

# **TimeTiger 2 API**

**Version 1.1.2**

---

## **TimeTiger 2 API: Version 1.1.2**

Copyright © 2011 Indigo Technologies Ltd.

All rights reserved. TimeTiger® is a registered trademark of Indigo Technologies Ltd. Other trademarks in this document are properties of their respective owners.

---

---

---

# Table of Contents

Extending the time tracking experience .....	x
Protocol and tools .....	x
Interaction process .....	x
Enabling the API for your application .....	x
Result codes .....	xi
Obtaining the TimeTiger WSDL file .....	xi
Sample applications and videos .....	xi
Contact Information .....	xii
1. Types .....	1
TimeTiger UIDs .....	1
StringPair .....	1
ArrayOfStringPair .....	1
LanguageCaption .....	2
EntityType .....	2
EntityNode .....	3
ToDo .....	3
TimeLog .....	4
CustomField .....	5
EntityMetrics .....	7
DefaultPick .....	8
RoleType .....	9
2. Database Normalized (DBN) Values .....	10
A standard string representation of data types .....	10
3. Session Management .....	12
Hello .....	12
Touch .....	13
Goodbye .....	14
LoginStandard .....	15
LoginWindows .....	16
4. Entity Types .....	19
GetEntityTypeList .....	19
GetValidChildTypes .....	20
5. Entities .....	22
GetEntity .....	22
GetEntities .....	23
GetAllEntities .....	25
CreateEntity .....	26
CopyEntity .....	27
DeleteEntity .....	29
FindEntitiesByName .....	30
GetEntityMetrics .....	32
6. Custom Fields .....	33
GetEntityCustomFieldList .....	33
GetCustomFieldValue .....	34
SetCustomFieldValue .....	35
7. Roles and Assignment .....	38
GetRoleTypeList .....	38
CreateAssignment .....	39
8. To Do Items .....	41
GetToDoList .....	41
SaveToDo .....	42

SelectToDo .....	44
SetToDoSecs .....	45
AddToDoWorkInProgress .....	46
DeleteToDo .....	47
9. Time Logs .....	49
GetTimeLogs .....	49
SaveTimeLog .....	51
DeleteTimeLog .....	53
DeleteTimeLogsForDay .....	54
GetTimeLogBilled .....	56
SetTimeLogBilled .....	57
Index .....	59

---

## List of Figures

1.1. StringPair .....	1
1.2. ArrayOfStringPair .....	1
3.1. Invoking HelloWorlds using Classic ASP .....	18

---

## List of Tables

1. Result Codes .....	xi
1.1. LanguageCaption .....	2
1.2. EntityType .....	2
1.3. EntityNode .....	3
1.4. ToDo .....	4
1.5. TimeLog .....	5
1.6. CustomField .....	6
1.7. EntityMetrics .....	8
1.8. DefaultPick .....	9
1.9. RoleType .....	9
2.1. Database Normalized (DBN) Representation .....	11
3.1. Input Parameters: Hello .....	12
3.2. Output Parameters: Hello .....	13
3.3. Result Codes: Hello .....	13
3.4. Language Codes .....	13
3.5. Input Parameters: Touch .....	14
3.6. Output Parameters: Touch .....	14
3.7. Input Parameters: Goodbye .....	15
3.8. Output Parameters: Goodbye .....	15
3.9. Input Parameters: LoginStandard .....	15
3.10. Output Parameters: LoginStandard .....	16
3.11. Result Codes: LoginStandard .....	16
3.12. Input Parameters: LoginWindows .....	16
3.13. Output Parameters: LoginWindows .....	17
3.14. Result Codes: LoginWindows .....	17
4.1. Input Parameters: GetEntityTypeList .....	19
4.2. Output Parameters: GetEntityTypeList .....	19
4.3. Input Parameters: GetValidChildTypes .....	20
4.4. Output Parameters: GetValidChildTypes .....	20
4.5. Result Codes: GetValidChildTypes .....	21
5.1. Input Parameters: GetEntity .....	22
5.2. Output Parameters: GetEntity .....	22
5.3. Result Codes: GetEntity .....	23
5.4. Input Parameters: GetEntities .....	24
5.5. Output Parameters: GetEntities .....	24
5.6. Input Parameters: GetAllEntities .....	25
5.7. Output Parameters: GetAllEntities .....	25
5.8. Input Parameters: CreateEntity .....	26
5.9. Output Parameters: CreateEntity .....	27
5.10. Result Codes: CreateEntity .....	27
5.11. Input Parameters: CopyEntity .....	28
5.12. Output Parameters: CopyEntity .....	28
5.13. Result Codes: CopyEntity .....	29
5.14. Input Parameters: DeleteEntity .....	29
5.15. Output Parameters: DeleteEntity .....	30
5.16. Result Codes: DeleteEntity .....	30
5.17. Input Parameters: FindEntitiesByName .....	31
5.18. Output Parameters: FindEntitiesByName .....	31
5.19. Result Codes: FindEntitiesByName .....	31
5.20. Input Parameters: GetEntityMetrics .....	32
5.21. Output Parameters: GetEntityMetrics .....	32

---

5.22. Result Codes: GetEntityMetrics .....	32
6.1. Input Parameters: GetEntityCustomFieldList .....	33
6.2. Output Parameters: GetEntityCustomFieldList .....	33
6.3. Result Codes: GetEntityCustomFieldList .....	33
6.4. Input Parameters: GetCustomFieldValue .....	34
6.5. Output Parameters: GetCustomFieldValue .....	34
6.6. Result Codes: GetCustomFieldValue .....	35
6.7. Input Parameters: SetCustomFieldValue .....	35
6.8. Output Parameters: SetCustomFieldValue .....	36
6.9. Result Codes: SetCustomFieldValue .....	36
6.10. Input Parameters: SetPasswordByUserName .....	36
6.11. Output Parameters: SetPasswordByUserName .....	37
6.12. Result Codes: SetPasswordByUserName .....	37
7.1. Input Parameters: GetRoleTypeList .....	38
7.2. Output Parameters: GetRoleTypeList .....	38
7.3. Result Codes: GetRoleTypeList .....	38
7.4. Input Parameters: CreateAssignment .....	39
7.5. Output Parameters: CreateAssignment .....	39
7.6. Result Codes: CreateAssignment .....	40
8.1. Input Parameters: GetToDoList .....	41
8.2. Output Parameters: GetToDoList .....	41
8.3. Result Codes: GetToDoList .....	42
8.4. Input Parameters: SaveToDo .....	43
8.5. Output Parameters: SaveToDo .....	43
8.6. Result Codes: SaveToDo .....	44
8.7. Input Parameters: SelectToDo .....	44
8.8. Output Parameters: SelectToDo .....	44
8.9. Result Codes: SelectToDo .....	45
8.10. Input Parameters: SetToDoSecs .....	45
8.11. Output Parameters: SetToDoSecs .....	45
8.12. Result Codes: SetToDoSecs .....	46
8.13. Input Parameters: AddToDoWorkInProgress .....	46
8.14. Output Parameters: AddToDoWorkInProgress .....	46
8.15. Result Codes: AddToDoWorkInProgress .....	47
8.16. Input Parameters: DeleteToDo .....	47
8.17. Output Parameters: DeleteToDo .....	47
8.18. Result Codes: DeleteToDo .....	48
9.1. Input Parameters: GetTimeLogs .....	50
9.2. Output Parameters: GetTimeLogs .....	51
9.3. Result Codes: GetTimeLogs .....	51
9.4. Input Parameters: SaveTimeLog .....	52
9.5. Output Parameters: SaveTimeLog .....	53
9.6. Result Codes: SaveTimeLog .....	53
9.7. Input Parameters: DeleteTimeLog .....	54
9.8. Output Parameters: DeleteTimeLog .....	54
9.9. Result Codes: DeleteTimeLog .....	54
9.10. Input Parameters: DeleteTimeLogsForDay .....	55
9.11. Output Parameters: DeleteTimeLogsForDay .....	55
9.12. Result Codes: DeleteTimeLogsForDay .....	55
9.13. Input Parameters: GetTimeLogBilled .....	56
9.14. Output Parameters: GetTimeLogBilled .....	56
9.15. Result Codes: GetTimeLogBilled .....	57
9.16. Input Parameters: SetTimeLogBille .....	57
9.17. Output Parameters: SetTimeLogBille .....	57

9.18. Result Codes: SetTimeLogBille ..... 58

---

# Extending the time tracking experience

The TimeTiger WebService API has been designed to allow application developers to easily extend the capabilities provided by the core TimeTiger 2 System Server. By using readily available languages and tools, developers can create add-ons or completely new projects that build on the features in TimeTiger.

These are just some examples of projects you can build using the TimeTiger WebService API:

- Custom user interface for mobile or shop-floor applications.
- Widget-style tracking interface for a different platforms and/or operating systems.
- Data aggregation, analysis and archival tool and data warehousing application.
- Integration conduit to a third-party issue tracking, billing, or project management system.

## Protocol and tools

The TimeTiger WebService API uses the SOAP protocol for message exchange. Any tool compatible with SOAP development should be suitable for use with the WebService API. Internally, we use Microsoft Visual Studio 2003 and 2005 for development and testing with the API.

