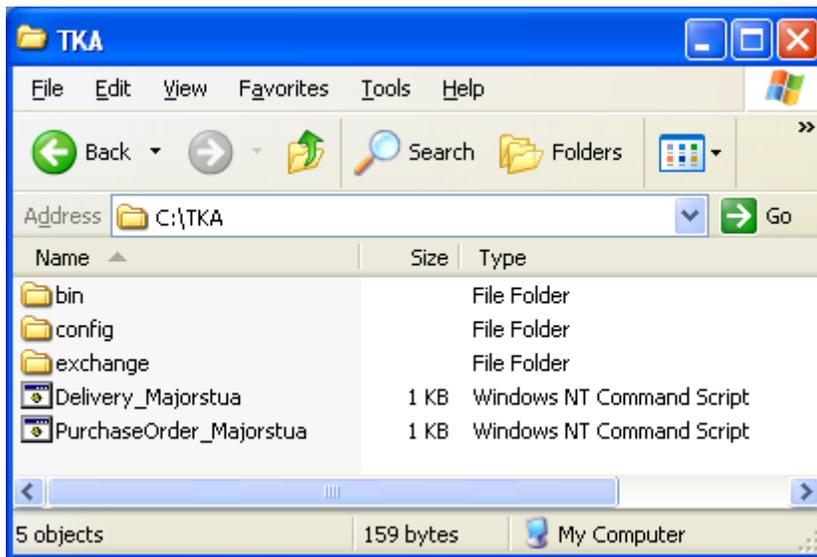
Date / Ver.	
19.05.2008: 0.01c	Changelog section added.
31.10.2008: 0.02a	Some text and graphics changes to documentation. Product delete has been added. Wrappers per response line and multiple datasets added.

Purchase List / Inventory

The Timekiosk Automation Purchase Module is designed to allow any party having granted access insight into the stock information where the Supplier in speak is in fact a designated Supplier.

In order to read a purchase list the following method may be utilized as described below.

Assuming a "C:\TKA" installation path the files and structure may be as shown here:



The "bin" directory contains the Automation binary files and should not be altered in any way. The "config" directory contains the XML configuration files. The "exchange" directory is a sample data exchange directory. The "CMD" files are sample Automation launch files.

Sample launch procedure:

```
c:  
cd \tka  
bin\AutomationClient.exe /config="config\PurchaseOrder_Maj.xml"
```

The launch procedure may be included in most batch routines, scripts and programs that do allow applications to be started.

Sample config file:

```
<?xml version="1.0" encoding="utf-8" ?>  
<Configuration>  
  
  <!-- Automation identifier and access code -->  
  <AccessCode>987u8f7s76gfs789fgssfg4232dfgdF7g</AccessCode>
```

```

<!-- Specify if local target file is to be DataSet encoded and sent
      as part of request or some other operation as added -->
<LocalRequestOperation>None</LocalRequestOperation>

<!-- Server URL path. -->
<URL>http://automation.timekiosk.com/AutomationServer.asmx</URL>

<!-- Auto Close after completion. -->
<AutoClose>Off</AutoClose>

<!-- Include header information in input/output file -->
<RequestHeaders>Off</RequestHeaders>
<ResponseHeaders>On</ResponseHeaders>

<!-- Wrapper to be used per tableset returned, -0- is the first,
      -1- is the second. Specifying a wrapper will suppress ResponseHeaders -->
<ResponseWrapper_0></ResponseWrapper_0>
<ResponseWrapper_1></ResponseWrapper_1>

<!-- The path and filename of the output or input file.
      Optional replaces: [] Year, Ye, Mo, Da, Ho, Mi, Se
      The target filename may also be set by command line argument,
      which then does have priority over this entry -->
<TargetPath>C:\TKA\exchange\PO_[Ye][Mo][Da]_[Ho][Mi][Se].txt</TargetPath>

<!-- Selection parms used by server system routines.
      Parms does vary with automation routine being utilized -->
<Parms>
  <BRAND></BRAND>
  <SERIES></SERIES>
  <NAME></NAME>
  <FILTERLVL>0</FILTERLVL>
</Parms>
</Configuration>

```

The configuration shown above will request a Purchase list from the Automation Server and save the resulting file into the "exchange" directory using date and time as the trailing part of the file name. No other input file is needed other than the XML configuration file as shown above to perform the operation. The resulting file may easily be read into Excel by simple copy-and-paste or may be read directly into legacy systems via their respective import routines.

If specific formatting of the response files is required the ResponseWrapper fields may be configured. Below is a sample where two datasets (non-standard) are used to create the purchase data file. The first dataset and single header line (ResponseWrapper_0) may be recognized by the importing party from the "OH" (OrderHeader) prefix. This line typically contains provider address and overall ordering information. Fields that may be used are agreed upon per Automation relationship.

