TFCompanion Technical Release 6.4.5 Update

Document date: April 15, 2026

Scope: detailed description of the major features, formulas, workflow updates, performance changes, and import/export additions introduced in this release cycle, starting with the incoherent-layer work and continuing through the material-model and Woollam import updates.

1. Optical Calculation Engine: Incoherent and Partial-Coherence Support

- Added explicit support for filmstacks containing an incoherent layer. The main FilmStack evaluation path now detects the incoherent-layer index and routes the calculation to a coherent, incoherent, or partial-coherence branch as appropriate.

- The degree of coherence is only meaningful when the filmstack contains an incoherent layer. UI behavior was updated so coherence is shown and used only in that physically relevant case.

- Fully incoherent behavior uses the dedicated IncoherentFilmstack path for structures of the form ambient / coherent top stack / thick incoherent slab / coherent bottom stack / backing medium.

- Partial coherence is implemented as a phase-jitter average around the nominal optical thickness of the incoherent layer, rather than as an arbitrary linear blend of coherent and incoherent intensities.

User coherence parameter and phase spread:

$$C = \exp(-\sigma\phi^2 / 2)$$

$$\sigma\phi = \sqrt{[\ -2 \ln(C)\ ]}$$

- For $0 < C < 1$, the code clones the filmstack, temporarily restores the incoherent layer to coherent behavior, sweeps the layer phase over one full 2π cycle, evaluates the coherent stack at each sample, and computes a weighted average with a wrapped-normal phase distribution.

Intensity averaging in the partial-coherence branch:

$$R_{partial} = (\Sigma_i w_i R_i) / (\Sigma_i w_i)$$

$$T_{partial} = (\Sigma_i w_i T_i) / (\Sigma_i w_i)$$

- The implementation falls back to the exact incoherent result when the phase-cycle thickness cannot be determined robustly or when the coherence parameter is effectively zero. When C is effectively one, the standard coherent evaluation path is used.

- Energy-conservation consistency was checked during development, and the corrected formulation produces physically sensible reflectance/transmittance totals for incoherent and near-coherent limits.

2. Calculation Speed and Parallel Residual Evaluation

- Optimized Marquardt-side residual evaluation by parallelizing point-wise optics evaluation in ElliFitFunc. The implementation uses worker filmstack copies so the expensive optics calculations can run concurrently without corrupting shared state.

- Added global calculation thread mode selection with an Auto mode and a Single Thread fallback mode.

- Removed a fixed synchronization delay that had been dominating total optimization time.

- Reduced worker initialization/update overhead and improved worker reuse between residual calls.

3. Model Refinement Dialog

- Added a separate Model Refinement workflow for improving an already-good filmstack fit when the target thickness is known and the remaining error is mainly dispersion-structure error rather than gross thickness error.

- The refinement dialog evaluates candidate model changes such as keeping the current model, adding an oscillator, splitting an oscillator, and removing suspicious oscillators.

- Residual analysis was added so the dialog can inspect the difference between measured and calculated data, identify local discrepancy regions, and present them to the user as candidate refinement zones instead of hard-coding a wavelength.

- Before a full Marquardt solve is launched for an added or modified oscillator, the seeded change is checked against the measured-versus-calculated discrepancy. Large or wrong-direction perturbations are reduced before handoff to Marquardt so the starting point stays closer to the baseline model.

4. Lorentz-Kim Oscillator Model

- Expanded Lorentz-Kim support and integrated it more deeply into both the UI and the fitting workflow.

- Added an explicit $\varepsilon\infty$ parameter to the Lorentz-Kim model, including calculation, display, XML import/export, text import/export, cloning, parameter-table integration, and Woollam import support.

- Added literature-based starting presets for α-NPD / NPB and Alq3.

Implemented Lorentz-Kim dielectric function:

$$\varepsilon(E) = \varepsilon\infty - \Sigma_i A_i \left[1 / (E - E_{o,i} + i\Gamma_i(E)) - 1 / (E + E_{o,i} + i\Gamma_i(E)) \right]$$

$$\Gamma_i(E) = G_i \exp\{ - [((E - E_{o,i}) / G_i)^2 / (1 + s_i^2)] \}$$

- Here E is photon energy in eV, A_i is oscillator amplitude, $E_{o,i}$ is center energy, G_i is the base broadening, and s_i is the Gauss-Lorentz switch parameter.

- Added staged Lorentz-Kim fitting in the Material dialog. The user can choose Direct Fit, Smart Fit, or Smart Fit + Cleanup. Smart Fit progressively builds the model instead of immediately exposing a large free parameter set to Marquardt.

