

WHEN YOU NEED TO BE SURE

SGS

CyFlex® Release Notes

Release Version 6.3.19

September 4, 2020

Developed by SGS North America, Inc.

Version History

Version	Date	Revision Description
1	8/31/2020	Initial publication for CyFlex 6.3.0
2	9/4/2020	Upgrade New Potential Format example and minor spelling/format corrections in <i>Update to Gantner and Snapio Drivers</i> on page 2

Document Conventions

This document uses the following typographic and syntax conventions.

- Commands, command options, file names or any user-entered input appear in Courier type. Variables appear in Courier italic type.
Example: Select the `cmdapp-relVersion-buildVersion.zip` file....
- User interface elements, such as field names, button names, menus, menu commands, and items in clickable dropdown lists, appear in Arial bold type.
Example: **Type**: Click **Select Type** to display drop-down menu options.
- Cross-references are designated in Arial italics.
Example: Refer to *Figure 1*...
- Click intra-document cross-references and page references to display the stated destination.
Example: Refer to *Overview of Changes* on page 1.

The clickable cross-references in the preceding example are *Overview*, and on page 1.

CyFlex Documentation

CyFlex documentation is available at <https://cyflex.com/>. View **Help & Docs** topics, **Release Info** topics or use the **Search** facility to find topics of interest.

For Cummins personnel, refer to the Cummins engineering wiki at:

<http://acizslpapp005.aciz.cummins.com:8005/display/glod/CyFlex+Documentation>. A Cummins account is required.

Table of Contents

Overview of Changes	1
KEY CHANGE IMPACTS AND BENEFITS.....	2
Update to Gantner and Snapio Drivers.....	2
New Modbus Driver	3
Compression Algorithm for History will Limit Disk Activity.....	3
New Error Manager Database (replacing flat files).....	4
gp_test Improvements Supporting Chain Events	4
Phased Timers for Better Real-time Behavior	4
Separated Engine and User Control Tasks	5
Changes to Array Variable Structure	6
Emissions Applications Component Feature Updates	7
ODOR (mercaptan) Composition Members and Related Handling.....	7
New DARTs Data-Point File Format.....	8
Method to Remove a PNP Controller	8
New ecat_mon Specification File Output Format	8
General Changes.....	9
System Status Log – Better Diagnostic Capability	9
CUTY Compatibility.....	10
Watchdog Trigger Popup Screen	10
Re-read Event Processing.....	10
POTENTIAL ISSUES AND WORKAROUNDS.....	11
Control Task.....	11
Changed, Added, and Removed Programs	12
CHANGED IN 6.3.19	12
ADDED TO 6.3.19	12
REMOVED FROM 6.3.19	13
Installation Changes	14

Overview of Changes

This release notes document identifies and describes the updates for CyFlex version 6.3.0. Refer to *Key Change Impacts and Benefits* on page 2 for details on some of the following changes.

- Scientific Linux 6.9 is required for this release. Refer to *CyFlex® Scientific Linux v.31 Installation Procedure*.
- Update to Gantner and Snapio Drivers
- New Modbus driver
- Compression algorithm for History will limit disk activity
- New error manager database (replacing flat files)
- `gp_test` improvements supporting chain events
- Phased timers for better real-time behavior
- Separated engine and user control tasks
- Changes to array variable structure
- Emissions applications component feature updates
- Added support for QT5; refer to *Installation* on page 14.
- ODOR (mercaptan) composition members and related handling
- New DARTs data-point file format
- General bug fixes

Refer to *Installation Changes* on page 14 for install-related change information.

The following is a link to a Jira query resulting in a list of all items addressed in this CyFlex release:

<https://max.cybermetrix.com/jira/issues/?filter=12330>

Key Change Impacts and Benefits

Update to Gantner and Snapio Drivers

This update enables getting new data faster. Previously, there were 2 timer events to process new data, but now the mentioned tasks will immediately update when new data comes in from a device.

