

WASE Administration

This document is intended for people who will manage the WASE system. Before reading this document, you should make sure you understand the basic architecture and function of WASE, as described in the WASE Overview document. It is particularly important that you understand the data structures that WASE manipulates (calendars, blocks, slots, appointments) as well as the various roles that users of WASE can take on (calendar owner, calendar manager, calendar member, appointment maker, guest). All of this is explained in the Overview document.

WASE includes a set of administrative utilities that can be used to query and maintain the system, as well as a parameters system that can be used to exert global control. In addition, WASE uses a super-password to permit administrators to login to the system using the credentials (e.g., userid) of any WASE user. Thus, for example, if a given user reports that there is a problem with their WASE calendar, an administrator can login as that user and inspect their calendar (and make any necessary corrections). A typical use of the super-password is to login as a professor and create a calendar for them. This ability to impersonate a user in WASE does not give the administrator any rights beyond the WASE system; it is purely internal to WASE.

The super-password is the MySQL password for the WASE MySQL database (although it can be set to something else). The username and password for your MySQL database is specified at installation time.

Parameters

The operation of WASE is controlled by a set of parameters which allow a privileged user (anyone who knows the super-password or is on the super-user list) to control the way WASE operates. Some of these parameters are set at installation time and should not be altered after that. Others can be altered on the fly (for example, you can turn the system on/off, or set an alert message). To see and/or set the available parameters, run the parms.php script, as follows:

<https://wase.princeton.edu/myuniversity/admin/parms.php>

OR

<https://wase.princeton.edu/myuniversity/admin/parms.php?secret=superlpassword>

The former URL requires that your userid be in the super-user list (itself a parameter). The latter only requires that you know the superpassword. Use the former URL if you are uncomfortable sending passwords over the web (even when SSL encrypted).

Impersonating a User in WASE

To login as any user on the system, at the login page:

1) In the box marked "e-mail address" (under "Guests"), enter the desired userid, followed by a space, followed by secret=superpassword (where you substitute the WASE MySQL password following the = sign). If you wish, you can set a special IMPERSONATE password in the WASE parameters file, and use that instead of, or in addition to, the superpassword. This lets you give impersonation privileges to staff (e.g., helpdesk) without giving away the mysql password. You can set this password by using the parms.php script (see below).

2) Click the "GUEST LOG IN" button.

For example, to login as a user with userid "smith", on a WASE system that has a MySQL superpassword of "officehr", you would type the following into the "e-mail address" box on the login page:

```
smith secret=officehr
```

then click the "GUEST LOG IN" button. You are now logged in as though you were user smith (but only for WASE; if you are using CAS or SHIB authentication, you are NOT authenticated to CAS or SHIB as user smith; you only get to act as "smith" within the WASE system).

WASE Utilities

WASE comes with a number of utility scripts, all in the "admin" directory, which you can use for a variety of purposes. You would typically invoke one of these scripts as follows:

```
https://wase.princeton.edu/myuniversity/admin/scriptname.php
```

You need to substitute the short name of your University (as decided when WASE was set up) and the script name (see below) in this URL.

Access to the admin scripts is controlled in one of two ways:

- 1) Each script has an associated set of userids which are allowed to access the script. When you invoke the script, WASE will take you through a login (unless you are already logged in) and will then check your userid against the allowed list. If your userid is in the list (or your userid is in a special "super-user" list) then the script will execute. These userids can be set by using the parms.php script.
- 2) As an alternative, you can invoke the script with the superpassword appended to the script URL, as follows:

```
https://wase.princeton.edu/myuniversity/admin/scriptname.php?secret=superpassword.
```

The scripts are as follows:

[parms.php](#):

This script lets you manipulate your institution's localization parameters, including things such as passwords. When WASE is set up, a "super-user" will have been defined. You can run the parms.php script by invoking it and logging in as the super-user. That is, you would point your browser to:

<https://wase.princeton.edu/myuniversity/admin/parms.php>

When prompted to login, use the super-user login that was assigned when WASE was set up.

[admindoc.php](#), [adminpdf.php](#):

These scripts display the WASE Administration document (this document!) in Word or PDF format.

