

Titanium Software, Inc.

P.O. Box 980788 Houston, TX 77098

713-942-7296

Technical Support: 281-443-3544 or Support@TitaniumSoftware.com

Titanium Schedule: Client Import Feature

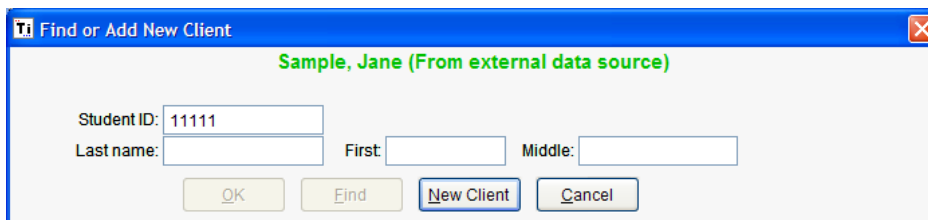
This feature is an add-in to the basic Titanium Schedule application. There is an additional charge to activate this feature.

The Client Import feature is designed to import information from a single outside data source into Titanium. Titanium does this one record at a time as needed. We don't import the entire external data source into Titanium since that would add thousands of records to the Titanium client table that will probably never be seen in the center.

How it works from a user perspective

When adding a new client:

When a user adds a new client by entering the **Student ID** number in the **Find or Add New Client** dialog, and then presses the tab key to leave that field, one of the following two things will happen. If that student ID is found in Titanium, then the student's name will be displayed in black at the top of the dialog as usual. If the student wasn't found in Titanium, the software will also look in the external data source. If the student is found there, then their name will be displayed in green at the top of the dialog along with a comment like *(From external data source)*.



If you then click the **New Client** button, a new client will be added to Titanium and some of the fields will be populated based on the information found in the external data source. (Depending on your configuration, it might also add a Data Form and populate some fields it.)

Updating an existing client:

There are some configuration options that control when an existing client is updated from the external data source. In most cases, this is completely transparent to the user. The most common setting is to have it update the client information the first time a client comes back to the center during a new semester. The first time an appointment is saved for a client in a semester, the client information will automatically be updated. There is no action required by the user.

Requirements

- The external data source must be an ODBC-compliant data source (Microsoft SQL Server and Oracle are the most commonly used data sources)
- All of the users of Titanium must have the appropriate ODBC driver installed. (The easiest data source to work with is a Microsoft SQL Server since the ODBC drivers for this are almost always already installed.)
- Titanium can only query a single table/view. So, if you need to pull data from multiple tables, you will need to combine those into a single view for use by Titanium.
- You will need to setup the security on your remote data source so that all of the Titanium users can view the query. (They won't need to update, insert, or delete rights.)
- The Client record and only one Data Form can be updated from the incoming information. (e.g. you can't update two different Data Forms at the same time. This means all the fields that you want to update must be on a single Data Form)

Configuration

First, your computer support people will need to setup the external data source and set the security on it.

Make sure all the users' computers have a required ODBC driver to connect to the external data source. Usually there is no reason to create an entry in the Windows ODBC data sources. The connection string can just be specified in Titanium.

The remaining configuration is done from inside Titanium.

Start Titanium and login in as a *System* level user.

From the main menu, select **Configure>System Configuration** then display the **Client Import** tab on the System Configuration screen. You should have a screen similar to the one below. Most of the items on the screen have popup help if you place your mouse cursor over the field label.

