# Bailey DCS Serial Communication Protocol

The Bailey driver implements communication with INFI 90 Distributed Control Systems via Serial communication. It operates as a Master on serial networks. Only the pooling commands are enabled for this driver.

## Summary Information

**Communication Driver Name:** Bailey

**Implementation DLL:** T.ProtocolDriver.Bailey

**Protocol:** Proprietary

**Interface:** Serial

**Supported Types:**

| Type | Read | Write | Description |
|---|---|---|---|
| AIL | Yes | Yes | Analog |
| DD | Yes | Yes | Device Driver |
| DI | Yes | Yes | Digital |
| MSDD | Yes | Yes | Multistate Device Driver |
| RCM | Yes | Yes | Remote Control Memory |
| RMCB | Yes | Yes | Remote Motor Control |
| RMSC | Yes | Yes | Remote Manual Set Constant |
| STN | Yes | Yes | Control Station |
| TEXT | Yes | Yes | Text Selector |
| BLK | Yes | Yes | Block Any Function Code |

---

## Channel Configuration

### Settings

Set the fields according to the device's serial port configuration

---

## Node Configuration

### Station Configuration

The syntax for the Bailey Station field is: <Ring>.<Node>.<Module>

Where:

- **<Ring>** indicates the Ring where the block is
- **<Node>** indicates the Node where the block is
- **<Module>** indicates the Module where the block is

E.g: 1.35.2

---

## Point Address Configuration

The syntax for the Bailey communication point is: <Block>:<Type>:<Value>

Where :

- **<Block>** indicates the Block that wants to communicate
- **<Type>** indicates the Function Code type. The valid values are:

| Type | Description |
|---|---|
| AIL | Analog |
| DD | Device Driver |
| DI | Digital |
| MSDD | Multistate Device Driver |
| RCM | Remote Control Memory |
| RMCB | Remote Motor Control |
| RMSC | Remote Manual Set Constant |
| STN | Control Station |
| TEXT | Text Selector |
| BLK | Block Any Function Code |

- **<Value>** indicates each Read or Write item

  E.g.:

  - 1363:DD:Outuput
  - 1632:STN:Spec4

---

## Troubleshoot

The status of the driver's execution can be observed through the diagnostic tools, which are:

• Trace window
• Property Watch
• Module Information

The above tools indicate if the operations have succeeded or have failed. A status of 0 (zero) means communication is successful. Negative values indicate internal driver errors, and positive values indicate protocol error codes.