The Great Reading Adventure Documentation

Release 4.1

Maricopa County Library District

Dec 13, 2018

Introduction

The Great Reading Adventure is a robust, open source software designed to manage library reading programs. The GRA is free to use, modify, and share. We hope you like it and will consider becoming a part of the community.

The code is open source, and is available on GitHub.

Note: This documentation is currently very incomplete. If you need help, please don't hesitate to contact developers and users on The Great Reading Adventure Forum.

This manual is split into several main sections:

- *Introduction* - What is The Great Reading Adventure? How does it work?
- *Installation* - installing and configuring The Great Reading Adventure software
- *Setup* - setting the software up to work with your reading program(s)
- *Technical Documentation* - technical documentation

CHAPTER 1

Overview

The Great Reading Adventure (GRA) is a robust, open source software designed to manage library reading programs. The GRA is free to use, modify, and share!

The GRA functions on computers, tablets, and mobile phones.

Features include:

1.1 Avatars

Engage participants by letting them put together a customized avatar that they can share via social media. The GRA includes almost 4,500 avatar image assets provided under a Creative Commons CC0 1.0 Universal License which can be interchanged to create billions of avatar combinations.

1.2 Badges

> "Digital Badges are an assessment and credentialing mechanism that is housed and managed online. Badges are designed to make visible and validate learning in both formal and informal settings and hold the potential to help transform where and how learning is valued." – MacArthur Foundation

Digital badges are an easy way to enhance your reading program. They can be used in addition to (or as a replacement for) physical incentive prizes. Digital badges go beyond the momentary happiness that a toy incentive provides, as they carry with them recognition of achievement, engagement, and personal identity. The Great Reading Adventure makes it easy to create badges for those who participate in library and other community experiences.

Here's an example of how it can work:

- A participant attends a program at your library.
- At the end of the program, you give the participant a secret code for attending.
- The participant enters the secret code into their online reading log.

- The participant receives the badge. The integrated badge wizard makes the process simple: create an event, upload or design a badge, and assign a secret code. It's a great new way to tie the online summer reading program to the events in your bricks-and-mortar locations.

1.3 Challenges

Participants can perform a number of tasks in order to complete challenges. Tasks are very flexible and participants can complete them away from the computer, by completing activities online, or by reading books. The system awards points and a badge when they complete a number of tasks in a challenge.

1.4 Drawings

Prize drawings can occur from a pool of all participants or from a specific set of participants based on drawing criteria (such as everyone who earned a certain amount of points over a certain date range or participants registered at a specific location). The system saves criteria for easy reuse.

1.5 Events

The built-in event system allows entering all events that occur during the program that helps publicize your events and possibly supplement an existing event management system. Events easily link to challenges or secret codes to relate them back to earning points and badges.

1.6 Mail

The GRA's built-in mail facilities allows direct engagement with participants without requiring them to provide an email address or leave the system. Any number of back-office staff can handle incoming messages from the threaded administration interface. The system also sends broadcast messages to all participants if desired.

1.7 Participant management

Participant management has different configurable permissions so that volunteers access a subset of features and access is restricted from personally identifiable information (if desired).

1.8 Triggers

Want to award a special badge when participants earn a certain amount of points? Configure a mega-badge when participants earn a set of several other badges or complete a number of challenges? Triggers are the answer! Triggers can award badges or prizes based on the criteria you configure.

1.9 Questionnaires

You put a lot of time and effort into your summer reading program- it is important to know that what you are doing has an impact. The GRA's comprehensive evaluation tools will help you do just that.

With our pre- and post-testing modules, you can deliver an assessment to participants and compare their scores to their activity during the program. The system auto-scores the multiple-choice assessments, which gives you all the data with none of the hassles.

The assessment module doubles as a survey tool, so you can simultaneously gather qualitative and quantitative data to tell a more impactful summer reading story.

1.10 Reporting

With The Great Reading Adventure, immediate access to your program's vital statistics is just a click away! At-a-glance stats give you a quick breakdown of registrations and completions and in-depth reporting allows you to drill down into the details.

All reports are exportable to HTML and Excel, putting all the summer reading data you need right at your fingertips.

CHAPTER 2

Planning for initial configuration

2.1 Hosting

As a Web application, the first decision that will need to be made is about hosting. Hosting requirements can be found in the system requirements section of this manual.

2.1.1 Self-hosting

If your organization has the proper Linux, macOS, or Windows server environment that is accessible from the Internet, hosting on your own systems can be an option. You can host the application on an existing Web server or host the application in a container using Docker.

