

SpecTcl Updates

Modifications from v32.1.1 to v32.4

ARIS / PID / optics reconstruction updates
6/27/2026

Current source location: /user/arisdaq/PID/Develop/v32

SpecTk application folder: /user/arisdaq/SpecTk_1.9.4

SpecTk window file shown in screenshot: /user/arisdaq/SpecTk/Oleg/o60_03-144Sm_June_DB2a.spk

SpecTcl definition file shown in screenshot: /user/arisdaq/vme/exp/def/Oleg/pid_v40_v5_144Sm-June_db23.tcl

Purpose. This note summarizes the practical source and calibration updates made after v32.1.1, ending with v32.4. The focus is on the PID Z-q correction, the new Dip23 optics reconstruction block, the new Dip23 delta_method=3 mode using Dip34.deltaP, and the matrix inverse cache optimization.

1. Executive summary

- **v32.1.1:** Merged the PID branch with the VME branch. This preserved the VME diamond/remapping changes and the PID/ToF updates from the v31.7.x line.
- **v32.2.0:** Added Z-q correlation correction controls: zmq_rotatePoint, Qzmq_rotateSlope, and Zzmq_rotateSlope. These provide event-by-event Z and q corrections based on the measured Z-q displacement.
- **v32.3.0 - v32.3.4:** Implemented and debugged Dip23 trajectory reconstruction. The important final step was delta_method=3, where Dip23 uses Dip34.deltaP instead of forcing the missing DB2 plane to a virtual X=0 point.
- **v32.4:** Added matrix inverse caching in ARISOpticalSegment. The inverse matrix is recalculated only when optics matrix coefficients change, rather than for every event.

2. Version timeline from v32.1.1 to v32.4

Date	Version	Who	Subject
6/25/2026	v32.1.1	OT	Merged PID and VME versions.
6/26/2026	v32.2.0	OT	Implemented CTreeVariable zmq_rotatePoint, Qzmq_rotateSlope, and Zzmq_rotateSlope.
6/27/2026	v32.3.0	OT	Implemented ARISOpticalSegment Dip23.
6/27/2026	v32.3.1	OT	Corrected CARis_optics, including inverse matrix vertical terms matr_inv.by and matr_inv.bb.
6/27/2026	v32.3.2	OT	Complete revision of ARISOpticalSegment::readVectorInit().
6/27/2026	v32.3.3	OT	Added ARISOpticalSegment reset and used it in ARIS::resetParamGlobal().
6/27/2026	v32.3.4	OT	Added Dip23 delta_method=3 using Dip34.deltaP.
6/27/2026	v32.4	OT	Added ARISOpticalSegment matrix inverse caching; kept v32.3.4 Dip23 delta_method=3 logic.

3. Source files touched between v32.1.1 and v32.4

File	Modification
source/CARis_include/CARis.h	Version string updated to Version 32.4; added ARISOpticalSegment Dip23 member.
source/CARis__main/CARis.cpp	Safer DB initialization loop; db[2] is skipped by continue instead of incrementing the loop counter manually. Added Dip23.Initialize("aris.Dip23").
source/CARis__main/CARisCalculator.cpp	Changed optics calibration order so Dip34 is calculated before Dip23. This is required because Dip23.delta_method=3 uses aris.Dip34.deltaP.
source/CARis__main/CARis_rate.cpp	Added per-event Reset() calls for Dip23, Dip34, and Dip45 in ARIS::resetParamGlobal().

File	Modification
source/CAris_optics/CAris_optics.h/.cpp	Main optics changes: Dip23 support, readVectorInit() logic, inverse matrix sign correction, delta_method=3, optics Reset(), and matrix inverse caching.
source/CAris_pid/CAris_pid.h/.cpp	Added ZmQrot and Z-q rotation variables; applied q and Z corrections when ZmQ is near the rotation point.
source/Makefile	Cleaned USERCXXFLAGS overwrite; added touch_sources helper for clock-skew warnings after copying/unzipping sources.
calibrPID/matrix/CB2_Dip23_208Pb.tcl and related def/calibration files	Dip23 settings now support delta_method=3 and DB2 virtual begin-side reconstruction using downstream deltaP information.

