

Version 1.0

# ReadyGo Administrator

*Course upload (and management) tool*



## Important Notice

Copyright © 2000-2010 ReadyGo Incorporated. All rights reserved.

ReadyGo Web Course Builder is a trademark of ReadyGo, Inc. Netscape is a registered trademark of Netscape Communications Corporation. Microsoft Internet Explorer is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries. Other product and company names mentioned in this document may be the trademarks of their respective owners.

The information in this manual is furnished for informational use only and is subject to change without notice.

Use of ReadyGo products is governed by the license agreement located in the file *license.txt* on the WCB CD. Highlights include:

- You are licensing the WCB software for use on your computers. Each license you purchase entitles you or your company to have one copy of the software running at a time. The basic idea is that you can install the software on more than one machine. However, if you have multiple authors developing courses at the same time, you should have one license for each course developer.
- Courses that you create are **your** intellectual property. You may serve them to as many students as you want. You may make as many copies of the produced courses (the HTML generated by the WCB) as you want. You may create as many courses as you want with WCB.
- You have a license to use the clip art provided by ReadyGo in your courses. However, you do **not** have permission to use these graphics for other purposes, unless you obtain such permission from ReadyGo, Inc.
- If you would like to receive updates to the software after your support period has expired, ReadyGo has several subscription programs available.

Published August 2010

## Contents

Contents .....	iii
<b>Summary .....</b>	<b>1</b>
<b>Process for the Course Developers .....</b>	<b>1</b>
<b>Tracking Configuration.....</b>	<b>1</b>
<b>Course Generation.....</b>	<b>3</b>
<b>Uploading the Course.....</b>	<b>4</b>
<b>Managing the Table of Contents/Course List.....</b>	<b>8</b>
<b>Administration of Upload Privileges .....</b>	<b>9</b>
<b>Installing and Configuring ReadyGo Administrator</b>	<b>10</b>
<b>Configuration of the rgdefines.php File.....</b>	<b>11</b>
<b>Configuration of the readygo.sst file .....</b>	<b>13</b>

## Summary

This document provides an outline of usage of the ReadyGo Administrator module. It also provides a step-by-step procedure to install the components on the customer's server.

The ReadyGo Administrator module is an add-on to ReadyGo Server Side Testing (SST). Its purpose is to provide an easier path for course creators to upload their produced courses to the hosting server. The module allows for the creation of several "uploader" accounts, each of which may upload to a separate directory on the server. An "uploader", from a technical point of view, is a user with a login name/password/upload-destination defined for them. The "uploader" logs in based on their login name/password, selects a zipped up ReadyGo course from their local hard drive, and clicks the "Upload" button. The file is received on the server. When the installation routine runs (periodically), the uploaded course will be expanded/installed at the uploader's destination directory, and optionally, a link will be added to a course index. If the uploaded course depends on SST, the SST configuration files will be placed in the appropriate location (under scripts/readygo or cgi-bin/readygo) automatically. The upload tool, however, can also be used independently of SST for courses that do not have tracking.

## Process for the Course Developers

For courses that track data using SST, the course developers need to follow the steps below in order to configure their courses to enable data storage. Once this is done, the generated course in concert with the software on the server will take care of the data storage.

The basic steps for the process are as follows:

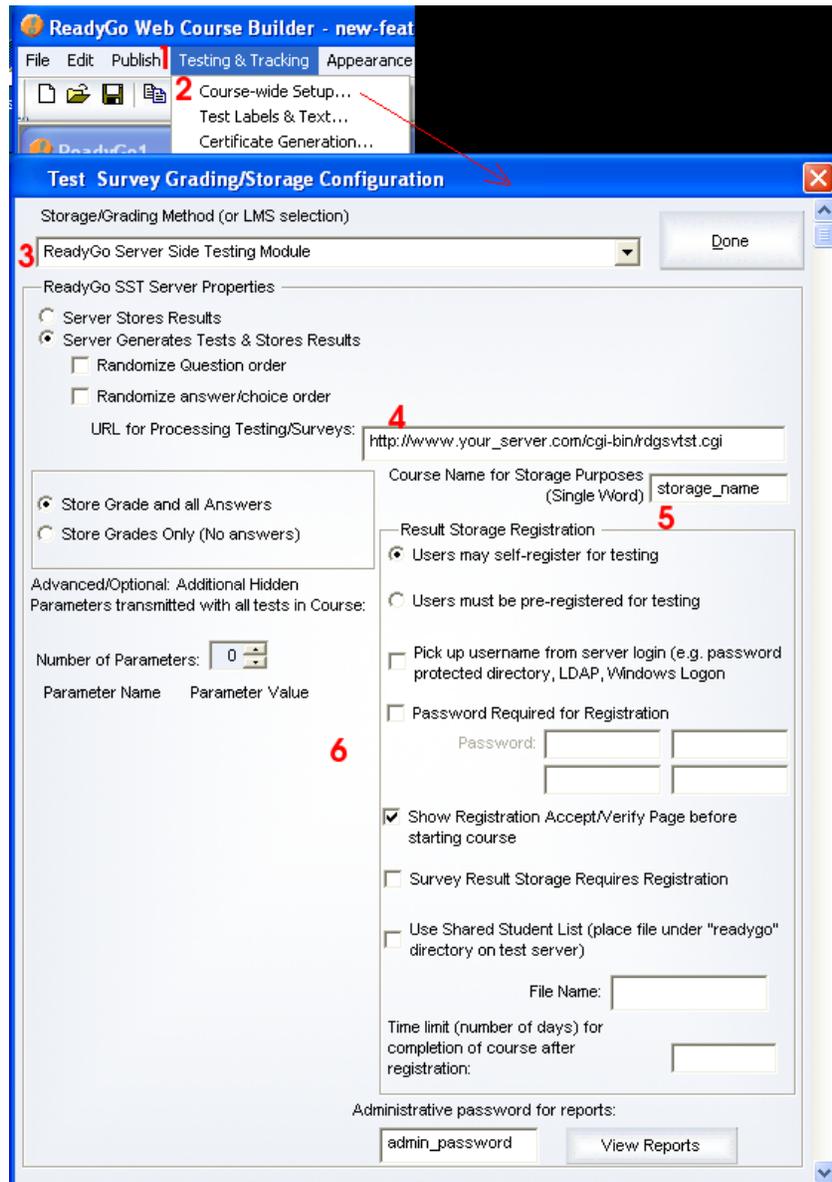
1. Input the course content and work with it until the course developer is satisfied with its appearance, flow, and experience. This may involve several iterations where the course is generated, reviewed, and updated.
2. When the course is ready for publication, the course developer changes the tracking state as described below. The course is then generated again with this tracking setup, and a compressed archive (single file) containing the course is created.
3. At the start of the upload step, the course developer can optionally instruct the upload component to add a link to the course in the start/index file for their courses.
4. The course developer uploads the content to the publication site, where it is moved to the proper directory by the upload tool. The course developer receives an e-mail when the course publication has been completed.