2.1.2 Paid hosting

Various providers are available which can host the GRA for a monthly fee.

2.2 Application configuration

2.2.1 Programs

Setting up multiple programs gives the flexibility to deliver targeted content to varying audiences keeping key elements unified for reporting and statistics. You have the freedom to customize the content of each program so that participants can access audience-appropriate activities, badges, and events. Users can self-select which program to enroll in or the system can automatically place them into an age-appropriate program, based on age or school grade.

Setting up multiple programs adds flexibility but increases complexity.

The default installation of the GRA sets up four programs defined by age:

- Prereaders (ages 4 and below)

- Kids (ages 5 to 11)
- Teens (ages 12 to 17)
- Adults (ages 18 and up)

These default programs are configured for participants to enter a number of minutes that they read every day with each minute equating to one point in the program. The translation of activity (minutes read) to points earned can be modified once installed, however at this time there is one translation for all programs.

If you would prefer the initial setup to be a single program which is configured to earn a single point for each book read you can set the `GraInitialProgramSetup` configuration value to "single".

Creating a **single reading program** is good if you:

- Are targeting a single age group
- Intend to have the same experience for all participants
- Aren't intending to require a literacy test for only some participants

Several age-specific reading programs are better if you:

- Want to report on age groups differently
- Intend to have Badges, Challenges, and Events specific to age groups

CHAPTER 3

System requirements

The GRA requires the following services to run. These services may all be hosted on the same machine or may be hosted on separate machines if desired. As an alternative to installing the GRA directly in a Web server environment, the GRA can be run from a Docker container using our official Docker images.

3.1 Web site

3.1.1 Web server requirements

Version 4 of the GRA runs in the Microsoft .NET Core 2.2 runtime environment. Downloads of the runtime environment are available from Microsoft for the following operating systems:

- macOS (10.12 "Sierra" and later versions)
- Linux (Red Hat Enterprise Linux, CentOS, Oracle Linux, Fedora, Debian, Ubuntu, Linux Mint, openSUSE, SUSE Enterprise Linux, Alpine Linux)
- Windows (x64 and x86)

As of this writing the latest release of the .NET Core 2.2 environment is Runtime v2.2.0. To host in a Windows environment you'll need the Runtime & Hosting Bundle, for other environments you will need to install the appropriate runtime. You can safely use any 2.2.x version of the runtime (though you probably want the latest as it will incorporate security fixes).

ASP.NET Core applications can be run behind a reverse proxy or directly connected to the Internet using the built-in ASP.NET Core Kestrel Web server.

- To host with IIS, Windows Server 2008 R2 or later is supported. Note that currently GRA reporting utilizes Web Sockets and in a Windows environment that requires IIS 8 and Windows Server 2012.
- To host with Apache or Nginx, a version of Linux which supports .NET Core 2.2 should be selected (see above).

3.1.2 Docker server requirements

Once Docker is installed in your environment (configured for Linux containers) you are good to go!

3.2 Database server requirements

The GRA version 4 supports the following database environments:

- Microsoft SQL Server 2008 or later using SQL Server authentication mode.

3.3 Mail server requirements

The ability to send Internet email, such as a service which accepts email via SMTP.

- The GRA sends mail in certain instances (such as helping users recover their lost passwords) and requires the ability to connect to an SMTP server.

CHAPTER 4

Requirements checklist

4.1 Configuration information you'll need

4.1.1 Configuration step 1: database configuration

- Database server name or IP address
- Database/catalog name
- Database owner user login (the user in the `db_owner` role)
- Database owner user password

4.1.2 Configuration step 2: mail server configuration

- The administrator's email address - you may want to set up a role address ahead of time so that system emails don't appear to come from your personal address
- Mail server - SMTP server to handle emails
- Mail server port (optional) - by default 25 will be used
- Mail server login (optional) - if you need to authenticate to send email
- Mail server password (optional)

4.1.3 Configuration step 3: select an initial program configuration

Your final decision is which initial program configuration to choose:

- You can opt to set up with a single reading program that **tracks by books read**
- You can opt to set up with four age-specific reading programs that **track by minutes read**

Once you set up in either configuration you can add or remove programs as you see fit.

For more information on these options, please review the planning section of this manual.

