

Data networks

From Starbase wiki



Contents

Summary

Terminology

Data network resources

Device fields

Field type

Value propagation

Summary

Data networks enable the creation, maintenance and control of large number of devices. Universal tool and cable tool are the tools generally used for managing data networks.

Data networks enable designers and pioneers to create, modify and maintain devices, such as power generators, programmable logic chips and more mundane devices such as doors and elevators.

The signals are controlled with programming language YOLOL that is accessible to everyone.

Terminology

Objects have physical **sockets** and virtual **device fields** inside them.

- **Sockets** are used to connect objects to data networks with cables or directly through other devices.
- **Device fields** are configured and accessed when objects share the same data network.
 - A device field controls one specific attribute or feature within a device.
 - The device field can for example be the amount of generated heat within a thruster, or a value that can be manually changed such as button states in buttons.
 - These device field names can generally be configured by using the appropriate tool.

Data network resources

Data networks are connected together with cables.

When devices are connected together, the data network can transfer two different resources.

1. **Electricity**
2. **Signals**

Electricity: all objects within the data network share power

- A generator attached to a data network tries to supply power to multiple devices or machines.
 - This is why it is important to have multiple generators if there are many devices connected to the data network that need power.

Signals: all objects within the data network are able to communicate with each other

- Cables can connect to each other directly.
- Sockets may have multiple connections and cables may branch.
- There are no limits to how many data networks can exist e.g. on one spaceship.
 - A branched data network doesn't break as easily during combat.

Device fields

Each device can have a set of device fields. This fields consist of a name and a value and both is customizable. Any field on the network can have any name you wish, "conflicting" names are allowed too, in fact sometimes they are used to build a network. Renaming a device field does not changes how the device manages it, swapping the name of two device fields' name within the same device does **not** actually swaps the two field. The devices' manages their field by their **position**, not by their name, e.g.: A box thruster's first field will be always the control field regardless of it's name, while the second field will always be the feedback.

Field type

These fields can be of 3 type: input, output or IO (input + output).

- **Input** fields are purely for controlling the device. The device itself does not changes the value of an input field but changes it's way of operation e.g.: ThrusterPowerLevel. The thruster increases or decreases it's current thrust based on what you set the ThrusterPowerLevel field to, but on it's own it will never change the value of the field.
- **Output** fields are the exact opposite. No matter what you write into these fields it will never affect how the device operates. Output fields are only for monitoring the devices' states. The devices periodically writes values into their output fields overwriting whatever was in there previously. An example is the "CurrentThrust" field of the thruster, you can not modify the thruster with that field, and whenever the thruster changes it's thrust level, it overwrites this field with the new value,.
- IO fields are the combination of both input and output. You can manually write a value in them and that will change the operation of the device but on the other hand some events within the device (e.g.: pulling a lever) will cause the device to also write a new value into the field.

Value propagation

Whenever a device changes its own fields' values, the change is propagated through the network. Every other device will update their fields of the same name to the new value. Updating the field through the Universal tool does not cause a change propagation but manual events (e.g.: pulling a lever) do. The propagation affects both input and output fields of every device on the same network. Thus you don't need Yolol to connect 2 devices, just name one device's input or IO field to the same name as another device's output or IO field. Yolol chips cause value propagation when you set a global name (e.g.: `:GlobalName = value`), but keep in mind that if the name does not exist on the network then the yolol line will fail. The previous yolol code is valid if there is at least one device that has a field with the given name (without the ':' prefix) or if there is a yolol memory chip with an entry with the given name.

Retrieved from "https://wiki.starbasegame.com/index.php?title=Data_networks&oldid=29043"

Category:

- Networks