Things

The cheatsheet

Class Things (int $type)

An object grouping class.
If $type is given, this class will be initialized with all objects of this type inside.
This class can be used without a `$type`, but that limits its function.

- **GetType () -> int**
  Get the type ID of this group. The class will contain objects only of this type.
- **SetType (int $type)**
  Sets the type ID of this group. It will also look for all objects in the database belonging to this group.
- **GetObjects (bool $refresh = false) -> array**
  retrieves all objects already in this group. If $refresh is true, **SetObjects** will be called, retrieving a new list of objects belonging to this group type.
- **GetObjectsTypes () -> array**
  collects the types and associated objects in the format
  ```
  [
    type_id_1 => [obj_id,obj_id,obj_id],
    type_id_2 => [obj_id,obj_id,obj_id], …
  ]
  ```
- **AddObjects (array/int $which)**
  adds $which, an array of object IDs, to the list of objects in this group.
- **SetObjectsRaw (array $ids)**
  replaces all object IDs in this group with $ids, an array of object IDs.
- **SetObjects (string $query_more = 'ORDER BY `type` ASC')**
  Called after **SetType**, this method loads matching objects (of said type) into this group. Access with **GetObjects**.
- **DelObjects (array/int $which)**
  RemoveObjects (array/int $which)
  remove $which, a list of object IDs, from the group's list of objects (if they exist).
- **DelAllObjects ()**
  RemoveAllObjects (mixed $which)
  Empty the list of objects in this group.
  $which is ignored.
- **DelObjectsProps (array $property_names)**
  If an object in this group contains any property name in the array $property_names, those properties will be removed from the object.
- **SetObjectsProps (array $properties)**
  apply given props to all objects in this group.
  Supply $properties in the form of
  ```
  [
    prop_1 => val_1
  ]
  ```
- **FindObject (string $name) -> array**
  returns the IDs of all objects with this name of this group's type, NOT just those in this group.
- **FilterByProp (string $prop, mixed $propval)**
  retains only those objects in the group, whose value for $prop equals to $propval.
- **FilterByProps (array $props, bool $any = false)**
  given $props, an array ('prop_name' => 'value', …), remove all objects in this class that does not meet the criteria, depending on
  - $any = false: if **ALL** properties match, the object is removed
  - $any = true: if **ANY** property matches, the object is removed
- **FilterByPreg (string $prop, string $preg)**
  retains only those objects in the group whose props[$prop] matches a PCRE regular expression.
- **PregReplace (string $prop, string $preg, string $replacement)**
  performs preg_replace on all child objects' property $prop.

Class Thing (int $oid)

Initialises an object of the id $oid.
If supplied $oid is negative, an object of the opposite type will be created. For example, new Thing (-123) will create an object of type 123 (if it exists).
If $oid is positive and the database does not contain an object of that ID, the script will halt.

- **public $oid**
  The object ID.
- `Replaceable` **Create ()**
  Insert the object into the database.
  This is called whenever a Thing is instantiated with a negative type ID.
- **Type (int $type_id = 0) -> int**
  If $type_id is nonzero, the function will set the object's type ID to that of $type_id.
  It will finish by returning its type ID.
- **GetType () -> int**
  returns the object's type ID.
- `Replaceable` **SetType (int $type_id)**
  Sets the type of this object to the one specified.
- **GetPropsRaw () -> array**
  retrieves all properties associated with that object, in table format:
  ```
  [
    [pid => …, oid => …, name => …, value => …],
    [pid => …, oid => …, name => …, value => …]
  ]
  ```
- **GetProps () -> array**
  retrieves all properties associated with that object, in PHP array format:
  ```
  [
    prop_1 => val_1,
    prop_2 => val_2
  ]
  ```
- **GetProp (string $name) -> string**
  retrieve a single value off the **GetProps** array. No speed advantage.
- **SetProp (string $prop, string $val)**
  Similar to **SetProps**.
- **SetProps (array $what) -> SQL resource**
  accepts an array $what ('name'=>'value','name'=>'value') and sets these properties to the object.
  Keys will be changed to lower case.
- **DelProps (array $what)**
  accepts an of names, e.g. array ('views','rating', 'status') and deletes all properties with any of those names.
- **DelPropsAll ()**
  DelAllProps ()
  removes all properties of this object.
- **GetChildren (int $type_id = 0, string $order_by = "`child_oid` ASC") -> array**
  returns all children object IDs associated with this one. You can arrange custom sorting schemes with $order_by, but this has no guaranteed effect.
  If $type_id is supplied, returns only children of that type.
- **SetChildren (array $what) -> mixed**
  appends new parent-child relationships into the hierarchy table.
  $what can be an array of child object IDs `[child1ID, child2ID, ...]`
  Returns can be either a SQL resource (if insertion took place) or a boolean (if no insertion took place).
- **SetChild (int $what) -> mixed**
  Singular form of **SetChildren**, accepting one child ID only.
- **DelChildren (array $child_ids) -> SQL resource**
  removes hierarchical data of some of this object's children, specified in $child_ids.
- **DelChild ($child_id)**
  Singular form of **DelChildren**, accepting one child ID only.
- **GetParents (int $type_id = 0, string $order = "ORDER BY `parent_oid` ASC") -> array**
  returns all parent objects associated with this one.
  You can arrange custom sorting schemes with $order, but this has no guaranteed effect.
  If $type_id is supplied, returns only parents of that type.
- **SetParents (array $what) -> mixed**
  appends new parent-child relationships into the hierarchy table. The object IDs in array $what will be parents of this object.
  Returns an SQL resource or a Boolean on success.
- **DelParents (array $parent_ids) -> SQL resource**
  removes hierarchical data where this object is the parent's child. The parent objects are not removed from the database.
  accepts `[parent1id, parent2id, ...]`
  Returns an SQL resource or a Boolean on success.
- **DelParent (int $parent_id)**
  Singular form of **DelParents**, accepting one parent ID only.
- **DelParentsAll () -> SQL resource**
  DelAllParents ()-> SQL resource
  removes hierarchical data where this object is someone's child. This effectively removes all of the object's parents.
  Parent objects are not removed from the database.
  Returns an SQL resource on success.
- **CreateObject (int $type, array $props = array ()) -> mixed**
  Creates a new object, with this object as the parent of the new one.
  Returns new object's ID on success, or `null` on failure.
- **ChangeID (int $nid) -> true**
  Attempt to change the ID of this object to the new ID, if available.
  This function only returns true. Script halts on failure.
- **Duplicate () -> int**
  creates a data-identical twin of this object. The duplicate will share the same parents and children.
  If any object property has information exceeding the server's limit, it will be created as a reference.
  Returns the dupe's object ID.
- **Destroy ()**
  deletes the object, the relationship with parents, and their children.
  This function needs to be executed with elevated privilege, where the user needs to be either an administrator or a parent of the object.