

adverb conjunction comma present progressive verb preposition definite
article noun preposition plural noun comma

sentence has a surface structure
sentence formed by transformations of a
modality component plus a propositional component
the latter based on a
predicate attended by one or more nouns

exordium

auxiliares

(hedge & bet circular binomial
as deflect, (ly) linear time frame)

estivate and xeroxed;
an economy of presence

vitamin c
vitamin e

b complex

hi — many apologies for this slow response, I didn't see any notifications so hadn't checked my mail, maybe email is better? ... very glad to hear that you'll submit something & many thanks for the kind words. If you'd like to discuss a project let

me know.

I think about corrosion sometimes. Automation, how to be absented. What about you?

hero track

Statements do not return values, but some kinds of statements can, under certain circumstances described below, generate errors.

The simplest kind of statement is the "null" statement, consisting of just a semicolon:

```
;
```

It doesn't do anything at all, but it does it very quickly.

gammaaminobutyric acid concentrated
periods of activity

@chown <object>=<player>. Changes the ownership of an object.

@create <name> [=<cost>]. Creates a thing with the specified name.

@describe <object> [=<description>].

@dig <name>. Creates a new room
@fail <object> [=<message>].
@find [name]. Displays the name and number ... whose name matches <name>.
@link <object>=<number>; @link <object>=here; @link <dir>|<room>=home.
@lock <object>=<key>.
@name <object>=<new name> [<password>]. Changes the name of <object>.
@ofail <object> [=<message>].
@open <dir>[;<other dir>]* [=<number>].
@osuccess <object> [=<message>].
@set <object>=<flag>; @set <object>=!<flag>. Sets (or, with '!', unsets)
@success <object> [=<message>].

and sulphur-coated authors
appropriating steady print,
emphasizing vicarious
qua prescriptive pulp
and digestion, fallen
forms reduce the growing,

@chown <object>=<player>. Changes the ownership of an object.
@create <name> [=<cost>]. Creates a thing with the specified name.
@describe <object> [=<description>].
@dig <name>. Creates a new room
@fail <object> [=<message>].
@find [name]. Displays the name and number ... whose name matches

<name>.
@link <object>=<number>; @link <object>=here; @link <dir>|<room>=home.
@lock <object>=<key>.
@name <object>=<new name> [<password>]. Changes the name of <object>.
@ofail <object> [=<message>].
@open <dir>[;<other dir>]* [=<number>].
@osuccess <object> [=<message>].
@set <object>=<flag>; @set <object>=!<flag>. Sets (or, with '!', unsets)

E_NONE No error

VERB
VERB DIRECT-OBJECT
VERB DIRECT-OBJECT PREPOSITION INDIRECT-OBJECT
say emote eval

the direct object, if any, and

the indirect object, if any.

EXPRESSION-1 ? EXPRESSION-2 | EXPRESSION-1

valent to the
conditional expression

EXPRESSION-1 ? EXPRESSION-1 | EXPRESSION-2

is performed before any movement takes place. If WHERE does not define an `accept' verb, then it is treated as if it defined one that always returned false.

The `location' property of WHAT is changed to be WHERE, and the `contents' properties of the old and new locations are modified appropriately. Let OLD-WHERE be the location of WHAT before it was moved. If OLD-WHERE is a valid object, then the verb-call.

Whenever the `create()' function is used to create a new object, that object's `initialize' verb, if any, is called with no arguments. The call is simply skipped if no such verb is defined on the object.

Symmetrically, just before the `recycle()' function actually destroys an object, the object's `recycle' verb, if any, is called with no arguments. Again, the call is simply skipped if no such verb is defined on the object.

Both `create()' and `recycle()' check for the existence of an `ownership_quota' property on the owner of the newly-created or -destroyed object. If such a property exists and its value is an integer, then it is treated as a "quota" on object ownership. Otherwise, the following two paragraphs do not apply.

