

## **TELL ME MORE ONLINE 10 – Standard portal integration -**

### Technical specification



This document describes possibilities of integration between an external portal and TELL ME MORE ONLINE Portal version 10 (TOL10).



## TELL ME MORE ONLINE 10 – Standard portal integration -

### Technical specification

TELL ME MORE Online provides several commands in order to enable integration between TOL10 and an external portal:

For general information about V10 concepts see **Appendix 5 – V10 Concepts**

### Table of contents

Common Parameters .....	3
Configuration File .....	5
Command 101 - User Creation .....	6
Command 102 - User Modification .....	8
Command 103 - User Removal .....	10
Command 104 - User Activation/Deactivation .....	11
Command 105 - User Virtual Classroom Time Upload.....	12
Command 107 – Create a Tutor and assign all the pupils of an administrative group.....	13
Command 201 - Group Creation .....	14
Command 202 - Group Modification .....	15
Command 301 - Group Assigning .....	16
Command 302 - Group Unassigning .....	17
Command 303 - Tutor Assigning (only for Corporate).....	18
Command 304 - Tutor Unassigning (only for Corporate) .....	19
Command 401 - Direct connection to the portal .....	20
Command 402 - Direct connection to the tests.....	21
Command 403 - User Creation/Modification and Direct connection to the portal .....	22
Command 501 - Enumeration of all the groups .....	24
Commands 502/503 - Host information export .....	26
Command 505 – Evaluation test result export .....	32
Command 506 – Pupil XML information by Administrative Group .....	34
Command 507 – Get Test result information for a day .....	42
Appendix 1 - Disciplines.....	44
Appendix 2 - Interface languages.....	44
Appendix 3 - Code Samples.....	46
Direct connection to TOL's portal by using PHP .....	46
Direct connection to TOL's portal by using ASP.NET.....	47
Computation of ptf_check in C# .....	49
Redirection using POST .....	51
Appendix 4 - Error Codes.....	52
Appendix 5 – V10 Concepts .....	53
UML Diagrams equivalence .....	55
UML Diagram for Corporate market .....	56
UML Diagram for Education market .....	57

At the end of this document, several examples show implementations of these commands, and how to enable these functionalities in TELL ME MORE Online.

## Common Parameters

First, the URL to call for all the commands available in the TOL API is the following:

<http://tmmserver/ApiHandler.ashx>

With tmmserver = the URL of the Tell Me More Online Server.

In addition to this URL, some parameters are needed:

- **ptf\_cmd** = the command to launch (Example: 301) as defined in the following sections
- **ptf\_login** = login of the user (Example: johnb)
- **ptf\_partner** = name of the partner (Example: alpha)
- **ptf\_timer** = number of seconds since 1970 January first (Example: 1098230427). This string must not be encoded. Time must be GMT (or UTC or TAI), not local time.
- **ptf\_method** = method used to authenticate the request (Example: md5). See the list below for the list of available methods. This parameter is case-insensitive.
- **ptf\_check** = calculated value to authenticate the request (Example: 57edf4a22be3c955ac49da2e2107b67a). See below to see how to calculate this value.

All these parameters, but also the parameters defined in the following pages must be UTF8, then URL encoded. All parameters are case-sensitive, except those marked explicitly as case-insensitive.

Example:

p@βwôrd → p@ÃŸwÃ´rd (UTF8 encoded) → p%40%C3%83%C5%B8w%C3%83%C2%B4rd (URL encoded)

Another parameter is provided to the partner but must not be given in the parameters: **ptf\_key** (Example: p@βwôrd). It is only used to calculate the ptf\_check.

**ptf\_check** is calculated according this formula:

**ptf\_check = ptf\_method(ptf\_login + ptf\_partner + ptf\_timer + ptf\_key)**

**Remark:** **ptf\_login** and **ptf\_partner** and **ptf\_timer** and **ptf\_key** are UTF8 encoded but not URL encoded when used in this formula.

For some commands, there is no ptf\_login parameter, so another value is used. See the specific documentation of each command for more information.

TOL10 server will check that ptf\_check is correct, and that ptf\_timer matches the server time (a difference of several minutes between servers is authorized; this value is set in the configuration file).

Example:

**ptf\_check** = md5(johnbalph1098230427p@ÃŸwÃ´rd)  
= EEB0B146D41F3BEA7D341761CB82A38A

The following values are supported for **ptf\_method** (case-insensitive):

- MD5
- SHA1
- SHA256
- SHA384
- SHA512

In case of error, the returned stream will be a XML string of this form:

```
<result>  
  <errorcode>3</errorcode>  
  <errortext>Can not find ptf_partner parameter</errortext>  
</result>
```

In case of success, the returned stream depends on the command. See the appropriate command documentation for more information.

## Configuration File

The configuration file is called **ApiPartnersParameters.xml** and is located under the directory “Settings” of the directory defined in the web.config of the website under the key “FileLibraryFolder”. It has the following form:

```
<?xml version="1.0" encoding="utf-8" ?>
<partners>
  <partner name="Auralog">
    <param key="code" value="Auralog" />
    <param key="delta" value="300" />
    <param key="prefix" value="Auralog_" />
    <param key="hosts" value="1;2" />
    <param key="commands" value="301;302" />
    <param key="HTTPStoHTTP" value="TRUE" />
  </partner>
</partners>
```

It should contain one “partner” tag for each authorized partner (the attribute “name” is the name of the partner, the same value that should be given as “ptf\_partner”).

- code (**required**): the value corresponding to the “ptf\_key” of the partner. This code should only contain ASCII characters.
- delta (**required**): the maximal time difference in seconds which is authorized between the TOL server and the portal server. We suggest using 300.
- prefix (**required**): a prefix that will be added to the login of all the created users.
- hosts (**required**): the list of hosts where the partner has the right to modify (add users, groups, etc). Hosts are identified by their unique number. Numbers can be separated by a semicolon “;” if more than one host is authorized.
- commands (**optional**): the list of commands available for the partner. It’s only useful for the command that need a special grant: the command 501 (the other commands don’t need this parameter). Commands are identified by their unique number. Numbers can be separated by a semicolon “;” if more than one special command is authorized.
- HTTPStoHTTP (**optional, default=“FALSE”**): if “TRUE” and if the API is called with HTTPS, the redirection to the portal will be done with HTTP (for cmd 401 and 403)

## Command 101 - User Creation

Value of ptf\_cmd: 101

Request type allowed: POST

Description: This command allows creating a user and gives him the specified role(s). If the user already exists and you want to give him another role (or change its role), use command 102.

In addition to the following parameters, all the parameters described in the section “Common Parameters” must be present.

Parameters are the following:

- **ptf\_role** (required) = the roles to give to the user. It's a value between 1 and 7:

ptf_role	HRManager	Tutor	Pupil
1			X
2		X	
3		X	X
4	X		
5	X		X
6	X	X	
7	X	X	X

- **ptf\_password** (required) = password of the user (limited to 16 characters)
- **ptf\_interfacelanguage** (required) = interface language. See Appendix 2 to know the correct values. (Example: fr-FR)
- **ptf\_firstname** (required) = user's first name (Example: John)
- **ptf\_lastname** (required) = user's last name (Example: Bell)
- **ptf\_email** (required) = user's email address (Example: johnb@auralog.com)
- **ptf\_client** (required for roles Pupil and HRManager) = id of the client containing the pupil (Example: 7)
- **ptf\_host** (required for roles Tutor) = id of the host containing the tutor (Example: 3)
- **ptf\_pupildiscipline** (required for roles Pupil) = desired discipline(s) for the pupil. See Appendix 1 to know the correct values. If multiple disciplines are desired, their codes must be separated by “;” (Example: ENGLISH;FRENCH)
- **ptf\_tutordiscipline** (required for roles Tutor) = desired discipline(s) for the tutor. See Appendix 1 to know the correct values. If multiple disciplines are desired, their codes must be separated by “;” (Example: ENGLISH; FRENCH)
- **ptf\_offername** (required for pupil creation, except if the system has only one offer) = name of the offer. An offer is a combination of settings that can be applied to a pupil. This settings make possible to fix the formation duration, the discipline learned and the possible accessed features that the user can use in his portal (ex: can use the Level test, the Language Achievement test, etc.). The offer should allow multiple disciplines or a free discipline choice.
- **ptf\_templatename** (optional, only for pupil creation) = name of the planning template to use for the pupil. A planning template is a group of tasks that can be applied to a pupil training planning. These tasks make possible to send email, generate detailed report etc occurring at specific dates.

- **ptf\_detailedResult** (optional) = should be 1 to return a detailed result if the creation succeeds.  
Default value is 0.

If ptf\_detailedResult is 0 and the user was correctly created, the returned stream will be a XML string of this form:

```
<result>Success</result>
```

If ptf\_detailedResult is 1 and the user was correctly created, the returned stream will be a XML string of this form:

XML Tag		Description
<result>		This is the root XML node
<user		Information relative to the user
	role="1"	Role of the user 1..7
	username="thierry1"	The user name of the user
	userid="ec28c1fd-acbf-406e-b3e5-a9f098b194ce"	The user internal ID in TELL ME MORE
	date="2011-02-25 18:01:59"	Date of pupil creation  Format: YYYY-MM-DD HH:MM:SS"
	>	
</result>		

Example of returned XML stream:

```
<result>
<user role="1" username="thierry1" userid="ec28c1fd-acbf-406e-b3e5-a9f098b194ce"
date="2011-02-25 18:01:59" />
</result>
```

## Command 102 - User Modification

Value of ptf\_cmd: 102

Request type allowed: POST

Description: This command allows modifying a user and changing his role(s). The user must imperatively exist.

In addition to the following parameters, all the parameters described in the section “Common Parameters” must be present.