Employees are now able to take the course. If changes need to be made to the course, the course developer can simply update the course, republish it, and re-upload it.

## Tracking Configuration

Once the course content has been assembled, the course developer follows the configuration as shown in Drawing 1, and listed using the red numbers.

1. Open the menu *Testing & Tracking*
2. *Select Course-wide setup...*



3. Change the setting in the “Storage/Grading Method” pull-down needs to one of the options that includes “ReadyGo SST”. This enables transmission of test results to the ReadyGo SST module.
4. The URL for tracking (where the SST results are to be sent) has to be set to:
 

---
5. A unique course name for storage purposes needs to be input. Since this name is used for creation of a folder on the server, the name should be a single word, with all letters in lower case. For example “course101”, “intro\_to\_hr\_2010”, and “preventing-abuse” are acceptable course names. Names such as “My First Course”, “Try 1”, “French/Test 1” are not acceptable.

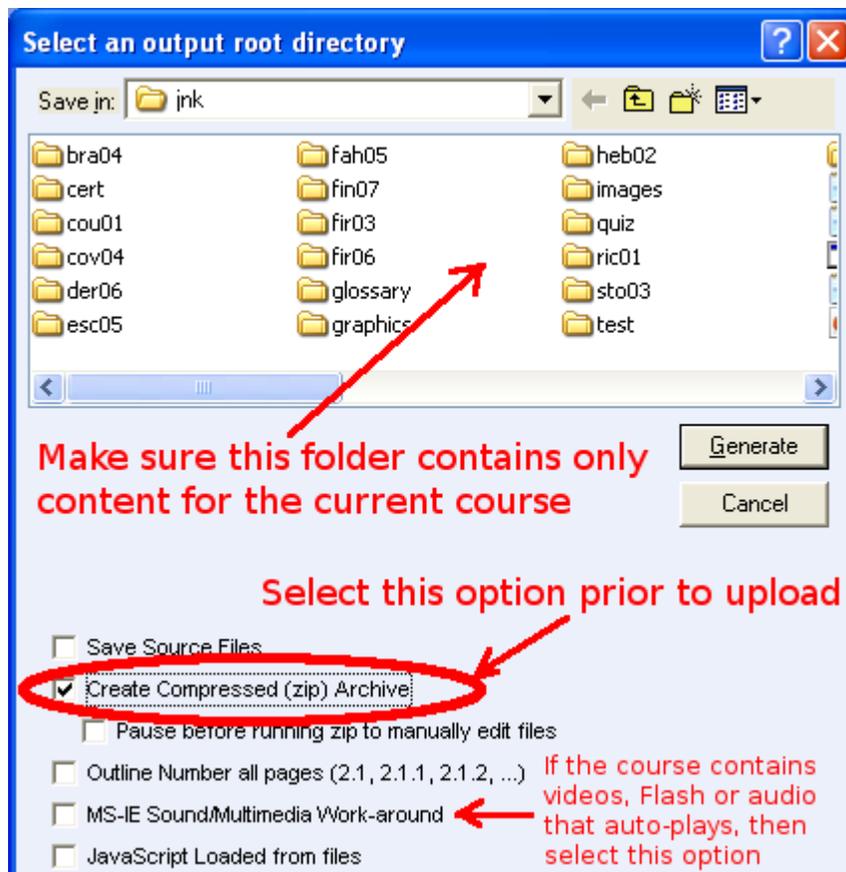
6. Other settings for tracking such as requirements for registration passwords,

## Course Generation

Once the configuration has been set, the course needs to be generated by going to the menu *Publish / Generate...*, or by clicking on the Generate button.



