AD FALCON API Manual

Element Activity (Birth / Kill)

Javad Ghorbani

March 26, 2026

Contents

1 Element Activity (Birth / Kill)

This page documents the input-file API for activating (birth) and deactivating (kill) elements during a multi-step analysis.

It covers: - The `% ElementActivity` section syntax - When activity changes are applied - Birth controls (stress-free birth, stiffness ramp-in) - UMAT birth controls (custom-variable reset, UMAT initialization on activation) - GiD output options when the active element set changes

---

1.1 `% ElementActivity` section

1.1.1 Placement

Place the element activity definitions anywhere in the input file. The section ends at `%%%` (or when the next `% ...` section starts).

```
% ElementActivity
...
%%%
```

1.1.2 Line syntax

Each non-empty line uses this form:

```
step <stepId> <action> <elemIdOrRange> [<elemIdOrRange> ...]
```

- `step`: keyword (case-insensitive).
- `<stepId>`: positive integer step ID.
- `<action>` (case-insensitive):
  - `activate` (alias: `birth`) → the listed elements become active.
  - `deactivate` (alias: `kill`) → the listed elements become inactive.
- `<elemIdOrRange>`: one or more element IDs and/or inclusive ranges.

Element lists accept: - IDs separated by whitespace, commas, or semicolons. - Ranges written as `lo-hi` or `lo:hi`.

Duplicates are ignored.

1.1.3 Required fields table

Field	Required	Default	Notes
`step`	Yes	—	Keyword introducing the line.
`<stepId>`	Yes	—	Must be a positive integer step ID.
`<action>`	Yes	—	`activate`/`birth` or `deactivate`/`kill`.
`<elemIdOrRange> ...`	Yes	—	One or more element IDs/ranges.

1.1.4 Example

```
% ElementActivity
step 2 deactivate 97-102
step 3 activate 97-102
%%%
```

1.2 When the change happens

Element activity is applied at **step start**, before the solver begins iterating for that step.
Practical effects:

- Deactivated elements do not contribute stiffness or internal forces.
- Newly activated elements contribute immediately, optionally with a ramp-in scale.
- If a node becomes disconnected from all active elements, its DOFs are automatically parked (removed from the active unknown set). When the node becomes connected again, those DOFs are revived unless the user explicitly restrained them.

1.3 Birth (activation) controls (step directives)

These directives are read inside `% Step Definitions` and are evaluated on steps that activate elements via `% ElementActivity`.

1.3.1 Controls table

Directive	Required	Default	Meaning
`@@ElementBirthStressFree`	No	`No`	Reset mechanical stress/strain state for newly activated elements (pore pressures/saturation are not reset). Alias: `@@ElementActivationStressFree`.
`@@ElementBirthRampTime`	No	`0`	Linearly ramp stiffness/internal-force contribution from `0` → `1` over this duration (simulation time units), starting at activation. Alias: `@@ElementActivationRampTime`.
`@@ElementBirthResetCustomVariables`	No	`No`	For newly activated UMAT elements, set all custom variable values to `0.0` (names preserved). Aliases: `@@ElementActivationResetCustomVariables`, `@@ElementBirthResetCustomState`, `@@ElementActivationResetCustomState`.

Directive	Required	Default	Meaning
`@@ElementBirthInitializeUMAT`	No	`No`	For newly activated UMAT elements, run UMAT initialization at activation time. Aliases: `@@ElementActivationInitializeUMAT`, `@@ElementBirthInitializeCustomVariables`, `@@ElementActivationInitializeCustomVariables`.

1.3.2 What these options mean

The directives above only apply to **newly activated elements** (elements that change from inactive → active at the start of the step).

- `@@ElementBirthStressFree`
  - **When `Yes`**: the element's *mechanical history* is cleared at activation. Conceptually, the element is introduced as a fresh piece of material with zero stored stress/strain history, so it starts contributing without a pre-existing mechanical state.
  - **When `No` (default)**: the element keeps whatever stress/strain history it currently has. This is useful when you are temporarily killing/reactivating elements and you want them to resume from their previous state rather than restarting.
  - **Important**: this option does **not** reset pore pressures, saturation, or other hydraulic state.
    * To control the baseline pore pressures for staged elements, define them via `%InitialAssignments` using `@PW:` / `@PA:` (these set `InitialPoreWaterPressure` / `InitialPoreAirPressure`).
    * If the elements are meant to be inactive from the start of the analysis, schedule them as `deactivate` in `step 1` so they don't contribute in the first step, but they still receive the initial assignment values.
    * See Initial Assignments for the `@PW`/`@PA` syntax.