Parameters are the following:

- **ptf\_role** (optional) = the role to give to the user. It's a value between 1 and 7:  
If the user has a role that is not in **ptf\_role**, this role is removed from the user.

ptf_role	HRManager	Tutor	Pupil
1			X
2		X	
3		X	X
4	X		
5	X		X
6	X	X	
7	X	X	X

- **ptf\_newlogin** (optional) = new login of the user. If the new login value already exists or if this value is the same as the current login, the result error code is UserAlreadyExists.
- **ptf\_password** (optional) = password of the user
- **ptf\_interfacelanguage** (optional) = interface language. See Appendix 2 to know the correct values. (Example: fr-FR)
- **ptf\_firstname** (optional) = user's first name (Example: John)
- **ptf\_lastname** (optional) = user's last name (Example: Bell)
- **ptf\_email** (optional) = user's email address (Example: johnb@auralog.com)
- **ptf\_client** (optional) = id of the client containing the pupil(Example: 7). This parameter is taken into account only when adding a new role to a user.
- **ptf\_host** (optional) = id of the host containing the tutor (Example: 3). This parameter is taken into account only when adding a new role to a user.
- **ptf\_pupildiscipline** (optional) = desired discipline(s) for the pupil. See Appendix 1 to know the correct values. If multiple disciplines are desired, their codes must be separated by “;” (Example: ENGLISH;FRENCH). Note that once added, disciplines can't be removed (this command will only add new disciplines).Warning: to add new disciplines, the offer must not define a fixed discipline.
- **ptf\_tutordiscipline** (optional) = desired discipline(s) for the tutor. See Appendix 1 to know the correct values. If multiple disciplines are desired, their codes must be separated by “;” (Example: ENGLISH;FRENCH). Note that once added, disciplines can't be removed (this command will only add new disciplines).
- **ptf\_offername** (required only for pupil role creation from existing account, except if the system has only one offer) = this parameter is used only for creating the Pupil role if it not already exists (ex: add the Pupil role to an Account that has already the Tutor role). However, if the system has only one offer, this parameter is ignored so it can be not used. This parameter is



not used if the pupil role already exists (it is not possible to add or replace an offer to an existing pupil).

- **ptf\_templatename** (optional, only for pupil) = name of the planning template to use for the user ((add this parameter only to add a NEW planning template to the user)

If the user was correctly modified, the returned stream will be a XML string of this form:

`<result>Success</result>`

## Command 103 - User Removal

Value of ptf\_cmd: 103

Request type allowed: POST

Description: This command will remove the selected role(s) from the user defined by ptf\_login and delete the user if he has no remaining role.

In addition to the following parameters, all the parameters described in the section “Common Parameters” must be present.

Parameters are the following:

- o **ptf\_role** (required) = roles to remove. It's a value between 1 and 7.  
If after removing these roles the user has no role, it is deleted.

ptf_role	HRManager	Tutor	Pupil
1			X
2		X	
3		X	X
4	X		
5	X		X
6	X	X	
7	X	X	X

If the user roles were correctly deleted, the returned stream will be a XML string of this form:

<result>Success</result>

## Command 104 - User Activation/Deactivation

Value of ptf\_cmd: 104

Request type allowed: POST

Description: This command will activate or deactivate the user defined by ptf\_login (it means the user has to already exist).

In addition to the following parameters, all the parameters described in the section “Common Parameters” must be present.

Parameters are the following:

- **ptf\_active** (required) = indicates if we have to activate or deactivate the user. It's a value between 0 and 1.
  - **0** = deactivate the user
  - **1** = activate the user

If the user role was correctly activated or deactivated, the returned stream will be a XML string of this form:

`<result>Success</result>`

## Command 105 - User Virtual Classroom Time Upload

Value of ptf\_cmd: 105

Request type allowed: POST

Description: This command will store in Tell Me More the time spent in a virtual classroom by the user defined by ptf\_login (it means the user has to already exist), along with other additional data concerning the classroom.

Grant: This command needs a special grant and "105" should be added in the section "commands" of the corresponding partner in the configuration file (see corresponding section).

Virtual Classroom session type is stored in TELL ME MORE with the value 7.

In addition to the following parameters, all the parameters described in the section "Common Parameters" must be present.

Parameters are the following:

- **ptf\_discipline** (required) = indicates the discipline of the pupil when he accessed the virtual classroom. See Appendix 1 to know the correct values.
- **ptf\_class\_id** (required) = the virtual classroom unique id (limited to 255 characters).
- **ptf\_class\_topic** (required) = the topic of the virtual classroom (limited to 255 characters)
- **ptf\_class\_level** (required) = the level of the virtual classroom (limited to 255 characters)
- **ptf\_start\_date** (required) = the date the user entered into the virtual classroom in UTC. Format should be as follows: "YYYY-MM-DD HH:MM:SS" (Example: 2011-02-25 18:01:59).
- **ptf\_duration** (required) = the time spent in the classroom, in seconds (Example: 300).
- **ptf\_connection\_breaks** (optional) = the number of times the user left the classroom during its session. Default value is 0.

If the time in the virtual classroom was correctly stored, the returned stream will be a XML string of this form:

<result>Success</result>

## Command 107 – Create a Tutor and assign all the pupils of an administrative group

Value of ptf\_cmd: 107

Request type allowed: POST

Description: This command creates a Tutor with ptf\_login username if he doesn't already exist. He then sets all the group pupils' disciplines to this tutor and assigns all the pupils of the administrative group. The pupils that are already assigned to the tutor but don't belong to the group anymore are unassigned.

In addition to the following parameters, all the parameters described in the section "Common Parameters" must be present.

Parameters are the following:

- **ptf\_password** (required) = password of the user (limited to 16 characters)
- **ptf\_interfacelanguage** (required) = interface language. See Appendix 2 to know the correct values. (Example: fr-FR)
- **ptf\_firstname** (required) = user's first name (Example: John)
- **ptf\_lastname** (required) = user's last name (Example: Bell)
- **ptf\_email** (optional) = user's email address (Example: johnb@auralog.com)
- **ptf\_groupnumber** (required) = number of the administrative group

## Command 201 - Group Creation

Value of ptf\_cmd: 201

Request type allowed: POST

Description: This command allows creating a new group among the following: administrative group, pedagogic group and classroom.

In addition to the following parameters, all the parameters described in the section “Common Parameters” must be present, except ptf\_login. Use ptf\_groupname instead of ptf\_login to calculate ptf\_check.

Parameters are the following:

- **ptf\_groupname** (required) = name of the group (Example: Classroom1)
- **ptf\_groupdescription** (optional) = description of the group (Example: Classroom for students learning English 1)
- **ptf\_discipline** (required except for administrative groups) = discipline of the group (Example: ENGLISH).
- **ptf\_client** (required) = id of the client (Example: 7)
- **ptf\_grouptype** (required) = type of group to create. It's a value between 1 and 3:
  - 1 = create an administrative group
  - 2 = create a pedagogic group
  - 3 = create a classroom

If the group was correctly created, the returned stream will be a XML string of this form:

```
<result>  
  <groupnumber>X</groupnumber>  
</result>
```

With X = the ID of the created group. This ID can be used to modify the group or assign users to the group.

## Command 202 - Group Modification

Value of ptf\_cmd: 202

Request type allowed: POST

Description: This command allows modifying an existing group among the following: administrative group, pedagogic group and classroom.

In addition to the following parameters, all the parameters described in the section “Common Parameters” must be present, except ptf\_login. Use ptf\_groupnumber instead of ptf\_login to calculate ptf\_check.

Parameters are the following:

- **ptf\_groupnumber** (required) = ID of the group as returned by the creation command (201) or the enumeration command (501) (Example: 7)
- **ptf\_groupname** (optional) = name of the group (Example: Classroom1)
- **ptf\_groupdescription** (optional) = description of the group (Example: Classroom for students learning English 1)
- **ptf\_grouptype** (required) = type of group to modify. It will not change the type of group, it is just used to indicate which group type the group number is corresponding to. It's a value between 1 and 3:
  - **1** = the group is an administrative group
  - **2** = the group is a pedagogic group
  - **3** = the group is a classroom

If the group was correctly modified, the returned stream will be a XML string of this form:

**<result>Success</result>**

## Command 301 - Group Assigning

Value of ptf\_cmd: 301

Request type allowed: POST

Description: This command will put the user defined by ptf\_login into the group defined by ptf\_groupnumber.

In addition to the following parameters, all the parameters described in the section “Common Parameters” must be present.

Parameters are the following:

- **ptf\_groupnumber** (required) = ID of the group as returned by the creation command (Example: 7)
- **ptf\_assigntype** (required) = type of assigning. It's a value between 1 and 6:
  - 1 = assign an administrative group to a pupil
  - 2 = assign a pedagogic group to a pupil
  - 3 = assign a classroom to a pupil
  - 4 = assign a classroom to a tutor
  - 5 = assign an administrative group to a HRManager
  - 6 = assign a classroom to a HRManager

If the user was correctly added to the group, the returned stream will be a XML string of this form:

<result>Success</result>



## Command 302 - Group Unassigning

Value of ptf\_cmd: 302

Request type allowed: POST

Description: This command will remove the user defined by ptf\_login from the group defined by ptf\_groupnumber.

In addition to the following parameters, all the parameters described in the section “Common Parameters” must be present.

Parameters are the following:

- **ptf\_groupnumber** (required) = ID of the group as returned by the creation command (201) or the enumeration command (501) (Example: 7)
- **ptf\_assigntype** (required) = type type of assigning. It's a value between 1 and 6:
  - 1 = remove a pupil from an administrative group
  - 2 = remove a pupil from a pedagogic group
  - 3 = remove a pupil from a classroom
  - 4 = remove a tutor from a classroom
  - 5 = remove a HRManager from an administrative group
  - 6 = remove a HRManager from a classroom

If the user was correctly removed from the group, the returned stream will be a XML string of this form:

`<result>Success</result>`

## Command 303 - Tutor Assigning (only for Corporate)

Value of ptf\_cmd: 303

Request type allowed: POST

Description: This command will attach the pupil defined by ptf\_login with the tutor defined by ptf\_tutorlogin. (In Education Market, the tutor should be attached to the classroom using command 301).

In addition to the following parameters, all the parameters described in the section “Common Parameters” must be present.

Parameters are the following:

- **ptf\_tutorlogin** (required) = login of the tutor (Example: tutor1)
- **ptf\_discipline** (required) = the tutor will be attached for this discipline (Example: ENGLISH)

If the pupil was correctly attached to the tutor, the returned stream will be a XML string of this form:  
<result>Success</result>

## Command 304 - Tutor Unassigning (only for Corporate)

Value of ptf\_cmd: 304

Request type allowed: POST

Description: This command will detach the pupil defined by ptf\_login from the tutor defined by ptf\_tutorlogin.

In addition to the following parameters, all the parameters described in the section “Common Parameters” must be present.

Parameters are the following:

- **ptf\_tutorlogin** (required) = login of the tutor (Example: tutor1)
- **ptf\_discipline** (required) = the tutor will be detached for this discipline (Example: ENGLISH)

If the pupil was correctly detached from the tutor, the returned stream will be a XML string of this form:

`<result>Success</result>`

## Command 401 - Direct connection to the portal

Value of ptf\_cmd: 401

Request type allowed: GET/POST

Description: This command will redirect the user defined by ptf\_login to the portal and automatically identify him.

In addition to the following parameters, all the parameters described in the section “Common Parameters” must be present.