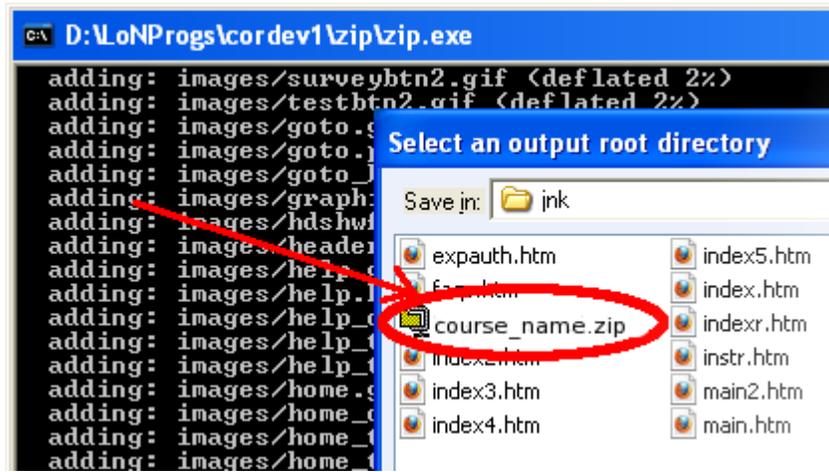
When the *Generate* button is clicked, a dialog appears giving the course developer the option as to where to generate the course. If this is the first time the course is generated, it should be generated into a new folder. In order to ease the upload process, the developer should select the “Create Compressed (zip) Archive” option. This option assembles the entire folder where the course was generated into a single .zip file that can be transferred. The course developer should be careful when selecting the folder where the course is generated to make sure that it does not contain material unrelated to the course.



*Drawing 1: Course Generation Dialog*

If the course developer has added video, Flash, audio, or other plug-in based content, and the employees are using Microsoft Internet Explorer as their browser, it is necessary to select the “MS-IE Sound/Multimedia Work-around” option. Without this option, MS-IE does not properly load and auto-start the additional content.

Once the settings have been selected, the developer should click the “Generate” button. At the end of the course generation process, a window with a black background should pop up momentarily as the course contents are zipped up. At the end of the archive creation process, there will be a new file with a .zip extension in the folder.



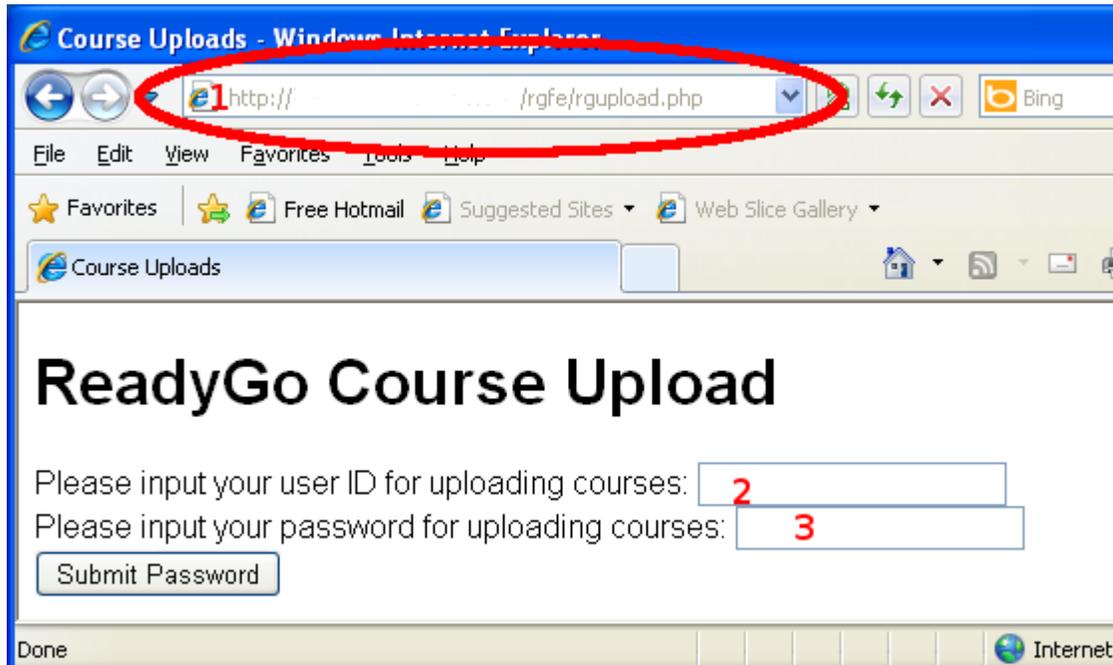
*Drawing 2: Course Archive Creation*

Later in the process, the course developer will need to find this zip file so that it can be uploaded. The directory path to the file can be found by opening the pull-down for the “Save in:” line on the course generation dialog (titled “Select an output root directory”).

