

NAME

changes – highlights of modifications for spec release 5.10

September 26, 2011 – RELEASE 5.10.01-10**Fix Of Bug In Previous Release That Leads To an Early Exit**

This minor update fixes a bug that was put into the previous release and that would cause spec to terminate during initial hardware configuration if there was a position discrepancy that was resolved in favor of the motor controller.

September 23, 2011 – RELEASE 5.10.01-9**Fix For `eval()` Return Of Associative Array**

The stricter validation for assignment of associative arrays implemented in spec release 5.09.2-2 disabled the ability to return associative arrays from the `eval()` function, which also broke the behavior when using the `remote_eval()` function with a spec server. That problem has been fixed.

Improvements For "Commanded" Position, Reversion Of `mvr` Macros

Several instances where the value for the last-commanded position of a motor was incorrect have been fixed. (The last-commanded position is available with the new `move_info()` function and a new option to `read_motors()`, both introduced in spec release 5.10.01-1. The last-commanded position saves the requested target position to the fully specified resolution, while the usual motor position is rounded to the motor step size. Using the last-commanded position to calculate target positions in relative moves allows one to perform successive relative moves in increments less than the motor resolution.) The last-commanded position will now be set to the current motor position on a fresh start (with the `-f` flag), when leaving simulate mode, and after correcting a position discrepancy where the software position or user offset was adjusted to match the hardware. Although the updated `mvr` and `umvr` included in release 5.10.01-1 should work fine for most users most of the time, the previous definitions have been restored for the time being while CSS works to make sure the last-commanded position feature is as robust as possible. The definitions that use the last-commanded position are still available as `mvr2` and `umvr2`.

September 15, 2011 – RELEASE 5.10.01-8**String Values Now Available For `epics_get()` Character Arrays**

When reading arrays of `DBF_CHAR` with `epics_get()`, if the optional argument that specifies the desired type of the return value is specified as "string", spec will now return an ordinary string. Previously, the function would return a single-row string data array. See the `epics` help file.

September 10, 2011 – RELEASE 5.10.01-7**Fix For Canberra Lynx MCA Issues After `^C`**

A problem with the Canberra Lynx MCA, where socket communications could become disrupted if a transaction was interrupted by a keyboard `^C`, has been fixed.

Fix For State File Locking On Mac OS X

Normally, spec prevents the same user from starting multiple instances of the same spec version. A problem on Mac OS X, where the locking mechanism didn't always work correctly with more than ten terminal windows open, has been fixed.

Fix For GPIB Sharing With Built-in NI GPIB Support

A problem where the sharing feature for GPIB controllers wasn't working with the built-in support for some of the National Instruments GPIB models has been fixed.

September 4, 2011 – RELEASE 5.10.01-6**Prompt Now Indicates Simulate Mode**

The spec prompt will now contain text to indicate when simulate mode is active, as in
123.FOURC_sim> .

Mu Pseudomotor Available For Six-Circle Geometry

The six-circle geometry, *sixc*, now recognizes a configuration where the mu rotation involves two real motors, muT and muR. See the *sixc* help file.

Fix For Recently Created Pseudomotor Issue

An oversight in spec release 5.10.01-3 that removed the ability to use `motor_par()` to retrieve arbitrarily configured unit and channel numbers for motors with the NONE controller type has been remedied.

Fix For EPICS Debugging Messages

A bug, where the values printed at debug level 128 for the `epics_put()` command were incorrect for non DBF_DOUBLE number-valued process variables, has been fixed. The bug only affected the debugging output, not the values sent through the EPICS channel access calls.

Update For Attocube Controller

The Attocube ANC350 controller support has been updated to work better with the firmware problem where the controller can take hundreds of milliseconds to report a motor is busy after spec sends a move command.

Fix For Number Of Channels For Huber SMC 9300 Controller

A bug, introduced in spec release 5.09.01-3 and associated with the update to the Huber SMC 9300 support for simultaneous motor starts, where the highest numbered motor channel wouldn't move unless the number of motor channels configured was one more than needed, has been fixed.

August 12, 2011 – RELEASE 5.10.01-5**Fix For Motor Sync Issue With spec Client**

A bug, where a ^C on a spec client could be ignored during a move of a motor on a spec server, has been fixed. The issue only occurred if the server had sent the client a "sync_check" message that initiated a motor discrepancy dialog on the client. (The bug was related to improper settings of the keyboard-input modes after the discrepancy dialog.)

Fix For Setting Certain MCA Parameters On Unresponsive Unit

The `mca_par()` commands "auto_clear", "auto_run" and "soft_preset" can now be used to set those modes on MCA devices that are configured, but not responsive. In particular, for the EPICS MCA (which isn't flagged as responsive until all the registered connection events arrive), the `mca_par()` commands can now be used immediately after reading the *config* file. For example, the commands can be included in the definition of `config_mac`, which will generally execute before the EPICS MCA is fully connected.

Fix For EPICS MCA Polling During `wait()`

A bug, where EPICS events were not processed during a call of `wait()` or `wait(0)` if the only EPICS device that was busy was the EPICS MCA, has been fixed.