4.2 Hosting in a Windows environment

- Ensure you have a Windows server running **Windows Server 2008 or newer**
- Note that for reporting to work you must be running **Windows Server 2012/IIS 8 or later with Web Sockets enabled**
- Ensure your server has the **.NET Core Runtime 2.2.x Hosting Bundle** installed
- Confirm that you can **create a new Web site in IIS** on this server
- Confirm that you will be permitted to configure it so that **Web site files can be writable by the Windows user who owns the IIS process (typically the `IIS_IUSRS` group or `DefaultAppPool` user)**.
- Ensure you have access to a **Microsoft SQL Server version 2008 or newer**
- Ensure that you'll be able to authenticate in **SQL Server authentication mode**
- Confirm that you'll be able to **create a database**
- Ensure that you have a **mail server with an accessible SMTP port or the ability to deliver mail from a service running on the Web server**
- The **latest release of the GRA software** downloaded from GitHub.

4.3 Hosting in a Docker environment

Setting up to run the Web site using Docker is much simpler as the environment is entirely contained in the Docker image.

- Ensure you have Docker installed properly using Linux containers. You can verify your install with:
  - The Docker Hello World sample (`docker run --rm hello-world`)
  - The Microsoft ASP.NET Core sample (`docker run --rm microsoft/dotnet-samples`)
- Ensure you have access to a **Microsoft SQL Server version 2008 or newer**
- Ensure that you'll be able to authenticate in **SQL Server authentication mode**
- Confirm that you'll be able to **create a database**
- Ensure that you have a **mail server with an accessible SMTP port or the ability to deliver mail from a service running on the Web server**

CHAPTER 5

Create the database

The GRA uses its own database to store reading program data. This database can be completely segregated from any other databases on the same SQL Server.

There are two options when creating the database: you can execute a script or manually create the database and users using SQL Server Management Studio (SSMS).

5.1 From a script

1. Connect to the database server to run a query (this can be done with SQL Server Management Studio by right-clicking on the server in the Object Explorer on the left and choosing **New Query**).
2. Copy the text from the database create script and paste it into the query window.
3. Click the **Execute** button in the toolbar.

5.2 From the user interface

5.2.1 Create the database

1. Launch SQL Server Management Studio and connect to the database server.
2. Double-click the server name to show the **Databases** folder.
3. Right-click on the **Databases** folder and select the **New Database...** option.
4. In the New Database window, enter a database name, for example: `SRP`.
5. If you wish to have the database files stored in a different location than the default, scroll the **Database files** list to the right and change the presented **Path** to where you'd like to store the database files.
6. Select **Options** from the page list on the left.
7. For the **Recovery model** drop-down, change the selection from `Full` to `Simple`.

8. Select **OK** to create the database.

5.2.2 Create the logins and users

1. Right-click the **Security** folder and select **New -> Login...**.
2. Enter the **Login name** for the database owner user, for example: `srp_owner`.
3. Select `SQL Server authentication`.
4. Enter a secure password for this user in the **Password** and **Confirm password** fields. Make a note of this password.
5. You can uncheck `Enforce password policy` so that the password for this account will not expire.
6. Select `User mapping` from the **Select a page** list on the left.
7. Locate the name of the database (this tutorial used `SRP` in *Create the database* step 4 above) in the **Users mapped to this login** list on the left.
8. Check the appropriate box in the `Map` column to map this login to a user in the database.
9. In the **Database role membership for:** list, select `db_owner`.
10. Select **OK** to create the login and user.

CHAPTER 6

Configuration

Initial configuration for version 4 of The Great Reading Adventure involves setting up database connection information and the initial signup authorization code. There are additional configuration settings but they are optional.

To customize settings in The Great Reading Adventure, create a file named `appsettings.json` and place it in the `shared` directory under your GRA installation. Ideally customization happens in this directory so that changes are not overwritten when updating the site with newer versions of the software. For a starting point, the `appsettings.json` file can be copied from the installation directory into the `shared` directory to provide a starting point. Alternately, a new text file can bre created as long as the extension of the file is `.json` and not `.txt` (be aware that by some operating systems (like Windows Server) tend to hide file extensions by default).

The critical settings to provide are the "SqlServer" setting under "ConnectionStrings" and the "GraInitialAuthCode" setting. Most of the other settings can be customized once you authenticate with an administrator-level account.

Please note: Application settings are configured in a JSON or "JavaScript Object Notation" file. This file can be edited with any text editor (such as notepad.exe) but must be in a specific format. You can find validators online which will help you ensure that the syntax of the file is correct. Also note that when a backslash (\) or double quote (") appears within quotes (for example in the database password) it must be escaped, meaning a backslash should appear prior to the escaped character (e.g. \\ or \").

