

Using XML Command Scripts

The Cryo-con Utility software package can be used to send configuration scripts to any Cryo-con instrument. These scripts consist mostly of standard remote commands and queries.

Scripts can be used to completely configure an instrument including setting custom sensor calibration curves and PID tables. They are commonly used in a manufacturing environment to set a baseline state for a target product. In the laboratory, scripts can be used to save and restore configurations for various experiments.

XML, or Extensible Markup Language, is used for the structure and format of script files. XML can be generated and edited with a standard text editor but advanced users may want to use one of the commonly available XML editors. Since it provides a structure and allows user documentation, it is easy to read and understand.

Configuration scripts have a file extension of .xml. These files are sent to an instrument by using the Operations->Send Command File function of the Cryo-con Utility Software.

Any remote command or query that is recognized by the instrument can be used in a script file. This includes commands that read and write user sensor calibration curves and PID tables. A complete description of available remote commands is given in the chapter titled Remote Programming Guide. The Remote Command Tree section is particularly useful for the advanced user.

Script File Structure

Header and Footer

Like all XML files, script files have the following header and footer:

```
<?xml version="1.0"?>
<Transactions>
.
.
.
</Transactions>
```

All user supplied information is placed between the Transactions tags.

Basic XML Tags

Comment: <! >

Inserts a comment in the file for documentation and readability. The comment within the angle brackets after the exclamation is ignored by the software.

```
<!Download User Curve 4>
<!----- Loop 1 ----->
```

Model: <Model> </Model>

Contains the Crycon instrument model number for source/destination verification.

```
<Model>Model24 Version 2.03</Model>
```

Remote Command: <Command> </Command>

Send a remote command to the instrument. Commands can be any of the instrument's commands as described in the Remote Programming Guide.

```
<Command>input c:sensor 2</Command>  
<Command>LOOP 1:SOURCE A;Setpt 20.0</Command>  
<Command>OVERTEMP:ENABLE ON</Command>
```

Query: <Query> </Query>

Query data from the instrument. Queries can be any of the instrument's commands as described in the Remote Programming Guide. Query is generally used with a Response tag to compare the instrument's response to an expected value. If there is no Response tag, the result of the query is printed but not tested for errors.

```
<Query>input c:sensor?</Query>  
<Query>input b:units K;units?</Query>
```

Response: <Response> </Response>

Identifies the expected response to a query. This tag must always follow a Query tag, otherwise, it is ignored. When the comparison fails, an error text message will be displayed and recorded to a file.

```
<Query>Relays? 0</Query>  
<Response>Lo</Response>
```

```
<Query>input c:units?</Query>  
<Response>K</Response>    <! Should be Kelvin. Error if not >
```

Floating Point Response: <Floatresponse> </Floatresponse>

Compare the response returned from the instrument against an expected floating point number. This tag must always follow a Query tag; otherwise, it is ignored. When the comparison fails, an error text message will display. The returned value passes the test if within +/-2.5% of the expected value.

```
<Query>input a:ALAR:High?</Query>  
<FloatResponse>200.000000</FloatResponse>
```

Pause: <Pause> </Pause>

Provide a pause for a specified number of milliseconds to allow the instrument to react to a command. Maximum 20 seconds. Generally, this is only used with the RS-232 serial interface where there is no hardware handshake.

```
<Pause>1000</Pause> <! Delay 1 second >
```

Group Tags

Any tag that is not defined is treated as a group tag. They are used to provide structure and enhance readability. Otherwise, they are ignored.

Complex Tags

Sending a user sensor calibration curve or a PID table to an instrument requires a complex tag because it can require many lines of data.

User Sensor Calibration Curve: <Calcur>

Send a sensor calibration curve to the instrument.

```
<!Download User curve 4>
  <CalCur>Calcur 4</CalCur>
  <!--Curve Name-->
    <CalCur>My Sensor</CalCur>
  <!--Curve Type-->
    <CalCur>Diode</CalCur>
  <!--Multiplier-->
    <CalCur>-1.000000</CalCur>
  <!--Units-->
    <CalCur>Volts</CalCur>
  <!--Curve Entries-->
    <CalCur>0.163300  475.000000</CalCur>
    <CalCur>0.173300  470.000000</CalCur>
    <CalCur>0.183400  465.000000</CalCur>
    <CalCur>1.866000   1.500000</CalCur>
  <!--Send the terminator character-->
  <CalCur>;</CalCur>
```

Transmission of the calibration curve starts with the first CALCUR tag and ends when the terminator character is sent. Comments are ignored.

