Tracking Module

The module that returns information about the current shipping status of a given package is implemented as a DLL that can be referenced by your Visual Basic or Visual C++ program.

To use this module, you create a `Tracking` object which gives you access to the following methods:

InitTracking(name As String, testingMode As Boolean) As Integer

This method initialized the Tracking module.

Parameters:

name	The full path and name of the database that contains the tracking information table.
testingMode	If this value it TRUE, smaller time units are used to determine the next change of status for a package's movement. You will want this value to be set to TRUE during all your testing.

Return Value:

Returns 0 if there is an error during the initialization procedure, otherwise, returns 1.

GetTrackingNumber(isRMA As Boolean) As Long

This method generates a new Tracking Number for each shipment and returns this new Tracking Number.

Parameter:

isRMA	Flag to determine if this shipment is an RMA or a normal shipment. If this value is TRUE, the tracking number is assigned to the RMA shipment and the initial status is WAITING FOR RETURN PICKUP. If this value is FALSE, the initial status assigned is WAITING FOR PICKUP.

Return Value:

Returns the long integer that denotes the newly assigned tracking number. This number is used to query the tracking module in the future to determine the current status of the package.

CheckStatus(trackingNumber As Long) As Long

The method returns the current status of package with the given tracking number.

Parameter:

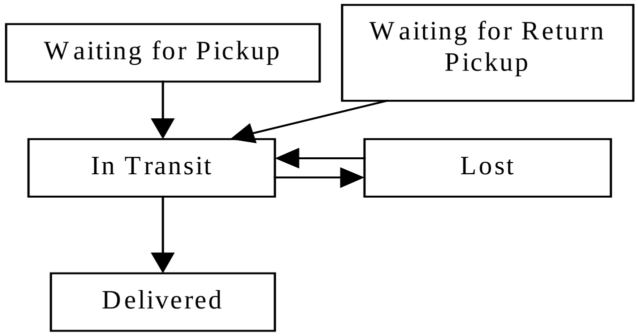
trackingNumber	The tracking number of the package which status you want to check.

Return Value:

Returns the current status of the package that is being shipped. The possible values are:

Status	**Value**
Waiting For Pickup	1
In Transit	2
Waiting For Return Pickup	3
Delivered	4
Lost	5

Transition of Shipping Status



This is a diagram of how the status of the package changes during the shipping process.

How to Incorporate this Module

Follow these steps to incorporate the tracking module into your PC ordering system.

What to add to your database

Add these two tables with the fields (names and properties) as described below: (*Note:* You should not need to modify these tables except to add them to your database)

Table: tblTrackingInformation

Field Name	**Data Type**	**Description**
TrackingNumber	AutoNumber	The primary key of this table that is the tracking number assigned to each shipment.
StatusID	Number(Long Integer)	The foreign key from `tblShipmentStatus` that designates the current status of the shipment.
TimeStamp	Date/Time	The date and time of the last change of status for this shipment.

The data for this table will be generated as the module is used and tracking numbers are assigned.

Table: tblShipmentStatus

Field Name	**Data Type**	**Description**
StatusID	AutoNumber	The primary key that uniquely identifies each possible shipment status.
StatusDescription	Text(50 characters)	The word description of the status.

Put the following data in this table when you first create it. The status id will be automatically generated as you put in the entries in the following order:

Status ID:	**Status Description:**
1	Waiting for Pickup
2	In Transit
3	Waiting for Return Pickup
4	Delivered
5	Lost

*Accessing the DLL within Visual Basic *** Changed from 1st document version****

1. Copy the directory `//ifilesrv1/students/molly/TrackingModule` to your working directory (all the files in this directory will come with it).
2. Go to your new version of the `TrackingModule` directory and open the `Tracking.vbp` Visual Basic project.
3. Choose `File|Make OLE DLL File` … from the menu bar.
4. Set the file name to `Tracking.dll` in your new `TrackingModule` directory and click `OK`. This will rebuild the Tracking.dll in your new `TrackingModule` directory.

From with the Visual Basic development enviroment:

1. Choose `Tools|References` from the menu bar.
2. Click on the `Browse` button.
3. Locate the `//[your working directory]/TrackingModule/Tracking.dll` file in the file dialog that pops up and click `OK`.
4. The Tracking Module will now appear in your list of available references with the check box checked. Click `OK`.
5. You can now declare a new clsTracking within your VB code to create a new instance of the tracking module. You can access the methods of the tracking module as shown below:

```
Dim myTrackingObject as new clsTracking
Dim returnValue As Integer
Dim newTrackingNumber As Long
Dim currentStatus As Long

returnValue = myTrackingObject.InitTracking ("order.db", TRUE)
newTrackingNumber = myTracking.GetTrackingNumber(FALSE)
currentStatus = myTracking.CheckStatus(newTrackingNumber)
```

Accessing the DLL within Visual C++

If you are using Visual C++, we will have to do some other stuff to get it to work with your code. I think that only one group is using this and I will work with them to get this working correctly.