If “HTTPStoHTTP” = TRUE in ApiPartnersParameters.xml file and if the API is called with HTTPS, the redirection to the portal will be done with HTTP.

Parameters are the following:

- **ptf\_endingurl** (required) = URL to return when the user logs off (Example: <http://www.auralog.com>)
- **ptf\_come\_from\_lms** (optional) = should be 1 if the Tell Me More should be displayed in “LMS mode” (as if the user comes from a LMS using Tell Me More LMS integration). 0 otherwise. Default value is 0.
- **ptf\_discipline** (optional) = discipline (Exemple: ENGLISH). This value is used after pupil authentication to automatically select the matching discipline and prevent any pupil manual discipline selection. Default value is Empty. In such case, there is no automatic discipline selection. If the pupil has several disciplines, he will need to manually select it in the portal.

## Command 402 - Direct connection to the tests

Value of ptf\_cmd: 402

Request type allowed: GET/POST

Description: This command will redirect the user defined by ptf\_login to the appropriate test.

In addition to the following parameters, all the parameters described in the section "Common Parameters" must be present.

The url of the test is read in the solution.xml file.

Parameters are the following:

- **ptf\_endingurl** (required) = URL to return when the test is finished (Example: <http://www.auralog.com>)
- **ptf\_testtype** (required) = type of test to display. It's a value between 1 and 2:
  - 1 = Position Test
  - 2 = Certification Test
- **ptf\_discipline** (required) = discipline of the test (Example: ENGLISH)

## Command 403 - User Creation/Modification and Direct connection to the portal

Value of ptf\_cmd: 403

Request type allowed: GET/POST

Description: This command allows creating a user (or modify him if he already exists) and redirects him to the portal automatically identified. It's a combination of commands 101/102 and 401

In addition to the following parameters, all the parameters described in the section "Common Parameters" must be present.

If "HTTPStoHTTP" = TRUE in ApiPartnersParameters.xml file and if the API is called with HTTPS, the redirection to the portal will be done with HTTP.

Parameters are the following:

- **ptf\_role** (required) = the role to give to the user. It's a value between 1 and 3:
  - 1 = pupil
  - 2 = tutor
  - 3 = pupil + tutor
- **ptf\_password** (required) = password of the user
- **ptf\_interfacelanguage** (required) = interface language. See Appendix 2 to know the correct values. (Example: fr-FR)
- **ptf\_pupildiscipline** (required for roles 1 and 3) = desired discipline(s) for the pupil. See Appendix 1 to know the correct values. If multiple disciplines are desired, their codes must be separated by ";" (Example: ENGLISH;FRENCH)
- **ptf\_tutordiscipline** (required for roles 2 and 3) = desired discipline(s) for the tutor. See Appendix 1 to know the correct values. If multiple disciplines are desired, their codes must be separated by ";" (Example: ENGLISH; FRENCH)
- **ptf\_firstname** (required) = user's first name (Example: John)
- **ptf\_lastname** (required) = user's last name (Example: Bell)
- **ptf\_email** (required) = user's email address (Example: johnb@auralog.com)
- **ptf\_client** (required for role 1 and 3) = id of the client containing the pupil (Example: 7)
- **ptf\_host** (required for role 2 and 3) = id of the host containing the tutor (Example: 3)
- **ptf\_endingurl** (required) = URL to return when the user logs off (Example: <http://www.auralog.com>)
- **ptf\_offername** (required for pupil creation when creating the user except if the system has only one offer, optional when modifying the user) = name of the offer (when the user already exists, add this parameters only to add a NEW offer)
- **ptf\_templatename** (optional, only for pupil) = name of the planning template to use for the user (when the user already exists, add this parameters only to add a NEW template)
- **ptf\_admingroup** (required for role 1 and 3 in Corporate) = ID of the administrative group to attach to the created user if it's a pupil. This parameter is only used to create the user (not to modify it).
- **ptf\_pedagroup** (required for role 1 and 3 in Corporate) = ID of the pedagogic groups to attach to the created user if it's a pupil. Separate the numbers by ";" if there are more than one group. This parameter is only used to create the user (not to modify it).
- **ptf\_classroom** (required in Education) = ID of the classrooms to attach to the created user (whether it's a pupil or a tutor). Separate the numbers by ";" if there are more than one

- classroom. This parameter is only used to create the user (not to modify it).

**Warning:**

- if **ptf\_role** value is **1** or **3**, the discipline of each classroom should correspond to one of the **ptf\_pupildiscipline** identifiers (the pupil should have a learning project corresponding to each classroom discipline)
- if **ptf\_role** is **2** or **3**, the discipline of each classroom should correspond to the one of the **ptf\_tutordiscipline** identifiers (the tutor should have a teaching project corresponding to each classroom discipline)

## Command 501 - Enumeration of all the groups

Value of ptf\_cmd: 501

Request type allowed: GET/POST

Description: This command will list all the groups of the system and give useful information about them.

Grant: This command needs a special grant and "501" should be added in the section "commands" of the corresponding partner in the configuration file (see corresponding section).

In addition to the following parameters, all the parameters described in the section "Common Parameters" must be present, except ptf\_login. An empty ptf\_login should be used to generate ptf\_check.

This command returns the groups for the Hosts that are defined in ApiPartnersParameters.xml file on the TellMeMore server side:

```
<param key="hosts" value="1;2" />
```

If no Host is defined in ApiPartnersParameters.xml, all the Hosts groups are returns.

Warning: ApiPartnersParameters.xml can contain Host ids that don't exist in the solution: they must be ignored.

The command returns an XML stream of this form:

```
<hosts>
  <host id="1" name="Host 1" lang="fr-fr|en-US" country="US" salesarea="North US"
salesrepresentative="commercial1" >
    <clients>
      <client id="1" name="Client 1">
        <admingroups>
          <admingroup id="1" name="Admin Group 1" />
          <admingroup id="2" name="Admin Group 2" />
        </admingroups>
        <classrooms>
          <classroom id="1" name="Classroom 1" discipline="2" grid="xxx" />
          <classroom id="2" name="Classroom 2" discipline="4" grid="yyy" />
        </classrooms>
      </client>
    </clients>
    <pedagroups>
      <pedagroup id="1" name="Pedagogic Group 1" discipline="1" grid="zzz" />
      <pedagroup id="2" name="Pedagogic Group 2" discipline="3" grid="" />
    </pedagroups>
    <offers>
      <offer name="Offer 1" days="180" offertype="AllDisciplines" totalassigned="3500"
totalconsumed="1022" />
      <offer name="Offer 2" days="365" offertype="OneSelectedByUser" totalassigned="1500"
totalconsumed="102" />
    </offers>
  </host>
</hosts>
```

The country field value ("US") corresponds to the code provided in § "2.4 Countries" of document 01-ListsOfIdentifiers).

Notice that the salesarea field value is returned in "en-US" interface language.



**Warning:** The V10 solution can be deployed for Corporate market or for Education market (no mixed configuration possible). So this example displays an xml content that contains information that could not exist at the same time for the same deployed Solution.

Notice that for Corporate market, the `<classrooms>` node is empty and for Education market, the `<admingroups>` and `<pedagroups>` nodes are empty. The `lang` attribute contains the available interface languages for pupils of the host.

## Commands 502/503 - Host information export

Value of ptf\_cmd: 502

Request type allowed: GET/POST

Description: This command will return the IDs of all the active pupils in the given host.

In addition to the following parameters, all the parameters described in the section “Common Parameters” must be present, except ptf\_login. **An empty ptf\_login should be used to generate ptf\_check.**

Parameters are the following:

- **ptf\_host** (required) = the ID of the desired host
- **ptf\_admingroup** (optional) = ID of the desired administrative group. We will return only the IDs of the pupils inside this administrative group.
- **ptf\_pedagroup** (optional) = ID of the desired pedagogic group. We will return only the IDs of the pupils inside this pedagogic group.
- **ptf\_classroom** (optional) = ID of the desired classroom. We will return only the IDs of the pupils inside this classroom.
- **ptf\_pupiltype** (optional) = the type of the pupils that are requested. It can be one of these values:
  - 0: All pupils (“Active” and “Not Active” Pupils are returned)
  - 1: Only the “Active” pupils are returned (this is the default value used if the parameter **ptf\_pupiltype** is not used).  
A pupil is considered as “Active” if the Pupil role is Active **AND** if there is at least one active learning project.
  - 2: Only the “Not Active” pupils are returned  
The pupil is considered as “Not Active” if the Pupil role is deactivated **OR** if all the learning projects are not active.
- **ptf\_timestamp** (optional) = the number of Ticks since Jan 01 1970 giving the UTC datetime. We will return only the IDs of the pupils who have the end date of their last connection session after or equal this timestamp.

The result will be as follows:

```
<result>
  <pupil_ids>1;2;3;4;5;6;7</ pupil_ids >
</result>
```

The list of the pupil ids separated by a semicolon (;).

Value of ptf\_cmd: 503

Request type allowed: GET/POST

Description: This command will return information about the given pupils.

In addition to the following parameters, all the parameters described in the section “Common Parameters” must be present, except ptf\_login. **An empty ptf\_login should be used to generate ptf\_check.**

Parameters are the following:

- **ptf\_pupil\_ids** (required) = ids of the pupils to return information on (limited to 20), separated by a semicolon ";" (Example: 1;2;3;4;5)

**Warning:** If the command 502 has returned more than 20 pupil ids, the call of the 503 command should be done several times using a maximum of 20 pupil ids at the same time. For example, if the command 502 returns 45 pupil ids (ex Id from #5000 to id #5044), the command 503 must be called 3 times like this:

- 1st call with the first 20 ids **ptf\_pupil\_ids = 5000;5001;5002;.....;5018;5019**
- 2nd call with the next 20 ids **ptf\_pupil\_ids = 5020;5021;5022;.....;5038;5039**
- Last call with the last 5 ids **ptf\_pupil\_ids = 5040;5041;5042;5043;5044**

Result example:

```
Work group;ID;First name;Surname;Status;Student group;E-mail;User name;Tutor's first
name;Tutor name;Lesson languages;Language;Language group;Date of project creation;Date of
first entry in personal objectives form ;Start date;Date of student's first session;End
date;Dates student account deactivated;Suspension dates;Number of days training
suspended;Placement Test results;Progress Test results;Total study time in between the
two tests;Path level;Programmed Learning Path;Total working time;Average working time (7
days);Date of last session;Date last message sent;Number of messages sent;Result of the
TOEIC Preparation Test/Language Achievement Test;Tutoring offer;Time spent studying since
objectives were set;Result of second Language Achievement Test;Language;24-hour customer
service;Conversation Classes;Paths worked on;% completed, % completed successfully
1;PUPIL1;Kent;Brockman;;;Ken@Brockman.com;pupil1;;;#STR_ENGLISH.en-US#;#STR_ANGLAIS.en-
US#;PedagGRP1;25/12/2010;26/12/2010;25/12/2010;27/12/2010;05/05/2011;01/05/2011;25/01/2011-
27/01/2011,12/02/2011-15/02/2011;5;;;;;;;;;0;;;No;No;;;;
```

## Important information

The returned result contains one line for each pupil id given **ptf\_pupil\_ids** parameter. However in the case a pupil is learning more than one disciplines (ex Italian and Spanish) the exported result contains one distinct line for each pupil's discipline even if the discipline is deactivated (2 lines for this pupil in the given example).