4. PID Z-q corrections and timing-shift observation

Motivation. The June PID analysis needed improved Z and q separation, especially in spectra using Z-q and A-2q style coordinates. v32.2.0 adds calibration parameters to apply small correlated corrections to Z and Q using the observed ZmQ displacement.

New PID parameters and variables added in CAris_pid.h/.cpp

```
CParameter ZmQrot;
CTreeVariable zmq_rotatePoint;
CTreeVariable Qzmq_rotateSlope;
CTreeVariable Zzmq_rotateSlope;
```

Correction logic. If ZmQ is valid, the code computes $dZmQ = ZmQ - zmq_rotatePoint$. For events close to the rotation point, q and Z are shifted using Qzmq_rotateSlope and Zzmq_rotateSlope. ZmQrot is then recalculated from corrected Z and Q. This keeps the raw ZmQ available while also providing a corrected coordinate for tuning.

Timing-shift note. The spreadsheet comparison still shows an unexplained timing/calibration shift between March and June analysis. In the highlighted row, run 4302 uses $tof_offset = 11.9$ and $Q_offset = -1.07$, while adjacent rows use $tof_offset = 16.4$ and $Q_offset = -0.74$. This should be treated as a calibration/history issue to verify, not as a confirmed source-code error yet.

Run range	tof_offset	Q_offset	Comment
4303-4313	16.4	-0.74	Reference row near June analysis setting
4302-4302	11.9	-1.07	Highlighted row; unexplained shift
4300-4301	16.4	-0.74	Neighboring row returns to reference-like offset

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	run_start	run_stop	Z_method	tof_metho	brho_meth	Z_slope2	Z_slope	Z_offset	Z_log_bet	tkebox	tof_offset	fine	EZ_meth	Q_offset	dbeta_me	dbeta_fix	dbrho_fix	L34_corre	L45_corre	exp #	fragment	notes	
23	4303	4313	0	153	45	0	6	0	1	5	16.4	1	1	-0.74	2	0.0013	0.019	0	0				117
24	4302	4302	0	153	45	0	6	0	1	5	11.9	1	1	-1.07	2	0.0013	0.019	0	0		Oleg play	changed fi	117
25	4300	4301	0	153	45	0	6	0	1	5	16.4	1	1	-0.74	2	0.0013	0.019	0	0				117
26	#4300	5000	1	153	45	0	6	-0.51	1	5	16.4	1	1	-0.74	2	0.0013	0.019	0	0				117

Figure 1. Spreadsheet screenshot showing the non-explained timing/Q-offset shift between March and June analysis.

5. PID performance after Z-q corrections

Result. The PID spectra show clearly separated Z and q groups after applying the correlation corrections. The ZmQ panel shows the charge-state bands and the red gates. The A/q, A-2Z, and A-2q panels are now visually consistent with improved Z and q resolution. This is the main practical validation of the v32.2.0 correction variables.

- AoQ vs Z shows diagonal isotope structure with improved separation.
- Am2Z and Am2Q plots show cleaner banding after Z/q tuning.
- ZmQ vs Z displays q-band structure directly and is useful for q-gate placement.
- TKE vs DB5 PIN dE remains a strong independent consistency check.