Another representational problem concerns compositionality. In this instance, the linguistic practices found on MUD metaphorically mimic social practices. The tense repetitive action is analogous to the twitching of muscle tissue.

Each action is a compound of many contractions and relaxations of muscles, and movements of limbs

Erikson, Erik H. Childhood and Society. New York: W. W. Norton & Company, 1985.

Foucault, Michel. Power, Truth, Strategy. Eds. Meaghan Morris and Paul Patton. Sydney: Feral, 1979.

Randomness says "Finally, I added hours and noticed that no
matter what I built, it went a way. I gave up."
Randomness grins at uwiz. "yeah, exactly."
Randomness says "Now I just build myself, and sleep ."

Randomness says "I wanted to show off stuff at the
beginning."
Dragon quotes something unfamiliar

Figure 2: A selection of scene from MUD adventure

```
                You are standing at the end of a road before a  
                small building.[2]  
  
            endif  
next  
next  
  
echo Your mud has been backed up to file:  
echo  
echo ${Back_Dir}/${date}.mud.tar.gz  
echo  
echo Backup Complete.  
echo Mud Backup
```

You fail to use a thing when you cannot hear it (because it's lock fails).
You fail to use an exit when you cannot go through it (because it's unlinked
or locked). You fail to use a person when you fail to hear them. You fail to
use a room when you fail to look around (because it's locked). See STRINGS,
and in the dictionary, @FAIL and @FAIL.

bandwidth in the silent sound system,
so many sour tastes

prevent the absorption of essential nutrients;

Electrolyte imbalances Nutrient deficiencies if not supplemented
Water intake 3.7 l or 125 oz (essentially a single gallon) acre 0.000003 barrel
(petroleum) 0.02327 fluid dram 1,001 (essentially a single gallon) Self-efficacy -
Outcome expectancy - Environmental factors

{"Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"}

* lists (of all of the above, including lists)

325.0 325. 3.25e2 0.325E3 325.E1 .0325e+4 32500e-2

VERB

VERB DIRECT-OBJECT

VERB DIRECT-OBJECT PREPOSITION INDIRECT-OBJECT

say emote eval

These operators are usually read as "and" and "or," respectively.

EXPRESSION-1 ? EXPRESSION-2 | EXPRESSION-1

EXPRESSION-1 ? EXPRESSION-1 | EXPRESSION-2

These two operators behave very much like "and" and "or" in English:

| | | |
|--------|---|---|
| 1 && 1 | → | 1 |
| 0 && 1 | → | 0 |
| 0 && 0 | → | 0 |
| 1 1 | → | 1 |
| 0 1 | → | 1 |
| 0 0 | → | 0 |

If WHERE is a valid object, then the verb-call

WHERE:accept(WHAT)

If moving WHAT into WHERE would create a loop in the containment hierarchy (i.e., WHAT would contain itself, even indirectly), then '_REMOVE' instead.

The `location' property of WHAT is changed to be WHERE, and the `contents' properties of the old and new locations are modified appropriately. Let OLD-WHERE be the location of WHAT before it was moved. If OLD-WHERE is a valid object, then the verb-call

OLD-WHERE:exitfunc(WHAT)

is performed and its result is ignored; it is not an error if OLD-WHERE does not define a verb named `exitfunc'. Finally, if WHERE and WHAT are still valid objects, and WHERE is still the location of WHAT, then the verb-call

WHERE:enterfunc(WHAT)

- Function: none delete_property (obj OBJECT, str PROP-NAME)
Removes the property named PROP-NAME from the given OBJECT and all of its descendants.

