

Releasing a guarantee does not erase its approval decision

One guarantee, two meanings: what was decided earlier and its current lifecycle status. This shows the ordinary approved-then-released flow.

Information	After approval	After release
Current status	Approved	Released
Recorded decision	Approved	Approved, retained
Decision date	Original approval date	Original approval date, retained
Release date	Not yet recorded	Release date recorded

The release callback preserves an existing decision

FinancialGuarantee.OnReleased · lines 209–215

```

L209   ReleasedDate = releasedDate;
L210
L211   if (Decision == FinancialGuaranteeDecision.None)
L212   {
L213       Decision = FinancialGuaranteeDecision.Released;
L214       DecisionDate = releasedDate;
L215   }
```

An Approved decision is not None. This conditional branch therefore does not overwrite the earlier decision or its date.

The assessment template uses both values

Exact expressions from Index.cshtml · not an application screenshot.

```

L31   @EA.Prdsd.Core.Helpers.EnumHelper.GetDisplayName(Model.Decision)
L35   @EA.Prdsd.Core.Helpers.EnumHelper.GetDisplayName(statusToUse)
```

These expressions occupy separate table rows for the decision and current status. The display logic on lines 13–16 retains Released when the decision is Approved.

[Inspect the actual template](#)

Consequence for a replacement: preserve both meanings and their relevant dates, or approve an explicit alternative mapping. A single simplified status value could discard information used by this consumer.

Source: DEFRA PRSD-IWS, commit 95f7eeae92f8. [FinancialGuarantee.cs](#) | [Index.cshtml](#)

The before/after table is a source-derived explanation, not measured application data. Annotations are Harten interpretation; source excerpts are separately identified.