In 6.3, `ai_transfer` is event-driven for `gantner`, `snapio`, `ethercat`, and `modbus` drivers rather than being timer-based. Modify `ai_transfer` in the `go.scp` to run at SLO and to only start one instance of it IF ONLY using the `gantner`, `snapio`, `ethercat`, and `modbus` drivers. If any other drivers are also used, like `cyrius` or MTL, then modify `ai_transfer` in the `go.scp` to run at FAS, MED, and SLO and to only start one instance of it. This is to make sure that the other drivers have the correct timers involved with them. An example change to the `/cell/go.scp`:

1. Old Format:

```
# spawn the I/O transfer tasks
ai_transfer 19 WARP +c &
ai_transfer 18 FAS +c &
ai_transfer 15 MED SLO +c &
di_logi_xfer &
```

2. New potential Format:

```
# for cells without MTL or Cyrius IO
ai_transfer 19 SLO +c &

# for cells with MTL or Cyrius IO
ai_transfer 19 WARP FAS MED SLO +c &
```

@Notes:

1. If using other drivers like `cyrius` or MTL, but in your `/specs/inpt_specs.xxx` file you do not use the MED rate, then you do not need to specify MED as an argument for `ai_transfer`. Conversely, if using WARP on some of the channels in `inpt_specs.xxx`, then add WARP to the arguments as well.
2. With 6.3 drivers, `ai_transfer` no longer uses the timers to get samples from the inputs, therefore removing the high-speed timers as in the example will have no effect on the performance of the inputs but will free up resources for other processes.

After modifying the `go.scp`, it is recommended to update all the analog inputs in `inpt_specs` to SLO for the `gantner`, `snapi_o`, `ethercat`, and `modbus` channels. This will significantly reduce the CPU consumption of `ai_transfer` and all the analog inputs will still update at the fastest rate possible. Keep all other channels at the same rate as before.

Before making any changes to the `go.scp` or `inpt_specs`, make a backup of the file in case you need to revert to 6.2.

New Modbus Driver

This change allows a `modbus` communication task the ability to use calibration tables which `gmodbus` does not easily allow.

Compression Algorithm for History will Limit Disk Activity

The `delta_hst` application in CyFlex versions prior to CyFlex.6.3 required modification of a “tolerance” parameter that is associated with each variable. This process was not well understood and the effect was that the `delta_hst` application was saving more data than necessary and thereby consuming a lot of disk space and continually doing a lot of disk I/O. Th 6.3 version has 2 significant modification to reduce disk I/O requirements.

1. An “autotuning” algorithm is used to continually adjust the “tolerance” setting based on a limitation in the rate of change of each variable. This limitation is also a user adjustable parameter.
2. Some variables can be excluded from the history files. There are certain variables which may be noisy or are of no interest for possible recovery. If these are excluded then the disk I/O requirements can be further reduced.

As a result:

- Data collection rates are greatly reduced, thereby reducing disk activity and disk size requirement.
- Collection rate is controlled by frequency of variable change but limited by user-tunable parameters.
- Collection rate for integer variables also frequency limited.

There are some variables which would not be of interest for diagnostics but change rapidly. You can now ignore these by modifying the `delta_specs` file. The `@IgnoreList` keyword can be added after the list of event names.

For a test cell using Gantner I/O and `asam3`, the following list is recommended:

- `@IgnoreList`
- `watch_dog`
- `asam3_1_Update_Interval1`
- `asam3_1_Update_Interval2`
- `asam3_1_Update_Interval3`
- `asam3_1_Update_Interval4`
- `$`

Sort by size so the largest files (variables) will be at the end of the list to determine which variables might be very noisy and changing very rapidly by A very large size may indicate a noisy channel or tolerance value that is too small. Enter:

```
cd /data/compressed/split
ls -lrS
```

This method applies to CyFlex.6.3 and earlier versions, but the file sizes should be smaller in 6.3 due to the auto-tuning algorithm which will continuously modify the tolerance specification, so it is not necessary to select a tolerance as in the earlier versions. If there are large files near the end of this output for variables that will never be of interest for recovery, exclude them by editing the specification file.

The auto-tuning feature can be adjusted to collect more or less data based on a frequency calculation. Two variables are available to adjust the auto-tuning algorithm, `HST_freq_lmt` and `HST_int_freq_lmt` for REAL and Integer variable, respectively. These two variables are created in the `perf_labels` file but can be overridden by an entry in the `/cell/cell_special` file or temporarily with the `svar` command.

Setting these variables to a larger value will result in lower tolerance settings and larger history files and vice-versa.

New Error Manager Database (replacing flat files)

The error manager has been modified to use a local database. This allows the storage of more and past errors, going back even weeks and multiple `gos`, and can eventually have intractable GUIs to ease filtering.