## Interaction process

In general, to connect with the a TimeTiger application using the WebService API, your application will issue the following method calls:

- **Hello** (to obtain a session ID)
- **LoginStandard** or **LoginWindows** (depending on the type of authentication supported by your TimeTiger application)
- (additional application-specific method calls)
- **Goodbye** (to release the session ID)

Your application may also call the Touch method periodically to keep a particular session ID alive. The session ID will stay valid on the server as long as you continue making API calls, but if your application goes for several minutes without making any API calls, it is possible that the session ID will time out. The exact timeout interval depends on server-specific settings, but in general delays of up to 10 minutes should not cause any issues.

It is good practice (although not required) to issue a Goodbye method to release the session when you are done with it. Eventually the server will time out any sessions for which Goodbye was not called.

## Enabling the API for your application

By default, the TimeTiger server will not allow API calls to a TimeTiger application until enabled by an administrator through the TimeTiger administrative console.

Log into the administrative console, choose **Applications > More**. Click the ... button beside your application, click **Change...** beside WebService API, and check the box to **Enable WebService API**. Click **Submit Changes**.



### Note

The sample application that is pre-installed with TimeTiger has its WebService API enabled by default.

## Result codes

All TimeTiger API methods use the following common result codes. Result codes -101 through -253 are reserved for method-specific error messages that are detailed for each function in the API reference below.

For all methods in the TimeTiger API, a ResultCode of  $\geq 0$  can be interpreted as success and a ResultCode of  $< 0$  is a failure.

**Table 1. Result Codes**

ResultCode	Meaning
0	Success (with no messages)
> 0	Success with messages/warnings. Specific result codes will be documented for each method.
-1	Error: invalid or expired session ID.
-2	Malformed request. The SOAP/XML message was incomplete or was formatted incorrectly.
-3	Unknown method. The SOAP/XML message attempted to execute a method that is not available through the TimeTiger API.
-4	Parameter missing. An expected parameter was not supplied.
-5	Unauthenticated session. The requested method requires the provided session to have been authenticated using a Login method but the session has not yet been successfully authenticated.
-6	This method requires valid Windows credentials but none could be obtained from the calling process.
-7 to -100	Reserved for future use.
-101 to -253	Method-specific errors, described further in the method description.
-254	Unknown error or failure. Please report this result to Indigo Technologies.

## Obtaining the TimeTiger WSDL file

The WSDL file can be obtained by issuing the following GET request to a valid, running application on the TimeTiger server:

```
http://myserver:myport/myapplication/ttxml?wsdl
```

## Sample applications and videos

The TimeTiger API Sample application, a Visual Studio .NET program written in C#, is available to demonstrate the functionality of the API and to provide sample code you can use in your own API client applications.

There are a number of Getting Started videos that demonstrate the use of the TimeTiger Sample application.

## Contact Information

For more information, contact Indigo Technologies technical support at 416-815-0579 or by e-mailing [support@indigo1.com](mailto:support@indigo1.com).

---

# Chapter 1. Types

## TimeTiger UIDs

All objects in TimeTiger are identified by a 16-character Unique Identifier (UID). When this document refers to EntityID, UserID, UID, etc., this unique identifier is actually what is being referred to.

## StringPair

A StringPair object contains a pair of strings typically representing an ID/Value pair. For example, a particular EntityType might be returned in a StringPair as follows:

**Figure 1.1. StringPair**

```
<StringPair>
  <ID>0001-000000000001</ID>
  <Value>User Group</Value>
</StringPair>
```

Many SOAP/XML-aware development environments insulate the developer from the specifics of the XML representation of this (and other) data types and will simply expose the StringPair type to you as a native language object.

## ArrayOfStringPair

When the API requires or returns a list of ID/Value pairs an ArrayOfStringPair is used. For example, the EntityTypes output parameter of the GetValidChildTypes method might return something similar to the following:

**Figure 1.2. ArrayOfStringPair**

```
<EntityTypes>
  <StringPair>
    <ID>0001-000000000001</ID>
    <Value>User Group</Value>
  </StringPair>
  <StringPair>
    <ID>0001-000000000051</ID>
    <Value>User</Value>
  </StringPair>
  <StringPair>
    <ID>0001-000000000071</ID>
    <Value>Client</Value>
  </StringPair>
  <StringPair>
    <ID>0001-000000000101</ID>
    <Value>Project</Value>
  </StringPair>
</EntityTypes>
```

# LanguageCaption

Some types described later in this chapter use captions that can depend on the language chosen by the user when they log into TimeTiger. TimeTiger includes one LanguageCaption node for each language configured for the application. It is safe to use the caption from the first such node (in an ArrayOfLanguageCaption element) if only a single language is configured for the application.

The following child nodes are included in each LanguageCaption node:

**Table 1.1. LanguageCaption**

Element	Description
LanguageID	The UID of the language this caption belongs to.
FieldName	If this caption refers to a CustomField, the FieldName is the field name that should be displayed to the user in the specified language. Will be blank if this caption does not refer to a CustomField.
Description	The caption (for non-field captions) or field description (for CustomField captions) in the specified language.

# EntityType

The various types of items in TimeTiger are referred to as EntityTypes. For example, Client, Project, and Task are all standard EntityTypes that may be present in a particular TimeTiger database. TimeTiger databases are customizable and in some cases other types may be present, and some common types (such as Client) may be absent. If your API application is designed to connect to a variety of different TimeTiger databases it is necessary to obtain a list of the EntityTypes present.

The EntityType structure describes a single EntityType in the TimeTiger database. An ArrayOfEntityType structure describes several EntityTypes.

The following elements are returned in an EntityType:

**Table 1.2. EntityType**

Element	Description
UID	The UID of the EntityType.
EntityTypeName	A textual description of the EntityType, e.g. Client.
PluralName	The plural textual name of the type, e.g. Clients.
SystemRef	A language-independent textual code used by TimeTiger for various built-in or standard EntityTypes. For example, 'client' is the SystemRef for the 'Client' EntityType, even when it has been renamed to "Customer" or something else meaningful to the end-user. Not every EntityType will have a SystemRef.
ParentTypes	An ArrayOfStringPair that lists the UIDs and Names of the possible parent types of this type. For example, the typical parent types of Task are Task and Project.
ChildTypes	An ArrayOfStringPair that lists the UIDs and Names of the possible child types of this type. For example, the possible child types of Client are typically Client (i.e. sub-Client) and Project.

# EntityNode

Basic entity information is returned in an EntityNode structure. Sets of entities are returned in an ArrayOfEntityNode collection (see ArrayOfStringPair for an example of how arrays are returned).

The following elements are returned in an EntityNode:

**Table 1.3. EntityNode**

Element	Description
UID	Entity UID, or for top level items, the EntityTypeID of the top level item and immediate children (e.g. Clients).
EntityName	The name of this entity (as displayed in the TimeTiger Explorer).
EntityTypeID	The UID of the entity type. Is the same as the EntityUID for top-level items.
EntityTypeName	A textual description of the entity type of this item.
ParentEntityID	The UID of the parent item, or blank if this is a top level item.
ChildCount	The number of direct children under this node or 0 if there are none. This count includes <i>all</i> direct children of this node, even if they would normally be hidden due to security limitations or criteria used in the API method used to retrieve this node.
ChildNodes	A collection of nodes representing the children of this entity. This collection is only populated when specifically requested in the method call, and is otherwise blank.
Hidden	“1” if this item would normally be hidden in the TimeTiger Explorer, “0” otherwise.

# ToDo

A To Do item is returned in an ToDo structure. Sets of To Do items are returned in an ArrayOfToDo collection (see ArrayOfStringPair for an example of how arrays are returned).

The following elements are returned in an ToDo:

**Table 1.4. ToDo**

Element	Description
UID	Unique identifier of the To Do item.
CreatedOn	The date/time (in DBN form) that this To Do was created.
UserID	The ID of the user to whom this To Do item belongs.
NothingToDo	“1” if this To Do item represents the (Nothing) item for the user, “0” for all other To Do items.
Seq	Numerical display sequence for the To Do item. TimeTiger displays To Do items in this order.
Description	The description of the To Do item as it appears in the TimeTiger To Do list.
Notes	Textual notes for the To Do item.
WorkInProgress	Notes used as the Status/Description when logging this To Do item through the Log Time window.
Allocation	An array of EntityNode items representing the lowest-level items (e.g. User, Task, Category) to which this To Do is assigned.
OnSince	If To Do item is currently the active, running, item, this is the date stamp (in DBN format) when the To Do item was started. Otherwise this is a blank string. Only one To Do item can be running for a user at any time.
StoredSecs	The number of seconds stored in the To Do item. The total time in a stopped To Do item is simply the StoredSecs. The total time in the running, active item can be calculated as StoredSecs + the number of seconds between OnSince and the current time.

## TimeLog

Basic time log information is returned in a TimeLog structure. Sets of time logs are returned in an ArrayOfTimeLog collection (see ArrayOfStringPair for an example of how arrays are returned).