For example, if the command 503 is called with 5 pupil ids and each of these pupils learns 2 disciplines, the returned result contains 10 lines (5x2).

Notice that:

- this command returns information for Active and Not active disciplines.
- All the dates are returned in UTC format

## Description of result columns:

1. **Work group**  
Host number (equivalent to Workgroup) where the pupil is registered
2. **ID**  
Pupil's Id
3. **First name**  
Pupil's First name

#### 4. Surname

Pupil's Last name

#### 5. Status

Pupil's training status. It can be the following values:

- **NotStarted**: The pupil hasn't started his study.
- **ObjectiveDefined**: The pupil already defined his objectives.
- **InProgress**: The pupil started to train or have worked\* in the Virtual classroom.
- **Completed**: The pupil has finished his study.

The 3 values "**InProgress**", "**ObjectiveDefined**" and "**NotStarted**" are depending on the Virtual classroom and Mobile Application working time. See <[UC0502](#)> § "**Training Status**" & § "**Learners Status**" for detailed information.

\*Virtual classroom is considered as "worked" only if there are Virtual classroom session duration greater than 0.

See also § "Working Time and deactivation of the Mobile Application" in document 0625-Pupil-LearningAchievementAdjustments\_BA\_UseCase in case of Mobile application deactivation.

#### 6. Student group

Administrative group name where the pupil is assigned for [CO] Corporate market. This field is empty in [EDU] Education market.

#### 7. E-mail

Pupil's email

#### 8. User name

Pupil's username

#### 9. Tutor's first name

Tutor's first name. This field is empty if there is no tutor assigned to the pupil.

#### 10. Tutor name

Tutor's last name. This field is empty if there is no tutor assigned to the pupil.

#### 11. Lesson language

Discipline (V10 official term) learned by the pupil.

#### 12. Language

Interface language of the pupil (ex: "fr-FR", "en-GB", etc.)

#### 13. Language group

According to the Market type:

- Corporate [**CO**] This is **Pedagogic group** name where the pupil is assigned for the matching Discipline (Lesson language)
- Education [**EDU**] This is **Classroom** name where the pupil is assigned for the matching Discipline (Lesson language)

14. **Date of project creation** (Format dd/MM/yyyy)  
Date (Format dd/MM/yyyy) at which the pupil Discipline has been created. ~~(added) to the pupil~~
15. **Date of first entry in personal objectives form** (Format dd/MM/yyyy)  
Date at which the pupil has defined his objectives in the grid.  
This field can be empty if the pupil has not defined his objectives.
16. **Start date** (Format dd/MM/yyyy)  
Date of the beginning of the pupil's training
17. **Date of student's first session** (Format dd/MM/yyyy)  
Date of the first pupil's working session.  
This date can be a TellMeMore working session (tracking), a Virtual classroom connection session or a Mobile Application connection session.  
This field can be empty if the pupil has not worked in activities.
18. **End date** (Format dd/MM/yyyy)  
Date of the end of the pupil's training
19. **Dates student account deactivated** (Format dd/MM/yyyy)  
Date at which the training has been deactivated. This field can be empty if the pupil is still in training (has not reached the training end date and the Discipline has not been manually deactivated by the Host Administrator).
20. **Suspension dates** (Format dd/MM/yyyy)  
No longer exists.
21. **Number of days training suspended** (integer)  
No longer exists.
22. **Placement Test results** (format float, 1 decimal after the point ex: "5.1")  
Placement test score. Can be empty if the pupil has not passed the placement test
23. **Progress Test results** (format float, 1 decimal after the point ex: "9.1")  
Progress test score. Can be empty if the pupil has not passed the progress test
24. **Total study time in between the two tests** (format HH:MM:SS ex "26:45:21")  
Working Time between Placement test and Progress test.  
This duration includes the time spent in Virtual classroom and Mobile Application.  
This field is empty if the duration is 0 (Placement Test and progress test have not been done).
25. **Path level** (string)  
Level selected in the Objective grid (the last selected one and not the initial one). Can be empty if the pupil has not defined his objective or if the grid is not configured to support level.
26. **Programmed Learning Path** (Name, list separated with " – ")  
Planned Catalogues assigned by the Host Administrator. This field can be empty if there is no

catalogue planned by the Host Administrator.

27. **Total working time** (format HH:MM:SS ex "26:45:21")  
Working time that corresponds to **Field A** defined in this [document](#) (no time spent in tests).  
This duration includes the time spent in Virtual Classroom and in the Mobile Application .  
This field is empty if the working time is 0.
28. **Average working time (7 days)** (format HH:MM:SS ex "26:45:21")  
This is the average Working time per week duration (based on **Field A** defined in this [document](#)) for that learning project. See <[UC502](#)> § "**Work Time**"/ "**GlobalAveragePerWeek & CurrentlyInTrainingAveragePerWeek**" rule description to learn more about the computation rule for a learning project.  
This average take into account the Mobile Application working time  
This field is empty if the average working time is 0.
29. **Date of last session** (Format dd/MM/yyyy)  
Date of the last working session.  
This date can be a TellMeMore working session, a Virtual classroom working session or a Mobile Application session .  
This field can be empty if the pupil has not worked in activities.
30. **Date last message sent** (Format dd/MM/yyyy)  
Date of the last message sent by the pupil. This field can be empty if the pupil has not sent a message to his tutor.
31. **Number of messages sent**  
Number of messages sent by the pupil. This field can be 0 if the pupil has not sent a message to his tutor.
32. **Result of the TOEIC Preparation Test/Language Achievement Test** (format: Integer for score / Integer for max score: ex: "250/400")  
This field can be empty if the pupil has not done the Achievement test.
33. **Tutoring offer**  
List of offers assigned to the pupil separated with "," character
34. **Time spent studying since objectives were set**  
**Empty:** Not implemented
35. **Result of second Language Achievement Test** (format: Integer for score / Integer for max score: ex: "250/400")
36. **Language**  
**Empty:** Not implemented
37. **24-hour customer service** (format string)  
**Yes** if the pupil has the 24/7 service in his offer, else **No**

**38. Conversation Classes** (format string)

**Yes** if the pupil has the Conversation classes service in his offer, else **No**.

**39. Paths worked on**

List of catalogues worked on by the pupil separated by a “-“ character. This field can be empty if the pupil has not worked on sequences.

**40. % completed, % completed successfully** (integer for %done, integer for %correct)

(ex: “50,75” for 50% done and 75 %correct)

Average of percentage done and percentage correct on the worked catalogues (Paths worked on)

## Command 505 – Evaluation test result export

Value of ptf\_cmd: 505

Request type allowed: GET/POST

Description: This command will return the result of the given evaluation test for the user defined by the parameter ptf\_login.

In addition to the following parameters, all the parameters described in the section “Common Parameters” must be present.

Parameters are the following:

- **ptf\_testtype** (required) = type of test to display. It's a value between 1 and 2:
  - 1 = Position Test
  - 2 = Certification Test
- **ptf\_discipline** (required) = discipline of the test (Example: ENGLISH)

If the test doesn't exist (not accessed by the pupil at all), the error code **24** (CannotGetTestResult) is returned in the result stream (see § [Common Parameters](#) for result error format).

If the test exists (pupil has accessed it), the command will return a XML string of this form:

Exemple when the test is finished:

```
<result>
  <status>Finished</status>
  <score>5</score>
  <scoreMax>10</scoreMax>
  <date>2012-08-20 23:59:59</date>
  <duration>00:45:21</duration>
</result>
```

Exemple when the test is not finished:

```
<result>
  <status>NotFinished</status>
  <score>0</score>
  <scoreMax>10</scoreMax>
  <date>2012-08-20 23:09:12</date>
  <duration>00:03:01</duration>
</result>
```

<status> can have the following values: Finished, NotFinished

<score> can be a decimal value (e.g. 5.2 or 8.3)

<scoreMax> will always be an integer value

<date> The date of the test in **UTC**: format : YYYY-MM-DD HH:MM:SS, It can be:

- The date at which the test is finished in case the status is Finished
- The date when the user has left the test in case the status is NotFinished



<duration> The duration spent in the test (sum of session durations): format HH:MM:SS ex “26:45:21” , It can be:

- The total time spent in the test (sum of all session durations) in case the status is Finished
- The current time spent in the test (sum of current session durations) in case the status is NotFinished

Technical requirement: This command needs the V9 and the V10 servers to be visible to each other (HTTP requests will be launched from the V10 server to the V9 server).

## Command 506 – Pupil XML information by Administrative Group

Value of ptf\_cmd: 506

Request type allowed: GET/POST

Description: This command will return XML information about pupils for the given Administrative group identifier (includes deactivated learning projects).

Grant: This command needs a special grant and "501" should be added in the section "commands" of the corresponding partner in the configuration file (see corresponding section).

In addition to the following parameters, all the parameters described in the section "Common Parameters" must be present, except ptf\_login. **An empty ptf\_login should be used to generate ptf\_check.**

Parameters are the following:

- **ptf\_grouppnumber** (required) = ID of the Administrative Group (Example: 1, this is the value returned in the enumeration command 501)
- **ptf\_pagenumber** (required) = number of the requested page (Example: 1 for first page, 2 for the second page, etc.). This value should not exceed the existing page count. Using a greater value will return the InvalidRequestParameter error.
- **ptf\_itemsperpage** (required) = number of pupil's information returned per page (Example: 20). This value can be greater than the existing count items, in such case, only the existing items are returned.
- **ptf\_ftp** (optional) = to decide whether the result should be transferred through HTTP or FTP. Default is "HTTP", can be set to "FTP".

A first call of the Command 501 - Enumeration of all the groups can be previously used to get list of the Administrative groups contained in the solution. However it is possible to use directly this 506 command if the Administrative group identifier is already known.