The screenshot shows the 'System Configuration' dialog box with the 'Client Import' tab selected. The dialog has a title bar with a close button. Below the title bar are several tabs: 'Users and Schedules', 'Appointments', 'Client', 'Schedule Viewer Layout', 'Schedule Viewer Colors', 'Reports', 'Security', 'Billing', 'Notes & Data Forms', 'Task List', 'Client Import', 'Miscellaneous', 'Exchange', and 'Web Component'. The 'Client Import' tab is active. A text box at the top of the tab contains a warning: 'The feature can reduce some data entry by pulling information into Titanium from an external data source. It is NOT designed to be used in conjunction with the Titanium Web Component since it is expected that information provided by the client a few minutes ago should not be overwritten by older information from an external data source.' Below this is a checked checkbox for 'Activate Client Import' and a text field for 'Client import activation code:'. The 'ODBC Connection string:' field contains 'DSN=MS Access Database; DBQ=C:\Ti\Devel\Schedule\Ver9\Source\SampleClient'. The 'Table/View name:' field contains 'STUDENTS'. There are 'Test' and 'Query' buttons. The 'Student ID expression:' field contains 'COLL_ID'. The 'Client name expression:' field contains 'rtrim(C_LAST)+', '+rtrim(C_FIRST)'. There are several dropdown menus: 'When to import client information:' (set to 'When adding a new client'), 'When to create a new Data Form:' (set to 'When adding a new client'), 'Data Form:' (set to 'Client Demographics'), and 'Default note type for a new Data Form:' (set to 'Client Demographics'). The 'When to check enrollment status:' dropdown is set to 'Never'. There is an 'Enrollment expression:' field. Two checkboxes are checked: 'Add a button to the client screen to let them manually import the client' and 'Add a button to the Find Client screen to view raw client import data'. There are buttons for 'Client Import Field Cross Reference Table' and 'Bulk Processing'. At the bottom are 'OK', 'Cancel', and 'Apply' buttons. A footer note states: 'The blue System settings may be overridden by User Configuration settings. Changes made here will not be visible to other users until the next time they start Titanium Schedule.'

Client import activation code:

Enter your client import activation code. Please contact us if you wish to purchase this feature and need a code. Or, if you have already purchased it and need to have your code sent to you. (You can configure this feature without an activation code, however the Titanium users won't be able to use it until the activation code is entered.)

ODBC Connection string:

This is the ODBC connection string to the external data source setup by your computer people. It can be in various formats depending on if your data source is Microsoft SQL Server, Oracle, or something else. For example, for Microsoft SQL Server it will be:

```
DRIVER=SQL Server;SERVER=SQLServerName;DATABASE=DatabaseName;uid=;pwd=;
```

The SQL Server name can be a name or an IP address and should contain the SQL Server instance name if there is one. You can also specify a port by following the instance name with a comma then the port number. For example, a computer called *OurServer* at IP Address *999.999.999.999* with an instance of SQL Server named *SQLExpress* and listening to port number *1433* would be entered as:

```
OurServer\SQLExpress,1433 or as 999.999.999.999\SQLExpress,1433.
```

With SQL Server, the **uid** (user name) and **pwd** (password) will be blank if you are using the preferred security approach on the SQL Server of Windows Authenticated security. If you use SQL Authenticated security, then you will need to specify the user name and password in the connection string.

Table/view name:

The client import routine will run a query to retrieve a single client record matching a specific Student ID from the external data source. This field contains the name of the table or view that will be queried for this record. The computer support person that setup the ODBC data source will be able to provide this information.

Test button:

Once you have the ODBC connection and table/view name setup, you can select this button to have Titanium try and connect to the external data source.

Query button:

Once you have the ODBC connection and table/view name setup, and the Test button successfully connects to your external data source, then you click this button to return some information to help with the mapping of fields between the external data source and Titanium (required in a later step). It will copy to the Windows clipboard a list of the data field names and data types, as well as a handful of client records. Normally, you will paste these into a Word document or text file for use later and then delete the file when you are finished with it. (See Appendix A for a sample output.)

Student ID expression:

The client import routine will run a query to retrieve a single client record matching a specific Student ID from the external data source. This field contains the name of field that will be searched to find the record for the student with the specified Student ID. Titanium is expecting this to be a character string, so if your data source uses a numeric data type for the StudentID field, you will need to enter an expression to convert it to a left justified string.

Client name expression:

When a matching record is found for the specified StudentID, this is the expression that is used to display the name to the user to confirm that the correct client was found. Please note that any function calls you specify here must be valid for the external data source. So, if your external data source is Microsoft SQL Server, you will use those functions and syntax. If it is Oracle, you will use Oracle commands and syntax.

Example:

If you look at the figure above of the configuration screen, the settings there specify that the following query will be run against the external data source to retrieve the clients name.

```
SELECT rtrim(C_LAST)+'', '+rtrim(C_FIRST) from STUDENTS where COLL_ID = ?
```

(The `rtrim()` function is valid in Microsoft SQL Server and MS Access to trim off spaces from the field so that it will display *Doe, John* to the users.)