The error manager program is now started in `cyflex.6.3.X` versions in the `go.scp` file with no arguments: `error_mgr &`.

gp_test Improvements Supporting Chain Events

A new `@CHAIN_EVENTS` keyword has been added to support features similar to `@FUEL_READING_SYNC`, but in test modes that are not taking a fuel reading. This provides support in `gp_test` for the synchronization of multiple data acquisition processes that must be managed within a test mode.

Phased Timers for Better Real-time Behavior

In CyFlex.6.3, it is possible to have standard timers which operate at the same frequency but are slightly delayed (phased). This makes it possible to control sequence of operation of related applications which are running at the same rate but must execute in a particular sequence to ensure that the results of one process are immediately available to a second one.

Separated Engine and User Control Tasks

Five new programs are now available for controlling and monitoring engine and user loops in CyFlex.

The programs are called `eng_ctrl_specs`, which will automatically launch the `eng_ctrl_task`, `user_ctrl_specs`, which will automatically launch multiple instances of `user_ctrl_task`, one for each loop, and a new `ctrl_disp` which will launch a QT version of control display variables and plots.

The old `ctrl_disp` and `ctrl_task` programs will not be supported in future major versions of CyFlex.

The new programs will replace the existing `ctrl_task` and `ctrl_specs` programs. They require the use of the new version of `ctrl_disp`, all of the engine control commands (`sp`, `to`, `dy`, `th`, etc.), and `gp_test` in order to be fully functional. All of these updated supporting control programs are included in this release.

Sample specification files (`eng_ctrl_specs.def`, and `user_ctrl_specs.def`) are available in the `/specs.def` directory.

The user control specification files are very similar (though not exact) to the present user specs in that they use the same keywords. However, more options are available with the new program than with the old. A separate instance of the `user_ctrl_task` will be launched for each user loop. Keep in mind is that individual instances of the user control program are not affected, locked, or aborted when processing or reprocessing a control specification file. Therefore, commenting out a loop specification and reprocessing a file does not abort the control loop. That must be done manually with the `slay`, `kill`, `remove_loop` or `remove_pnp` command.

The engine control specification file is now of the keyword style with information grouped in much the same way as it was before. Again, more options are available in how values can be specified.

The control display program will now bring up a QT based template and will populate the template with variables that are pertinent to the loop/controller in question. The template contains tabs for variables and others for plots.

Many of the display values, such as target, tolerance, gains, ramp rate, etc. are live and can be changed at any time. More control display variables are now created and may also be changed on the fly unless the user has specified that a CyFlex expression be used. In that case, the expression will override the display variable. The specification files can be updated with changes in gain using the new GUI.

Important:

All control loops in the new programs must be referenced by their name. Loop numbers do not exist in the new programs. Therefore, if the user has a `gp_test` or a computed expression that is checking the open/closed loop mode, these tests will need to be modified.

For example, the following computed expression, which is used for running transient tests, will fail:

```
Throt3DGross    %      2      -      FAS      OFF      1
if( @strcmp_lbl_lit( CtrlMode_2, 'OPEN_LOOP' ) ) \
  then ( 0.0[%] ) \
else ( @long_3d_comp( GrossTorqTrgt, 9[none] ) )
```

The test will need to be modified as shown below, assuming the control device is named Throttle:

```
Throt3DGross    %      2      -      FAS      OFF      1
if( @strcmp_lbl_lit( Throttle_MD, 'OPEN_LOOP' ) ) \
  then ( 0.0[%] ) \
else ( @long_3d_comp( GrossTorqTrgt, 9[none] ) )
```

The general-purpose test program (`gp_test`) has been modified to look for the new programs and will send events to them if they exist or to the old `ctrl_task` if they do not exist.

A document describing the new engine control program is attached with Jira Issue [CFBUG-1428](#).

When using the new engine control task, the `comp_ctrl` program should no longer be started in the `go` script. This functionality has been moved into the engine control task.

Execute the following steps to implement:

1. Modify the `go` script to remove calls to `ctrl_task`, `ctrl_specs`, and `comp_ctrl`.
2. Create a new `eng_ctrl_specs.NNN` file.
3. Create one or more `user_ctrl_specs` files. These files are not automatically read. All loops can be defined in one file or the loops can be defined in many different files.
4. Add a call to `eng_ctrl_specs [file_name]` in the `go` script in a place where the necessary control loop name variables have already been created.
5. Add calls to `user_ctrl_specs [file_name] [loop_name]` as desired

Changes to Array Variable Structure

In prior versions of CyFlex, the data structure which contained the value members of the array were required to all have the same units. In CyFlex.6.3, the default case is for all members to have the same units, but this can be overridden by a command line utility so that individual members can have different units.