6.1 Connection string

```
"ConnectionStrings": {
  "SqlServer": "Server=<servername>;Database=<databasename>;user id=<username>;
↪password=<password>;MultipleActiveResultSets=true"
},
```

In the above example you'd replace the following:

- `<servername>` - Hostname or IP address of your SQL Server
- `<databasename>` - Name of the SQL Server database
- `<username>` - SQL Server login to use

- <password> - SQL Server password to use

For password generation, please consider using a utility like pwgen in a Linux environment or something similar to the online Diceware password generator. If you can create a long and complex password without backslash (\) or double quote (") in it you will not have to worry about escaping those characters in the configuration file.

6.2 Authorization code

```
  "GraInitialAuthCode": "<authorizationcode>"
```

This is the code that you will use when you set up your administrator account to grant you full access to Mission Control (the administrative interface) of the software. Please change this value to ensure that people who come across your site cannot grant themselves full administrator access!

6.2.1 Sample configuration file

Here's what your appsettings.json file in your shared directory might look like if you are changing those two required configuration settings:

```
{
  "ConnectionStrings": {
    "SqlServer": "Server=<servername>;Database=<databasename>;user id=<username>;
↪password=<password>;MultipleActiveResultSets=true"
  },
  "GraInitialAuthCode": "<authorizationcode>"
}
```

6.3 More configuration options

The Application Settings section of the manual provides a comprehensive list of settings that can be configured in the appsettings.json file.

CHAPTER 7

Install the software

The final installation step is to install and run the GRA software.

7.1 Install and run the GRA on a Windows Server

The hosting Windows Server should have IIS installed and the Web Sockets feature enabled. You do not need to install the Windows Authentication IIS feature.

Also, ensure the server has the proper .NET Core v2.2 Hosting Bundle installed.

7.1.1 Set up the GRA as the only site on the server

If this Web server is configured solely for the Great Reading Adventure, you can utilize the default Web site for your GRA installation.

1. Delete existing default files placed in `c:\inetpub\wwwroot\` such as `iisstart.htm` and `welcome.png`.
2. Unzip the GRA files into `c:\inetpub\wwwroot\`. Ensure that the files are placed in that directory directly and not in a subdirectory (you should see files such as `appsettings.json` and `GRA.dll` in `c:\inetpub\wwwroot\`).
3. Create a new folder named `shared` in `c:\inetpub\wwwroot`.
4. Right-click on the `c:\inetpub\wwwroot\shared` directory and select **Properties**.
5. Choose the **Security** tab and then click the **Edit** button next to *To change permissions, click Edit*.
6. If you don't see the `IIS_IUSRS` group listed, select **Add** in the **Permissions for wwwroot** window and in the **Enter the object names to select** box, enter in `IIS_IUSRS` and click **OK** (the `IIS_IUSRS` group is a local group on this machine, so ensure that the **From this location** box has the name of the Web server and not your domain - you can change this by clicking **Locations...** and select the server). You may also use an AppPoolIdentity account instead of the `IIS_IUSRS` group if required in your environment for security reasons.

7. Ensure all the checkboxes (including Modify) are selected except **Full control** and **Special permissions** in the **Allow** column.
8. Select **OK** to close this window and **OK** to close the **wwwroot Properties** window.
9. Ensure you have updated the configuration in the `appsettings.json` file - or if you are using a configuration override file, ensure it's named `appsettings.json` and placed into the `shared` folder. See the configuration section of this document for more details.
10. Launch a Web browser on the server and navigate to the URL `http://localhost/`.
11. At this point you should see the initial GRA setup screen.
12. You can continue the GRA setup process either directly in this Web browser or you can navigate to the Web server name or IP address in a browser on another system to continue.

7.1.2 Set up the GRA as an additional site on the server

If you are deploying the Great Reading Adventure to a Web server which is already hosting one or more Web sites, you must create a new Web Site specifically for the GRA.

Because this Web server is already serving out files through the default Web Site, you must differentiate your GRA Web site somehow. Methods for doing so include:

- Setting up a separate host name for your GRA Web site (for example: `http://gra.<your domain>`. This method utilizes the HTTP Host Header Field.
- Setting up a separate IP address on the Web server and putting your GRA site there.
- Setting up the GRA on a different port number so that you'll access it via `http://<your Web server>:<GRA port>/`.

Your system administrator can help you select the correct approach.