The following elements are returned in a TimeLog:

**Table 1.5. TimeLog**

Element	Description
UID	Unique identifier for the TimeLog.
LogDate	The date to which this log is attributed, in the format <code>yyyymmdd</code> . Not necessarily the date the time log was actually entered into the system.
LogSeconds	The number of seconds attributes to the log.
StatusDescription	A textual description (as entered by the user) of the work that was done for this log.
ToDoID	If the log's allocation matches that of an existing ToDo item for the user, the UID of that ToDo item is included.
Allocation	An <code>ArrayOfEntityNode</code> , with each element representing a lowest-level entity to which this log is allocated. For example, the array with include an <code>EntityNode</code> for the User but not the User Group (as User Group is implied by User).
PortionComplete	The value of the <code>% Complete</code> field for this particular time og. Presented as a <code>VT_FRACTION</code> .
Custom1	For future use.
Custom2	For future use.
Custom3	For future use.

## CustomField

Complete custom field information is returned in a `CustomField` structure. Note that this structure represents the *definition* of a custom field rather than the actual value for a particular custom field in the system.

Sets of custom fields are returned in an `ArrayOfCustomField` collection (see `ArrayOfStringPair` for an example of how arrays are returned). Most applications will not require many of these structure elements but they are provided for completeness.

The following elements are returned in a `CustomField`:

**Table 1.6. CustomField**

<b>Element</b>	<b>Description</b>
UID	Unique identifier for the CustomField.
CustomFieldGroupID	The various “tabs” shown in the Properties view of the TimeTiger Explorer are internally referred to as CustomFieldGroups. This value identifies the CustomFieldGroup, or tab, to which the field belongs.
Seq	The numerical display sequence of the custom field within the CustomFieldGroup.
Restricted	Flag indicating whether this field is considered Restricted for display and editing purposes. Different security role permissions can be granted to users to allow them to view/edit restricted and non-restricted fields.
Visible	Controls whether this field is displayed, by default, in certain situations. Possible values:  0. hidden by default  1. visible on both time logs and reports  2. visible on reports
ValueType	Numerical code representing the type of value stored in the field. See the section on Database Normalized Values (DBN) below.
ValueEntityTypeID	If the ValueType of this custom field is VT_ENTITY, the ValueEntityTypeID indicates which type of entity can be stored in the field. For example, if only Project IDs can be stored in the custom field, the ValueEntityTypeID will be the EntityTypeID of Project (as returned by GetEntityTypeList). Otherwise ValueEntityTypeID is blank.
ValueListID	If the possible values in this custom field are limited by the values of a particular user-maintained list, the ValueListID is the UID of the list containing the possible values. Otherwise ValueListID is blank.
DefaultValue	The default value of the custom field, or blank.
LinkType	Indicates whether changes to the value of a custom field affect other values in the system. Possible values: 0 – none#1 – linked bi-directionally. Changes in either field affect the corresponding field.
LinkField	The CustomFieldID to which this custom field is linked, or blank.
AggregateType	Indicates whether changes to the value of a custom field aggregate up to a different totaling field. Possible values: 0 – none#1 – sums up to totaling field#2 – the first value found is stored in the totaling field#3 – the last value found is stored in the totaling field#4 – averaged up to the totaling field#5 – averaged up but weighted by the AggregateWeightFieldID
AggregateWeightFieldID	If the AggregateType is AGG_WEIGHTEDAVG_UP, the CustomFieldID that determines the weighting of this value for averaging purposes.
ValidationRules	Custom validation rules for this field.
System	“1” if this is considered a system field, that cannot be deleted and is restricted from certain modifications. “0” otherwise.
SystemRef	The string reference code for a system field.
LanguageCaptions	An ArrayOfLanguageCaption element that includes one LanguageCaption node for every language configured for this TimeTiger application.

## EntityMetrics

Calculated metrics for a particular entity at a particular moment in time are returned in an EntityMetrics structure. Many of these metrics are calculated using Earned Value formulas. For more information about Earned Value and interpreting these values, see the TimeTiger Reference Guide.

All metrics are returned in DBN format.

**Table 1.7. EntityMetrics**

Element	Data Type	Description
AsOf	VT_DATETIME	The moment at which the metrics are current.
TotalSecs	VT_DURATION	The total number of seconds logged to the entity up to the AsOf instant.
ChildCount	VT_NUMBER	The number of direct and indirect children of this entity in the tree.
FirstLogOn	VT_DATETIME	The entry timestamp of the first (oldest) time log used in calculating these metrics.
LastLogOn	VT_DATETIME	The entry timestamp of the last (newest) time log used in calculating these metrics.
CompleteSecs	VT_DURATION	The number of seconds considered to be complete (% complete * estimated seconds).
PlannedSecs	VT_DURATION	The number of seconds that should be complete if progress exactly matched the plan.
EstimatedSecs	VT_DURATION	The total number of seconds of work estimated for this entity.
EstimatedStartOn	VT_DATE	The date on which work on this entity was expected to start.
EstimatedEndOn	VT_DATE	The date on which work on this entity was expected to be completed.
EstimatedCost	VT_CURRENCY	The total amount budgeted for work on this entity.
ActualCost	VT_CURRENCY	The cost to date of time logged to this entity.
PercentComplete	VT_TWODECIMAL	The calculated percent complete for this item.
PlannedPercentComplete	VT_TWODECIMAL	The percent complete that would be expected if progress exactly matched the plan.
EarnedValue	VT_CURRENCY	The Earned Value (EV) for this entity up to the AsOf instant.
PlannedValue	VT_CURRENCY	The Planned Value (PV) for this entity up to the AsOf instant.
CPI	VT_TWODECIMAL	The Cost Performance Index (CPI) as the AsOf instant.
SPI	VT_TWODECIMAL	The Schedule Performance Index (SPI) at the AsOf instant.
PercentUnderBudget	VT_TWODECIMAL	The percent by which progress is beating the budget plan, or negative if progress is behind the budget plan.
PercentAheadSchedule	VT_TWODECIMAL	The percent by which progress is beating the schedule plan, or negative if progress is behind the schedule plan.
ForecastedCost	VT_CURRENCY	The projected cost of this entity at completion based on current progress.
ForecastedCompleteOn	VT_CURRENCY	The projected completion date of this entity based on current progress.

## DefaultPick

A Default Pick is a directive to TimeTiger that when a particular Entity is chosen through the user interface, the corresponding Default Pick should be chosen as well. This allows administrators to specify a default Category for each Task, for example. Default Picks can be optional (the user is allowed to override the default) or forced (no override is permitted).

There are specific rules for the Default Picks that can exist for a particular Entity. For example, it would not make sense to create a default Client for a particular Project since Client must be chosen before Project in the TimeTiger UI. The definitive guide for the Default Picks that can be created for an Entity is what is possible through the TimeTiger Explorer user interface, but in general most common-sense associations are possible. Note that only lowest-level Entities (Entities with no sub-Entities) can be Default Picks.

**Table 1.8. DefaultPick**

Element	Description
EntityID	The UID of the Entity that should be picked.
Type	The type of Default Pick. If Type = 0, this Default is considered optional and a different Entity can be chosen by the user through the user interface. If Type = 1, this Default is forced and no alternative is permitted through the UI.

## RoleType

Information about a specific TimeTiger security role.

**Table 1.9. RoleType**

Element	Description
UID	The UID of the RoleType.
Flags	An integer value representing the bitwise-OR of all the permissions granted to this role.
System	1 if this role is a system-standard role (e.g. Self), 0 if it is a custom user-created role.
SystemRef	The string reference code for system-standard roles (e.g. 'self', 'admin').
FilterType	1 if this role is limited to certain Entity Types. 0 if this role can be assigned for any Entity Type.
FilterInclude	If FilterType = 1, a FilterInclude node is included containing the EntityTypeID of each Entity Type to which users can be assigned in this role.
LanguageCaptions	An ArrayOfLanguageCaption element that includes one LanguageCaption node for every language configured for this TimeTiger application.

---

# Chapter 2. Database Normalized (DBN) Values

## A standard string representation of data types

In order to maximize compatibility with various different languages, locales, and platforms, the TimeTiger API requires and returns many values in a standard textual notation we refer to as Database Normalized (DBN) form. DBN form is a string-based format that each different TimeTiger ValueType can be unambiguously converted to and from. The following table lists the different ValueTypes supported by TimeTiger and the corresponding DBN specification.

Not all API methods require or return DBN-formatted values. This document will explicitly indicate where DBN form is used in input or output parameters.

Note that native string values are stored unchanged in DBN format. So, for string values, the DBN representation is the same as the original string representation and no conversion is required.

