# PowerGrid Class Documentation

## Bot class

This class is meant to represent the player. It contains mostly high-level operations such as “Travel to <point>” and “Go to nearest <destination>”. It also provides basic information about the player such as its position, as well as its state.

## PowerGrid class

This class controls PowerGrid’s main functions. It is the class that loads and starts all PowerGrid modules and also starts RSBot itself. It provides methods for starting and stopping PowerGrid, and also holds a small Logging facility that can be used to log messages to the console.

## Control package

### IOHandler class

This class provides an easy way of reading and writing files. It provides two versions for each operation: one version through Java’s I/O classes and one version where the shell is used for the operation. This last version is useful, since RSBot blocks certain File operations. Having the shell perform the actual operation prevents RSBot’s SecurityManager from blocking it.

### Mapper class

This class manages updating PowerGrid’s world map. The world map is an essential part of PowerGrid, since it serves as the primary resource for path finding operations, and also for looking up interactions, NPC’s and basically every other object in the Runescape world.

### PathFinder class

The PathFinder class contains functionality to compute paths throughout the Runescape world. The PathFinder class takes along interactions, teleports and transports in the computation, ensuring that the computed path is indeed the most efficient route to get to a certain destination. The PathFinder class uses A\* to compute the paths.

### ScriptLoader class

The ScriptLoader class can interface with RSBot and provides commands to programmatically start and stop scripts. The ScriptLoader can work with either the fully qualified name of an ActiveScript subclass, the Class object corresponding to an ActiveScript subclass, or even an ActiveScript instance.

### TaskManager class

The TaskManager class queues Tasks, and executes them in order of priority. The TaskManager runs in RSBot as an ActiveScript, and therefore it can be started and stopped on demand. The class contains various methods for checking the state of the queue and the Task being executed.

### XMLToolBox class

Many sorts of data in PowerGrid are defined in XML files. This class deals with reading and parsing these files to a tree structure, after which the required data can be easily extracted. Since most XML data gets parsed by PowerGrid classes, it is advised that script / plugin developers use these classes instead of directly parsing the XML data.

### UIControls package

#### Noticeboard class

This class deals with operations regarding the Runescape Noticeboard. It can open the different tabs on the Noticeboard, and provide information on the currently opened tab.

#### Quests class

This class loads the status for each quest and stores it, so that information on the status of certain Quests can be retrieved.

#### WidgetManager class

The WidgetManager class manages opening and closing of commonly used widgets. It is PowerGrid’s class that provides access to most of Runescape’s user interface.

## Data package

This package contains data files for PowerGrid. It contains information on where certain special locations such as banks, stores, water sources, mining sites and such are located, and what these positions offer, too. For example, not every mining site has the same types of ore available, so for the sake of selecting the correct destination, it must be known what that destination has to offer.

## Images package

This package contains PowerGrid’s images, logos and icons.

## Model package

### Destination class

This class represents a destination in the Runescape world. A Destination object is basically a mapping from a String to a Point. A list of registered destinations is maintained in this class. Custom destinations can be set and retrieved, allowing the user to provide quick-access to frequently visited places.

### GameObject class

The GameObject class is the superclass of basically any object that exists in the Runescape world. It provides methods for getting its location and type, and is the basic building block for any class representing an object from the Runescape environment.

### Grid class

This class acts as the data structure that the world is saved in. it has an O(1) runtime for get and set operations, meaning that retrieving the object at a certain position, or setting an object at a certain position is instantaneous.

### Item class

This class represents an item in the player’s bank or inventory. It currently acts as a wrapper around the RSItem class, but this will be extended or rewritten in a later version of PowerGrid.

### OutOfReachException class

An OutOfReachException can be thrown whenever PowerGrid fails to reach a certain destination or perform a certain action because of physical restrictions in the Runescape world. Exceptions of this type are for example thrown when the PathFinder cannot compute a path to the given destination; the player cannot reach the object it needs to interact with, and so on.