5. Woollam-Style Gaussian Oscillator Model

- Added a new Gaussian Oscillator material model based on the Woollam Gen-Osc style, primarily for HTL-type organic dispersion modeling.

Implemented model structure:

$\varepsilon(E) = \varepsilon\infty + A_UV / (E_UV^2 - E^2) - A_IR / E^2 + \varepsilon_gauss(E)$

$\varepsilon_2,gauss(E) = \Sigma_i A_i \{ \exp[-((E - E_i)/B_i)^2] + \exp[-((E + E_i)/B_i)^2] \}$

$\varepsilon_1,gauss(E)$ is obtained numerically from the Kramers-Kronig transform of $\varepsilon_2,gauss(E)$.

- Below the Urbach transition energy E_t, the implementation switches the imaginary part to an Urbach-tail continuation matched to the Gaussian response at E_t:

$\varepsilon_2,Urbach(E) = A_U \cdot \varepsilon_2(E_t) \cdot \exp[(E - E_t) / E_u], \quad for\ E \le E_t$

- The real part is cached on an energy grid and interpolated during normal evaluation, which keeps the model practical for repeated fitting.

- Added a predefined HTL starting set based on the supplied 10-oscillator HTL parameterization, plus a generic HTL-style starting point.

6. CPExciton and Related Material-Model Updates

- Reviewed CPExciton dimensionality handling and corrected the UI/table behavior so dimensionality values other than the default case are editable and propagate correctly.

- Added CPExciton preset-list handling consistent with other oscillator families.

7. CauchyK-Exp Update

- Updated CauchyK-Exp absorption to use the intended micron-based exponential law.

Implemented absorption expression:

$k(\lambda) = k_amp \cdot \exp[p (1/\lambda - 1/\lambda_ref)]$

- All wavelengths in this expression are in microns. This is aligned with the Woollam Cauchy import path, where imported reference wavelengths are normalized to the micron convention when required.

8. Batch / Production Workflow and Statistics

- Fixed batch-processing cases where running from a parameterization template without an explicit calibration/measurement option could still reach correlation/statistics code paths with too few points and cause exceptions or empty exports.

- Added and exposed new statistical outputs in the batch results workflow.

Raw regression residual standard deviation: fit $M = kT + b$ with

$k = \Sigma (T_i - \bar{T})(M_i - \bar{M}) / \Sigma (T_i - \bar{T})^2$

$b = \bar{M} - k\bar{T}$

$\sigma_reg = \sqrt{[\ \Sigma (\ M_i - (kT_i + b)\ )^2 / (n - 2)\ ]}$

Step fidelity: for all point pairs (i, j), define $\Delta T_{ij} = T_j - T_i$ and $\Delta M_{ij} = M_j - M_i$, then compute

$k_pair = \Sigma (\Delta T_{ij}\ \Delta M_{ij}) / \Sigma (\Delta T_{ij}^2)$

$\sigma_step = \sqrt{[\ \Sigma (\Delta M_{ij} - k_pair\ \Delta T_{ij})^2 / N_pairs\ ]}$

- Updated non-calibration wording so the summary uses SlopeCorr and SlopeCorr s.d. instead of calibration-specific labels when no calibration is actually present.

9. Woollam Model Import

- Added dedicated Woollam model import for materials and filmstacks.

- Supported material conversions in the Woollam-model parser are currently: CAUCHY → CauchyK-Exp, OSC (Lorentz-type layers) → Lorentz-Kim, and GENOSC → Gaussian Oscillator.

- The parser now matches exact layer tags such as start_Layer0 / end_Layer0 and no longer mistakes blocks such as start_Layer1 Misc Parms for real filmstack layers.

- Library Manager now includes Woollam Model as an import option. For material import, all supported materials discovered in selected Woollam files are listed and can be selected before import. The material name is then prompted with the Woollam-provided name as the default so the user can rename it before saving.

- Woollam filmstack import was added to Library Manager as a parallel workflow. The user is prompted for the filmstack name before the stack is saved.

- Filmstack construction rule: if Layer0 contains a Substrate Thickness entry, Layer0 is treated as a finite film layer placed on a Void substrate. Otherwise Layer0 is treated as the substrate and higher-numbered layers are added on top of it.

10. UI and Workflow Fixes

- Enabled direct editing of filmstack-local material names in the main TFCompanion filmstack table. Editing the Material cell changes only the material name inside the current filmstack copy and does not rename any material stored in the database.

- Added relative constraint input support in parameter-constraint tables. Users can type values such as −5% or +5%, and the software converts them to absolute constraint values internally before re-displaying the stored absolute values in the table.

- Fixed material-copy paths for Tabular-based materials so operations such as duplicating or using a layer material as the seed for EMA creation do not fail because of missing copy-constructor behavior.

End of document.