**Table 2.1. Database Normalized (DBN) Representation**

Value	Type Name	Formatting	Example
0	VT_STATUSCODE	UID of status code	0001-12345678901
1	VT_MEMO	(raw text)	Alpha beta kappa
2	VT_TEXT50	(raw text)	Alpha beta kappa
3	VT_TEXT255	(raw text)	Alpha beta kappa
4	VT_ENTITY	UID of entity	0001-12345678901
5	VT_LISTENTRY	UID of list entry	0001-12345678901
6	VT_NUMBER	+/- followed by number left-padded to 20 digits	+ 6
7	VT_CURRENCY	+/- followed by dollar value left-padded to 20 digits followed by cents value zero-padded to 2 digits	+ 165099
15	VT_ONEDECIMAL	+/- followed by whole number portion left-padded to 20 digits followed by single decimal digit	+ 12348
16	VT_TWODECIMAL	+/- followed by whole number portion left-padded to 20 digits followed by two decimal digits	+ 356025
8	VT_DATE	Standard date format	20021031
9	VT_TIME	Standard time format	120000
10	VT_YESNO	Yes = 1, No = 0	1
11	VT_DATETIME	Concatenation of VT_DATE and VT_TIME	20021031120000
12	VT_DURATION	Number of seconds, formatted as VT_NUMBER	+ 180
13	VT_DAYOFWEEK	Sunday = 0, Saturday = 6	2
14	VT_FRACTION	+/- followed by numerator (left padded to 20 digits) then / then denominator (left padded to 20 digits). 42 characters total.	+ 25/ 100

---

# Chapter 3. Session Management

## Hello

### Syntax

```
Hello(string APIVersion, string PreferredLanguage)
```

### Description

The Hello command initiates communication with the TimeTiger server and requests a session ID for the interaction. Hello must be the first command issued in an interaction as all other operations require a valid session ID.

### Input Parameters

**Table 3.1. Input Parameters: Hello**

Element	Type	Sample	Description
APIVersion	String	1.0.0	The version of the TimeTiger API requested by the client. The server will produce an error result if the requested APIVersion is not supported.
PreferredLanguage	String	ENUS	The language the client wishes to use for server interactions. This primarily affects the language in which error messages, prompts and some data are transmitted back to the client. If this application does not support the requested language code the server will use its default language and return an appropriate response notifying the client.

## Output Parameters

**Table 3.2. Output Parameters: Hello**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. “Success” is a typical result representing completion.
SessionID	String	(30 chars)	A 30-character unique session identifier for use in future interactions.
LanguageCode	String	ENUS	The language code the server has chosen to use for subsequent interactions. May not be the language requested by the client if the application is not configured to support it.

## Method-specific result codes

**Table 3.3. Result Codes: Hello**

Result	Description
1	Success. The requested language is not available and the default application language will be used.
-101	Failed. The requested API version is out of date. The client software must be updated to use a newer version of the API.
-102	Failed. The requested API version is newer than the version supported by the server. The server must be updated to support the API version requested by the client.

## Additional information

The following language codes are currently supported. Additional languages will be supported in future TimeTiger releases. Note that in order to use a language code for a given TimeTiger application that language must first be enabled for the application through the TimeTiger Setup and Configuration console.

**Table 3.4. Language Codes**

Code	Language Name
ENUS	English
FREN	French

## Touch

### Syntax

Touch(**string** *SessionID*)

## Description

Keeps the session from timing out due to inactivity. By default TimeTiger will time out any session that has not been used for over an hour (although this is a configurable setting). If you expect that your application will only communicate with the TimeTiger server periodically but would like to use the same session, Touch your session at regular intervals to prevent it from timing out.

## Input Parameters

**Table 3.5. Input Parameters: Touch**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.

## Output Parameters

**Table 3.6. Output Parameters: Touch**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.

## Method-specific result codes

None.

## Goodbye

### Syntax

Goodbye(**string** *SessionID*)

### Description

The Goodbye command logs the particular Session ID out of TimeTiger and releases resources. Strictly speaking a Goodbye is not required as inactive sessions will eventually time out (based on the TimeTiger server session timeout settings), but doing so will release resources and is considered good programming practice.

## Input Parameters

**Table 3.7. Input Parameters: Goodbye**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.

## Output Parameters

**Table 3.8. Output Parameters: Goodbye**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. “Success” is a typical result representing completion.

## Method-specific result codes

None.

## LoginStandard

### Syntax

`LoginStandard(string SessionID, string UserName, string Password)`

### Description

Attempts to log into the TimeTiger application using the specified user name and password. Use only for TimeTiger applications configured for Login/Password security. If your TimeTiger application uses Windows integrated authentication, use the LoginWindows request instead.

## Input Parameters

**Table 3.9. Input Parameters: LoginStandard**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
UserName	String	DAVE	The name of an active TimeTiger user in the database.
Password	String	secret	The clear-text password of the TimeTiger user.

## Output Parameters

**Table 3.10. Output Parameters: LoginStandard**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.
UserName	String	DAVE	The name of the user that logged in or attempted to log in.
UserID	UID	(16 chars)	The unique TimeTiger user ID of the user that logged in. Blank if login failed.

## Method-specific result codes

**Table 3.11. Result Codes: LoginStandard**

Result	Description
-101	Failed. The session has been logged in previously.
-102	Failed. The TimeTiger application does not support this type of login authentication. Use LoginWindows instead.
-103	Failed. Login failure described in the Message output parameter.

## LoginWindows

### Syntax

```
LoginWindows(string SessionID)
```

### Description

Attempts to log into the TimeTiger application using Windows Integrated Security (via SSPI/NTLM). Use only for TimeTiger applications configured for Windows Integrated Security and when the client is already authenticated on an appropriate Windows domain. If your TimeTiger application uses standard login/password security, use the LoginStandard request instead.

## Input Parameters

**Table 3.12. Input Parameters: LoginWindows**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.

## Output Parameters

**Table 3.13. Output Parameters: LoginWindows**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.
UserName	String	DAVE	The name of the user that logged in or attempted to log in.
UserID	UID	(16 chars)	The unique TimeTiger user ID of the user that logged in. Blank if login failed.

## Method-specific result codes

**Table 3.14. Result Codes: LoginWindows**

Result	Description
-101	Failed. The session has been logged in previously.
-102	Failed. The TimeTiger application does not support this type of login authentication. Use LoginStandard instead.
-103	Failed. Login failure described in the Message output parameter.

## Additional information

Using TimeTiger with Windows Integrated Authentication can get particularly tricky when your TimeTiger interface code resides, for example, on a web page that itself requires authentication (Windows integrated or otherwise). If you receive a 401 (Unauthorized) HTTP response from your LoginWindows method call, the most likely cause is that no Windows credentials are being transmitted to the TimeTiger server.

For example, consider a Classic ASP invocation of the LoginWindows method using the MSXML2.ServerXMLHTTP object. Here is some sample code that invokes LoginWindows with a known session ID (obtained through a Hello method call).

**Figure 3.1. Invoking HelloWindows using Classic ASP**

```
Dim oXMLHTTP
Dim sLoginRequest, sLoginResponse
Set oXMLHTTP = Server.CreateObject("Msxml2.ServerXMLHTTP")

oXMLHTTP.open "POST", "http://timetigerserver/app/ttxml", False
oXMLHTTP.setRequestHeader "Content-Type", "text/xml"
oXMLHTTP.setRequestHeader "SOAPAction", _
"http://www.timetiger.com/LoginWindows"

sLoginRequest = "<soap:Envelope>" & _
"<soap:Body>" & _
"<LoginWindows>" & _
"<SessionID>" & sSessionID & "</SessionID>" & _
"</LoginWindows>" & _
"</soap:Body>" & _
"</soap:Envelope>"

oXMLHTTP.send sLoginRequest
sLoginResponse = oXMLHTTP.responseText
```

Regardless of the IIS Application security configuration (Windows Integrated or Anonymous), and even if the account used for Anonymous access matches a TimeTiger user account, the LoginWindows call will fail with a 401 (Unauthorized) result code. In fact, the MSXML2.ServerXMLHTTP object's open call must be hardcoded with a valid Windows domain user account and password and the domain user account specified must match a valid TimeTiger user account in order for the method call to succeed.

```
oXMLHTTP.open "POST", "http://timetigerserver/app/ttxml", False, _
"DOMAIN\USER", "password"
```

In summary, do not assume that your web server or development platform is passing your domain credentials through your SOAP marshaling tool or object. In some cases, you must explicitly provide your SOAP mechanism with valid domain login credentials (that match a TimeTiger user name) in order for Integrated Windows Authentication to succeed.

---

# Chapter 4. Entity Types

## GetEntityTypeList

### Syntax

```
GetEntityTypeList(string SessionID, string BottomTypesOnly)
```

### Description

### Input Parameters

**Table 4.1. Input Parameters: GetEntityTypeList**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
BottomTypesOnly	String	1	If this value is 1 only the bottom-most types are returned. The bottom-most types are the ones required for recording time logs in the system (for example, User is typically bottom-most but User Group is not because when User is specified the User Group can be automatically determined by TimeTiger).

### Output Parameters

**Table 4.2. Output Parameters: GetEntityTypeList**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.
EntityTypes	ArrayOfEntityType	(array)	An array of EntityType objects representing each entity type in the application.

### Method-specific result codes

None.

# GetValidChildTypes

## Syntax

```
GetValidChildTypes(string SessionID, string SubjectID)
```

## Description

Returns a list of valid entity types (e.g. Client, Project, Task, etc.) for the specified Subject ID. Subject ID can be either a valid EntityID or a valid EntityTypeID. Also returns a result code indicating whether the user has sufficient privileges to create a new item underneath the specified EntityID (or if a EntityTypeID has been specified, whether the user has sufficient privileges to create a top-most level entity of the specified entity type).

