



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# Performance Management Advisors Deployment Guide

Purge Key Action Reports and Historical Alerts

12/20/2025

# Purge Key Action Reports and Historical Alerts

**NEW** Advisors alert and action management features can generate historical alert and action management report data that the Advisors application never removes automatically. An Advisors database administrator can delete or purge the data with a scheduled job or a manual operation.

The historical data purge process relies on properties recorded in the Platform database table `CONFIG_PARAMETER`:

- `keyactions.purging.timeframe.months`: Applicable to CCAdv and WA only.
- `keyactions.purging.successrating.value`: Applicable to CCAdv and WA only. *Success rating* refers to the success rating system used in Key Action Reports (see [Key Action Reports Table](#) for more information about Key Action Reports).
- `keyactions.purging.successrating.range`: Applicable to CCAdv and WA only. *Success rating* refers to the success rating system used in Key Action Reports (see [Key Action Reports Table](#) for more information about Key Action Reports).
- `fa.archive.purging.timeframe.months`: applicable to FA only; purges archived threshold violations

Your database administrator can modify the parameters to address your enterprise's needs and data cleanup policies.

Parameter Name	Parameter Value	Description
keyactions.purging.timeframe.months	12	Number of months to purge historical data.
keyactions.purging.successrating.value	2	The value of the success rating system used in Key Action Reports.
keyactions.purging.successrating.range	<	The range of the success rating system used in Key Action Reports.
fa.archive.purging.timeframe.months	12	Number of months to purge historical data.

Alert and Action Management Report Parameters

The Figure "Alert and Action Management Report Parameters" shows the default values for the relevant parameters in the `CONFIG_PARAMETER` table.

The `keyactions.purging.successrating.value` and `keyactions.purging.successrating.range` parameters depend on each other. If one of the parameters is not defined, the other is ignored.

## Example

To trigger the purge of data related to Action Management reports that have a success rating less than two, you must set `keyactions.purging.successrating.value` to 2 and `keyactions.purging.successrating.range` to `<`.

If the `keyactions.purging.successrating.value` and `keyactions.purging.successrating.range` parameters are not defined, *all* records are removed based on the `keyactions.purging.timeframe.months` parameter setting.

If, in affected Action Management report data, the success rating is not defined (that is, NULL), the records are removed if the related historical alerts meet the `keyactions.purging.timeframe.months` condition.

### Calling Stored Procedures to Purge Key Action Reports and Historical Alerts

Regardless of the method used to purge key action reports and historical alerts (manual operation and/or scheduled job), the process must contain a call of the stored procedure that removes the data from all related tables. The stored procedure has no input parameters. The procedure purges the data based on the criteria generated from the related configuration parameters present in the `CONFIG_PARAMETER` table at the time of procedure execution.

#### MSSQL procedure call

```
EXEC      [spPurgeAMHistory]
          @p_AmrPurged = @p_AmrPurged OUTPUT,
          @p_HstAlertsPurged = @p_HstAlertsPurged OUTPUT,
          @p_HstFATHresholdsPurged = @p_HstFATHresholdsPurged OUTPUT,
          @p_AmrEndDate = @p_AmrEndDate OUTPUT,
          @p_HstAlertEndDate = @p_HstAlertEndDate OUTPUT,
          @p_HstFaThresholdEndDate = @p_HstFaThresholdEndDate OUTPUT,
          @r = @r OUTPUT,
          @m = @m OUTPUT,
          @r1 = @r1 OUTPUT,
          @m1 = @m1 OUTPUT,
          @r2 = @r2 OUTPUT,
          @m2 = @m2 OUTPUT,
          @r3 = @r3 OUTPUT,
          @m3 = @m3 OUTPUT

SELECT    @p_AmrPurged as N'@p_AmrPurged',
          @p_HstAlertsPurged as N'@p_HstAlertsPurged',
          @p_HstFATHresholdsPurged as N'@p_HstFATHresholdsPurged',
          @p_AmrEndDate as N'@p_AmrEndDate',
          @p_HstAlertEndDate as N'@p_HstAlertEndDate',
          @p_HstFaThresholdEndDate as N'@p_HstFaThresholdEndDate',
          @r as N'@r',
          @m as N'@m',
          @r1 as N'@r1',
          @m1 as N'@m1',
          @r2 as N'@r2',
          @m2 as N'@m2',
          @r3 as N'@r3',
          @m3 as N'@m3'
```

GO

#### Oracle procedure call

```
SET SERVEROUTPUT ON
SET FEEDBACK OFF
DECLARE
  P_AMRPURGED NUMBER;
  P_HSTALERTSPURGED NUMBER;
  P_HSTFATHRESHOLDSPURGED NUMBER;
  P_AMRENDDATE DATE;
  P_HSTALERTENDDATE DATE;
  P_HSTFATHRESHOLDENDDATE DATE;
```

```
R NUMBER;
M NVARCHAR2(2000);
R1 NUMBER;
M1 NVARCHAR2(2000);
R2 NUMBER;
M2 NVARCHAR2(2000);
R3 NUMBER;
M3 NVARCHAR2(2000);
BEGIN

SPPURGEAMHISTORY(
  P_AMRPURGED => P_AMRPURGED,
  P_HSTALERTSPURGED => P_HSTALERTSPURGED,
  P_HSTFATHRESHOLDSPURGED => P_HSTFATHRESHOLDSPURGED,
  P_AMRENDDATE => P_AMRENDDATE,
  P_HSTALERTENDDATE => P_HSTALERTENDDATE,
  P_HSTFATHRESHOLDENDDATE => P_HSTFATHRESHOLDENDDATE,
  R => R,
  M => M,
  R1 => R1,
  M1 => M1,
  R2 => R2,
  M2 => M2,
  R3 => R3,
  M3 => M3
);

END;

/
```