When to import client information:

These settings should be set to **When adding a new client** if you only want the client information to be imported when you are creating new clients. Select **New client and first appointment of the semester** if you want the client contact information updated each semester if the client comes back to the center. Please realize that an update will overwrite existing client contact information. So, if you have the client import feature configured to update phone numbers, and you change a phone number after adding a client, then the next time that client is updated from the external data source your changed phone number will be overwritten by the one in the external data source. Sometimes this is the behavior you want, sometimes it isn't, and that is why there are two settings.

When to create a new Data Form:

The client contact information goes on the Client screen. Any additional information you want to import, like client demographics, will go on a Data Form. There is only one client record, so updating that information will overwrite the existing information. You can have as many Data Forms as you want so this setting specifies how often to create a new Data Form instead of updating the old one. If you are going to import any information into a Data Form, then normally this would be set to **New client and when schedule first appointment of the year**. That way a new set of information will be created every year so things that change from year to year like (Freshman, Sophomore, Junior, etc.) will be recorded correctly for each year. This helps ensure that historical reports will be more accurate.

Data Form:

If you are going to import information to a Data Form, this is the name of the Data Form that will be used. Most often this is the Data Form you use to record client demographic information.

Default note type for a new Data Form:

Data Forms are attached to Client Notes. So, if Titanium needs to create a new Data Form, it will need to create a client note to hold that Data Form. Since all client notes require a note type, you must select one here.

When to check enrollment status:

You can optionally have Titanium check the enrollment status of a student when appointments are made. This setting determines how often that happens.

Enrollment expression:

If this field is blank, then just the existence of the client in the external data source will consider them as enrolled. If there is a particular field in the external data source that must be evaluated to confirm enrollment, then you need to enter an expression here that evaluates to TRUE to tell Titanium how to test for enrollments. For example, if your external data source has a field named *Enrolled* and it is set to 1, if they are enrolled then the expression would be "Enrolled=1".

Add a button to the client screen to let them manually import the data:

Normally the client import happens automatically. If you select this option, then there will also be a button on the Client screen that can be used to manually trigger an import of the client. This option is most commonly used if the client import feature was turned-on after there was already some clients in Titanium to give the users a way to force a client import to backfill the information on a particular client.

Add a button to the Find Client Screen to view raw client import data:

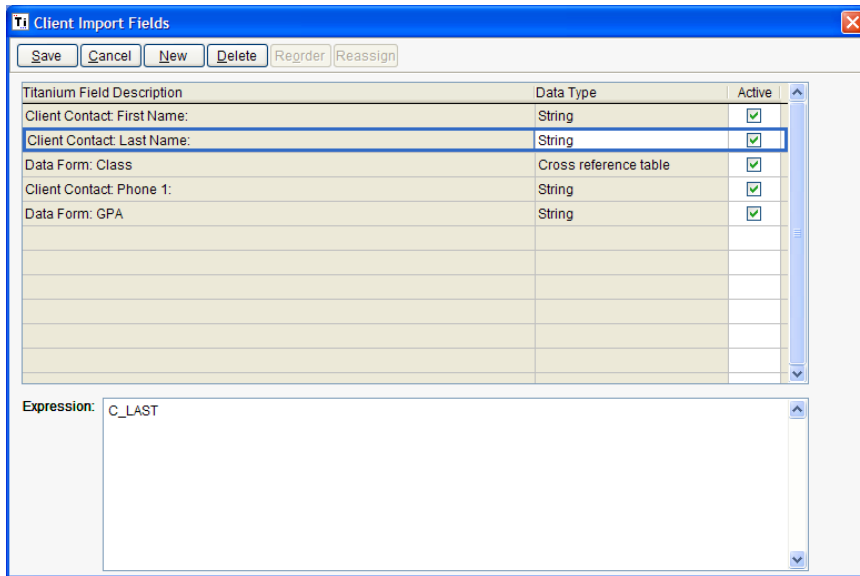
This feature will add a button to the dialog where you normally find or add clients. If you enter a Student ID number, you can press the button to view the raw data from the external data source. It is often used during testing of a new client import setup. It may also be useful if there is some information in the raw data that isn't being imported into Titanium that the users might want to view occasionally.