- Each element can have different units in 6.3 but all are initialized the same way.
- Use `sarru` to change the units for an individual element.

Example: `sarru my_arr:4 psi`

Emissions Applications Component Feature Updates

Several new emissions-related components are added to CyFlex 6.3 to support the flow of information that will be coming from the new Test Request Interface (TRI) application, which is a stand-alone application separate from CyFlex. All the previous emissions-related components (that supported the flow of information from the current `CVS_CI` application) are retained in CyFlex 6.3. Existing emissions test cells migrating to CyFlex 6.3 will continue to use the previous emissions-related components. The TRI data flow will be prototyped at the target test cells at Cummins Technical Center (CTC) in their Emissions Measurement Lab (EML). These new emissions-related components in CyFlex 6.3 will be used in that process.

ODOR (mercaptan) Composition Members and Related Handling

CyFlex.6.3 supports measurement systems which monitor the concentration of chemical components (mercaptan) which produce the noticeable odor that we associate with natural gas. The composition variable structure supports the different constituents related to this.

The newly-supported Mercaptan odor measurement device monitors the following components:

- Hydrogen Sulfide
- Methyl Mercaptan
- Ethyl Mercaptan
- Dimethyl Sulfide
- 2-Propyl Mecaptan
- Butyl Mercaptan
- Methyl Ethyl Sulfide
- 1-Propyl Mercaptan
- 2-Butly Mercaptan
- Diethyl Sulfide
- TetraHydroThiophene

New DARTs Data-Point File Format

This eases transmission of files to the DARTS database rather than taking the old PAM files and converting to DARTS. This will also allow the use of ECM and Aux parameters in the specfile making the files easier to work with.

Method to Remove a PNP Controller

Previously it was required to download and execute certain commands when using a PNP device. Among other things, this may have included processing a control loop specification file.

An instance of the `user_ctrl_task` should be launched with the command:

```
user_ctrl_specs /path/spec_file_name [loop_name]
```

There are no longer hidden methods to launch user control loops.

A new `remove_pnp` command is provided to kill a specific instance of the user control program.

Example:

```
remove_pnp loop_name
```

The script contains a **kdiallog** that requests confirmation, but this may be commented out if the command is to run in the background.

New `ecat_mon` Specification File Output Format

The `ecat_mon` specification file has new format for outputs.

If communicating with a device via EtherCAT and using the `ecat_mon` application, modify your `ecat_mon` specification file to specify the variable type for the output. Inputs already had this feature, but the variable type was not specified before in the outputs.

Example of specfile changes:

1. Old Format:

```
$OutputVariables
# variable cyflex device update rate [ms] master slave slave multiplier
# name units units OR event name index position channel default=1
Speed_RF rpm rpm 10 0 0 4 8.19175
```

2. New Format:

```
$OutputVariables
# variable cyflex device update rate [ms] master slave slave type multiplier
# name units units OR event name index position channel default=1
Speed_RF rpm rpm 10 0 0 4 f 8.19175
```

Check the `/cyflex/specs.def/ecat_mon.unico` for an example specfile.

General Changes

- The trace files are now located in `/specs/gp/trace/` directory. Before the upgrade, make sure there is no file in the `/specs/gp` directory called 'trace' in all lowercase or this can cause issues. Refer to *Installation* on page 14 for additional information.
- There will now be a small-time delay when running `errs` which is normal, that is because it is now a database.
- The `use` function has changed to `.html` format so it can be maintained in one location for CyFlex and CyFlex.com. Appearance of `use` is also revised in this release.

System Status Log – Better Diagnostic Capability

The CyFlex System Status Log has been improved to provide more information about the CyFlex applications running on a system and the ability log information from scripts. A couple examples of information that is logged: core dumps, CyFlex watchdogs, and 'Go'.

System information is logged: Kernel versions, CyFlex version, CPU, Video Card.

In the near future, the types of hardware communicating with CyFlex drivers will be logged.

This will allow us to notify customers when improvements or bugs have been discovered that effect a particular device

CUTY Compatibility

Some users reported issues with `asam3` automatically reconnecting after upgrading from 6.2.x to 6.3. Automatic reconnect is handled by CUTY. If using CyFlex 6.3.x and CUTY is running on a Windows 10 machine, use version 7.1.0.20016 of CUTY.