Fix For TANGO Commands

Implementation of data-array type conversions for `tango_put()` input arguments has been completed. Previously, if the type of the data array passed to `tango_put()` didn't match TANGO's expected data type, incorrect data could be sent. Also, a bug with the `tango_io()` and `tango_put()` functions with 64-bit spec builds, where incorrect values were sent when converting elements of an input associative array to TANGO long (32-bit) data types, has been fixed.

ser_par() "queue" Implemented For TACO Serial Devices

The `ser_par()` "queue" option is now implemented for the TACO serial device server.

July 1, 2011 – RELEASE 5.10.01-4

Fix For OMS Position Maintenance With Negative Encoder Ratio

A problem, where the Oregon Micro Systems position-maintenance mode would not work if the encoder-step-size parameter had the opposite sign of the standard step-size parameter, has been fixed.

June 22, 2011 – RELEASE 5.10.01-3

Fix For Calculational Pseudomotors In Server Mode

A problem, where `spec` would get stuck waiting for a move to finish in the seldom encountered configuration of running in server mode with a calculational pseudomotor (using the macro hardware feature) having an associated real motor configured as controller type `NONE`, has been fixed.

June 20, 2011 – RELEASE 5.10.01-2

Fix For GPIB-ENET Issues On ^C

An issue with some built-in hardware controlled over GPIB using the National Instruments GPIB-ENET interface, where a `^C` interrupt would lead to a breakdown in communications with the GPIB-ENET, has been fixed.

June 6, 2011 – RELEASE 5.10.01-1

New `move_info()` Function

A new `move_info()` function returns information about what would happen on a subsequent `move_all` command given the current motor positions and current values in the `A[]` array. The information returned can include a list of motors that would be moved. See the new `move_info` help page for details.

New "Commanded" Motor Position Value Available With `read_motors()`

`spec` now retains the value of the commanded motor position passed to the move code via the motor position `A[]` array. The value is retained to the full precision specified, which can be higher precision than the normal motor positions, which are rounded to the step size of the motor. A new option to the built-in `read_motors()` function will return this commanded position. If called as `read_motors(0x10)`, the commanded positions for all motors will be placed in the `A[]` array. If called as `read_motors(0x10, mne)`, the function will return the commanded position for motor `mne`, leaving the `A[]` array unchanged. The commanded position is set to the current (rounded) position on start up, after hardware reconfiguration, at the end of a homing operation, after hitting a limit, with a `chg_offset()` command (called by the `set` macro) and when a move is aborted by `^C` or an emergency stop.

Revised `_mvr` Macro Uses Commanded Position

The standard `_mvr` macro (used by `mvr` and `umvr`) has been updated to use the new commanded-position return value of `read_motors()` to calculate the target position of the relative move.

New `encode()` and `decode()` Functions

New built-in functions `encode()` and `decode()` are available to convert between `spec` data types and data-serialization formats to aid in exchanging data with other processes. See the new `encode` help file for details.

New Macro Hardware MCA Functionality

The macro hardware feature now includes support for MCA devices, in addition to the existing support for motors and counter/timers. See the `mac_hdw` help file for details.

Restored Auto-Raise Functionality to X Window Plots

The focus-stealing-prevention functionality that has appeared in window managers over the last years broke the auto-raise feature of `spec`'s X Window plots. With the auto-raise feature enabled, hidden or partially hidden plot windows rise to the top when `spec` updates the plot. This release restores the auto-raise functionality. See `spec`'s `x11` help file for details on setting plot window options.

Fix For Early Read Of Counters In Server Mode

In server mode, while waiting for command input, `spec` will periodically issue calls to read the active scalers and cache the values in order to be able to respond to client requests for scaler values without needing to access the hardware. Previously, as part of this procedure, the counters would be read immediately after starting the time count interval. Now, that first read won't occur immediately. The change will be noticed particularly in the sequence of calls to macro counter `_cmd()` functions.

Enhanced Support For 2D Associative Arrays With `var in` Syntax

The syntax related to querying 2D associative arrays with a known second element, as in

```
for (var in arr[][key]) print arr[var][key]
```

and

```
if (var in arr[][key]) ...
```

is now supported. Previously, such usage produced a syntax error.

Support For `DEV_ENCODED` `tango_io()` Type

This `spec` release supports the current implementation of the new `DEV_ENCODED` data type for the `tango_io()` function as used at ESRF and elsewhere.

Fix For Crash When Using Empty Macro Functions

A segmentation fault that sometimes occurred when using macro functions defined as an empty string (no curly brackets) in particular contexts has been fixed.

Fix For TACO MCA Access Of Disabled Units

A bug, where `spec` would access the device state on a call of disabled (via the "disabled" key for the `mca_par()` function) TACO MCA device-server units, has been fixed.

Fix For Mistaken "Duplicate Channel" Error

An issue with a handful of hardware controllers (EPICS motor record, ESRF VCT6, JVL SM120B, Munich BR-tronik IPS, Newport Agilis and Sigmatech FC-501A), where a "duplicate channel" error message would sometimes be erroneously displayed during hardware configuration when a channel number was the same as the configured number of channels, has been fixed. The issue with the VCT6 was introduced with the code update in `spec` release 5.09.02-1. The issue with the other controllers dates generally from the introduction of the particular controller support to `spec`.