## Uploading the Course

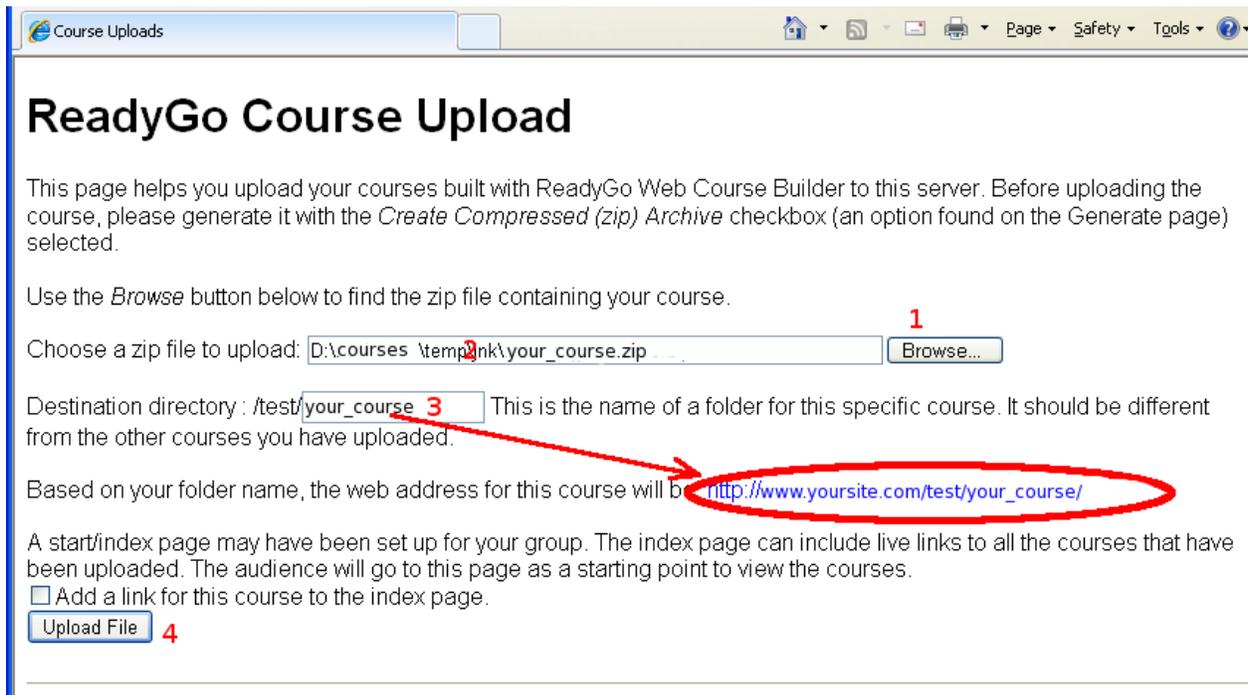
Once the course has been generated, it is time to post it to the web site (\_\_\_\_\_ ) so that it can be served to the employees and their results can be tracked. To post the course to the server, open your browser and put the following in the address/URL bar:

[http://\\_\\_\\_\\_\\_/rgfe/rgupload.php](http://_____/rgfe/rgupload.php)



*Drawing 3: Course Upload Start*

As shown in Drawing 4, once the address has been input (1), the course developer needs to input (2) the user ID provided to them for uploads and (3) the associated password. Upon clicking the “Submit Password” button, the developer is taken to the screen shown in Drawing 5.



### Drawing 4: Filling in the Course Upload Form

First (step 1), the developer clicks the “Browse” button, and finds the zip archive file previously created in the course generation directory. The name of the file is placed in the entry area (2). Next, the “Destination Directory” for the course (3) needs to be filled in. This folder name will control the directory location where the course is published. Its value also controls the Web Address for the course that is shown circled (and in blue text) in Drawing 5. When ready, the “Upload File” button (4) is pressed. Note that there is a limit to the size of the course that can be uploaded. The site administrator may need to adjust this limit (*upload\_max\_filesize* and *post\_max\_size* in the *php.ini* file) if courses will contain graphics/videos such that the zip file is larger than the current limit. If the file is larger than the current limit, when the user tries to upload, the upload process will terminate very quickly.

If the course author wishes to add an entry to the table of contents/index for their upload directory, they can click on the “Add a link for this course to the index page” check-box. This opens the dialog:

Add a link for this course to the home page.

Title:

Summary:

Link will be added to:

For courses with SST tracking, send user to the registration page.

This course uses ReadyGo SST for user registration and all tracking. (Do not use this option if your course uses another tracking system, even if it is in addition to SST.)

If the course has already been uploaded, the new text in the entry area will replace the existing link material.

When the “Upload File” button is pressed, a pop-up should appear as shown in Drawing 6. This pop-up should be available for 15 seconds or more. If it disappears within 1 second, it means that the zip file is too large, and the file transfer has not occurred.



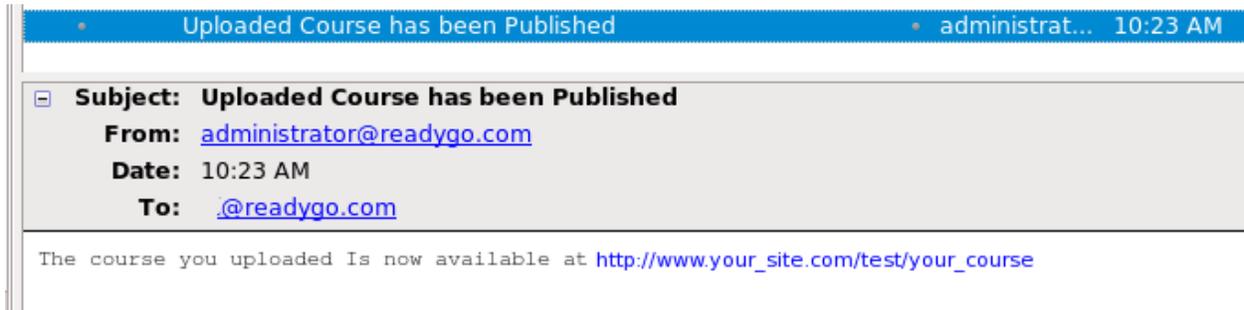
*Drawing 5: Active Course Transfer*

Once the file transfer is complete, the developer is presented with the upload information as shown in Drawing 6



*Drawing 6: Upload Completion*

At this point, the course has been queued up for publication. Periodically (with a recommended interval of 5 minutes), a routine checks the queue for uploaded courses and if it finds any in the queue, it transfers the course to the proper location. Once the publication of the course has been accomplished, the developer will receive an e-mail (at the address listed in Drawing 6 with the details that the course is ready. A sample of this e-mail is shown in Drawing 8.