[didyouknow.php](#):

The WASE home page includes a "Did You Know" panel which you can use to advertise features of WASE. This script lets you add/edit/remove entries from the "Did You Know" panel

[calendar.php](#):

Use this script to create, edit, or extend the daytype calendar built into WASE. This calendar tells WASE how to classify all of the days of any given academic year (so that professors can schedule recurring office hours that only occur on, for example, teaching days). You can run this script at any time (against a live system) to update the academic calendar. You can set/reset individual days, or groups of days. The daytypes for your system are set during the installation of WASE.

[mailusers.php](#):

Use this script to send email to all or to various subsets of WASE users (e.g., everyone who has a calendar in WASE, or anyone who has appointments scheduled in WASE, or other subsets of WASE users). A form prompts you for the target population, as well as the subject and text of the email.

[listactiveowners.php](#):

This script provides a tabular view of "active" WASE calendar owners/members (people who have blocks in WASE with start dates greater than or equal to the current, or specified, date). It tries to classify the users based on a default ('title') or specified directory attribute. To specify an alternative directory (e.g., ldap) attribute, invoke the script as:

https://wase.princeton.edu/myuniversity/admin/listactiveowners.php?attr=alternative_attribute

[listactivemakers.php](#):

This script provides a tabular views of "active" WASE appointment makers (people who have

appointments in WASE with start dates greater than or equal to the current, or specified, date). It tries to classify the users based on a default ('status') or specified directory attribute. To specify an alternative directory (e.g., ldap) attribute, invoke the script as:

https://wase.princeton.edu/myuniversity/admin/listactivemakers.php?attr=alternative_attribute

Note: specifying an attribute is only effective if your LDAP server is accessible to WASE.

[remind.php:](#)

This script sends out appointment reminders. This script should be run at regular intervals, and this is set up during the installation procedure. You can, however, run it whenever you like. By default, it sends out reminders for appointments scheduled on the following day. You can provide a 'date' argument, in which case it will send out reminders for appointments on the specified date. For example, to have reminders go out for appointments scheduled on December 22, 2015, you would invoke the script as:

<https://wase.princeton.edu/myuniversity/admin/remind.php?date=2015-12-22>

The remind script will send an email summarizing its actions to the sysadmin email specified in the parameters file.

[reown.php:](#)

This script allows you to change the ownership of a calendar, and, optionally, of the blocks associated with that calendar. It puts up a form where you can fill in the calendar id and current owner, then verifies the ownership, then lets you reset the ownership.

<https://wase.princeton.edu/myuniversity/admin/reown.php>

As with all admin scripts, you either specify a secret= parameter on the URL, or you login to WASE as the super user.

[Local Calendar Integration](#)

Be sure you have read the section on local calendar integration in the WASE Overview document (WASE.docx or WASE.pdf) before you read this section.

[Exchange:](#)

To enable Exchange integration, you must select or create an Exchange user account to which WASE will login to update other user's Exchange calendars. When a WASE user selects Exchange as their local

calendar sync preference, they are instructed to give this special user account rights to update their calendar.

You do this by filling in the following parameters, using the parms.php utility (see above):

```
# Exchange server host name
EXCHANGE_HOST: "host.institution.edu"
# The email address for login to Exchange (an Exchange email address)
EXCHANGE_EMAIL: "user@institution.edu"
# The username for login to Exchange
EXCHANGE_USER: "Wuser"
# The password for login to Exchange
EXCHANGE_PASSWORD: "password"
# The type of Exchange integration: direct or not (by invitation)
EXCHANGE_DIRECT: 1
```

You would substitute the appropriate institution values for all of the values listed above. WASE can also support a second Exchange server (for example, if you are migrating from a local exchange server to O365) – contact serge@princeton.edu for details.

Google:

Google integration requires no administrator action. When a WASE user selects Google as their local calendar sync preference, they are taken through the standard OAuth2 Google authorization sequence so that they can give WASE permission to update their Google calendar.

iCal:

iCal integration also requires no administrator action. When a WASE user selects iCal as their local calendar sync preference, WASE includes an iCal attachment in the notification and reminder emails it sends to the user. Users can also generate iCal streams by clicking on the sync icons displayed in WASE.