## Input Parameters

**Table 4.3. Input Parameters: GetValidChildTypes**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
SubjectID	UID	(16 chars)	The EntityID or EntityTypeID for which we would like to get child information.

## Output Parameters

**Table 4.4. Output Parameters: GetValidChildTypes**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.
EntityTypes	ArrayOfStringPair	(Array)	An array of ID, Value pairs where ID represents the TimeTiger UID for the entity type and Value is the name of the entity type in the currently selected language (e.g. Client).
CanAdd	String	1	If 1, sufficient privileges are available to add a new entity of a valid type underneath the specified Entity ID (or as a top-most item if the SubjectID refers to an EntityType).

## Method-specific result codes

**Table 4.5. Result Codes: GetValidChildTypes**

<b>Result</b>	<b>Description</b>
-101	Failed. Invalid SubjectID (it does not represent a valid EntityID or EntityTypeID).

---

# Chapter 5. Entities

## GetEntity

### Syntax

```
GetEntity(string SessionID, string EntityID, string ShowSubChildren)
```

### Description

Returns details of the specified Entity.

### Input Parameters

**Table 5.1. Input Parameters: GetEntity**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
EntityID	UID	(16 chars)	The entity in which we are interested.
ShowSubChildren	String	1	If 1 the ChildNodes collection of the EntityNode object is recursively populated with all the children of the selected entity. Otherwise the ChildNodes object is left blank.

### Output Parameters

**Table 5.2. Output Parameters: GetEntity**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.
EntityNode	EntityNode	(EntityNode)	A single EntityNode object representing the selected entity.

## Method-specific result codes

**Table 5.3. Result Codes: GetEntity**

<b>Result</b>	<b>Description</b>
-101	Failed. Missing or invalid EntityID.
-102	Failed. Insufficient permissions to get information about the specified entity.

## GetEntities

### Syntax

```
GetEntities(string SessionID, string ParentEntityID, string ShowSubChildren, string ShowHidden, string PermissionFilter)
```

### Description

Returns a list of entities in the TimeTiger database. The direct children of the specified ParentEntityID are displayed, and if ShowSubchildren is set to 1, each returned EntityNode has a populated ChildNodes collection all the way down through the tree of entities. If ShowSubChildren is 0 the ChildNodes collection is empty. This method can be called recursively to drill down through the entire entity tree.

Only general entity information of the type EntityNode is provided. For detailed field information for a particular entity call GetEntityDetails.

To obtain a complete list of entities visible to the logged in user, set ParentEntityID = blank, ShowSubChildren = 1, ShowHidden = 1 and PermissionFilter = 0.

## Input Parameters

**Table 5.4. Input Parameters: GetEntities**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
ParentEntityID	UID	(16 chars)	The parent entity underneath which entities will be returned. If ParentEntityID is blank, the top level Entity placeholder nodes will be returned (e.g. User Groups, Clients, Categories).
ShowSubChildren	String	1	If 1, recursively includes all child entities in the ChildNodes member of the EntityNode structure of each returned entity. Otherwise, ChildNodes is left blank (although ChildCount is still populated).
ShowHidden	String	1	If 1, entities marked with a status code that would normally hide them will also be displayed. This is equivalent to clicking the Show Hidden Items checkbox in the TimeTiger Explorer.
PermissionFilter	String	0	If 0, shows all entities the user is entitled to see. If 1, only shows entities for the user is entitled to create new time logs (has the CAN_LOG permission).

## Output Parameters

**Table 5.5. Output Parameters: GetEntities**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.
EntityNodes	ArrayOfEntityNode	(array)	An array of EntityNode objects representing the selected entities.

## Method-specific result codes

None.

# GetAllEntities

## Syntax

```
GetAllEntities(string SessionID, string ShowHidden, string
PermissionFilter)
```

## Description

Returns a complete, hierarchical list of entities in the TimeTiger database. Only general entity information of the type EntityNode is provided. For detailed field information for a particular entity call GetEntityDetails.

## Input Parameters

**Table 5.6. Input Parameters: GetAllEntities**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
ShowHidden	String	1	If 1, entities marked with a status code that would normally hide them will also be displayed. This is equivalent to clicking the Show Hidden Items checkbox in the TimeTiger Explorer.
PermissionFilter	String	0	If 0, shows all entities the user is entitled to see. If 1, only shows entities for the user is entitled to create new time logs (has the CAN_LOG permission).

## Output Parameters

**Table 5.7. Output Parameters: GetAllEntities**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.
EntityNodes	ArrayOfEntityNode	(array)	An array of EntityNode objects representing the selected entities.

## Method-specific result codes

None.

# CreateEntity

## Syntax

```
CreateEntity(string SessionID, string EntityTypeID, string
ParentEntityID, string EntityName)
```

## Description

Creates a new Entity underneath the specified Parent and of the specified Type, if possible.

## Input Parameters

**Table 5.8. Input Parameters: CreateEntity**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
EntityTypeID	UID	(16 chars)	The ID representing the type of Entity to create, as returned by GetEntityTypeList.
ParentEntityID	UID	(16 chars)	The ID of the parent Entity underneath which to create this Entity. If blank, will create a top-level Entity.
EntityName	String	General Admin.	The name of the new entity to create. This name must be unique at the level of the tree in which it is being created and (in the case of Users) unique throughout the system.
CustomFieldValues	ArrayOfStringPair		<b>FOR FUTURE USE.</b> Currently this must be a blank ArrayOfStringPair.
DefaultPicks	ArrayOfDefaultPick		Include any Default Picks you would like associated with this Entity upon creation. This allows you to specify a default Category for a new Task, for example. As with all parameters, you must include a DefaultPicks node even if it is empty (has no child DefaultPick items).

## Output Parameters

**Table 5.9. Output Parameters: CreateEntity**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.
EntityID	UID	(16 chars)	The EntityID of the newly created Entity.

## Method-specific result codes

**Table 5.10. Result Codes: CreateEntity**

Result	Description
-101	Failed. Missing or invalid EntityTypeID.
-102	Failed. Blank entity name is not allowed.
-103	Failed. The ParentEntityID you specified cannot have children of type EntityTypeID, or, if you have not specified a ParentEntityID, the EntityTypeID you specified is not a top-level type and must have a parent.
-104	Failed. You do not have permissions to create child nodes under ParentEntityID.
-105	Failed. The entity name you have specified would create a duplicate underneath the parent, or this is a duplicate entity of a type that requires unique names throughout the system (such as User).
-106	Failed. Missing or invalid DefaultPicks. Even if you do not provide any DefaultPicks, you must still include an (empty) <DefaultPicks> node in the method call.
-107	Validation error. The Entity could not be saved because information you provided violates standard or custom business rules.

## CopyEntity

### Syntax

```
CopyEntity(string SessionID, string EntityID, string ToParentEntityID,
string EntityName)
```

### Description

Creates a complete copy of the Entity and all sub-Entities underneath the specified parent. All properties of the original entity are copied, but no time log data, just as if the entity was copied using the TimeTiger Explorer.

## Input Parameters

**Table 5.11. Input Parameters: CopyEntity**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
EntityID	UID	(16 chars)	The ID of the entity to copy.
ToParentEntityID	UID	(16 chars)	The ID of the parent Entity underneath which to place the copy.
EntityName	String	Copy of General Admin	The name of the copy. If this parameter is left blank TimeTiger will use the name of the original unless that would create a duplicate name under the parent, in which case TimeTiger will attempt to find a similar but unique name to use. On the other hand, if a non-blank value of EntityName is provided that would create a duplicate, an error is raised.

## Output Parameters

**Table 5.12. Output Parameters: CopyEntity**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.
NewEntityID	UID	(16 chars)	The EntityID of the new copy.
NewEntityName	String	Copy of General Admin	The name of the new copy, as requested or determined by TimeTiger.

## Method-specific result codes

**Table 5.13. Result Codes: CopyEntity**

Result	Description
-101	Failed. Missing or invalid EntityID or ToParentEntityID.
-102	Failed. You do not have permission to see all, including restricted, properties on EntityID. You need to be able to see every property of EntityID in order to copy it..
-103	Failed. The ToParentEntityID you specified cannot have children of the type you are copying.
-104	Failed. You do not have permissions to create child nodes under ToParentEntityID.
-105	Failed. The entity name you have specified would create a duplicate underneath the parent, or this is a duplicate entity of a type that requires unique names throughout the system (such as User).

## DeleteEntity

### Syntax

```
DeleteEntity(string SessionID, string EntityID, string ForceDeleteChildren, string ForceDeleteLogs)
```

### Description

Permanently removes the specified entity from TimeTiger, optionally removing all sub-items and associated time logs.

### Input Parameters

**Table 5.14. Input Parameters: DeleteEntity**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
EntityID	String	(UID)	The EntityID of the entity to delete.
ForceDeleteChildren	String	1	If 1, recursively deletes all children of this entity. Otherwise, only deletes this entity if there are no children.
ForceDeleteLogs	String	1	If 1, deletes this entity along with all time logs (if any). Otherwise, only deletes this entity if there are no associated logs.