Watchdog Trigger Popup Screen

This release includes a new popup screen that appears when the watchdog is triggered. It includes the registered name, the current task name, the time of the incident, and the time interval of the running task. See the example below.



Re-read Event Processing

The prior release included support for processing of a re-read event to avoid the need to slay and restart tasks to read modified specification files. The executables with this support are `subsonic` and `gas_prop`. Release 6.3 adds this support to the following executables:

<code>lc_scales</code>	<code>cf_scales</code>
<code>subsonic</code>	<code>gas_prop</code>
<code>gasfl</code>	<code>volef</code>
<code>cfv_1065</code>	<code>add_water</code>

Potential Issues and Workarounds

Control Task

Running too many control loops at FAS that do not need to operate at a high speed can cause the old `ctrl_task` to become overloaded in 6.3 when it sometimes did not in previous versions. An indicator of this is that some of the variables created by `ctrl_task` (`loop_name_CM`, `loop_name_RF`, `loop_name_IT`, etc.) will stop updating. This can often be fixed by re-evaluating the speed necessary for the control loop. For example, control loops controlling the temperature of liquid rarely need to be updated faster than MED due to the slow response time of heating or cooling the liquid. Often times, they can even run at SLO without any difference in performance. If you're still experiencing issues after re-evaluating the control loops and slowing some of them down, you can go through and change some control loops to FAS1, FAS2, and FAS3. This will change the phase of when these loops are executed, helping to eliminate 'timer storms'.

Changed, Added, and Removed Programs

Changed in 6.3.19

dspec_qt_conv is now dspec_qt5_conv
 mk_data_files is now mk_config_files.12

Added to 6.3.19

ai_modbus_serial	ai_modbus_tcp	ao_modbus_serial	ao_modbus_tcp
blk_num_mgr.12	candbc_daq	candbc_full_list	changeDartsKeyword
changeDartsTestid	changeDartsUnits	ci2fur12	dartsAgeSite
darts_datapoint	dartsResend	dartsResendSite	dartsStatus
dartsStatusSite	di_modbus_serial	di_modbus_tcp	do_modbus_serial
do_modbus_tcp	ecg.12	ecg_header.12	email_mgr
emcon_comm	esvd_pre_star5.12	esvd_star5.12	esvd_star5_12.raw
filterdata.12	filterdata12.raw	get_errs	make_tri_arrays
medordata	MEDOR_set	mk_config_files12	mk_ecm_files12
modbus_srvr_serial	modbus_srvr_tcp	modbus_strm_serial	modbus_strm_tcp
msu_clear_states	parse_a21	qt5loggerplot	re-read
run_num_mgr.12	sarru	setDARTSpoint	setDARTStestid
set_eng_specs12	setevparam	show_hst_data	simple_elb
star1.12	star2_7.12	star_pream.12	tca.12
tca_header.12	tca_logr.12	tri_logr_spec.12	update_eng_specs
update_user_specs			

Removed from 6.3.19

ABpanel	ABpanel_specs	adv_th	ai_ni6220
asset2MC	asset_ver	at	bl
bl_setup	brandt	cart_node	cell_mail
cflxstart	ci_ni6624	ck_cal_type	comp_meterlog
conevreal	co_ni6624	di_ni6220	di_ni6528
di_no6220	di_no6528	down_code_report	err_stats
err_stats_rpt	fix_fuel_reset	fix_fuel_step	fix_hr_reset
floger2	get_pam_data	get_racfid_name	hsda_link
hsda_results	hsda_var	hst_list	hst_off
hst_on (history flags are on by default in 6.3)	ipv	ipv_specs	limx
mtrMonitor	niFPGA	niFPGA_specs	occMenu
PPN_report	push_reset_stat	read_DAV_ambient	resource_mgr
rmgr_rqst	set_tc_name	task_err_log	test_ct
transt_isa_noKM	transt_warn	weekly_down_report	wtrm_barcode
wtrm_testcell	wtrm_valid		

Installation Changes

gp_test Trace Files Directories

In CyFlex.6.3.x, `gp_test` trace files are no longer in the `/specs/gp/` directory. They cannot be arbitrarily placed by changing the `@TRACE_FILENAME` keyword in the header file.

Instead, all trace files go into the `/specs/gp/trace/` directory that will be automatically created by the first `gp_test` instance launched in 6.3.