PID tables are sent to the instrument by using standard command tags. For example:

```
<PIDtable> <!--Group tag is for documentation only-->
  <Command>PIDTABLE 0:TABLE</Command>
  <Command>PID table 1</Command>
  <Command>320.00 2 10 1 LOW</Command>
  <Command>300.00 2 10 1 LOW</Command>
  <Command>150.00 2 10 1 LOW</Command>
  <Command>55.00 2 10 1 LOW</Command>
  <Command>40.00 2 10 1 MID</Command>
  <Command>30.00 2 10 1 MID</Command>
  <Command>25.00 2 8.5 1 MID</Command>
  <Command>20.00 2 4 1 MID</Command>
  <Command>15.00 2 3 1 MID</Command>
  <Command>10.00 2 2 1 MID</Command>
  <!--Send terminator-->
  <Command>;</Command>
</PIDtable>
```

Script File Example

```
<?xml version="1.0"?>
<Transactions>
  <Model>Model44 Version 3.06</Model>
  <Input>
    <!-- CHA -->
    <Command>input a:sensor 20</Command> <!-- Set to PT100>
    <Query>input a:temp?</Query>
    <Command>input a:sensor 21</Command> <!-- Set to PT1K>
    <Query>input a:temp?</Query> <!-- Ignore response>

    <!-- CHB -->
    <Command>input b:sensor 20</Command>
    <Query>input b:temp?</Query>
    <Response>K</Response>
    <Command>input b:sensor 21</Command>
    <Query>input b:temp?</Query>
  </Input>
  <Loop>
    <!-- Loop 1 -->
    <Command>Loop 1:SetPt 250</Command>
    <Command>Loop 1:Type MAN</Command>
    <Query>Loop 1:Type?</Query>
    <Command>Loop 1:Pman 20</Command>
    <Query>Loop 1:Pman?</Query>
    <Command>control</Command>
    <Query>Loop 1:Outp?</Query>
    <Command>Stop</Command>
  </Loop>
  <PIDTable>
    <!-- Download to table 6 -->
    <Command>PIDTABLE 5:table</Command>
    <!-- Table Name -->
    <Command>LOOP1 Htr</Command>
    <!-- Table Entries -->
    <Command>310.00 1.60 160.00 40.00 HI</Command>
    <Command>280.00 1.50 150.00 30.00 HI</Command>
    <Command>260.00 1.40 140.00 30.00 HI</Command>
    <Command>240.00 1.30 130.00 30.00 HI</Command>
    <Command>220.00 1.20 120.00 30.00 HI</Command>
    <Command>200.00 1.10 110.00 20.00 HI</Command>
    <Command>180.00 1.00 100.00 20.00 MID</Command>
    <Command>160.00 0.90 90.00 20.00 MID</Command>
    <Command>140.00 0.80 80.00 20.00 MID</Command>
    <Command>120.00 0.70 70.00 10.00 MID</Command>
    <Command>100.00 0.60 60.00 10.00 MID</Command>
```

```
<Command>80.00 0.50 50.00 10.00 MID</Command>
<Command>60.00 0.40 40.00 10.00 LOW</Command>
<Command>40.00 0.30 30.00 0.00 LOW</Command>
<Command>20.00 0.20 20.00 0.00 LOW</Command>
<Command>;</Command>
```

```
</PIDTable>
```

```
<SensorCurve>
```

```
<!User curve 4>
```

```
<CalCur>CALCUR 4</CalCur>
```

```
<!Curve Name>
```

```
<CalCur>Test S700</CalCur>
```

```
<!Curve Type>
```

```
<CalCur>Diode</CalCur>
```

```
<!Multiplier>
```

```
<CalCur>-1.000000</CalCur>
```

```
<!Unit>
```

```
<CalCur>Volts</CalCur>
```

```
<!Curve Entries>
```

```
<CalCur>0.163300 475.000000</CalCur>
```

```
<CalCur>0.173300 470.000000</CalCur>
```

```
<CalCur>0.183400 465.000000</CalCur>
```

```
<CalCur>0.193500 460.000000</CalCur>
```

```
<CalCur>0.203800 455.000000</CalCur>
```

```
<CalCur>0.214100 450.000000</CalCur>
```

```
<CalCur>0.224600 445.000000</CalCur>
```

```
<CalCur>0.235100 440.000000</CalCur>
```

```
<CalCur>0.245800 435.000000</CalCur>
```

```
<CalCur>0.256500 430.000000</CalCur>
```

```
<CalCur>0.267300 425.000000</CalCur>
```

```
<CalCur>0.278100 420.000000</CalCur>
```

```
<CalCur>0.289100 415.000000</CalCur>
```

```
<CalCur>0.300100 410.000000</CalCur>
```

```
<CalCur>0.311100 405.000000</CalCur>
```

```
<CalCur>0.322200 400.000000</CalCur>
```

```
<CalCur>0.333400 395.000000</CalCur>
```

```
<CalCur>0.344600 390.000000</CalCur>
```

```
<CalCur>0.355800 385.000000</CalCur>
```

```
<CalCur>0.367100 380.000000</CalCur>
```

```
<CalCur>0.378400 375.000000</CalCur>
```

```
<CalCur>0.389700 370.000000</CalCur>
```

```
<CalCur>0.401100 365.000000</CalCur>
```

```
<CalCur>0.412500 360.000000</CalCur>
```

```
<CalCur>0.423900 355.000000</CalCur>
```

```
<CalCur>0.435300 350.000000</CalCur>
```

```
<CalCur>;</CalCur>
```

```
</SensorCurve>
```

```
</Transactions>
```



Cryogenic Control Systems, Inc.
PO Box 7012
Rancho Santa Fe, CA 92067-7012

Telephone: 858 756 3900
FAX: 858 759 3515

techsupport@cryocon.com
www.cryocon.com