

PARTs NetworkTables

Explanation & History

Sending and getting data from the dashboard and other network devices requires a lot of tedious work which clutters up our code as we work and test. This utility of the library fixes that issue by making the process extremely easy. Not only does the utility make it easy to send and receive data through the robots network, it also optimizes the data it handles. This is important when multiple subsystems are all sending data periodically.

Code Example

```
package;  
  
import org.parts3492.partslib.network.PARTsNT;  
  
public class MyClass {  
  
    // The PARTsNT instance for this class.  
    PARTsNT partsNT;  
  
    // Flag to allow sending and receiving the real network data.  
    // This is useful for conditionally enabling network access. E.g. sending data  
    // to the dashboard only in debug mode.  
    boolean allowNetworkAccess = true;  
  
    // My cool boolean.  
    boolean myBoolean = true;  
  
    public MyClass() {  
        // Initialize the PARTsNT instance for this class.  
        partsNT = new PARTsNT("MyClass");  
  
        // Send myBoolean to the network. This will create a new entry in the network  
        // table with the key "MyClass/MyBoolean" and the value of myBoolean.  
        partsNT.putBoolean("MyBoolean", myBoolean, allowNetworkAccess);  
    }  
}
```

```
// Get modified myBoolean from the network. This will return the value of the
// entry with the key "MyClass/MyBoolean" if it exists, or the default value of
// myBoolean if it doesn't exist. This allows us to modify myBoolean from the
// network and have it update in our code.
myBoolean = getMyNetworkBoolean();
}

// Method to get myBoolean from the network.
public boolean getMyNetworkBoolean() {
    return partsNT.getBoolean("MyBoolean", allowNetworkAccess);
}
}
```

Revision #1

Created 2 April 2026 21:03:37 by Rowan Buchecker

Updated 2 April 2026 21:20:59 by Rowan Buchecker