Client Import Field Cross Reference Table:

This will open the **Client Import Fields** screen where most of the work is done to map the external data fields to existing data fields in Titanium. If you don't have documentation on the data structure and content of the external data source, you can use the **Query** button to get some information about the external data that will be helpful. The **Query** button will copy that information to the Windows clipboard so you should paste it into a Word document so you can view it. The first part of the information will be a field of the field names and their data type. The rest of it is a few client records so you can see what the incoming data looks like and have some Student ID numbers to use for testing. (See Appendix A for a sample of the output.) (If you create any clients in Titanium during testing you should delete them when you are done, however make sure you don't delete a valid client.)

Client Import Fields Screen

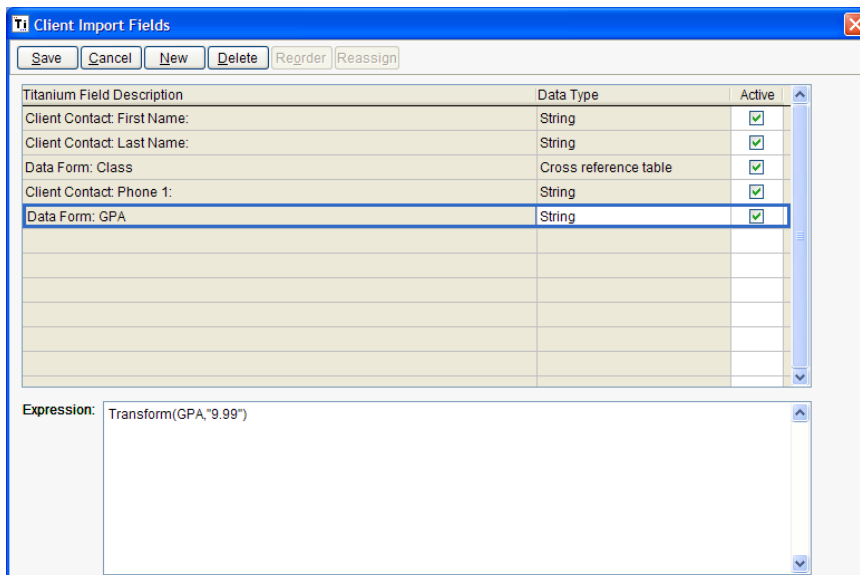
This is where you will add fields that you want to import into Titanium. You first specify what Titanium field you want to populate by selecting **New** followed by either the **Client Contact Information** (e.g. the Contact tab of the Client screen) or the **Data Form** you specified in the confirmation section. Then you will select a field on that screen/Data Form. After closing the New dialog you will see the field you added as well as the Data Type. The Data Type is important because the incoming data must have the same data type or an error will be generated by the client import routine. In the **Expression** section at the bottom of the screen you will enter a field or expression based on the incoming data. If the Titanium data type is *String* and the incoming data type is also a character string, then all you normally have to do is enter the field name. See the figure below that shows that the Titanium Last Name field will be populated by the incoming field **C_LAST**.



The Address fields in Titanium are a single field with multiple lines of text. If you are importing addresses that are in multiple separate fields you will need to combine them into a single expression. You will need to add a Carriage Return, CHR(13), and New Line, CHR(10), characters to force a new line. The easiest way to do this is to use a STRCAT(cDelimiter, cExpression1, cExpression2,...) function that will take multiple arguments and concatenate them while separating them with the delimiter. It will also correctly ignore NULL values and empty strings as well as trimming trailing spaces. So an expression like: STRCAT(CHR(13)+CHR(10),Address1, Address2, STRCAT(" ",City, State, Zip)) which will result in:

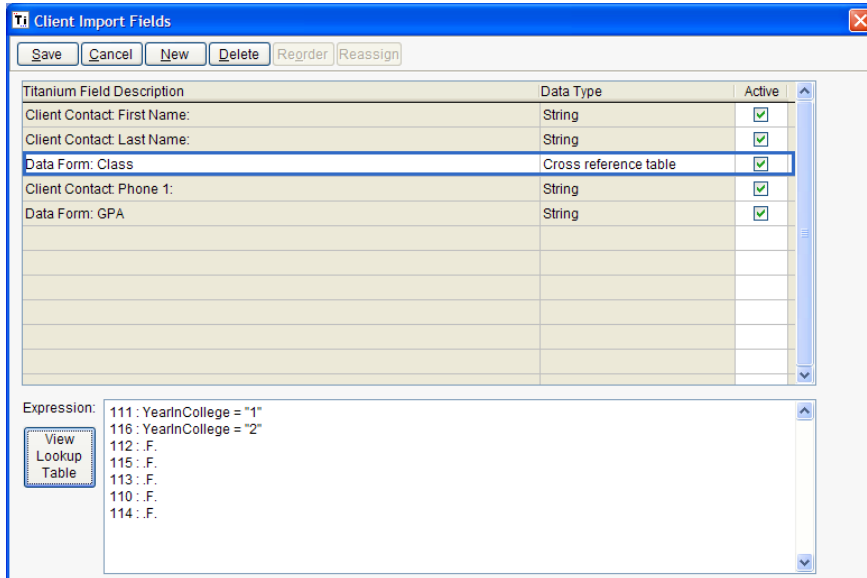
```
Address1
Address2 (This line will be removed if it is null or blank.)
City State Zip
```

Sometimes the incoming data will have to be manipulated to match the data type required by Titanium. The following figure shows that the GPA field that is on a Data Form in Titanium is a String and the incoming field name GPA is a numeric value so the expression **Transform(GPA,"9.99")** is used to convert it from a numeric value to the correctly formatted string. See **Appendix B** for some valid functions you can use to manipulate your incoming data. If you have a particular difficult manipulation, please contact us for guidance.

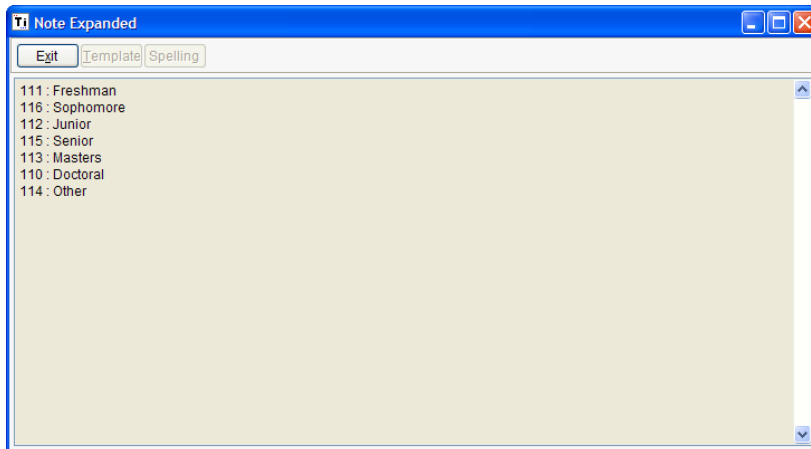


The most difficult manipulation you will need to do is if the Titanium field is based on a drop-down list and you will have to evaluate one or more fields in the incoming data to decide which of the drop down list

answers should be selected. These Titanium fields will have a data type of **Cross reference table** as shown in the figure below. The **Expression** will be a series of numbers on the left that represent the drop down list answers followed by a colon and then an expression based on the incoming data that evaluates to TRUE if that answer should be selected. In the example below for Class (Freshman, Sophomore, Junior, etc), the number *111* means Freshman and in the incoming data Freshman was specified by the YearInSchool contains the character "1". The reason the 1 is in quotes is because the incoming data had a data type of character. If it was a numeric value, the quotes would have been omitted. So the entry is *111 : YearInCollege = "1"*. There are normally entries for every incoming value that has a match in Titanium. If there is an entry in Titanium that doesn't have an incoming match, it can either be omitted from the Expression, or it can be entered as a **.F.** after the colon as shown below.