1. Create a directory on the Web server to contain the GRA files. For example, `c:\inetpub\gra\`.
2. Unzip the GRA files into that directory. Ensure that the files are placed in that directory directly and not in a subdirectory (you should see files such as `appsettings.json` and `GRA.dll` in `c:\inetpub\gra\`).
3. Right-click on directory created in step 1 above and select **Properties**.
4. Choose the **Security** tab and then click the **Edit** button next to *To change permissions, click Edit*.
5. If you don't see the `IIS_IUSRS` group listed, select **Add** in the **Permissions for ...** window and in the **Enter the object names to select** box, enter in `IIS_IUSRS` and click **OK** (the `IIS_IUSRS` group is a local group on this machine, so ensure that the **From this location** box has the name of the Web server and not your domain - you can change this by clicking **Locations...** and select the server). You may also use an AppPoolIdentity account instead of the `IIS_IUSRS` group for security reasons.
6. Ensure all the checkboxes (including Modify) are selected except **Full control** and **Special permissions** in the **Allow** column.
7. Select **OK** to close this window and **OK** to close the **Properties** window.
8. Ensure you have updated the configuration in the `appsettings.json` file - or if you are using a configuration override file, ensure it's named `appsettings.json` and placed into the `shared` folder. See the configuration section of this document for more details.
9. Open up the **Internet Information Services Manager** on the Web server.
10. Expand the Server under **Connections**.
11. Right-click on **Sites** and choose **Add Web Site...**.

12. Enter an appropriate site name such as `SummerReading`.
13. Enter the physical path where you put the GRA files under **Physical path** (we used `c:\inetpub\gra` above).
14. In the **Binding** section you will either need to assign an IP address, a host name, or select a different port as defined above.
15. Select **OK** to close the **Add Web Site** window.
16. Launch a Web browser on the server and navigate to the URL you defined for this install of the GRA.
17. At this point you should see the initial GRA setup screen.
18. You can continue the GRA setup process either directly in this Web browser or you can navigate to the Web server using the URL you defined for this install of the GRA.

7.1.3 Troubleshooting a Windows installation

1. Did you modify the `appsettings.json` or provide an override file in the `settings` directory to configure the database connection and authorization code?
2. Did you install the .NET Core 2.2.x Hosting Bundle?
3. Did you restart IIS after installing it?
4. Are you sure the `shared` directory has write permissions for the process running IIS? Once the application starts, it will create a "logs" directory there so you will know if the Web server can write to this directory once you see the "logs" directory present. Check that "logs" directory to see if there are any errors written out to the log file.
5. If the Web process can write to the `shared` directory, you can edit the `Web.config` file `aspNetCore` tag to set `stdoutLogEnabled="true"` and `stdoutLogFile=".\shared\stdout"`. When you restart IIS it should write a log file starting with "stdout" that you can examine to see if any errors are being written to it.
6. Microsoft's Troubleshoot ASP.NET Core on IIS page will walk you through some typical troubleshooting steps.

7.2 Install and run the GRA in Docker

If you don't currently have a Web site running on your Docker server you can forward port 80 from the server directly into the Docker container.

1. Create a directory on the Web server to contain the shared GRA files, for example `/gra/shared` in Linux or `c:\gra\shared\` in Windows.
2. Place your `appsettings.json` file into the shared directory that you just created.
3. From a prompt (a `bash` shell, `command prompt` or `PowerShell` window) start the Docker image with a command similar to `docker run -d -p 80:80 --name gra --restart unless-stopped -v /gra/shared:/app/shared mcld/gra` - details of that command:
    - `-d` tells Docker to run the container in the background
    - `-p 80:80` says to forward port 80 from the local server to port 80 in the container
    - `--name gra` provides a name for the container to make it easier to reference while it's running (e.g. if you have to stop the container you can with `docker stop gra`)
    - `--restart unless-stopped` will restart the container if it should stop unless you explicitly stop it

- `-v /gra/shared:/app/shared` tells Docker to share the `/gra/shared` directory on the local server with the `/app/shared` directory inside the container
- `mcld/gra` is the image to run - this will download and run the `mcld/gra:latest` image from Docker Hub

4. Launch a Web browser on the server and navigate to the URL you defined for this install of the GRA (for the above example, `http://localhost/` will work).
5. At this point you should see the initial GRA setup screen.
6. You can continue the GRA setup process either directly in this Web browser or you can navigate to the Web server using the URL you defined for this install of the GRA.

In the case that there is already a Web site running on your server you will need to forward a different port into the Docker container. If you chose to forward port 2001 from your server into the GRA container you'd use `-p 2001:80` in step 3 above and then in step 4 you'd access the site by navigating to `http://localhost:2001/`.

The software should create a "logs" directory inside the `shared` directory which you can review to see if there are any errors written out to the log file.