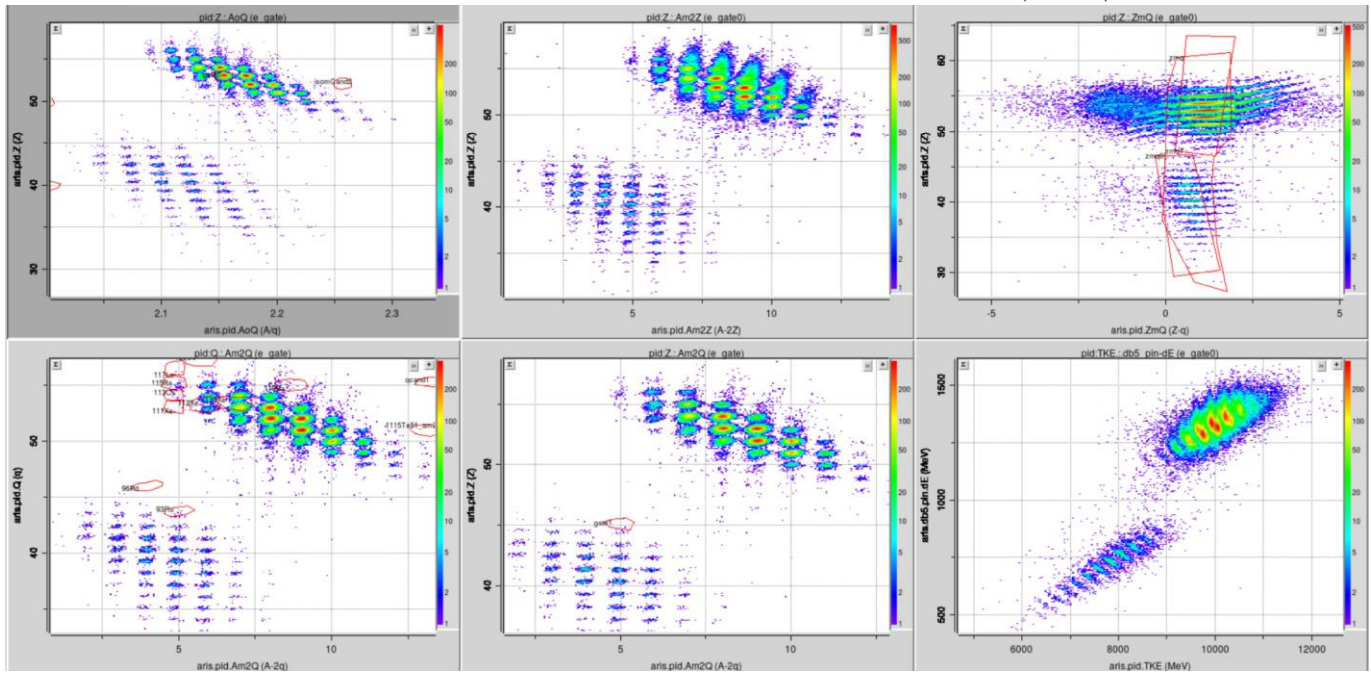


Figure 2. PID figures demonstrating Z and q resolution after implementation of Z-q correlation corrections.

6. Dip23 optics reconstruction updates

Why Dip23 needed special handling. Dip23 corresponds to transport from DB2 to DB3. In the current setup DB2 is not initialized/measured, but Dip23 still needs to reconstruct the upstream spatial distribution at the dispersive plane. If the begin side is treated as a virtual point and the code solves deltaP by forcing X=0, the reconstructed DB2 X collapses artificially. That was the problem observed before v32.3.4.

readVectorInit() revision. v32.3.2 made `use_x < 0` a true virtual-object mode. This prevents uninitialized `db[2]` from being touched and sets X, Y, A, B, L, D, and DA to zero for the virtual point. For measured modes, invalid X/Y/A/B values are explicitly invalidated so stale event values are not reused.

Virtual object mode for Dip23 when DB2 is absent

```
if(use_x < 0)
{
    vi.X = 0.0;
    vi.Y = 0.0;
    vi.A = 0.0;
    vi.B = 0.0;
    vi.L = 0.0;
    vi.D = 0.0;
    vi.DA = 0.0;
    return;
}
```

Inverse matrix correction. v32.3.1 corrected the vertical inverse matrix terms. For the Y/B block, the inverse must use `-by` and `+yy` in the lower row. This affects reverse vertical reconstruction and cleans up `b_vc.Y` and `b_vc.B` logic.