*Drawing 7: Course Upload e-mail Confirmation*

If the e-mail is not received within 15 minutes of the course upload, please contact your site administrator. The course can be viewed by clicking on the link provided in the e-mail; however, it is normal to receive error messages because this URL does not properly launch the course for tracking purposes.

### Managing the Table of Contents/Course List

The course author wishes to modify the order of the courses listed in the index/table of contents or to remove files from that listing. At the bottom of the “rgupload.php” page, there is a table that allows this. The table is presented in Drawing 8.

#### Manage Index file

Course URL	Title	Summary	Order in Index	
<a href="http://localhost/uploads/blue/yellow">http://localhost/uploads/blue/yellow</a>	Green blue Yellow	Temp course blue	1 ▾	Remove
<a href="http://localhost/uploads/blue/green">http://localhost/uploads/blue/green</a>	Green title	Temp course blue	2 ▾	Remove
<a href="http://localhost/uploads/blue/red">http://localhost/uploads/blue/red</a>	red	Temp course blue	3 ▾	Remove
<a href="http://localhost/uploads/blue/cPolicy">http://localhost/uploads/blue/cPolicy</a>	CPolicy	A summary for cPolicy	4 ▾	Remove
<a href="http://localhost/uploads/blue/yell">http://localhost/uploads/blue/yell</a>	yell yell yell	yell	5 ▾	Remove
<a href="http://localhost/uploads/blue/blue">http://localhost/uploads/blue/blue</a>	Green blue title	Temp course blue	6 ▾	Remove
			Modify order	

*Drawing 8: Manage Listing for Courses*

The “Remove” button at the end of each row will remove that course from the listing. To modify the order in which the courses appear, the user can change the numbering under the “Order in Index” column. Once a new ordering has been established, the user can click the “Modify order” button. Changes to the index/table of contents will not take place until the processing routine which runs periodically executes. The course author will receive an e-mail when the listing table has been updated.

## Administration of Upload Privileges

To set up the list of users who are given access to upload courses to the server, the site administrator can visit the page [http://\\_\\_\\_\\_\\_/rgfe/rgupadmin.php](http://_____/rgfe/rgupadmin.php). Access to the upload list is limited by the overall password for SST for the site. An explanation on setting and modifying this password is provided in **Error! Reference source not found.**

Once the overall password has been provided, a table is presented with the existing uploaders as showing in Drawing 9.

Username	Password	Upload Directory	E-mail	Index File	SST Result Dir		
blue	●●●●●●●●	/uploads/blue	al@readygo.com	/uploads/blue/index.htm	/var/www/cgi-bin/readygo/	Update	Delete
purple	●●●●●●●●	/courses/purple	purple@purple.com	/courses/purple/index.htm	/var/www/cgi-bin/readygo/	Update	Delete
yellow	●●●●●●●●	/uploads/yellow	yello@readygo.com	/uploads/yellow/index.htm	/var/www/cgi-bin2/readygo/	Update	Delete
test	●●●●●●●●	courses/test	test@test.com	/courses/index.htm	/var/www/cgi-bin/readygo/	Update	Delete
red	●●●●●●●●	/courses/red	red@red.com	/courses/index.htm	/var/www/cgi-bin/readygo/	Update	Delete
anita	●●●●●●●●	/uploads/anita	anita@readygo.com	/uploads/anita/index.htm	/var/www/cgi-bin/readygo/	Update	Delete
1	2	3	4	5	6	Add	7

*Drawing 9: Controlling Course Uploaders*

To add a new user, the course administrator can fill in the details in the last row of the table:

1. Username: This will be the login name used by the uploader. This can be a simple name or can be an e-mail address. The uploader needs to know this name in order to log into the upload screen.
2. Password: A password must be established for the user. The password is stored in an encrypted form. A password for an existing user can be modified from the upload page even though it does not display once the user has been registered. Since the uploader does not have a way to modify this password, it must be set in agreement with the uploads administrator.
3. Upload directory: Each uploader can post courses to a separate folder. It is possible to allow multiple uploaders to transfer courses to the same directory. If a single uploader is to be able to transfer courses to multiple different directories, the recommended approach is to set up multiple usernames for that person. The path to the upload folder for the uploader is relative to the root directory of the web server. So, if the physical folder for the root is /var/www/html, the user "blue" will be able to post courses to /var/www/html/uploads/blue. The URL to these course will then be http://.../uploads/blue. Note that if "BASEDIR" is left blank or does not end with a "/" in the "rgdefines.php" folder, the upload directory name must start with a "/".
4. E-mail: Correspondence regarding a course being ready will be sent to the e-mail address input for the user. It is important that this e-mail address be correct.
5. Index File: If the uploader will be allowed to have a listing of all the courses that have been uploaded to the upload directory, the path to the index file should be provided. Note that if multiple users are uploading to the same folder, it is possible to have separate index files or they may share the index files.
6. SST Result Dir: In most cases, this value can be left blank. The system will automatically fill in the value based on the default SST Directory for all users. However, if the site has multiple SST directories (e.g. if there are multiple sub-domains), the SST result directory for the different sub-domains can be assigned based on the uploader name. If the courses being uploaded by the

user use SST for tracking, the test configuration files for the course will be placed into the “SST Result Dir” under the storage name for the course.