The missing piece to this is a way to find out what numbers Titanium is using for the answers in the drop down list. In my example above, how did I know that 111 means Freshman? The solution is the **View Lookup Table** button. If you are working with a Titanium field that has a lookup table of answers, then this button will be available. When you click it, you will get the figure below listing the numeric keys as well as the plain text descriptions of the answers. When you exit the screen, it will offer to paste this table into your **Expression** field. The first time you start working with this type of field, you should do so then replace the text descriptions with the appropriate expressions from the incoming data.



APPENDIX A

Sample Listing from the Query Button

(Your incoming data will have different field names and data types and that is ok.)

This first part is the field in the data source as well as the data type.

Field data types can be:

C = Character

Y = Currency

D = Date

T = DateTime

B = Double

F = Float

G = General (Cannot be imported)

I = Integer

L = Logical

M = Memo (unlimited text string)

N = Numeric

Q = Varbinary (Cannot be imported)

V = Varchar and Varchar (Binary)

W = Blob (Cannot be imported)

DATA STRUCTURE

C_SSN, M, 4, 0

COLL_ID, V, 7, 0

C_LAST, M, 4, 0

C_FIRST, M, 4, 0

ADDRESS, M, 4, 0

CITY, M, 4, 0

STATE, M, 4, 0

POSTALCODE, M, 4, 0

PHONENUMBER, M, 4, 0

EMAILADDRESS, M, 4, 0

DOB, T, 8, 0

SEX, M, 4, 0

ETHNICBACKGROUND, M, 4, 0

YEARINCOLLEGE, M, 4, 0

MARTIAL, V, 10, 0

CATEGORY, V, 10, 0

GPA, B, 8, 4

DATA

RECORD 1

COLL_ID: 11111

C_LAST: Sample

C_FIRST: Jane

ADDRESS: 1 Main Street

CITY: Houston

STATE: TX

POSTALCODE: 77005

PHONENUMBER: 1231231234

EMAILADDRESS: test@yahoo.com

DOB: 01/01/1980 12:00 AM

SEX: F

ETHNICBACKGROUND: WHITE, NOT-HISPANIC

YEARINCOLLEGE: 1

MARTIAL: MARRIED

CATEGORY: STUDENT

GPA: 1.1100

APPENDIX B

Some functions that can be used in the **Expression** field on the **Client Import** screen to manipulate the incoming data.

CTOD(cExpression) - Converts a character expression to a date expression.

LOWER(cExpression) - Returns the specified character expression in lowercase.

LTRIM(cExpression) - Removes all leading spaces from the specified character expression.

PROPER(cExpression) - Returns a character expression string capitalized as appropriate for proper names.

RTRIM(cExpression) - Removes all trailing spaces from the specified character expression.

STR(nExpression [, nLength [, nDecimalPlaces]]) - Returns the character equivalent of a numeric expression.

STRCAT(cDelimiter,cExpression1,cExpression2,...) - Returns a character expression that concatenates the expressions while separating them with cDelimiter. Empty strings and NULLS are ignored and trailing spaces on the expressions are removed.

UPPER(cExpression) - Returns the specified character expression in uppercase.

VAL(cExpression) - Returns a numeric value from a character expression composed of numbers.

TRANSFORM(eExpression, [cFormatCodes]) - Returns a character string from an expression in a format determined by a format code.

cFormatCodes	Description
@D	Converts Date and DateTime values to the current SET DATE format.
@T	Trims leading and trailing spaces from character values.
@Z	If 0, converts currency or numeric values to spaces.
@!	Converts an entire character string to uppercase.
@\$	Adds the current currency symbol specified by SET CURRENCY to currency and numeric values. By default, the symbol is placed immediately before or after the value. However, the currency symbol and its placement (specified with SET CURRENCY), the separator character (specified with SET SEPARATOR) and the decimal character (specified with SET POINT) can all be changed.
X	Specifies the width of character values. For example, if cFormatCodes is 'XX', 2 characters are returned.
Y	Converts logical True (.T.) and False (.F.) values to Y and N, respectively.
!	Converts a lowercase character to uppercase in the corresponding position in a character string.
.	Specifies the decimal point position in currency and numeric values.
9	Specifies width of numeric values. For example, if cFormatCodes is '999.99', the numeric characters are formatted with three characters to the left of the decimal point, and two characters to the right. The format code must be at least the size of the value you want to display or asterisks will be returned instead to indicate an overflow.