LTI Integration

WASE can act as an LTI 1.1 “tool producer”, allowing it to be launched from an LTI 1.0 or 1.1 compatible LMS without the user having to re-authenticate.

The LTI integration has only been tested with Blackboard 9.1. The basic functionality should work with any LTI 1.0+ capable LMS. This basic functionality allows a user to access WASE directly from the LMS. In addition, WASE can immediately display the WASE calendars of instructors in a course (without the user having to look them up in WASE) if the LMS provides roster information (LTI 1.1). For Blackboard,

this extended roster functionality currently requires the installation of Stephen Vickers' LTI building block. Instructions for doing this are documented at:

<http://www.spvsoftwareproducts.com/bb/basiclti/>

[Note: this is optional for Blackboard; the access to WASE works without the building block].

To configure your LMS to access WASE:

- 1) IN WASE: Set the LTIKEY and LTISECRET parameters using the parms.php script (see above) to anything you like. The LTILAUNCH parameter should be modified by substituting your institution short name for 'institution-short-name' in the URL.
- 2) IN THE LMS:

If you are using the Vickers' Basic LTI Tools Building Block

1. Click on Basic LTI Tools in the System Admin panel.
2. Click **Register New Tool**.
3. Give tool a Name and Description.
4. Enter in Launch URL, Consumer Key, and Shared Secret. The Consumer Key and Shared Secret must match the values specified for LTIKEY and LTISECRET in WASE. The launch URL is: <https://WASE.princeton.edu/yourinstitution/views/pages/ltilaunch.php>
5. Leave all "Message Types" unchecked.
6. Set *Memberships* to "Required by tool", but leave *Limit Membership List* unchecked. All other settings can remain at the default.
7. NOTE: the following should be checked under Basic LTI Tools:
 - a. Status, Course Tool, Context ID, Context Title, Use ID, User Name, Email, Roles.
 - b. Services should be set to "xMS:."
8. Under *Personal Data*, make sure that you set the "Value to use for user ID" to the LMS field that corresponds to the WASE login userid.
9. Click Submit to return to the list of LTI tools.
10. Select the WASE tool, choose the *Course Tools* menu and select "Add as Course Tool", then choose *Status* menu and select "Enable"

If you are using Blackboard's Built-In LTI (Release 9.1+)

1. In the System Admin panel, Go to Building Blocks → LTI Tool Providers.
2. Click **Register Provider Domain**.
3. Enter Provider Domain (e.g. WASE.princeton.edu)
4. Under *Default Configuration*, select "Set globally" and enter the *Tool Provider Key* and *Tool Provider Secret*. The *Tool Provider Key* and *Tool Provider Secret* must match the values specified for LTIKEY and LTISECRET in WASE.
5. Under *Personal Data*, make sure that you set the "Value to use for user ID" to the LMS field that corresponds to the WASE login userid.
6. Under *Institution Policies*, set *Send User Data* to a choice other than "Never" depending on whether you are using SSL or not. Also, for *User Fields to Send*, check all three boxes.

7. Click Submit to return to the list of LTI tools.
8. Click on the down arrow next to the WASE tool and select “Approve”, then select “Manage Placements”.
9. Click Create Placement.
10. Enter a name and description for the tool, give it a *Handle*, such as “WASE Tool”, and set the *Type* to Student Tool.
1. Enter in Tool Provider URL. The URL is:
<https://WASE.princeton.edu/yourinstitution/views/pages/ltilaunch.php>
11. All other settings remain at the default.
12. Click Submit.

WASE should now appear as a tool in the Course Tools section of each Blackboard course. To make WASE available only in specific courses, go to System Admin panel, select Tools, find the WASE tool, and set it to off by default. Then go to a specific course, select Customizations, select Tool Availability, and make the WASE tool available.

When clicked in a Blackboard course, the WASE Tool link the user will be taken into WASE already authenticated. If the user is enrolled in courses whose instructors have a WASE calendar, and you are using the Vickers building block, then they will be shown a list of those calendars to facilitate their making an appointment with any of these instructor(s).

The LTI integration should work natively with Canvas and Sakai (I would appreciate hearing from anyone who tries this).