If OBJECT is not valid, then does not have write permission on OBJECT, If OBJECT does not directly define a property named PROP-NAME (as opposed to inheriting one from its parent)

- Function: int is_clear_property (obj OBJECT, str PROP-NAME)
- Function: none clear_property (obj OBJECT, str PROP-NAME)

- Function: list verb_args (obj OBJECT, str VERB-DESC)

These two functions get and set (respectively) the direct-object, preposition, and indirect-object specifications for the verb as specified

{DOBJ, PREP, IOBJ}

where DOBJ and IOBJ are strings drawn from the set ``"this"`, `"none"`, and `"any"`, and PREP is a string that is either `"none"`, `"any"`, or one`

- Function: list set_verb_code (obj OBJECT, str VERB-DESC, list CODE)
These functions get and set (respectively)

- Function: list disassemble (obj OBJECT, str VERB-DESC)
Returns a (longish) list of strings giving a listing of the server's internal "compiled" form of the verb as specified by VERB-DESC on OBJECT. This format is not documented and may indeed change from release to release

`"client-echo"

Send `WONT ECHO' or `WILL ECHO' command, depending on whether VALUE is true or false, respectively.

`"binary"

If VALUE is true, then both input from and output to can contain arbitrary connections. Input from a connection in binary mode is not broken into lines at all; it is delivered to either the read function or the built-in command parser as "binary strings", in whatever comes back from the operating system.

Whenever the ``create()`` function is used to create a new object, that object's ``initialize`` verb, if any, is called with no arguments. The call is simply skipped if no such verb is defined on the object.

Symmetrically, just before the ``recycle()`` function actually destroys an object, the object's ``recycle`` verb, if any, is called with no arguments. Again, the call is simply skipped if no such verb is defined on the object.

Both ``create()`` and ``recycle()`` check for the existence of an ``ownership_quota`` property on the owner of the newly-created or -destroyed object. If such a property exists and its value is an integer, then it is treated as a "quota" on object ownership.

The ``create()`` function checks whether or not the quota is positive; if so, it is reduced by one and stored back into the ``ownership_quota`` property on the owner. If the quota is zero or negative, the quota is considered to be exhausted and ``create()`` raises ``E_QUOTA``.

The ``recycle()`` function increases the quota by one and stores it back into the ``ownership_quota`` property on the owner.

Appendix B: Further Examples

Randomness grins "People have no loyalties to a server, and don't build massively anymore, because they know that servers are ephemeral.
Randomness never builds *anything* anymore. "Only what I can carry. It's like being a refugee." He grins.
Garrett builds.
Randomness hasn't the time.
Randomness used to build his home and stuff. "I was unemployed, then."
Randomness says "Then, I started working part time, and I built just my home."
Randomness says "Finally, I added hours and noticed that no matter what I built, it went a way. I gave up."

```
next
next
endproc
```

```
echo Your mud has been backed up to file:
echo
echo ${Back_Dir}/${shdate}.mud.tar.gz
echo
echo Backup Complete.
echo Mud Backup
```

FAILURE

You fail to use a thing when you cannot take it (because it's lock fails).
You fail to use an exit when you cannot go through it (because it's unlinked or locked). You fail to use a person when you fail to rob them. You fail to use a room when you fail to look around (because it's locked). See STRINGS

and in the dictionary, @FAIL and @FAIL

@chown <object>=<player>. Changes the ownership of an object.
@create <name> [=<cost>]. Creates a thing with the specified name.
@describe <object> [=<description>].
@dig <name>. Creates a new room
@fail <object> [=<message>].
@find [name]. Displays the name and number ... whose name matches <name>.
@link <object>=<number>; @link <object>=here; @link <dir>|<room>=home.
@lock <object>=<key>.
@name <object>=<new name> [<password>]. Changes the name of <object>.
@ofail <object> [=<message>].
@open <dir>[;<other dir>]* [=<number>].
@osuccess <object> [=<message>].
@set <object>=<flag>; @set <object>=!<flag>. Sets (or, with '!', unsets)
@success <object> [=<message>].
@unlink <dir>; @unlink here.
@unlock <object>. Removes the lock on <object>.

Excedrin headache
(O blue
blood, O 10W40, O lubricated with nonoxynol, O engine engine #9;