There is no limitation of number of pupils per page, nevertheless a time-out might occur if too high values are used. In case this API is used inappropriately, Auralog reserves the right to limit the number of pupils per page.

This command can also be used pupil per pupil by using an item count per page set on 1.

### Note on FTP option:

- To use this command, additional information should have been set for the client in the configuration file: ftp site, connection parameters, success callback and error callback. Those values should be provided by the client to the administrator prior to the usage of the API. If the ftp parameters or the callbacks values are not set, success and error information can be found in logs. If the ftp parameters are not set the file is not generated.
- If the value is set to FTP, only the summary header will be sent back by the http request, and the success callback will be called at the end of the process (file generation and transfer to the ftp site is completed).
- The fact that the callbacks available and operational is the client's responsibility.
- The system will call the callbacks with a parameter set in GET mode, with the name of the file in the URL such as : **success\_callback?Name="Name\_of\_the\_file"** or **error\_callback?Name="Name\_of\_the\_file"&errcode=errorcode**

To configure the FTP option, you need to fill in the following values in the ApiPartnersParameters.xml file on the TellMeMore server side:

```
<param key="FtpServer" value="" />
<param key="FtpPort" value="" />
<param key="FtpDirectory" value="" />
<param key="FtpUsername" value="" />
<param key="FtpPassword" value="" />
<param key="SuccessCallback" value="" />
<param key="ErrorCallback" value="" />
```

## Warning:

- In case a limitation is set, an InvalidRequestParameter error will be returned
- In case a deletion/insertion is done on the database in the meantime, the number of pupils would change, and the list of pupils per page could be different from one call to another using the exactly the same parameters.

The limitation is set by adding in the ApiPartnersParameters.xml file on the TellMeMore server side:  
<param key="MaxPupilsForCommand506" value="xxx" />

The returned result contains a summary header with additional information, followed by the data block. This header (starting with the `<summary>` tag) contains amongst other information the total count of pupils (**TotalPupilCount**).

The caller of this command can use this value to specify the correct **ptf\_itemsperpage** and **ptf\_pagenumber** passed in each call as described in the following example (Administrative group contains 42 pupils):

The 1st call with the first 20 items and the first page on group number 1 would imply to set the following variables:

- ptf\_groupnumber =1
- ptf\_pagenumber=1
- ptf\_itemsperpage=20

The returned result contains the first 20 pupils and their data (TotalPupilCount="42", CurrentPageNumber="1" and CurrentPageItemCount=20, pupil #1 to #20 in the order of the page).  
**Please note that the total count of items is known by checking the return result field: **TotalPupilCount="42"**.**

2nd call with the next 20 items

- ptf\_groupnumber =1
- ptf\_pagenumber=2
- ptf\_itemsperpage=20

The returned result contains the next 20 pupils (TotalPupilCount="42", CurrentPageNumber="2" and CurrentPageItemCount=20, pupil #21 to #40 in the order of the page).

Last call with the next 20 items (but only 2 are remaining)

- ptf\_groupnumber =1
- ptf\_pagenummer=3
- ptf\_itemsperpage=20

Notice that the returned result contains the 2 last pupil items (TotalPupilCount="42", CurrentPageNumber="3" and CurrentPageItemcoun=2, pupil #41 and #42 in the order of the page).

## Xml Structure:

The header is constructed as followed:

XML Tag	Description
<result>	Root node
<summary	Header of the returned result containing the summary
dateofgeneration="2011-05-20 23:59:59"	Date and hour of the generation of the page (UTC time)
groupid="1"	Current Administrative Group Id
totalpupilcount="42"	Total count of pupils contained in the current Administrative group
currentpagenumber="1"	The current requested page number
currentpageitemcount="20"	The count of items contained in the returned result
filename="Name">	<p>Name is empty if <b>ptf_ftp</b> is set to HTTP. Proposed name structure is : <a href="#">client_name_GroupId_PageId_Date.xml</a></p> <p>Date is : <a href="#">YYYYMMDD_Sec</a></p> <p>Where Sec is the time if the day in seconds at the time of the generation of the summary. If two files with the same name are dropped on the ftp sites, the previous one will be discarded. It is the responsibility of the customer to copy and paste to another location.</p>

The XML section containing the information for the requested pupils follows in the file and is constructed as described in the following table:

XML Tag	Description
<pupils>	List containing the information of the pupils
<pupil	Information relative to one pupil
pupild="5000"	The pupil identifier
hostid="3"	The Host identifier where the pupil is registered
clientid="26"	The Client identifier where the pupil is assigned
firstname="Thierry"	The pupil's first name
lastname="Longval"	The pupil's last name
username="thierryl"	The pupil's username
email="thierry.longval@auralog.com"	The pupil's email
administrativegroup="AuralogSA">	The Administrative group name where the pupil is assigned
<disciplines>	List of the Disciplines (Learning language of the pupil)
<discipline	Information relative to one Discipline of the pupil
disciplineid="AMERICAN_ENGLISH"	Identifier of the Discipline learned by the pupil
name ="American English"	Name of the Discipline learned by the pupil
trainingstartdate="2011-05-20" trainingenddate="2011-08-20"	Training start date and end date for the Discipline
pedagogicgroup="AME English Pedag.Group"	Pedagogic group where the pupil is assigned
status="ObjectiveDefined"	<p>Pupil's training status. It can be the following values:</p> <p><b>NotStarted:</b> The pupil hasn't started his study</p> <p><b>ObjectiveDefined:</b> The pupil already defined his objectives.</p> <p><b>InProgress:</b> The pupil started to train.</p> <p><b>Suspended:</b> The pupil is currently suspended</p>

XML Tag	Description
	(holidays ...). <b>Completed:</b> The pupil has finished his study.
placementtestscore="5.6" placementtestmaxscore="10"	The placement test score & score max
progresstestscore="6.7" progresstestmaxscore="10"	The progress test score & score max
achievementtest1score="547.5" achievementtest1maxscore="800.0"	The achievement test #1 score & score max
achievementtest2score="623.2" achievementtest2maxscore="800.0">	The achievement test #2 score & score max
<offers>	List of offers assigned to the pupil's Discipline
<offer name="eCoaching">	Information relative to one offer of the pupil's Discipline
<catalogues>	List of the catalogues assigned to the pupil for the Discipline
<catalogue name="Beginner-A1" percentdone="0" percentcorrect="0" workingTime="0:00:00">	Information relative to one catalogue assigned to the pupil.
<catalogues>	List of catalogues that are contained in the current catalogue (recursive)
<sequences>	List of the sequence contained in the current catalogue
<sequence name="Company-Strategy" percentdone="50" percentcorrect="55" workingTime="1:00:00" />	Information relative to one sequence

## Xml sample file when ptf\_ftp is set to HTTP:

```
<?xml version="1.0" encoding="utf-8"?>
<result>
<summary dateofgeneration="2011-05-20 23:59:59"
  groupid="1"
  totalpupilcount="42"
```

```

currentpagenumber="1"
currentpageitemcount="20"
filename=""
</summary>
<pupils>
  <pupil pupilid="5000"
    hostid="3"
    clientid="26"
    firstname="Thierry"
    lastname="Longval"
    username="thierry1"
    email="thierry.longval@auralog.com"
    administrativegroup="AuralogSA">
</pupils>
<disciplines>
  <discipline disciplineid="AMERICAN_ENGLISH" name="American English"
    trainingstartdate="2011-05-20"
    trainingenddate="2011-08-20"
    pedagogicgroup="AME English Pedag.Group"
    status="ObjectiveDefined"
    placementtestscore="5.6" placementtestmaxscore="10"
    progresstestscore="6.7" progresstestmaxscore="10"
    achievementtest1score="547.5" achievementtest1maxscore="800.0"
    achievementtest2score="623.2" achievementtest2maxscore="800.0">
</disciplines>
<offers>
  <offer name="eCoaching"></offer>
</offers>
<catalogues>
  <catalogue name="My training catalogs" percentdone="7"
    percentcorrect="60" workingtime="1:10:00">
    <catalogues>
      <catalogue name="Beginner A1" percentdone="0"
        percentcorrect="0" workingtime="0:00:00">
        <catalogues></catalogues>
        <sequences></sequences>
      </catalogue>
      <catalogue name="Advanced B2" percentdone="15"
        percentcorrect="60" workingtime="1:10:00">
        <catalogues>
          <catalogue name="Situations" percentdone="25"
            percentcorrect="60" workingtime="1:10:00">
            <catalogues>
              <catalogue name="Presentations and Meetings III" percentdone="25"
                percentcorrect="60" workingtime="1:10:00">
                <catalogues></catalogues>
                <sequences>
                  <sequence name="Company Stategy" percentdone="50"
                    percentcorrect="55" workingtime="1:00:00"></sequence>
                  <sequence name="Executive commitee Meeting" percentdone="20"
                    percentcorrect="65" workingtime="0:00:00"></sequence>
                  <sequence name="Virtual conversation: Organize a meeting" percentdone="0"
                    percentcorrect="0" workingtime="0:00:00"></sequence>
                  <sequence name="A project update Organize a meeting" percentdone="0"
                    percentcorrect="0" workingtime="0:00:00"></sequence>
                  <sequence name="The Year in review" percentdone="0"
                    percentcorrect="0" workingtime="0:00:00"></sequence>
                  <sequence name="Policies" percentdone="0"
                    percentcorrect="0" workingtime="0:00:00"></sequence>
                  <sequence name="Site visit" percentdone="0"
                    percentcorrect="0" workingtime="0:00:00"></sequence>
                  <sequence name="Virtual conversation" percentdone="0"
                    percentcorrect="0" workingtime="0:00:00"></sequence>
                </sequences>
              </catalogue>
            </catalogues>
          <catalogue name=" Interpersonal Skills and Negotiation III" percentdone="50"
            percentcorrect="55" workingtime="1:00:00">
            </catalogues>
          </catalogue>
        </catalogues>
      </catalogue>
    </catalogues>
  </catalogue>
</catalogues>