## Output Parameters

**Table 5.15. Output Parameters: DeleteEntity**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.

## Method-specific result codes

**Table 5.16. Result Codes: DeleteEntity**

Result	Description
-101	Failed.
-102	Failed. You do not have sufficient permissions to delete this entity.
-103	Failed. This entity has time logs associated with it. You must set ForceDeleteLogs = 1 in order to delete this entity.
-104	Failed. This entity has child entities. You must set ForceDeleteChildren = 1 in order to delete this entity.

## Additional information

It is not possible to delete an entity from TimeTiger while there are sub-entities or logs associated with that entity. This method gives you the option of forcibly deleting all sub-entities and logs, or simply failing if sub-items and/or logs exist.

A safe way to implement entity deletion functionality is to first execute this method with ForceDeleteLogs = 0 and ForceDeleteChildren = 0, and if it fails, present the user with the option to delete forcibly. Once an entity is deleted, it cannot be recovered without restoring your TimeTiger database from backup.

## FindEntitiesByName

### Syntax

```
FindEntitiesByName(string SessionID, string NameStartsWith, string
EntityTypeID, string ParentEntityID, string RecurseIntoChildren)
```

### Description

Returns the entity or entities whose name begins with the specified string. Optionally limits the search to entities of a particular type or entities underneath a particular parent entity. The search is case-sensitive.

## Input Parameters

**Table 5.17. Input Parameters: FindEntitiesByName**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
NameStartsWith	String	DAVE	The string to search for (case-sensitive). This can be a full Entity name or the beginning of an Entity name.
EntityTypeID	UID	(16 chars)	If not blank, The ID of the type of entity we're seeking. If blank, finds entities of all types.
ParentEntityID	UID	(16 chars)	If not blank, limits search to the children of the specified parent entity. If blank, considers all entities in the tree.
RecurseIntoChildren	String	0	If 1, searches sub-children, sub-sub-children etc. Otherwise search is limited to the top-most level underneath ParentEntityID (if specified).

## Output Parameters

**Table 5.18. Output Parameters: FindEntitiesByName**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.
EntityNodes	ArrayOfEntityNodes		The set of entities found matching the requested criteria.

## Method-specific result codes

**Table 5.19. Result Codes: FindEntitiesByName**

Result	Description
-101	Failed. The EntityTypeID you specified does not represent an actual EntityType in the system.
-102	Failed. The ParentEntityID you specified does not represent an actual entity in the system.
-103	Failed. No matching entities found.

# GetEntityMetrics

## Syntax

```
GetEntityMetrics(string SessionID, string EntityID, string AsOf)
```

## Description

## Input Parameters

**Table 5.20. Input Parameters: GetEntityMetrics**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
EntityID	UID	(16 chars)	The entity for which to calculate metrics.
AsOf	String	20070113235900	If not blank, the date/time in DBN form at which to calculate metrics. If blank, uses the present instant in time for metrics calculation.

## Output Parameters

**Table 5.21. Output Parameters: GetEntityMetrics**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.
EntityMetrics	EntityMetrics	(EntityMetrics)	The EntityMetrics structure containing the calculated metrics.

## Method-specific result codes

**Table 5.22. Result Codes: GetEntityMetrics**

Result	Description
-101	Failed. The EntityID specified does not represent an actual entity in the system.
-102	Failed. The AsOf date/time you specified is invalid

---

# Chapter 6. Custom Fields

## GetEntityCustomFieldList

### Syntax

```
GetEntityCustomFieldList(string SessionID, string EntityTypeID)
```

### Description

Returns a list of the custom fields available for the specified entity type. There is a Custom Field to store each discrete piece of data (e.g. Name, E-mail address, etc.) for each entity in the system. The API can be used to modify the values in the custom fields using the FieldID returned by this method.

### Input Parameters

**Table 6.1. Input Parameters: GetEntityCustomFieldList**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
EntityTypeID	UID	(16 chars)	The entity type in whose custom fields we are interested. Each entity type has its own unique set of custom fields.

### Output Parameters

**Table 6.2. Output Parameters: GetEntityCustomFieldList**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.
CustomFields	ArrayOfCustomField	(array)	An array of CustomField structures (see the Types section for more details).

### Method-specific result codes

**Table 6.3. Result Codes: GetEntityCustomFieldList**

Result	Description
-101	Failed. Invalid EntityTypeID (it does not present a valid EntityTypeID for this TimeTiger application).

# GetCustomFieldValue

## Syntax

```
GetCustomFieldValue(string SessionID, string CustomFieldID, string EntityID)
```

## Description

Returns the DBN-formatted value of the specified custom field for the specified EntityID.

## Input Parameters

**Table 6.4. Input Parameters: GetCustomFieldValue**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
CustomFieldID	UID	(16 chars)	The ID of the custom field in whose value we are interested. A list of CustomFieldIDs can be obtained using the method GetEntityCustomFieldList
EntityID	UID	(16 chars)	The ID of the entity in whose information we are interested.

## Output Parameters

**Table 6.5. Output Parameters: GetCustomFieldValue**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.
FieldValue	String	+ 180	The retrieved field value, in DBN form.

## Method-specific result codes

**Table 6.6. Result Codes: GetCustomFieldValue**

Result	Description
-101	Failed. The CustomFieldID you specified does not exist.
-102	Failed. The EntityID you specified is not valid.
-103	You do not have sufficient permissions to view the specified field value for the specified entity. The field may be marked as Restricted and you may not have rights to view Restricted properties, or you may simply not be assigned to any role on the entity at all.

## SetCustomFieldValue

### Syntax

```
SetCustomFieldValue(string SessionID, string CustomFieldID, string
EntityID, string FieldValue)
```

### Description

Sets the value of the specified custom field for the specified entity to the passed FieldValue (in DBN form).

### Input Parameters

**Table 6.7. Input Parameters: SetCustomFieldValue**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
CustomFieldID	UID	(16 chars)	The ID of the custom field whose value you wish to change. A list of CustomFieldIDs can be obtained using the method GetEntityCustomFieldList.
EntityID	UID	(16 chars)	The ID of the entity whose information we want to modify.
FieldValue	String	+ 180	The DBN-formatted value we wish to set.

## Output Parameters

**Table 6.8. Output Parameters: SetCustomFieldValue**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.

## Method-specific result codes

**Table 6.9. Result Codes: SetCustomFieldValue**

Result	Description
-101	Failed. The CustomFieldID you specified does not exist.
-102	Failed. The EntityID you specified is not valid.
-103	You do not have sufficient permissions to set the specified field value for the specified entity. The field may be marked as Restricted and you may not have rights to edit Restricted properties, or you may simply not be assigned to any role on the entity at all.

## SetPasswordByUserName

### Syntax

```
SetPasswordByUserName(string SessionID, string UserName, string ClearTextPassword)
```

### Description

Sets the password for the specified User in the system.

### Input Parameters

**Table 6.10. Input Parameters: SetPasswordByUserName**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
UserName	String	DAVE	The TimeTiger user name whose password we wish to change.
ClearTextPassword	String	abcdefg	The password, in clear-text, we wish to set.

## Output Parameters

**Table 6.11. Output Parameters: SetPasswordByUserName**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.

## Method-specific result codes

**Table 6.12. Result Codes: SetPasswordByUserName**

Result	Description
-101	Failed. Missing or invalid UserName.
-102	Failed. You do not have permission to set the password for the specified UserName.

---

# Chapter 7. Roles and Assignment

## GetRoleTypeList

### Syntax

```
GetRoleTypeList(string SessionID)
```

### Description

Retrieves a list of all available security roles, both built-in and custom.

### Input Parameters

**Table 7.1. Input Parameters: GetRoleTypeList**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.

### Output Parameters

**Table 7.2. Output Parameters: GetRoleTypeList**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.
RoleTypes	ArrayOfRoleTypes	(array)	An array of the available RoleTypes. See the Types section of this document for more information about the RoleType type.

### Method-specific result codes

**Table 7.3. Result Codes: GetRoleTypeList**

Result	Description
-101	Failed. No RoleTypes were found. This signifies an error condition, as every TimeTiger database will contain at least the Self role.

# CreateAssignment

## Syntax

```
CreateAssignment(string SessionID, string FromEntityID, string
ToEntityID, string RoleID)
```

## Description

Assigns the User or User Group specified in *FromEntityID* to the item specified in *ToEntityID* in the role specified by *RoleTypeID*. The list of available roles can be obtained using *GetRoleTypeList*.

## Input Parameters

**Table 7.4. Input Parameters: CreateAssignment**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
FromEntityID	UID	(16 chars)	The ID of the User or User group that should be assigned.
ToEntityID	UID	(16 chars)	The item to which the User or User Group should be assigned.
RoleTypeID	UID	(16 chars)	The role in which the assignment should be made.