1.4 Step-triggered initial fields for staged activation (`% Step Initial Assignments`)

Use this section to override initial fields at the **start of a given step**. It runs at **step start** (before birth/kill and before the Newton solve) and **overwrites** the current Gauss-point state.

It is commonly used in activation/deactivation workflows: - active elements receive the override immediately at step start, and - elements scheduled to be activated on that step can be seeded so they start with the desired baseline fields.

1.4.1 Syntax

```
% Step Initial Assignments
@Step <stepId>:
  @PW:   H <a1> values <v1>   H <a2> values <v2>   ...
  @PA:   H <a1> values <v1>   H <a2> values <v2>   ...
  @Void: H <a1> values <v1>   H <a2> values <v2>   ...
%%%
```

Notes:

- Section name variants such as `% StepInitialAssignments` are accepted.
- `@Step` headers are flexible (`@Step 3`, `@Step 3:`, `@Step: 3` all accepted).
- Each variable needs at least **two** `H ... values ...` groups (even for constants).
- Supported headers match `% Initial Assignments`: `@PW`, `@PA`, `@Void`, `@Stress`, `@TotalStress`, `@Damping`, `@Alpha_p_c`, and `@MyCustomVar` (UMAT custom variables).
- Do not mix `@Stress` and `@TotalStress` within the same step.
- `@@ElementBirthRampTime`
  - **When `0` (default)**: the element's stiffness and internal-force contribution turns on immediately at full strength.
  - **When `> 0`**: the element is activated smoothly. Its stiffness/internal-force contribution is scaled from 0 to 1 over the specified duration (in simulation time units) starting from the activation time. This is mainly to reduce sudden stiffness jumps in staged construction/backfill problems.
  - **Note**: the ramp scales stiffness/internal force contributions; it is not a general "material softening" feature and it does not change the element mass/inertia terms.
- `@@ElementBirthResetCustomVariables` (UMAT only)
  - **When `Yes`**: all UMAT custom state-variable values for the newly activated elements are set to `0.0` at activation (variable names remain defined).
  - **When `No` (default)**: custom variables are preserved (reactivated elements resume with their previous custom-state history).
- `@@ElementBirthInitializeUMAT` (UMAT only)

- **When `Yes`**: the UMAT initialization routine is executed for the newly activated elements at activation time (useful if your UMAT expects a dedicated initialization step).
- **When `No` (default)**: no extra initialization is performed at activation; the element continues with its current UMAT/internal state.
- If both `@@ElementBirthResetCustomVariables` and `@@ElementBirthInitializeUMAT` are enabled, the reset happens first and initialization runs afterward.

1.4.2 Example

```
% Step Definitions
@Step 3:
@@ElementBirthStressFree: Yes
@@ElementBirthResetCustomVariables: Yes
@@ElementBirthInitializeUMAT: Yes
@@ElementBirthRampTime: 1000
%%%
```

1.5 Step Materials (`% Step Materials`)

Use `% Step Materials` when you need to overwrite the material assignment of selected elements at a specific step start by referencing a material ID already defined in `% Materials` (common in staged workflows).

See Step Materials for the full syntax, constraints, and restart behavior.

1.6 GiD output options when activity changes

These settings affect GiD mesh/results output when the active element set changes. They apply only when the postprocessor is GiD.

1.6.1 Controls table

Directive	Required	Default	Meaning
@@GIDOmitInactiveElements	No	No	Omit inactive elements from the written GiD mesh connectivity (.post.msh). Alias: @@GIDOmitInactiveElementsInMesh.
@@GIDExcludeElements	No	*(empty)*	Element IDs/ranges that are never written to the GiD mesh connectivity.
@@GIDSplitOnElementActivity	No	No	Start a new GiD segment at the start of a step where % Element Activity schedules any birth/kill.
@@GIDSplitPerStep	No	No	Start a new GiD segment at each step start (step 1 uses base filenames).
@@GIDStartNewFiles	No	No	Start a new GiD segment at this step start only.

Notes:

- @@GIDSplitOnElementActivity and @@GIDSplitPerStep are global modes (the last value read is used).
- @@GIDStartNewFiles is per-step.

1.6.2 Example (split on activity + omit inactive connectivity)

```
% Step Definitions
@Step 1:
@@GIDSplitOnElementActivity: Yes
@@GIDOmitInactiveElements: Yes
%%%
```

1.6.3 Example (exclude elements)

```
% Step Definitions
@Step 1:
@@GIDExcludeElements: 97-102 250
%%%
```