The second line (ResponseWrapper_1) in the sample below is the template for the repeated orderlines. When using such a wrapper most fields are usually not included in the order detail lines and this may make the file more easy to read and adapt to some import scenarios.

```

<ResponseWrapper_0>"OH",14456,"[PROVIDEDNAME]","[ADDRESS1]",[ZIP]
</ResponseWrapper_0>
<ResponseWrapper_1>"OL","[ITEMNO]","[PURCHASECOUNT]"</ResponseWrapper_1>

```

The fields "BRAND", "SERIES" and "NAME" may be used to filter out the purchase/stock list to include only desired products. The value used as a filter will be interpreted as a wildchar (%vlu%) term when retrieving the data records.

The field "FILTERLVL" may be used to limit the returned records by requiring the number of items being ordered to exceed some specific level. Hence, specifying a value of "5" will return a purchase list containing only products for which the number items with an ordering count exceeding "5". If specifying "-1" all stock controlled products will be returned as all "0" order count products are also included.

The fields in the repeated orderlines created as a result of the standard Automation Purchase routine are as specified below:

PRODUCTID	Timekiosk Unique Product ID
BRAND	The Brand of the product
SERIES	The Series description of the product
PRODUCTNAME	The name of the product
BARCODE	The Primary Barcode set for the product
ITEMNO	The Supplier Item number / Product number
COSTPRICE	The Cost price as registered in the Provider database
PRICE	The street/retail price
NUMBER_INSTOCK	Number of items in stock
ORDER_POINT	The target stock level
ORDER_COUNT	The preferred order count. Set as zero for top-up ordering
PURCHASECOUNT	The actual number being requested/purchased
STOCKID	The unique ID of the stock being checked
SUPPLIERID	The Timekiosk unique supplier ID
SUPPLIER	Name of the supplier

Other information may be added as agreed upon per Automation relationship.

Stock / Product update.

The stock and product Automation module is designed to allow a product Supplier to add stock and product information to the databases. This includes adding products, updating product information, adding stock level parameters and adding stock level changes to the stock register.

Assuming a "C:\TKA" installation the procedure to update product and/or stock information is as described below.

Sample launch procedure:

```
c:  
cd \tka  
bin\AutomationClient /config="config\Delivery_Maj.xml"
```

The launch procedure may be included in most batch routines, scripts and programs that do allow applications to be started.

Sample config file:

```
<?xml version="1.0" encoding="utf-8" ?>  
<Configuration>  
  
  <!-- Automation identifier and access code -->  
  <AccessCode>jksdf78768fs8678sdfgsd87fsdf</AccessCode>  
  
  <!-- Specify if local target file is to be DataSet encoded and  
    sent as part of request or some other operation as added -->  
  <LocalRequestOperation>TargetFileToDataSet</LocalRequestOperation>  
  
  <!-- Server URL path. -->  
  <URL>http://automation.timekiosk.com/AutomationServer.asmx</URL>  
  
  <!-- Auto Close after completion. -->  
  <AutoClose>Off</AutoClose>  
  
  <!-- Include header information in input/output file -->  
  <RequestHeaders>Off</RequestHeaders>  
  <ResponseHeaders>On</ResponseHeaders>  
  
  <!-- The path and filename of the output or input file.  
    Optional replaces: [] Ye, Mo, Da, Ho, Mi, Se -->  
  <!-- The target filename may also be set by command line argument,  
    which then does have priority over this entry -->  
  <TargetPath>C:\TKA\exchange\Delivery.txt</TargetPath>  
  
  <!-- UPDATE_TYPEENT: 6=ItemNumber, 7=Primary Barcode -->  
  <!-- UPDATE_REFERENCE may also be specified as a command line argument -->  
<Parms>  
  <UPDATE_TYPEENT>6</UPDATE_TYPEENT>  
  <UPDATE_REFERENCE>PO# 98234789</UPDATE_REFERENCE>  
</Parms>  
  
</Configuration>
```

The configuration file as defined above will read a text file and submit this as part of the request to the Automation Server, in order for the Automation routine to perform update operations. The resulting file will be saved in a file named using a trailing "_RESPONSE" text after the original filename.

The field "UPDATE_TYPEENT" may contain the number "6" for selecting Item number matching against the central databases. Specifying a value of "7" will tell the Automation routine to match against the primary Barcode set per product.

"UPDATE_REFERENCE" is typically the order number related to a delivery update operation. The value set will be added to each stock level change transaction allowing future tracking of stock level changes.

Automation Product Update mode parameters:

The Automation routine will be configured centrally to allow or disallow the creation of products and/or the creation of product to supplier to product relationship. Please do contact Timekiosk support to discuss details regarding these configuration options.

The options have the following meaning:

Create products	If full product definitions are included with the update any product that is not already in the database will be created using the data specified. If products should be created, this option needs to be set. If this option is set also the "Create supplier" option needs to be set.
Create supplier	If the updating process of stock/products also should add any missing Supplier to Product/Provider relationships, this option is required.

Input data to Stock / Product update.

Any data input should be formatted according to the local culture used in the Windows operating system in speak. Hence, when adding data to the system in Norway any floating point number should use a comma (",") as a decimal separator. Any data correctly input will be converted to the system format during import. Each column of data is separated by a single TAB character, which is the default separating character when pasting data to and from Excel.