## Output Parameters

**Table 7.5. Output Parameters: CreateAssignment**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.

## Method-specific result codes

**Table 7.6. Result Codes: CreateAssignment**

<b>Result</b>	<b>Description</b>
-101	The FromEntityID does not exist or you do not have sufficient permissions to assign it.
-102	The ToEntityID does not exist or you do not have sufficient permissions to assign items to it.
-103	The RoleTypeID does not exist.
-104	This role assignment already exists.
-105	The RoleTypeID you specified cannot be used to items of the type of the ToEntityID.
-106	The FromEntityID and ToEntityID cannot be the same.
-107	You cannot assign items in the Self role.

---

# Chapter 8. To Do Items

## GetToDoList

### Syntax

```
GetToDoList(string SessionID, string UserID)
```

### Description

Retrieves the complete To Do list stored on the server for the specified user, as well as any unlogged time captured in the To Do items.

### Input Parameters

**Table 8.1. Input Parameters: GetToDoList**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
UserID	UID	(16 chars)	The ID of the user whose To Do items we are interested in. If left blank, obtains To Do list for the current user. We must have a status of System Admin in order to see any To Do list other than our own.

### Output Parameters

**Table 8.2. Output Parameters: GetToDoList**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.
ToDoS	ArrayOfToDo	(array)	An array of to do items for the specified user. See the Types section of this document for more information about the ToDo type.

## Method-specific result codes

**Table 8.3. Result Codes: GetToDoList**

<b>Result</b>	<b>Description</b>
-101	Failed. No To Do items were found, suggesting that an invalid User ID was passed (as every user should have at least the Nothing To Do item).
-102	Failed. Insufficient permissions. Only System Admins can retrieve To Do lists for other users.

## SaveToDo

### Syntax

```
SaveToDo(string SessionID, string ToDoID, string UserID, string Seq, string Description, string Notes, string WorkInProgress, ArrayOfStringPair Allocation, string OnSince, string StoredSecs)
```

### Description

Creates a new To Do item or updates an existing To Do item in the system.

## Input Parameters

**Table 8.4. Input Parameters: SaveToDo**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
ToDoID	UID	(16 chars)	The ID of the To Do item to overwrite or blank string to create a new To Do item.
UserID	UID	(16 chars)	The UID of the user to whom the To Do item will belong.
Seq	String	45	Numerical sequence of this To Do item in the user's To Do list. Leave blank or pass "0" to add the To Do item to the bottom of the list.
Description	String	Arbirter Support Research	The description of the To Do item as it should appear in the To Do list. Cannot be blank.
Notes	String	Complete all research to handle support query.	Additional notes for the To Do item.
Allocation	ArrayOfStringPair	(Array)	An array of StringPair objects. For each StringPair, the ID should be the EntityTypeID of a bottom-most Entity Type (returned by GetEntityTypeList) and the Value should be the EntityID of a bottom-most entity of the specific type (returned by GetEntities). Note that the User element of the Allocation, if supplied, will be ignored in favor of the value passed as the UserID parameter.

## Output Parameters

**Table 8.5. Output Parameters: SaveToDo**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.
ToDoID	UID	(16 chars)	The UID of the newly created To Do item (or the UID of the updated To Do item).

## Method-specific result codes

**Table 8.6. Result Codes: SaveToDo**

Result	Description
-101	Failed. Invalid parameter value. Details in the Message output parameter.
-102	Failed. Insufficient permissions to modify the existing To Do item or create the new To Do item for the UserID you have specified.
-103	Failed. You have attempted to modify the (Nothing) To Do item. This To Do item cannot be modified.

## SelectToDo

### Syntax

SelectToDo(**string** SessionID, **string** ToDoID)

### Description

Selects the specified To Do as the currently running To Do on the server.

### Input Parameters

**Table 8.7. Input Parameters: SelectToDo**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
ToDoID	UID	(16 chars)	The ID of the To Do item to select. If this ID is blank, selects the (Nothing) To Do item (the (Nothing) To Do item can also be selected by specifying its ID)

### Output Parameters

**Table 8.8. Output Parameters: SelectToDo**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.

## Method-specific result codes

**Table 8.9. Result Codes: SelectToDo**

Result	Description
-101	Failed. The To Do ID specified does not represent a valid To Do item in the system.

## SetToDoSecs

### Syntax

`SetToDoSecs(string SessionID, string ToDoID, string Seconds)`

### Description

Sets the time stored in the specified To Do item. This method will work on any To Do item, including the currently running one.

### Input Parameters

**Table 8.10. Input Parameters: SetToDoSecs**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
Seconds	String	5400	The string representation of the number of seconds to set the To Do item to. This is not in DBN format and is simply a number of seconds stored in string format. Pass "0" to clear the time stored in the To Do item.

### Output Parameters

**Table 8.11. Output Parameters: SetToDoSecs**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.

## Method-specific result codes

**Table 8.12. Result Codes: SetToDoSecs**

Result	Description
-101	Failed. The To Do ID specified does not represent a valid To Do item in the system.

## AddToDoWorkInProgress

### Syntax

```
AddToDoWorkInProgress(string SessionID, string ToDoID, string
WorkInProgress, string Append)
```

### Description

Sets (or optionally appends to) the WorkInProgress field of a To Do item. The WorkInProgress field is used to accumulate a detailed description of a user's work for use when the time is eventually logged.

### Input Parameters

**Table 8.13. Input Parameters: AddToDoWorkInProgress**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
WorkInProgress	String	Prepared preliminary report.	The string to set/append to the WorkInProgress field of the To Do item.
Append	String	1	If "1", appends WorkInProgress to the existing value in the To Do item. Otherwise replaces any existing WorkInProgress.

### Output Parameters

**Table 8.14. Output Parameters: AddToDoWorkInProgress**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.

## Method-specific result codes

**Table 8.15. Result Codes: AddToDoWorkInProgress**

Result	Description
-101	Failed. The To Do ID specified does not represent a valid To Do item in the system.
-102	Failed. You cannot set the WorkInProgress of the (Nothing) To Do item.

## DeleteToDo

### Syntax

DeleteToDo(**string** SessionID, **string** ToDoID, **string** EvenWhenUnlogged)

### Description

Deletes a To Do item from the TimeTiger server, optionally even if there is unlogged time for the item.

### Input Parameters

**Table 8.16. Input Parameters: DeleteToDo**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
ToDoID	UID	(16 chars)	The ID of the To Do item to remove. If this item does not belong to us we must be either a System Admin or the original creator of the item in order to delete it.
EvenWhenUnlogged	String	1	“1” if item should be deleted regardless of whether there is unlogged time. “0” only deletes the item if there is no unlogged time (i.e. the To Do item is not currently running and there is no time stored in the item).

### Output Parameters

**Table 8.17. Output Parameters: DeleteToDo**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. “Success” is a typical result representing completion.

## Method-specific result codes

**Table 8.18. Result Codes: DeleteToDo**

<b>Result</b>	<b>Description</b>
-101	Failed. The To Do ID specified does not represent a valid To Do item in the system.
-102	Failed. You do not have sufficient permissions to remove this To Do item.
-103	Failed. There is unlogged time for the item. If you wish to remove it, set the value of EvenIfUnlogged to "1"
-104	Failed. You have attempted to delete a (Nothing) To Do item. The (Nothing) To Do item cannot be deleted.

---

# Chapter 9. Time Logs

## GetTimeLogs

### Syntax

```
GetTimeLogs(string SessionID, string SubjectID, string FromDate,  
string ToDate, string ApprovalFilter, string MarkAsBilled, string  
AdditionalExportType)
```

### Description

Retrieves a list of time logs from the database and optionally creates an export file of the specified format that can be downloaded separately.

## Input Parameters

**Table 9.1. Input Parameters: GetTimeLogs**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
SubjectID	UID	(16 chars)	The ID of the specific entity (Client, Project, User, etc.) whose time logs should be retrieved, or blank string for all entities.
FromDate	String	20070101	The start date from which logs should be retrieved, or blank string for no start date. Must be blank or in DBN format (yyyymmdd).
ToDate	String	20071231	The end date to which logs should be retrieved, or blank string for no end date. Must be blank or in DBN format (yyyymmdd).
ApprovalFilter	String	1	Flags representing which of the following types of logs should be retrieved. 1 = approved, 2 = not reviewed, 4 = declined. For example, to retrieve declined and not reviewed logs (not not approved logs), pass 6. For all logs, pass 7. This parameter is ignored if time approval is not enabled.
MarkAsBilled	String	2	Controls both which logs are retrieved and whether retrieved logs are marked (or unmarked) as having been billed. Pass one of the following values: <ul style="list-style-type: none"> <li>0. Display all logs</li> <li>1. Display unmarked logs only but not mark them as billed</li> <li>2. Display unmarked logs and mark them as billed</li> <li>3. Display marked logs only</li> <li>4. Display marked logs only and unmark them</li> </ul> This parameter is ignored if Mark as Billed functionality is not enabled.
AdditionalExportType	String	IIF	When non-blank, creates an additional export file of one of the following types depending on the parameter passed. Currently only <b>IIF</b> is supported: this creates an Intuit QuickBooks IIF export file.

## Output Parameters

**Table 9.2. Output Parameters: GetTimeLogs**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. “Success” is a typical result representing completion.
TimeLogs	ArrayOfTimeLog	(array)	An array of time logs that match the passed criteria. See the Types section of this document for more information about the TimeLog type.
ExportFileURL	String	http://server/application/tempfile.htm	A temporarily URL that can be used to download the requested export file (if AdditionalExportType was specified).