CHAPTER 8

Upgrading

There is no upgrade path from versions 2 or 3 to this release due to significant architectural changes. Follow these instructions to upgrade from a previous release of version 4 (i.e. 4.0, 4.0.0-beta1, and 4.0.0-beta2). When upgrading from version 4.0 to 4.1 note that you'll need to ensure you have the appropriate version of the .NET Core Hosting Bundle on the server.

Note that during the upgrade there will be an interruption in service so it may be ideal to schedule this upgrade in off hours.

8.1 Upgrading a Web server

1. Back up the database and the files that comprise the Web site.
2. If you have modified any files in the application (such as `.cshtml` Razor template files), please make a note of which changes you have made, these changes will have to be performed again. Changes to the `shared` directory (such as to `style.css` or uploaded files) will be maintained.
3. Replace the application files with the files from this release. **Ensure that your `shared` directory is not overwritten when replacing application files**.
4. The `appsettings.json` file in the application folder will be replaced when you copy in files from the release. If you modified this file please compare to the file you backed up in the first step above to ensure any configuration changes you made initially are set in the new file. If you added an `appsettings.json` file in the `shared` directory with your settings it will **not be overwritten** and you shouldn't need to change any settings.
5. Load the site in your browser. It will take longer than normal as the database is upgraded (you can see evidence of the database upgrade in the log file in `shared/logs`).

8.2 Upgrading a docker instance

1. Get the name of the current Docker container (`docker ps -a`). We'll assume it's named `gra` for this example.
2. Stop the current Docker container (`docker stop gra`).
3. Remove the current Docker container (`docker rm gra`). Data will be saved in your database and any uploaded files are contained in your `shared` directory.
4. Look for any GRA docker images you have on your system (`docker image ls`).
5. Remove any GRA docker images on your system (`docker rmi mcld/gra:latest`).
6. Use the command that you initially used to run the Docker container to download the latest container and run it (for example `docker run -d -p 80:80 --name gra --restart unless-stopped -v /gra/shared:/app/shared mcld/gra`, see the Install the Software section of this manual for more information).
7. Load the site in your browser. It will take longer than normal as the database is upgraded (you can see evidence of the database upgrade in the log file in `shared/logs`).

CHAPTER 9

Setting up your administrator account

Once you can access your installation of the software, it's time to set up the initial administrator account. To do this, you'll need the initial Authorization Code (configured as `GraInitialAuthCode`) that you put into the `appsettings.json` file.

9.1 Joining with an Authorization Code

The easiest approach is to join and grant yourself administrative rights at the same time.

1. Visit `/Join/AuthorizationCode/` in your installation
2. Supply the Authorization Code specified as `GraInitialAuthCode` in your configuration file
3. Once you've completed the process you'll be able to click the rocket in the navigation bar at the top to access Mission Control

9.2 Joining with a regular account

1. Select the "Join" link or visit `/Join/` in your installation.
2. Complete the process to create your account.
3. Visit `/MissionControl/` in your installation
4. Supply the Authorization Code specified as `GraInitialAuthCode` in your configuration file
5. The system will grant you full rights and place you in Mission Control

CHAPTER 10

Adding avatars

Artwork from the game Glitch was made freely available under a Creative Commons CC0 1.0 Universal License license (essentially a "no rights reserved" license). The artwork used to create in-game characters has been adapted to work with The Great Reading Adventure as the participant's avatar.

10.1 Loading the avatars

You must load the avatars into the software before they can be used.

1. Access **Mission Control** on your installation.
2. Choose the **picture frame icon** from the navigation menu and select "**Avatars**".
3. Click the "**Add default avatars**" button.
4. Wait. **This takes a while** - it may take so long that the Web page itself times out. If you want to see the progress, look in the `shared/logs` directory on your server, it should output as it's importing the various items. You'll see "Default avatars added in *x* seconds." in the log file when it is done.
5. Navigate through the avatar section of the software and ensure that all of the available assets are agreeable for your program(s) and audience.
6. The `assets` directory can be removed (if desired) once avatars are loaded.
7. Optional: when participants share their avatars via Open Graph markup or Twitter Cards their avatar is placed upon a stage to make it better fit into the image sizes necessary for Open Graph and Twitter: the file named `background.png` in the `assets` directory will be loaded into the GRA as that stage. There are other options in the `backgrounds` folder - if you would like to use one of those just rename the one you select `background.png` and overwrite the current `background.png` file.

CHAPTER 11

Site customizations

Many parts of the GRA can be customized for your specific situation.