Correct Y/B inverse matrix block

```
matr_inv.yy = matr.bb * rdet_yb;
matr_inv.yb = -matr.yb * rdet_yb;
matr_inv.by = -matr.by * rdet_yb;
matr_inv.bb = matr.yy * rdet_yb;
```

delta_method=3. v32.3.4 adds a special Dip23 mode: use Dip34.deltaP as the external momentum deviation. This avoids solving deltaP by forcing DB2 X/A to zero. As a result, Dip23.b_vc.X now shows a physical spatial width and a sensible correlation with deltaP.

Dip23 uses Dip34.deltaP

```
set aris.Dip23.delta_method 3

// In source:
if(delta_method == 3 && m_sName == "aris.Dip23" && aris.Dip34.deltaP.isValid())
{
    b_vi.D = aris.Dip34.deltaP;
    e_vi.D = aris.Dip34.deltaP;
}
```

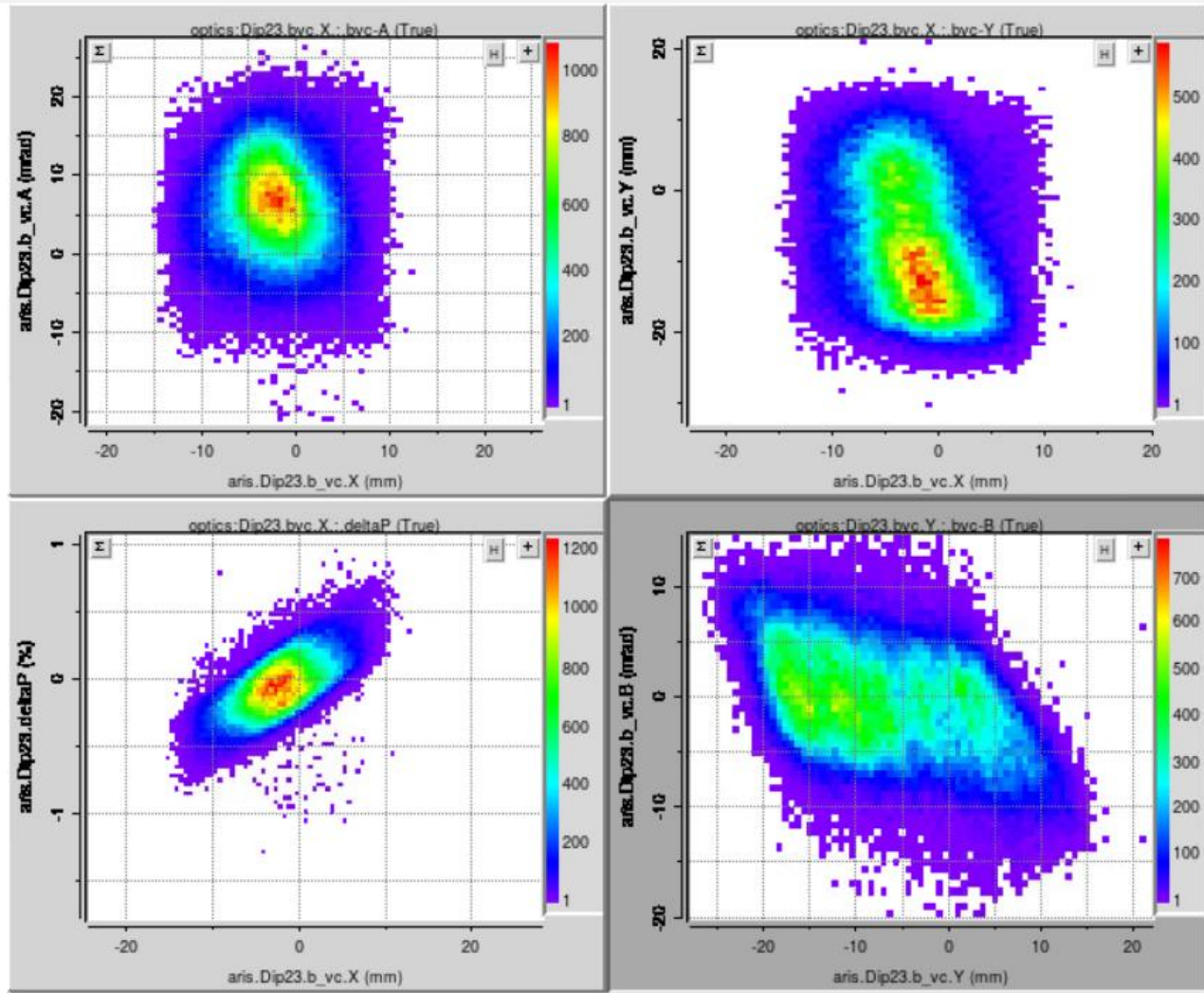


Figure 3. DB2/Dip23 spatial plots after v32.3.4. The reconstructed Dip23.b_vc.X is no longer artificially collapsed to zero, and the X-deltaP correlation is restored.

7. v32.4 matrix inverse caching

Original behavior. The inverse optics matrix was recalculated during every event. This is unnecessary because matrix coefficients come from Tcl/calibration settings and normally remain constant for a run.

New behavior. v32.4 adds a cache inside ARISOpticalSegment. The inverse matrix is recalculated only when one of the matrix coefficients changes. The cache is initialized/reset at segment initialization, but it is not cleared by the per-event Reset() method.

- matrix_cache_valid records whether the cached inverse is usable.
- last_xx, last_xa, last_xd, last_xdd, last_ax, last_aa, last_ad, last_yy, last_yb, last_by, last_bb, last_lx, last_la, and last_ld store the matrix coefficients used for the cached inverse.
- MatrixChanged() compares current Tcl-controlled coefficients to the stored values.
- RememberMatrix() stores the coefficients after recalculation.
- CalcInverseMatrix() exits immediately when MatrixChanged() is false.

v32.4 cached inverse-matrix logic

```
void ARISOpticalSegment::CalcInverseMatrix()
{