7. Add: Once the information for the user has been input, the “Add” button can be clicked to commit the changes.
8. Update: Changes can be made to the information for each uploader, and then the “Upload” button can be clicked. To apply changes to a specific user, click on the “Update” button for that uploader's row.
9. Delete: Individual uploaders can be removed from the list by clicking the respective “Delete” button.

If an index file for a specific user does not exist before they upload their first course, the upload file will be automatically created. Once it is created, it can be customized by editing it in-place. The existing index file is used as a template. The template parser looks for the commented tag:

```
<!-- RG_COURSELIST -->
```

as an indicator of where the course list starts. The course list ends at the commented tag:

```
<!-- RG_COURSELIST_END -->
```

Each course title will be surrounded by `<div class="coursetitle">` and `</div>`. Similarly, the course summary is surrounded by `<div class="coursesumm">` and `</div>`. So, the formatting of the course title and listing can be modified using the style sheet entries:

```
.coursetitle {}  
.coursesumm {}
```

Outside the commented tags, any content/decoration can be added.

## Installing and Configuring ReadyGo Administrator

ReadyGo Administrator requires PHP 5.0 or later. Installation of ReadyGo Administrator consists of the following steps:

1. Unzip the file containing the PHP files into a web-view-able directory on the server. For this installation process, the folder *rgfe* will be used. The path, then is `http://_____/rgfe`.
2. Check that the sub-folder *private* has been created under the installation directory (e.g. *rgfe/private*). Configuration and temporary data is stored here.
3. The *private* folder will need to allow the remote/web user to have full read/write privileges; however, it is very important that this folder not be web-view-able. Set the privileges to the folder as “777”. Open a terminal, change directories to the folder with the PHP files (*rgfe*), and then set the privileges. If your “web” account is “apache”, privileges could also be set up so that the “apache” user has full access to the folder.

```
chmod 777 private
```

4. If you are using Apache as the web server, check that the file *private/.htaccess* exists. This is configured to prevent browsing into this folder. Its contents should be:

deny from all

5. Set up the file *readygo.sst* which is placed in the primary SST directory. The main field needed for ReadyGo Administrator is the administrative password (*admin\_pw*). Note that the rest of the configurations in this file can affect storage of SST courses. The default *readygo.sst* file included in the zip file should be adequate once the administrative password is changed. Instructions for SST features configurable through the *readygo.sst* file are available in the *readme.doc* for SST. The configuration instructions are repeated below for convenience.
6. Copy the *rgdefines.php.tmp* provided in the zip file to *rgdefines.php*. Configure the file *rgdefines.php* so that ReadyGo Administrator knows where to find the *readygo.sst* file, where to place the SST Test Configuration files, what addresses to use for e-mails, and several other functions. A detailed explanation of configuration of this file is presented below.
7. Set up a periodic (cron) execution of the *rgupproc.php* file. Typically, the “web” or “apache” user does not have write privileges to the folders where the courses are hosted (under *www/html* for example) and to the folders where the SST data is stored (under *cgi-bin/readygo* for example). In order to get past this problem without degrading server security, the easiest approach is to have a separate process (*rgupproc.php*) that runs as administrator or as a more privileged local user, and let this process unzip and move the uploaded files into the web-view-able area. The following “cron” table will make the function run every 5 minutes, and will log the output to the file *cronlog.txt*. Please contact [support@readygo.com](mailto:support@readygo.com) if you need assistance setting up the *cron* process:

```
*/5 * * * * cd /var/www/html/rgfe:/var/www/html/rgfe/rgupproc.php >> private/cronlog.txt
```

8. In the *php.ini* file for the installation, increase the maximum size of file allowed for uploads. The variables *upload\_max\_filesize* and *post\_max\_size* should be set to the largest file that is to be allowed. ReadyGo courses are typically less than 1MB of text and standard graphics. However, if developers add videos or other multimedia, the entire course can end up being several hundred megabytes.
9. Test the installation:
  - 9.1. Set up one or more users by using a browser to go to [http://\\_\\_\\_\\_\\_/rgfe/rgupadmin.php](http://_____/rgfe/rgupadmin.php)
  - 9.2. Create an SST-based course using Web Course Builder. When the course is generated, select the “Create compressed (zip) archive” option.
  - 9.3. Test the upload: Open a browser, and go to [http://\\_\\_\\_\\_\\_/rgfe/rgupload.php](http://_____/rgfe/rgupload.php) Log in as one of the users authorized to upload zip files. Select the zip file created by Web Course Builder, and upload it.
  - 9.4. Wait a few minutes to see if an e-mail is received from the server informing that the course is now available for testing.
  - 9.5. Once the course is available for testing, log into the course directly (or through the uploader's table of contents/index page). Verify that results are stored correctly.