11.1 Site settings

By default site settings coded into the software or provided in the configuration file are used to define things like the name of the site and the page footer. These can all be customized in Mission Control by selecting the gear icon on the right side of the navigation bar and choosing "Site management" from the drop-down menu.

11.2 Custom landing pages and dashboard

The GRA has two operating modes, either without schedule or with scheduled dates.

1. The default operation of the GRA means that as soon as you install the software the program is open and running for registration and for participants to log activities. In this instance the software considers the program to always be in the "open" state and uses that template for the landing page.
2. Through "Site management" in Mission Control a schedule can be configured dividing the program up into the following stages:
   (a) Before registration opens
   (b) Registration open
   (c) Program starts
   (d) Program ends (but accounts are still accessible)
   (e) Access closed

11.2.1 Templates

When the GRA starts up, it will create a `templates/Home` directory in the `shared` directory containing template files which are used for displaying the landing page(s) and dashboard. If you want to customize these pages you can copy the files from `templates/Home` into `views/Home` and then modify them.

- `IndexBeforeRegistration.cshtml` is shown during stage 1 above, before registration has opened.
- `IndexRegistrationOpen.cshtml` is shown during stage 2 above, once registration is opened but before participants can log activity.
- `IndexProgramOpen.cshtml` is shown during stage 3 when the program is open for participants.
- `IndexProgramEnded.cshtml` is shown during stage 4 when the program has closed, however participants may need to access the site to retreive prize codes or review their mail.
- `IndexAccessClosed.cshtml` is shown during stage 5 when participants may no longer sign in.
- `Dashboard.cshtml` is shown when a participant logs in.

11.3 Global styles and scripts

Starting with version 4.1.1, custom styles and scripts can be configured which will be added to every page site-wide. Note that the GRA will only check if these files have been changed every 60 minutes by default. The site setting "Check for site.css and site.js changes on disk" under Web in Site management can be changed to a number (in minutes) that you'd like to check for these files to have changed. Set the value to 0 while editing the files to see them refresh every time you hit reload.

11.3.1 Styles

Additional CSS styling can be added in the `shared` directory: create a subdirectory called `styles` and place a `site.css` file in it. This CSS file is loaded last so any changes provided in it should take precedence over built-in CSS styles. As an example: if you'd like to make the background of the navigation bar light blue, place the following in `shared/styles/site.css`:

```
.gra-navbar { background-color: Azure; }
```

11.3.2 Scripts

Custom JavaScript that you'd like injected into the site can be added in the file: `shared/scripts/site.js`. This file is loaded after all of the other JavaScript so elements should be available for you to access or modify.

CHAPTER 12

Application Settings

The GRA checks several locations for configuration settings:

1. First the `appsettings.json` file in the deployed application directory (where the `GRA.dll` and `GRA.Web.dll` files are (**we recommend that you don't edit this file as it may be overwritten during software upgrades**)
2. Next, the `shared/appsettings.json` in the deployed application directory - settings in this file override any settings in the top level `appsettings.json` file
3. Finally the GRA checks environment variables - any configured environment variables are passed into the software. If you don't wish to put sensitive information (such as your configuration string) into a file in the application directory you can configure those items via environment settings.

Please note: Application settings are configured in a JSON or "JavaScript Object Notation" file. This file can be edited with any text editor (such as notepad.exe) but must be in a specific format. You can find validators online which will help you ensure that the syntax of the file is correct. Also note that when a backslash (`\`) or double quote (`"`) appears within quotes (for example in the database password) it must be escaped, meaning a backslash should appear prior to the escaped character (e.g. `\\` or `\"`).

Any settings below not marked with a version number were added in v4.0.

12.1 Connection strings

One connection string is required (either `SqlServer` or `SQLite`).

- `SqlServer` - A SQL Server connection string
- `SQLite` - SQLite connection information (typically the path to the SQLite database file)
- `SqlServerSessions` - *optional* - A SQL Server connection string for storing session data in a SQL Server database (necessary for multiple Web servers answering requests for the same site)
- `SqlServerSerilog` - *optional* - A SQL Server connection string used for storing SQL Server application logs; the user should have database owner access (at least initially) so that it can create the proper table for logging