```

```

        <catalogues></catalogues>
        <sequences></sequences>
    </catalogue>
    <catalogue name="Business Partner Relations III" percentdone="0"
        percentcorrect="0" workingtime="0:00:00">
        <catalogues></catalogues>
        <sequences></sequences>
    </catalogue>
</catalogues>
<sequences></sequences>
</catalogue>
<catalogue name="Video themes" percentdone="0"
    percentcorrect="0" workingtime="0:00:00">
    <catalogues></catalogues>
    <sequences></sequences>
</catalogue>
</catalogues>
<sequences></sequences>
</catalogue>
</catalogues>
<sequences></sequences>
</catalogue>
</catalogues>
<discipline>
<discipline disciplineid="SPANISH_LATAM" name="Spanish (Latin american)"
    trainingstartdate="2011-05-18"
    trainingenddate="2011-08-18"
    pedagogicgroup="LAT Span Pedag.Group"
    status="NotStarted"
    placementtestscore="0" placementtestmaxscore="0"
    progresstestscore="0" progresstestmaxscore="0"
    achievementtest1score="0" achievementtest1maxscore="0"
    achievementtest2score="0" achievementtest2maxscore="0">
    <offers>
        <offer name="eLearning"></offer>
    </offers>
    <catalogues></catalogues>
</discipline>
</disciplines>
</pupil>
<pupil pupilid="5001"
    hostid="3"
    clientid="26"
    firstname="Tom"
    lastname="Pouce"
    username="tomp"
    email="tomp@free.fr"
    administrativegroup="AuralogSA">
<disciplines>
    <discipline disciplineid="AMERICAN_ENGLISH" name="American English"
        trainingstartdate="2011-05-20"
        trainingenddate="2011-08-20"
        pedagogicgroup="AME English Pedag.Group"
        status="NotStarted"
        placementtestscore="0" placementtestmaxscore="0"
        progresstestscore="0" progresstestmaxscore="0"
        achievementtest1score="0" achievementtest1maxscore="0"
        achievementtest2score="0" achievementtest2maxscore="0">
        <offers>
            <offer name="eCoaching"></offer>
        </offers>
        <catalogues></catalogues>
    </discipline>
</disciplines>
</pupil>

```



```
</pupils>  
</result>
```

## Xml sample file when ptf\_ftp is set to FTP:

```
<?xml version="1.0" encoding="utf-8"?>  
</result>  
<summary dateofgeneration="2011-05-20 23:59:59"  
  groupid="1"  
  totalpupilcount="42"  
  currentpagenumber="1"  
  currentpageitemcount="20"  
  filename="client_name_1 _1_20110520_12344455.xml">  
</summary>  
</result>
```

## Command 507 – Get Test result information for a day

Value of ptf\_cmd: 507

Request type allowed: GET/POST

Description: This command gets the entire pupil finished test result information for a specific day date.

- Only the finished tests (test having a score) are reported.
- The tests that are begun but not finished at this date are not reported.

**Warning:** this command might take a long time to process for a new day date the first time. Once this day date has been processed, the system keeps the corresponding day date information in order to answer faster the other times this day date is requested.

In addition to the following parameters, all the parameters described in the section “Common Parameters” must be present, except ptf\_login. **An empty ptf\_login should be used to generate ptf\_check (value provided in parameter if exist is ignored).**

Parameters are the following:

- **ptf\_date** (required) = the day date corresponding to the desired test results. Format should be as follows: “YYYY-MM-DD” (Example: 2011-11-25). The day date is considered in the time zone that is defined in the solution.

The time zone by default is “SA Pacific Standard Time” and can be changed in the ApiPartnersParameters.xml file on the TellMeMore server side:

```
<param key="TimeZoneForCommand507" value="SA Pacific Standard Time" />
```

### Xml Structure:

The result is constructed as followed:

XML Tag		Description
<result>		This is the root XML node
<pupil>		Information relative to one test for one pupil
	username= "thierryl "	The user name of the pupil
	discipline= "ENGLISH"	The discipline of the test. <b>Warning:</b> The test is realized in an external site that doesn't have all the various disciplines defined in § “Appendix 1”.

	<p>The returned discipline is:</p> <ul style="list-style-type: none"> <li>• “ENGLISH” for all pupils having accessed the test for a learning language such: “BRITISH_ENGLISH”, “AMERICAN_ENGLISH” and “ENGLISH”.</li> <li>• “SPANISH” for all pupils having accessed the test for a learning language such: “SPANISH_SPAIN”, “SPANISH_LATAM” and “SPANISH”</li> <li>• “FRENCH” for all pupils having accessed the test for a “FRENCH «learning language.</li> <li>• Other disciplines values are not supported yet.</li> </ul>
<code>score="5"</code>	The Score obtained at the test
<code>scoremax="10"</code>	The maximum reachable score of that test
<code>testtype="1"</code>	<p>The test type:</p> <p><b>1</b> for Position Test</p> <p><b>2</b> for Certification Test</p>
<code>date="2012-08-20 23:59:59"</code>	<p>The date of the test in UTC</p> <p>Format : <b>YYYY-MM-DD HH:MM:SS</b></p>
<code>duration="00:45:21"</code>	<p>The duration spent in the test (sum of session durations)</p> <p>Format <b>HH:MM:SS</b> (ex “26:45:21”)</p>

## Example of returned XML stream:

```
<result>
<pupil username="thierry1" discipline="AMERICAN_ENGLISH" score="5" scoremax="10"
testtype="1"/>
  <pupil username="pascalb" discipline="SPANISH_LATAM" score="450" scoremax="800"
testtype="2" date="2012-08-20 23:59:59" duration="00:45:21" />
</result>
```

## Appendix 1 - Disciplines

Here are the values to use in ptf\_discipline:

Discipline	Value (must be formatted like here)
British English	BRITISH_ENGLISH
French	FRENCH
German	GERMAN
American English	AMERICAN_ENGLISH
Italian	ITALIAN
Spanish	SPANISH_SPAIN
Latin American Spanish	SPANISH_LATAM
Dutch	DUTCH
English (American + British)	ENGLISH
Spanish (Spain + Mexico)	SPANISH
Arabic	ARABIC
Chinese	CHINESE
Japanese	JAPANESE
All	MULTI

## Appendix 2 - Interface languages

Here are the values to use in ptf\_interfacelanguage:

Interface Language	Value
English	en-US
French	fr-FR
German	de-DE
Italian	it-IT
Spanish	es-ES
Albanian	sq-AL
Arabic	ar-SA
Portuguese (Brazil)	pt-BR
Bulgarian	bg-BG
Hungarian	hr-HR
Czech	cs-CZ
Danish	da-DK
Dutch	nl-NL
English (UK)	en-GB
Finnish	fi-FI
Greek	el-GR
Hebrew	he-IL

Hungarian	hu-HU
Icelandic	is-IS
Japanese	ja-JP
Korean	ko-KR
Norwegian (Bokmal)	nb-NO
Polish	pl-PL
Portuguese (Portugal)	pt-PT
Romanian	ro-RO
Russian	ru-RU
Slovak	sk-SK
Swedish	sv-SE
Thai	th-TH
Turkish	tr-TR
Vietnamese	vi-VN
Chinese simplified	zh-CN
Chinese traditional	zh-TW
Basque	eu-ES
Slovenian	sl-SI
Ukrainian	uk-UA
Spanish (Mexico)	es-MX
French (Canada)	fr-CA
English (Canada)	en-CA
Catalan	ca-ES

## Appendix 3 - Code Samples

### *Direct connection to TOL's portal by using PHP*

To connect to tell me more Portal using PHP we can use the following example:

```
<?PHP
// Please remark:
//          utf8_encode(),gmdate(),MD5(),urlencode() are PHP native
functions

$Login=utf8_encode("user"); // Login name for this example
$Partner=utf8_encode("partner"); // Partner for this example
// the following PHP function calculates the number of seconds since
01/01/1970
$Timer=(string)gmdate("U");

// valid private key in this example is: p@ßwôrd,
// please remark that the key must be utf8 encoded before
// using it to calculate the MD5 value
$PrivateKey = utf8_encode("p@ßwôrd");
//result utf-8 value for this example: "p@ÃÿwÃ´rd";

// calculates the MD5 value to be validate in the TELL ME MORE portal
// using the MD5 PHP native function
$Check=MD5($Login.$Partner.$Timer.$PrivateKey);

// URL encode to redirect in JavaScript
$Login = urlencode($Login);
$Partner = urlencode($Partner);

$Redirect="http://<TOL10PORTAL>:<PORT>/ApiHandler.ashx?ptf_cmd=401&
ptf_login=$Login&ptf_partner=$Partner&
ptf_endingurl=http%3A//www.auralog.com&
ptf_timer=$Timer&ptf_method=MD5&ptf_check=$Check";

?>
<HTML>
<HEAD>
<!-- we use the Meta refresh in the client side to redirect to TOL10
portal -->
<meta http-equiv="refresh" content="2;url=<?PHP echo $Redirect; ?>">
</HEAD>
<BODY>
</BODY>
</HTML>
```

## Direct connection to TOL's portal by using ASP.NET

```
<%@ Import Namespace="System.Security.Cryptography" %>
<%@ Import Namespace="System.Text" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.SqlClient" %>
<%
Dim sLogin,sEncodedLogin, sPartner, sEncodedPartner, sTimer, sPrivateKey
as String
sLogin = "user"
sPartner = "partner"
sPrivateKey = "p@ßwôrd"

'The encoder class used to convert sToCrypt to an array of bytes
'Please be careful about this, if your strings are already utf8
'then you dont have to make the conversions
Dim encoder as New System.Text.UTF8Encoding()
Dim unicodeBytes as System.Byte()

unicodeBytes = encoder.GetBytes(sLogin)
sEncodedLogin = Server.UrlEncode(encoder.GetString(unicodeBytes))

unicodeBytes = encoder.GetBytes(sPartner)
sEncodedPartner = Server.UrlEncode(encoder.GetString(unicodeBytes))

'Get the number of seconds since 01/01/1970 in GMT
Dim localDateTime as System.DateTime
localDateTime = now()
sTimer = datediff("s","01/01/1970", localDateTime.ToUniversalTime())

Dim sToCrypt as String
sToCrypt = sLogin + sPartner + sTimer + sPrivateKey

'The array of bytes that will contain the encrypted value of sToCrypt
Dim hashedDataBytes as System.Byte()

'Create an instance of the MD5CryptoServiceProvider class
Dim md5Hasher as New MD5CryptoServiceProvider()

'Call ComputeHash, passing in the plain-text string as an array of bytes
'The return value is the encrypted value, as an array of bytes
hashedDataBytes = md5Hasher.ComputeHash(encoder.GetBytes(sToCrypt))

'Get the string hexadecimal value for the array of Bytes
Dim sCkeckString, sHex as string
Dim i as Integer
For i = 0 To hashedDataBytes.Length - 1
    sHex = Hex(hashedDataBytes(i))
    if(len(sHex)=1)
        sHex = "0" + sHex
    end if
end if
```

```
sCkeckString = sCkeckString + sHex
next i
Dim sRedirect as string
sRedirect =
"http://<server>:<port>/cmd303/ApiHandler.ashx?ptf_cmd=401&ptf_login="+
    sEncodedLogin+"&ptf_partner="+sEncodedPartner+
    "&ptf_endingurl=http%3A//www.auralog.com&ptf_timer="+
    sTimer+"&ptf_method=MD5&ptf_check="+sCkeckString
%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
    <HEAD>
        <title>Connect to TOL10</title>
        <!-- we use the Meta refresh in the client side to redirect to
TOL10 portal -->
        <meta http-equiv="refresh" content="2;url=<%=sRedirect%>">
    </HEAD>
    <body>
    </body>
</HTML>
```