### Point class

This class is the PowerGrid equivalent of RSBot’s Tile class. In contrast to Tile, the Point class offers many arithmetic operations on points. This makes it easier to perform more complex mathematical computations. Point also contains methods to convert from and to Strings (in the format (x, y) or (x, y, z)), polar coordinates, or Tiles. The Point class implements the Locatable interface, meaning that Points can be used in nearly all RSBot methods that would normally require Tile objects.

### XMLNode class

This class is the main unit for XML trees, as used in the XMLToolBox class. It contains various methods for reading and filtering the contents of the XML tree, as well as producing XML that can be written to a file. XMLNode implements the Iterable<XMLNode> interface, so that it is possible to iterate over the XMLNode’s children using a for-each structure (in Java 7).

### Interact package

#### ComplexInteractable interface

#### Gate class

#### Interactable interface

#### Lodestone class

#### Portal class

#### Shortcut class

#### TeleportItem class

#### Teleportable class

#### Transportable class

#### Handlers package

##### EntranaBoat class

##### IcebergBoat class

##### JatizsoBoat class

##### LunarIsleShip class

##### MiscellaniaShip class

##### MusaPointShip class

##### NeitiznotBoat class

##### PiratesCove class

##### ShiloCart class

##### WaterbirthIsle class

##### Manifold package

###### Canoe class

###### Chartership class

###### FairyRing class

###### GnomeGlider class

###### HotAirBalloon class

###### MagicCarpet class

###### Minecart class

###### SpiritTree class

### World package

#### Door class

#### Enemy class

#### Entity class

#### Person class

#### Wall class

#### Elevator package

##### Elevator class

##### Hatch class

##### Stairs class

#### Resources package

##### AbstractResource class

##### Fishspot class

##### Ore class

##### Tree class

## Plugins package

### Plugin interface

### PluginInfo annotation interface

### PluginLoader class

## Tasks package

### ActiveScriptTask class

This task is essentially a wrapper around an ActiveScript instance. By wrapping the ActiveScript in a Task, the ActiveScript can be executed in the PowerGrid TaskManager. This provides a bridge between native RSBot scripts and PowerGrid tasks.

### RestTask class

This task rests the player until a certain level of energy has been achieved. It can be used to quickly recover energy when walking long distances, or can be run as a task with a low priority to recover energy when PowerGrid is not doing anything else.

### ReturnTask class

This task can run another task, and then returns to the position of the player before the task was executed. It is useful for ensuring that the player is at certain position after a task completes. It basically combines the task with a TravelTask.

### StepTask class

This task works in the same way as an ActiveScript, first executing the start method, then looping the step method until done, and lastly run the finish method. It provides an easy way of creating tasks that consist of multiple, similar steps.

### Task class

This is the basic class for all Tasks. It defines execute and cancel methods, which can be overridden to create custom tasks. Tasks can be assigned a priority that specifies the importance of the Task. Tasks of higher priority will be executed before other, lower priority Tasks in the Task Queue. Tasks can either be run from the TaskManager’s task queue, or manually using the execute method.

### TaskDeque class

This class is not a Task, but can act as a storage container for tasks. Tasks can be added to the beginning and end of the TaskDeque, and any number of Tasks can be executed from this TaskDeque in sequence. Tasks in a TaskDeque can also be moved to the TaskManager’s Task Queue.

### TravelNearestTask class

This task travels to the nearest location matching specific criteria. It can be used to travel to special locations such as banks, mining sites, stores, or basically any other location labeled on the Runescape world map.

### TravelTask class

This task travels to a given point using the Pathfinder to determine the shortest route. It is the standard way of travelling using PowerGrid, and

## View package

### ControlPanel

This panel provides end-user access to PowerGrid’s core functionality. It can be used to show the World Map, show PowerGrid’s status, and show or manipulate the TaskManager’s Task Queue.

### DestinationPanel

This panel shows a list of available destinations, and also offers the possibility to travel to them with the click of a button. It is the common way of using PowerGrid for simply moving from place to place.

### MapViewer

This Canvas shows PowerGrid’s world map. The used colors can be customized through code, and users can click a location on the map to travel there.

### TravelPanel

This panel provides a search box for the DestinationPanel, which allows the user to filter the available destinations.