12.2 General settings

- `GraConnectionStringName` - which connection string to use (either `SqlServer` or `SQLite`)
- `GraCulture` - *optional* - defaults to "en-US", the culture to use for displaying things like dates and times - for valid options see the language tags listed in the Microsoft National Language Support (NLS) API Reference
- `GraInitialAuthCode` - the Authorization Code entered to grant users full access to the site - **it's important that you change this!**
- `GraInitialProgramSetup` - *optional* - defaults to "multiple" which creates four age-based programs and sets up a point translation of one minute read equals one point, can also be set to "single" which creates one program and sets up a point translation of one book read equals one point
- `GraMaximumAllowableActivity` (v4.1.0) - *optional* - for reporting purposes, limit participant activity above this amount to the "achiever" level configured for the participant's program
- `GraReverseProxyAddress` - *optional* - if provided, internally the software will disregard proxy IP addresses
- `GraRollingLogPath` - defaults to "shared/logs", a path to save a daily-rotating log file - if `GraInstanceName` is specified in `appsettings.json` it will be included in the log file name
- `GraSqlServer2008` - *optional* - if you are using SQL Server 2008, put text into this setting (any text will do)

12.3 Default settings

These settings are used when the program runs for the first time to insert some reasonable defaults into the database. All of these settings are optional. All of these settings can be configured in the Site Settings area of Mission Control.

- `GraDefaultSiteName` - defaults to "The Great Reading Adventure", what the site refers to itself as
- `GraDefaultPageTitle` - defaults to "Great Reading Adventure", set in many page titles
- `GraDefaultSitePath` - defaults to "gra", this is used for tenancy (which is not implemented yet)
- `GraDefaultFooter` - the footer output on every web page
- `GraDefaultOutgoingMailHost` - the hostname or IP address of the outgoing mail server
- `GraDefaultOutgoingMailLogin` - login name for the mail server (if needed)
- `GraDefaultOutgoingMailPassword` - password for the mail server (if needed)
- `GraDefaultOutgoingMailPort` - defaults to "25", port to connect to for relaying SMTP emails

12.4 Static file settings

- `GraContentDirectory` - defaults to "shared/content", the path to the shared content files for this instance of the application
- `GraContentPath` - defaults to "content", the URL path to the files in the `GraContentDirectory` (e.g. by default accessing /content/ with your Web browser serves files off the disk from the content/shared directory)

12.5 Distributed cache and multiple front-end settings

When operating in a load-balanced environment these settings are used to configure instances to keep settings and data shared or unique as necessary.

- `GraApplicationDescriminator` - defaults to "gra", application discriminator to use for data protection
- `GraDataProtectionPath` - defaults to "shared/dataprotection", location to save the data protection key if not using Redis as a distributed cache
- `GraDistributedCache` - *optional* - select a system to use for distributed cache: "Redis" or "SqlServer", anything else uses an in-memory distributed cache
- `GraInstanceName` - the name of this deployed instance
- `GraRedisConfiguration` - *optional* - address of a Redis server for distributed cache, only used if `GraDistributedCache` is set to "Redis"
- `GraSqlSessionSchemaName` - *optional* - the schema to use for the SQL Server distributed cache table, defaults to "dbo"
- `GraSqlSessionTable` - *optional* - the table to use for the SQL Server distributed cache, defaults to "Sessions"

12.6 Developer settings

These settings are primarily of interest to developers working on The Great Reading Adventure source code.

- `GraDatabaseWarningLogging` (v4.1.1) - *optional* - when set to any value write relational database warnings to the log
- `GraEmailOverride` - *optional* - override any emails and send them to this address

12.7 Logging with Serilog

Customization can be done to the way Serilog works by adding a "Serilog" section to the log file. For example, logging to Slack can be added by putting the following configuration section in (and replacing `<webhook URI>` with the actual Slack incoming webhook URI):

```
"Serilog": {
  "MinimumLevel": "Debug",
  "Enrich": [ "FromLogContext" ],
  "WriteTo": [
    {
      "Name": "Slack",
      "Args": {
        "webhookUri": "<webhook URI>",
        "restrictedToMinimumLevel": "Warning"
      }
    }
  ]
}
```

More information about customizing Serilog in `appsettings.json` can be found in the serilog-settings-configuration project on GitHub.

CHAPTER 13

About the GRA and manual

13.1 The Great Reading Adventure

The Great Reading Adventure was initially developed by the Maricopa County Library District with support by the Arizona State Library, Archives and Public Records, a division of the Secretary of State, with federal funds from the Institute of Museum and Library Services.

The Great Reading Adventure source code is distributed under The MIT License.

13.2 Colophon

The content of this manual is part of the open-source project on GitHub. It is generated using the following tools and technologies:

- reStructuredText and CommonMark (a fork of Markdown) text markup languages.
- Sphinx generates the manual from source files.
- GitHub provides version control and source code management.
- Read The Docs generates and hosts the online manual in PDF and HTML formats.