    if(!MatrixChanged()) return;

    // calculate inverse matrix only here
    ...

    RememberMatrix();
    matrix_cache_valid = true;
}
```

Why this is safe. The per-event variables b_vi, e_vi, b_vc, e_vc, brho, deltaL, and deltaP are still reset every event. Only the inverse matrix is cached. If a Tcl script reloads the matrix during a SpecTcl session, MatrixChanged() detects the coefficient change and forces a recalculation.

8. Validation, files, and working configuration

Item	Path / file
Current source	/user/arisdaq/PID/Develop/v32
SpecTk version folder	/user/arisdaq/SpecTk_1.9.4
SpecTk file from screenshot	/user/arisdaq/SpecTk/Oleg/o60_03-144Sm_June_DB2a.spk
SpecTcl definition folder	/user/arisdaq/vme/exp/def/Oleg
SpecTcl definition file	pid_v40_v5_144Sm-June_db23.tcl

Validation status. The v32.3.4 logic was visually validated using the Dip23 DB2 spatial plots shown above. The v32.4 change is a performance optimization; after installation it should be compiled and checked with the same PID and Dip23 plots to confirm identical physics output. A useful additional test is to compare spectra before and after v32.4 with the same run and calibration files.

9. Recommended follow-up checks

- Verify the unexplained March/June timing shift: confirm whether the tof_offset=11.9 and Q_offset=-1.07 row is intentional, copied from an older calibration, or run-specific.
- Check that ZmQrot improves charge-state separation over the full run range, not only the displayed gate.
- For Dip23, compare Dip23.deltaP and Dip34.deltaP for method 3 and verify that the DB2 X-deltaP correlation remains stable for neighboring runs.
- Benchmark v32.4 against v32.3.4 on the same file. The matrix cache will remove repeated inverse calculations, but the total speedup depends on whether optics reconstruction is a visible fraction of total event processing time.
- Keep the matrix cache out of per-event Reset(); otherwise the optimization is lost.