## Method-specific result codes

**Table 9.3. Result Codes: GetTimeLogs**

Result	Description
-101	Invalid parameter value. Details in the Message output parameter.

## SaveTimeLog

### Syntax

```
SaveTimeLog(string SessionID, string TimeLogID, string LogDate, string
LogSeconds, string StatusDescription, ArrayOfStringPair Allocation,
string PortionComplete, string Custom1, string Custom2, string Custom3,
string MergeForDay)
```

### Description

Creates a new Time Log or overwrites an existing Time Log in the system.

## Input Parameters

**Table 9.4. Input Parameters: SaveTimeLog**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
TimeLogID	UID	(16 chars)	The ID of the Time Log to overwrite or blank string to create a new Time Log.
LogDate	String	20071231	The date to which the Time Log should be attributed. Must be in DBN format (yyyymmdd).
LogSeconds	String	5400	The number of seconds for the Time Log, expressed as a string. There are 3600 seconds in each hour so, for example, 1.5 hours would be expressed as "5400" For convenience, DBN format is not used for this parameter.
StatusDescription	String	General troubleshooting	The status/description to associate with the Time Log.
Allocation	ArrayOfStringPair		An array of StringPair objects. For each StringPair, the ID should be the EntityTypeID of a bottom-most Entity Type (returned by GetEntityTypeList) and the Value should be the EntityID of a bottom-most entity of the specific type (returned by GetEntities). Note that the Allocation should include an entry for the User type as well, even if the Time Log is for the current user ID.
PortionComplete	String	+ 25/ 100	A VT_FRACTION representation of the % Complete field for this time log. The denominator does not necessarily have to be 100. Passing a blank string is equivalent to entering 0% Complete.
Custom1	String		For future use
Custom2	String		For future use
Custom3	String		For future use
MergeForDay	String	1	If "1", replaces any existing time logs for the same day and the same allocation and attempts to make the total time logged for the day equal to LogSeconds (given that some logs may not be removable because they have been approved or exported). Also merges the Status/Description and other custom fields appropriately. This is useful for merging a set of time logs for one day into a single log.

## Output Parameters

**Table 9.5. Output Parameters: SaveTimeLog**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. “Success” is a typical result representing completion.
TimeLogID	UID	(16 chars)	The UID of the newly created Time Log (or the new UID of the replaced Time Log).
TotalSecsForDay	String	5400	Only returned when MergeForDay is “1”. Returns the actual seconds left in the system for the day and the allocation after the merge operation. This may be > LogSeconds if some existing time logs could not be removed.

## Method-specific result codes

**Table 9.6. Result Codes: SaveTimeLog**

Result	Description
-101	Failed. Details in the Message output parameter.
-102	Invalid parameter value. Details in the Message output parameter.

## DeleteTimeLog

### Syntax

```
DeleteTimeLog(string SessionID, string TimeLogID)
```

### Description

Attempts to remove a single time log from the system. If the log could not be removed (for permissions reasons, time period lockout, approval status, etc.) this method will fail.

## Input Parameters

**Table 9.7. Input Parameters: DeleteTimeLog**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
TimeLogID	UID	(16 chars)	The ID of the time log to remove.

## Output Parameters

**Table 9.8. Output Parameters: DeleteTimeLog**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.

## Method-specific result codes

**Table 9.9. Result Codes: DeleteTimeLog**

Result	Description
-101	Failed. Details in the Message output parameter.
-102	Invalid parameter value. Details in the Message output parameter.

## DeleteTimeLogsForDay

### Syntax

```
DeleteTimeLogsForDay(string SessionID, string LogDate,
ArrayOfStringPair Allocation)
```

### Description

Attempts to remove all time logs for the specific allocation for the specified day. As some time logs may not be removable (for permissions reasons, time period lockout, approval status, etc.) this method returns the total time left unremoved for the day and the allocation.

## Input Parameters

**Table 9.10. Input Parameters: DeleteTimeLogsForDay**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
LogDate	String	20071231	The date to which the Time Log should be attributed. Must be in DBN format (yyyymmdd).
Allocation	ArrayOfStringPair		An array of StringPair objects. For each StringPair, the ID should be the EntityTypeID of a bottom-most Entity Type (returned by GetEntityTypeList) and the Value should be the EntityID of a bottom-most entity of the specific type (returned by GetEntities). Note that the Allocation should include an entry for the User type as well, even if the removal is for the current user ID.

## Output Parameters

**Table 9.11. Output Parameters: DeleteTimeLogsForDay**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.
RemainingSecsForDay	String	0	Returns the number of seconds that could not be removed for the day and the allocation specified.

## Method-specific result codes

**Table 9.12. Result Codes: DeleteTimeLogsForDay**

Result	Description
-101	Failed. Details in the Message output parameter.
-102	Invalid parameter value. Details in the Message output parameter.

# GetTimeLogBilled

## Syntax

```
GetTimeLogBilled(string SessionID, string TimeLogID)
```

## Description

Returns the date and time the time log was marked as billed (exported), and the User ID that performed the billing operation if any. If the log has not been billed, returns a blank string for BilledOn.

Note that to reliably test for whether the log has been billed, check BilledOn and ignore BilledBy as BilledBy could be blank even if the time log has been billed (for example, if the user ID who originally performed the billing operation is no longer in the TimeTiger system).

## Input Parameters

**Table 9.13. Input Parameters: GetTimeLogBilled**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
TimeLogID	UID	(16 chars)	The ID of the Time Log in which we are interested.

## Output Parameters

**Table 9.14. Output Parameters: GetTimeLogBilled**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. "Success" is a typical result representing completion.
BilledOn	String	20071231	The DBN-form date and time when this log was billed (exported), or a blank string if the log has not been billed.
BilledBy	UID	(16 chars)	The UserID of the user that performed the export, or blank string if the log has not been exported or if the UserID is no longer in the system.

## Method-specific result codes

**Table 9.15. Result Codes: GetTimeLogBilled**

Result	Description
-101	Failed. Tracking billed logs is not enabled for this application.
-102	Failed. The TimeLogID you specified is not a valid time log in the system.
-103	You do not have sufficient permissions to view the billing information for the specified time log. The fields may be marked as Restricted and you may not have rights to view Restricted properties, or you may simply not be assigned to any role on the items to which the log is allocated.

## SetTimeLogBilled

### Syntax

```
SetTimeLogBilled(string SessionID, string TimeLogID, string Billed)
```

### Description

Marks or unmarks the specified time log as having been billed (exported) by the current user at the current time.

### Input Parameters

**Table 9.16. Input Parameters: SetTimeLogBilled**

Element	Type	Sample	Description
SessionID	String	(30 chars)	A 30-character unique session identifier originally created using the Hello command.
TimeLogID	UID	(16 chars)	The ID of the Time Log to mark as having been billed.
Billed	String	1	“1” to mark the log as billed, “0” to unmark/clear billing information for the log.

### Output Parameters

**Table 9.17. Output Parameters: SetTimeLogBilled**

Element	Type	Sample	Description
ResultCode	String	0	A numerical code (transmitted as a SOAP string) representing the result of the operation.
Message	String	Success	A textual description of the result of the operation or any error encountered. “Success” is a typical result representing completion.

## Method-specific result codes

**Table 9.18. Result Codes: SetTimeLogBille**

<b>Result</b>	<b>Description</b>
-101	Failed. Tracking billed logs is not enabled for this application.
-102	Failed. The TimeLogID you specified is not a valid time log in the system.
-103	You do not have sufficient permissions to modify the billing information for the specified time log. The fields may be marked as Restricted and you may not have rights to modify Restricted properties, or you may simply not be assigned to any role on the items to which the log is allocated.

---

# Index

## A

AddToDoWorkInProgress, 46

## C

CopyEntity, 27  
CreateAssignment, 39  
CreateEntity, 26  
Custom Fields, 33

## D

Database Normalized (DBN) Values, 10  
DeleteEntity, 29  
DeleteTimeLog, 53  
DeleteTimeLogsForDay, 54  
DeleteToDo, 47

## E

Entities, 22  
Entity Types, 19

## F

FindEntitiesByName, 30

## G

GetAllEntities, 25  
GetCustomFieldValue, 34  
GetEntities, 23  
GetEntity, 22  
GetEntityCustomFieldList, 33  
GetEntityMetrics, 32  
GetEntityTypeList, 19  
GetRoleTypeList, 38  
GetTimeLogBilled, 56  
GetTimeLogs, 49  
GetToDoList, 41  
GetValidChildTypes, 20  
Goodbye, 14

## H

Hello, 12

## L

LoginStandard, 15  
LoginWindows, 16

## R

Roles and Assignment, 38

## S

SaveTimeLog, 51  
SaveToDo, 42  
SelectToDo, 44  
Session Management, 12  
SetCustomFieldValue, 35  
SetPasswordByUserName, 36  
SetTimeLogBilled, 57  
SetToDoSecs, 45

## T

Time Logs, 49  
To Do Items, 41  
Touch, 13  
Types, 1