## Computation of *ptf\_check* in C#

To compute the **ptf\_check** parameter we can use the following example:

### Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Security.Cryptography;

// Calculation of ptf_check for Standard Portal Integration of TOL10
// (C) Auralog 2009
namespace Sharp303
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        public static UInt32 GetSecondsSince1970()
        {
            DateTime dt1 = new DateTime(1970, 1, 1, 0, 0, 0, 0,
DateTimeKind.Utc);
            DateTime dt2 = DateTime.UtcNow;
            Int64 iDiff = dt2.Ticks - dt1.Ticks;
            return Convert.ToUInt32(Math.Abs(iDiff / 10000000.0));
        }

        public static string CP1252ToUTF8(string value)
        {
            byte[] old = Encoding.GetEncoding(1252).GetBytes(value);
            // Encoding.GetEncoding("ISO-8859-1")
            byte[] lst = Encoding.Convert(Encoding.GetEncoding(1252),
Encoding.UTF8, old); // Encoding.GetEncoding("UTF-8")
            return Encoding.GetEncoding(1252).GetString(lst);
        }

        // Hash an input string and return the hash as
        // a 32 character hexadecimal string.
        static string getMd5Hash(string input)
        {
            // Create a new instance of the MD5CryptoServiceProvider
            object.
            MD5 md5Hasher = MD5.Create();

            // Convert the input string to a byte array and compute the
            hash.
        }
    }
}
```

```

        byte[] data =
md5Hasher.ComputeHash(Encoding.Default.GetBytes(input));

        // Create a new StringBuilder to collect the bytes
        // and create a string.
        StringBuilder sBuilder = new StringBuilder();

        // Loop through each byte of the hashed data
        // and format each one as a hexadecimal string.
        for (int i=0; i<data.Length; i++)
        {
            sBuilder.Append(data[i].ToString("x2"));
        }

        // Return the hexadecimal string.
        return sBuilder.ToString();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        // Login, partner, etc.
        string sLogin      = CP1252ToUTF8(textBoxLogin.Text);
        string sPartner    = CP1252ToUTF8(textBoxPartner.Text);
        string sPrivateKey = CP1252ToUTF8(textBoxPrivatekey.Text);
        string sServerAddr = textBoxServerAddr.Text;
        string sServerCmd  = textBoxServerCmd.Text;

        // Number of seconds since 01/01/1970
        UInt32 nDiff = GetSecondsSince1970();
        string sTimer = Convert.ToString(nDiff);

        // ptf_check calculation
        string sCheck = getMd5Hash(sLogin + sPartner + sTimer +
sPrivateKey).ToUpper();

        // ptf_check display
        MessageBox.Show("ptf_check: " + sCheck);
    }
}

```

## ***Redirection using POST***

An code sample showing how to perform a redirection using POST protocol in ASP.NET is exposed here: <http://www.codeproject.com/Articles/37539/Redirect-and-POST-in-ASP-NET>

The trick is to load a page containing an invisible form, and to submit this form automatically using JavaScript method submit(). Then the user will be redirected to TMM without any action needed from him, and parameters will not be visible directly in the URL.