### Configuration of the *rgdefines.php* File

The *rgdefines.php* file that is placed in the folder where the ReadyGo Administrator components are extracted is used to ensure restricted access to uploading, to control administration access, and to provide several other settings needed for the proper operation of the ReadyGo Administrator module. A sample file is provided with an explanation of its variables below.

```

<?php
define("BASEDIR", "");
define("BINDIR", "../..../cgi-bin/");
//define("SSTDIR", BINDIR."readygo/");
define("SSTDIR", "/var/www/cgi-bin/readygo/");
define("SSTURL", "/cgi-bin/rdgsvtst"); /* web path */
define("iUtf", "7");
define("ALLOW_MULTI_SST", "0"); // If there is more than one cgi-bin for SST
define("iDebug", "0");
define("user", "root");
$admin_email = "administrator@readygo.com";
$from_email = "administrator@readygo.com";
$contact_email = "support@readygo.com";
// ----- No need to configure below this line -----
if (iDebug == "1") {
    error_reporting(E_ALL);
    ini_set("display_errors", 1);
} else {
    ini_set("display_errors", 0);
}
if (!(function_exists('fputcsv'))) {
    function fputcsv($fh, $arr, $delim=',', $enc='') {
        if ($fh) {
            $i = 0;
            foreach ($arr as $value) {
                $outstr = sprintf($enc."%s".$enc, addslashes($value));
                fwrite($fh, $outstr);
                if ($i < count($arr)) {
                    fwrite($fh, $delim);
                }
                $i++;
            }
            fwrite($fh, "\n");
        }
    }
}
?>

```

Variables in this file are defined as follows:

**BASEDIR:** If all the courses are to be uploaded to a single folder, or all uploaders' directories reside under one master folder, this location could be defined in BASEDIR. The default value is empty. The value of BASEDIR will prepend the PHP variable `$_SERVER['DOCUMENT_ROOT']` when there is an effort to unzip a course. Recommended setting: `BASEDIR = ""`;

**BINDIR:** This is the relative path from the folder containing the `rgdefines.php` file to the main cgi-bin (or `scripts`) folder where the SST modules are installed.

SSTDIR: The SSTDIR variable contains the absolute, physical path to the primary *readygo* folder containing the *readygo.sst* file. This is also the parent folder to the default for all the course-specific SST data storage folders.

iDebug: If detailed debug information is desired, set this value to "1". The recommended value is "0".

user: This variable is not currently used.

SSTURL: SSTURL contains the web-path to the primary instance of the SST registration/testing/tracking executable (*rdgsvtst*, *rdgsvtst.exe*, *rdgsvtst.cgi*). It is used when a course is purchased, and it is necessary to provide a direct URL to the registration module for the end-user who purchased the course.

IUTF: For most courses in English, leaving this value as "7" is adequate. A value of "8" will be used for multibyte character encoding (when the course's specific encoding has not been determined). Further a value of "16" is used if UNICODE encoding is to be used throughout.

ALLOW\_MULTI\_SST: (default value: "0") Sometimes it is necessary to have multiple SST directories (and therefore multiple uploaders) where the data needs to be kept separate. One example would be when courses for several customers are maintained on the same server. If multiple SST directories (meaning several installations of the SST files, each with its own *readygo* sub-folder) are desired, this value can be set to "1".

\$admin\_email: This is the e-mail address where any error messages or progress messages are sent by the upload routines.

\$from\_email: This is the return address for any e-mails sent to either the system administrator or to the uploader (when a course publication has been completed.)

\$contact\_email: When automated messages are sent to the end-users/course participants (in the case of courses that are purchased through ReadyGo Merchant, this e-mail address is provided if they have any questions.

Below the horizontal line, there is no need to edit the file. Below the line, there are some detailed settings for the debug setting along with a version of the PHP routine *fgetcsv* for older versions of PHP that do not contain it.

### **Configuration of the *readygo.sst* file**

SST the ability to store a summary of student events from all courses in a central file. It also has the ability to transmit a report of these events to a given e-mail address. The events of interest are:

- Student Registration (or denial)
- Test Request (or denial)
- Test Grading (or denial)
- Certificate Grant (or denial)

The interface to control these events is a configuration file placed in the "readygo" folder. The file is named "readygo.sst". For e-mail transmission of events, the server must have a command-line e-mail program like "sendmail" on Un\*x/Linux systems or "blat" on MS-Windows. Blat is available in source code and executable from [http://sourceforge.net/project/showfiles.php?group\\_id=81910](http://sourceforge.net/project/showfiles.php?group_id=81910)

This file is also used to control an “administrative password” for SST-related actions. With this password, actions such as reprinting a user’s certificate, deleting a single test result event from a test, control of uploaders through ReadyGo Administrator, and configuration of user lists and sales through ReadyGo Merchant can be enabled.