The trace file names are automatically assigned using the name of each instance of `gp_test`. If you are running a “test” instance and a “smoke” instance and enter `ls -ltr` in the `/specs/gp/trace/` directory to view results similar to following:

```
-rw-rw-r-- 1 tc102 users 1611 Feb 25 13:33 smoke.02.26.07.56
-rw-rw-r-- 1 tc102 users 2167 Feb 25 14:33 test.02.26.07.56
-rw-rw-r-- 1 tc102 users 1611 Feb 26 07:56 smoke.02.26.09.01
-rw-rw-r-- 1 tc102 users  370 Feb 26 07:56 test.02.26.09.01
-rw-rw-r-- 1 tc102 users 1611 Feb 26 09:01 smoke
-rw-rw-r-- 1 tc102 users 1267 Feb 26 09:04 test
```

The “test” and “smoke” files will contain the most recent mode changes. When that file fills up, instead of being renamed to “test.old”, etc., it will be renamed to “test.MM.DD.HH.MM” (month/day/hour/minute) and that file will not be removed until a cleanup operation is executed. Consequently, the system will save old trace files for as long as desired or until your disk fills up. The default is 30 days and a max of 100 files PER INSTANCE. The cleanup specs are defined in the header file of each instance.

@TRACE_FILENAME

```
#filename no longer used          max-mode-transitions    max_age-in-days    max_num files
                                30                      100                 1000
```

The cleanup operation is only performed when the `gp_test` app is launched, usually whenever a `go` is performed, so the number and age of files may grow considerably. For this reason, it is recommended to add a line similar to the following to the `usercron` specification file.

```
58 1 * * * /cyflex/bin/cleanup 30 500 "/specs/gp/trace/*" 1>/dev/null 2>&1/dev/null
```

This will operate on all the files in the `/specs/gp/trace/` directory based on the age and number of files and will accomplish: .

- 1) Removal of any orphans from instances that were used previously, but no longer are in use.
- 2) If you run for a very long time between `gos` it will keep the number of trace files from building up from one `go` to the next, since the “instance-specific” cleanup only happens when you first launch `gp_test`.

MTL Kernel Modules Location Change

The kernel modules for MTL, vcanpcid and tc9513 have been moved out of the /cyflex/bin/modules directory to keep them CyFlex version independent. The files are now in the /usr/local/share/cyflex/kernel_modules directory. The /etc/rc.d/rc.local file has been modified to reflect the new location of the kernel module files. A backup of the previous rc.local file is in the /etc/rc.d/ directory as rc.local.pre63. The cyflex-kernel-module RPM will take care of installing the kernel module files and modifying the rc.local file.

verify_testcell_config.sh script includes Qt4 and Qt5

The verify_testcell_config.sh script file has been modified to include the version of Qt4 and Qt5. For CyFlex.6.3.0, the Qt4 version must be 4.8.7 and the Qt5 version must be 5.12.3. Execute the verify_testcell_config.sh script file from the command line to determine the version numbers. Contact SGS if the version numbers are incorrect.

QT Version Upgrade

With the CyFlex 6.3.0 release, the Qt4 version is updated from 4.7.4 to 4.8.7 which is the last release of the Qt4 API. We have also started but not yet completed migrating applications to the Qt5 API.

The reasons we have switched from Qt4 to Qt5 are support and sustainability. Qt4 was no longer being updated by the vendor. In addition, the Qt5 API is faster and provides additional capabilities to be leverage in the future.

No changes are required to current .ui files to use the new runqtui using the Qt5 API. If you discover a bug with the Qt5 version runqtui, you may revert to the older runqt4ui version until the bug is fixed.

The migration to Qt5 does not change the look and feel of any of the GUIs.

Applications using Qt5 API	Applications using Qt4 API
runqtui	qtloggerplot
quick_display	tcal_util_64pts
qtxyplot	abate_* applications
cyberwizard	wtrm_* applications
valuemanagerui	runqt4ui
qtstripchart	

cell.def and specs.def files Location Change

The `cell.def` and `specs.def` files have been moved to directories under the `/cyflex` directory. The purpose of these directories residing under the `/cyflex` directory is to keep their file contents version-specific. There is no longer a need to maintain a `/cell.def` and a `/specs.def` directory. When a CyFlex.6.3.# version is installed, the `cell.def` and `specs.def` directories will also be installed.