Input data column/field data specification:

Column	Content / description
1	Specify a number between 1 and 4 for stock update operation as follows: 1=IN (add to current level / count) 2=OUT (subtract from current level / count) 3=SALE (subtract from current level / count) 4=SET (set the actual level / count for the product) Option 3 and 4 should normally not be used by Suppliers, unless they do perform actual sales and/or stock accounting on behalf of the Provider.
2	Stock level change. Always specify as a positive integer number.
3	The value used for looking up the product. This should be set according to the selection made in the "Update indextype" header field (see above). This may be the Suppliers internal item number, the barcode or the product name valid for the product in speak. The value is case sensitive and any discrepancy in format or value will hinder the lookup of the appropriate product from the database.
Optional fields used if "Create Products" option has been configured	
4	Brand. String value. This is the top-level category text used by the system.
5	Series. String value. This is the second-level category text used by the system.
6	Name of product. String value.
7	Barcode. String value.
8	Item number. String value.
9	Price. Floating point value.
10	Purchase price. Floating point value.
11	VAT. Floating point value (Eq. 12% = 0,12 or 0.12). If using default VAT value this value may be ignored/omitted by specifying an asterix ("*") as the field value or specifying no value.
Optional add-on fields used to set Stock Parameters controlling purchase	
4 or 12	Order Point value. Specifies the low trigger level for purchase.
5 or 13	Order Count value. The count of items to order. If set to -0- the Order Point will be considered the Stock Level target when calculating the order count.

Result report:

The Import Response report file does have one or more of the entries/lines being a status report from the Automation routine being run. The three TAB separated columns included are as follows:

- #1: The status code where -0- means OK
- #2: The status text
- #3: The Item number / Barcode referencing the update

The status word(s) has the following meanings:

- NOT FOUND: The system was not able to find the product specified (no update)
- OK: The system did find a single product (update ok)
- MULTIPLE: The system did find two or more products (no update)
- OK/NEW: The system created the product (update ok)
- ERROR: Some error prevented the system from updating the product

If using Excel as a source of import data one may copy-and-paste the report back into the spreadsheet in order to easily get an overview of successful and failed import data and sort the spreadsheet to eliminate successful updates.

Delete products.

The delete products Automation module is designed to allow a product Supplier to delete products from the Provider database. Only products for which the supplier has a defined relationship as a Supplier may be deleted.

Assuming a "C:\TKA" installation the procedure to delete product information is as described below.

Sample launch procedure:

```
c:  
cd \tka  
bin\AutomationClient /config="config\Delete_Maj.xml"
```

The launch procedure may be included in most batch routines, scripts and programs that do allow applications to be started.

Sample config file:

```
<?xml version="1.0" encoding="utf-8" ?>  
<Configuration>  
  
  <!-- Automation identifier and access code -->  
  <AccessCode>jksd9878df7gfd6g786fdb87fsdf</AccessCode>  
  
  <!-- Specify if local target file is to be DataSet encoded and  
    sent as part of request or some other operation as added -->  
  <LocalRequestOperation>TargetFileToDataSet</LocalRequestOperation>  
  
  <!-- Server URL path. -->  
  <URL>http://automation.timekiosk.com/AutomationServer.asmx</URL>  
  
  <!-- Auto Close after completion. -->  
  <AutoClose>Off</AutoClose>  
  
  <!-- Include header information in input/output file -->  
  <RequestHeaders>Off</RequestHeaders>  
  <ResponseHeaders>On</ResponseHeaders>  
  
  <!-- The path and filename of the output or input file.  
    Optional replaces: [] Ye, Mo, Da, Ho, Mi, Se -->  
  <!-- The target filename may also be set by command line argument,  
    which then does have priority over this entry -->  
  <TargetPath>C:\TKA\exchange\Delete.txt</TargetPath>  
  
  <!-- UPDATE_TYPEENT: 6=ItemNumber, 7=Primary Barcode -->  
<Params>  
  <UPDATE_TYPEENT>6</UPDATE_TYPEENT>  
</Params>  
</Configuration>
```

The configuration file as defined above will read a text file and submit this as part of the request to the Automation Server, in order for the Automation routine to perform delete operations. The resulting file will be saved in a file named using a trailing "_RESPONSE" text after the original filename.

The field "UPDATE_TYPEENT" may contain the number "6" for selecting Item number matching against the central databases. Specifying a value of "7" will tell the Automation routine to match against the primary Barcode set per product.

Input data to Product delete.

Any data input should be formatted according to the local culture used in the Windows operating system in speak. The product that is to be deleted should be specified as a column single value on a separate line. Do not add any TAB characters or other special characters other than a standard lineshift at the end of each line.

Example content:

```
234234234  
891892-S  
234442-SA
```

Result report:

The Product Delete Response report file does have one or more of the entries/lines being a status report from the Automation routine being run. The three TAB separated columns included are as follows:

- #1: The status code where -0- means OK
- #2: The status text
- #3: The Item number / Barcode referencing the update