An example configuration file “readygo.sst” is provided below:

---

```
# Configuration file for SST

# Administrative password to allow an administrator #to delete entries from
# all courses.
admin_pw=testme

# In order to allow storage of data without all SST files, (e.g. course
# launched from LMS), it is necessary to allow this with a value of
# 'yes' or 'true'. Default is to not allow LMS launched course storage.
# allow_lms=true
allow_lms=false

# If central file storage is desired, set name here. #This is a relative name,
# so absolute names like "c:\home\all.txt" will not #work. It is relative to the
# "cgi-bin/readygo" or "scripts\readygo" folder, #where this file is located.
# Leave blank if no central storage file desired.
# e.g.: central_file=readygo.out
central_file=allres.txt

# If central file storage is on, entries from the #user registration file
# can be replicated in the output. These will be #placed after the
# standard built-in fields. Use internal storage #names from the
# registration setup. Separate these with vertical #bar: |
# e.g.: central_fields=last_name|email
central_fields=

# If storing data in a central file, need to define #what actions
# cause data to be saved. This is a list separated #by vertical bars (/).
# Possible values are: #all|register|test|cert|test_deny|test_request
# e.g.:central_actions=register|cert
# default is none.
central_actions=all

# The following is an internal code that can be set #for specific
# behaviors based on company name.
# e.g.: company=abc
company=

# If an e-mail is to be sent out when someone #registers, takes a test,
# gets a certificate, etc. insert the destination e-#mail address here.
# e.g.: email_to=you@readygo.com
email_to=

# If an e-mail_to value has been given, need to #define what actions
# cause an e-mail to be sent. This is a list #separated by vertical bars (/).
# Possible values are: #all|register|test|cert|test_deny|test_request|test_pass|test_fail
# e.g.: email_actions=register|cert
# default is none.
email_actions=test_deny|cert

# If e-mail_to and email_actions values have been #given, the command-line for
# transmitting an e-mail has to be provided. #Replaceable fields are the
# location of the temporary file containing the #outgoing e-mail ($file),
# the e-mail address ($email_to), from address #($email_from) and the subject ($subject)
```

```

# e.g: for sendmail, the command could be:
# email_command=/bin/mail -s $subject $email_to -- -#f "$email_from" < $file
# for blat (on windows) the command could be:
# email_command=d:\temp\blat262\full\blat.exe $file #-to $to_address -u you@yoursmt.com -pw
yourpasswd #-subject "$subject"
email_command=/bin/mail -s "$subject" $email_to -- -f "$email_from" < $file

# If email transmission is on, entries from the user #registration file
# can be replicated in the email.
# Use internal storage names from the
# registration setup. Separate these with vertical #bar: |
# e.g.: central_fields=last_name|first_name|email
email_fields=last_name|first_name

# If an e-mail is to be sent out when someone #registers, takes a test,
# gets a certificate, etc. insert the "from" address #here. Note that with some servers, this
will alert #the
# receiving computer that the e-mail may be fake.
# e.g.: email_from=you@readygo.com
email_from=

# If email transmission is on, the subject line can
# be customized.
# Internal names from registration can be used.
# Separate these with
# a vertical bar. The default subject line is:
# SST $action $action_mod on course: $course_name
# Possible fields are:
# $action: Replaced with the e-mail action above
# $mod_action: Used on some fields like test to
# modify e.g. Pass/Fail
# $course_name: Name of the course
# $test_name: Name of the test taken/request (if testing)
# any of the email_fields
email_subject=Report for $first_name $last_name on $course_name: $action $mod_action $test_name

# To override any of the settings above for a #specific course (e.g. "test1"), prefix the setting
#with
# the internal storage name and an underscore (e.g. #"test1_")
test1_email_to=support@readygo.com
test1_email_actions=all

```

---

The syntax of the file is as follows:

Any line starting with '#' is a comment line. Fields and their values are given one per line, with an '=' between the field name and its value. If a field has multiple values, these values are separated by '|'. There are comments before each field name that describe how it should be configured.

The admin\_pw is an administrative password for specific result deletion (discussed in the next section). The company\_name field is used for special customizations that can be purchased from ReadyGo for your organization's needs.

Certain LMS-interface packages in Web Course Builder allow user registration to occur through a Learning Management System (LMS), with enhanced tracking through SST. This means that the requirement that a user must be registered through SST to enable storage of their results needs to be loosened. By setting the variable **allow\_lms** to "true" or "yes," SST will allow storage of data without checking that the user's student\_id has been set in the registration file for that course. Whatever student identifier that the LMS passes back to the course for that student, will be picked up and used to tag each user result set. Reports will present the raw data.

To enable central storage of events, the field `central_file` needs to have a non-empty value. The value for `central_file` is the name of the file where results should be stored. The path is relative to the “readygo” directory. If `central_file` has no value, no storage will be carried out.

To enable e-mail transmission of summaries, the field for `email_to` must have the e-mail address of the recipient/course administrator. It is also important that the line for `email_command` be configured properly. ReadyGo recommends that the command line e-mailer be tested manually to help develop the syntax of the command for the `readygo.sst` file.

One feature of this file is that settings can be customized for each course separately. To override the settings for a specific course, repeat the field name, but pre-pond the course storage name and an underscore before the field name. For example, if there is a desire to transmit the summary for course “test1” to a different administrator, include the line:

**`test1_email_to=othername@yourdomain.com`**

Then, for course “test1”, the summary will be e-mailed to the alternate address. If e-mail is not desired for the test1 course, leave the value of `test1_email_to` blank.

The e-mail report is of the form:

To: [al@readygo.com](mailto:al@readygo.com)

Subject: *SST Registration accepted on course: tst1ab*

```
student_id=newn0001
from_ip=127.0.0.1
date_time=2008-02-21 12:51:59
action=registration
status=registered
mastery=0.000000
last_name=newname
first_name=blueman
```

For different events the value of the action will be different.