```
<form id="myForm" name="myForm" action="http://server/ApiHandler.ashx" method="POST">
  <input type="hidden" name="ptf_cmd" value="401" />
  <input type="hidden" name="ptf_login" value="johnb" />
  <!-- insert all parameters here -->
</form>
<script type="text/javascript">
  var vForm = document.myForm;
  vForm.submit();
</script>
```

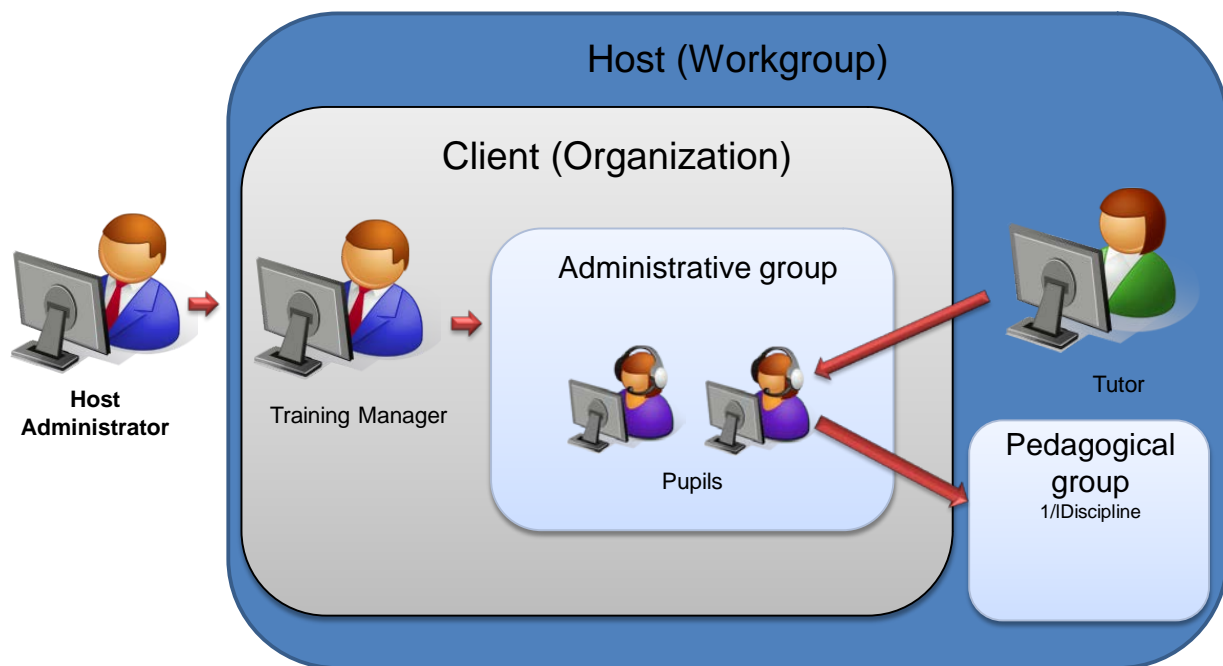
## Appendix 4 - Error Codes

InternalError	= 0,
MissingRequestParameter	= 1,
InvalidRequestParameter	= 2,
OnlyPostRequestsAreAllowed	= 3,
OnlyGetRequestsAreAllowed	= 4,
UnknownCommand	= 5,
CannotReadPartnerParameters	= 6,
CannotFindApiConfigurationFile	= 7,
CannotFindPartnerInConfigurationFile	= 8,
InvalidPartnerParameterDelta	= 9,
InvalidPartnerParameterHosts	= 10,
InvalidPartnerParameterCommands	= 11,
InvalidPartnerParameterPrefix	= 12,
InvalidPartnerParameterCode	= 13,
NotAuthorizedHost	= 14,
NotAuthorizedCommand	= 15,
UserAlreadyExists	= 16,
InvalidUserName	= 17,
ParameterTooLong	= 18,
BadTypeOfUser	= 19,
PupilAlreadyAttached	= 20,
CannotGetTestUrl	= 21,
UserHasNotTheCorrectRole	= 22,
InvalidInterfaceLanguage	= 23,
CannotGetTestResult	= 24,
InvalidPartnerParameterMaxPupilsForCommand506	= 25,
CommandNotAvailableWithMarketType	= 26,
FtpUploadFailed	= 27,
DuplicatedSession	= 28,

## Appendix 5 – V10 Concepts

The V10 solution is built to work for 2 distinct markets: Corporate market (Pro) or Education market (Edu).

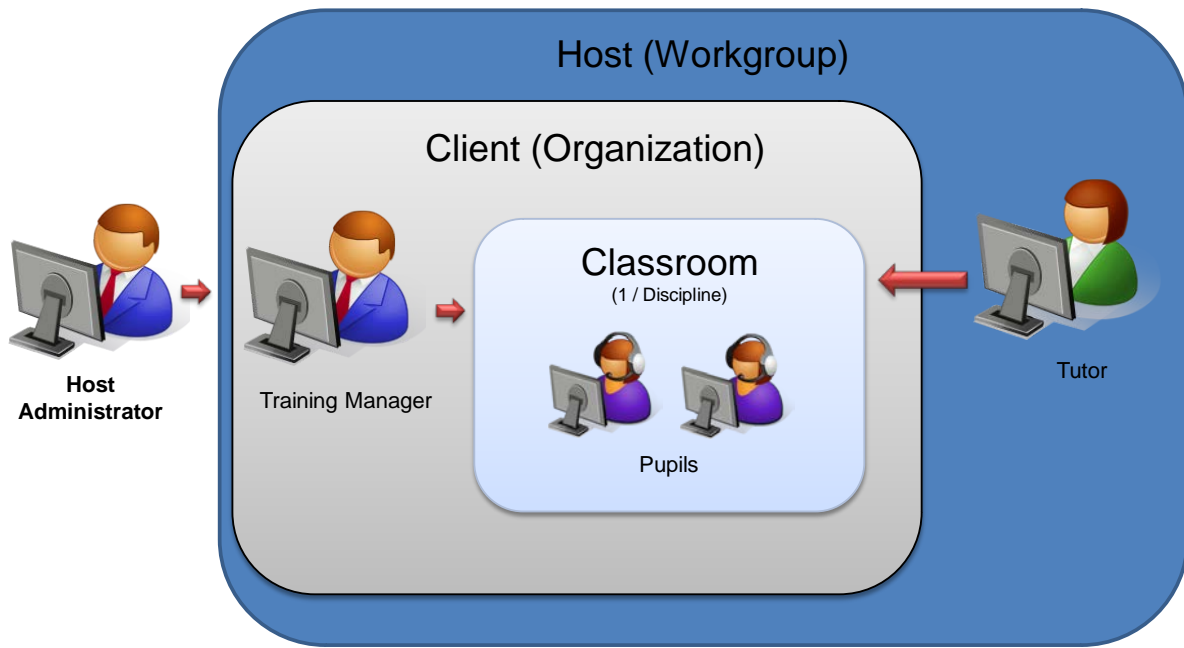
### V10 architecture - Corporate



In this architecture, the **Host Administrator** manages one or more **Hosts** (Workgroup).  
A **Host** contains one or more **Clients** (Organization).  
A **Host** contains one or more **Pedagogic groups** (a Pedagogic group is for only one discipline).  
A **Client** contains one or more **Administrative groups** (does not depend on the Discipline).  
The **Pupils** are assigned to one **Pedagogic group** for each of their learned discipline. However the **Pupils** are assigned to only one **Administrative group**.  
The **Tutors** are assigned to one or more **Hosts**. However, the **Tutors** are directly assigned to the **Pupils** (The Tutor can be different for each Discipline learned by the Pupil).

See also UML equivalence diagram below for technical details.

## V10 architecture - Education

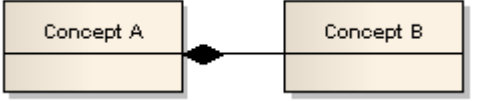
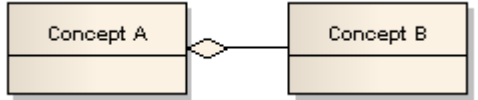


In this architecture, the **Host Administrator** manages one or more **Hosts** (Workgroup).  
A **Host** contains one or more **Clients** (Organization).  
A **Client** contains one or more **Classroom** (a Classroom is for only one discipline).  
The **Pupils** are assigned to one **Classroom** for each of their learned disciplines.  
The **Tutors** are assigned to one or more **Hosts**. However, the Tutors are directly assigned to the **Classroom** (The Tutor can be different for each Classroom).

See also UML equivalence diagram below for technical details.

## UML Diagrams equivalence

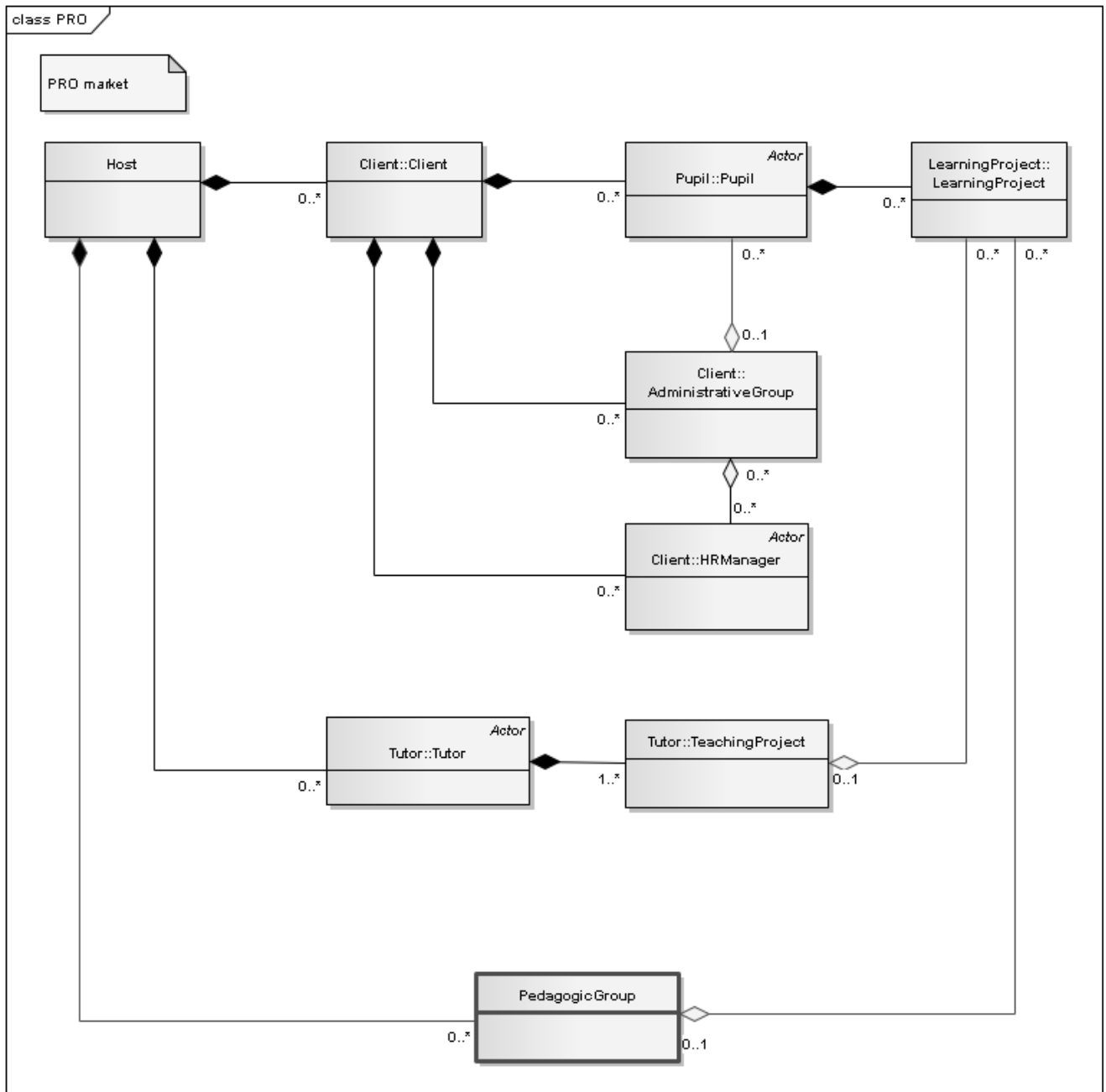
To understand the following diagrams, please consider the legend below:

	<p>The <b>Concept A</b> contains the <b>Concepts B</b> (B is created inside A). If A is deleted, B is also deleted.</p>
	<p>The <b>Concept A</b> references the <b>Concepts B</b> (B is not created inside A). If A is deleted, B still exist but it is no longer assigned to A.</p>

Example of UML usage found In the 2 UML diagrams below:

- The **Host** contains 0 or more **Clients**
- The **Host** contains 0 or more **Tutors**
- The **Tutor** contains 0 or more **Teaching projects** (the Discipline taught by the Tutor)
- The **Client** contains 0 or more **Pupils**
- The **Pupil** contains 0 or more **Learning projects** (Discipline learned by the pupil)

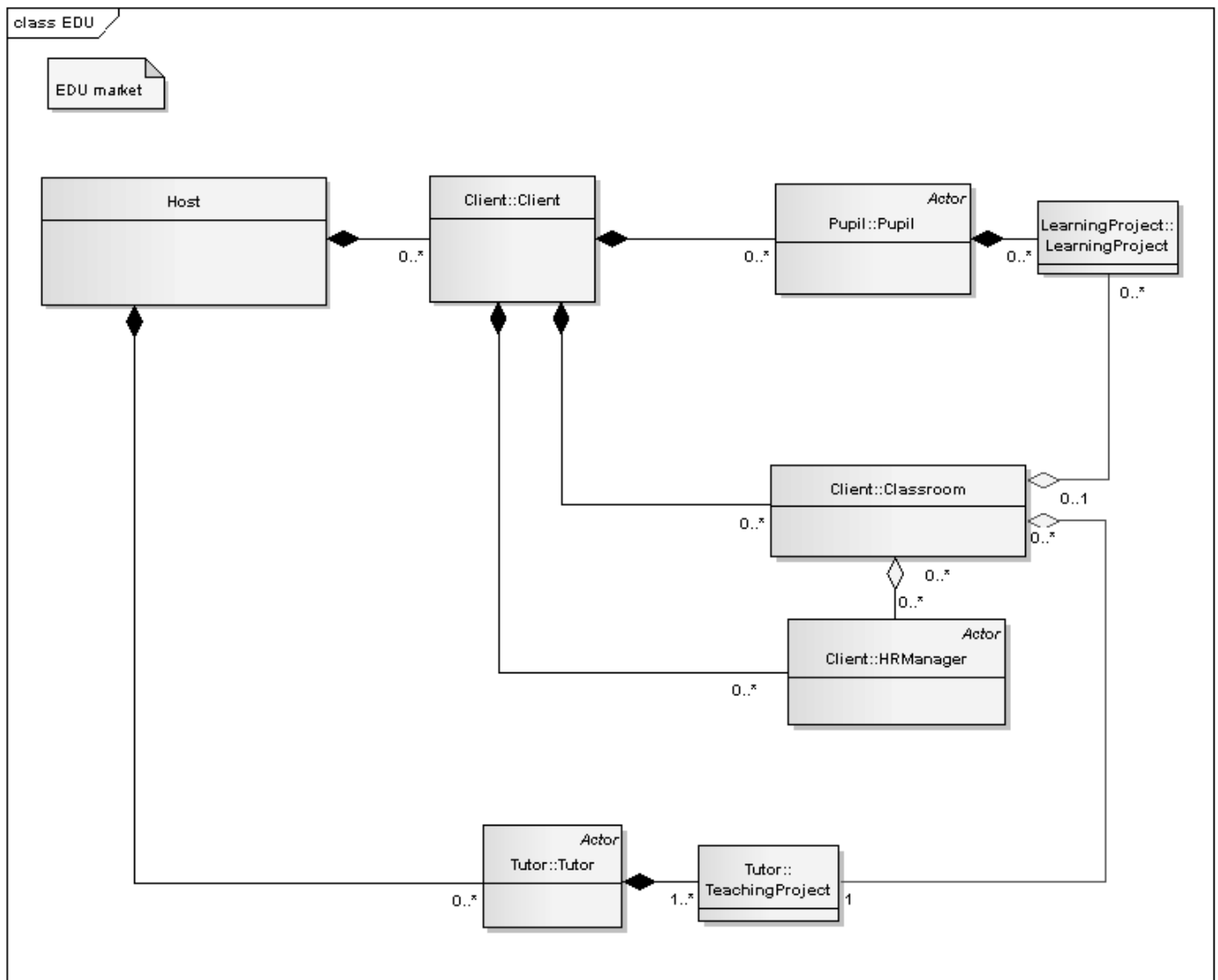
## UML Diagram for Corporate market



In **Corporate market** the **Tutors** and the **Pupils** are assigned using the **Teaching project** of the **Tutor** (Discipline taught by the Tutor) and the **Learning project** of the **Pupil** (Discipline learned by the Pupil).



## UML Diagram for Education market



In **Education market** the **Tutors** and the **Pupils** are assigned using the **Classroom**. The **Classroom** references the Tutor's **Teaching project** (Discipline taught by the Tutor) and the **Classroom** references the pupil's **Learning projects** (Discipline learned by the Pupils).