

**CaptureDBQueryPlans**

**Purpose:**

The CaptureDBQueryPlans stored procedure is designed to capture and manage query plans from the environment. It provides a flexible way to gather detailed information about query execution, helping you optimize performance and diagnose issues. The stored procedure supports viewing and inserting data into the QueryPlanHistory table, with various configuration options such as filtering by stored procedure name, database, or execution threshold. This document is for DBAs only. The document that is for NOC and System Admins is located here: [PerfStats – QueryPlans.docx](https://alliedsolutions.sharepoint.com/:w:/r/teams/DatabaseServices/Shared%20Documents/AlertSupport/PerfStats%20%E2%80%93%20QueryPlans.docx?d=wb9545ebdfdd14da59bc65b3b4697fd52&csf=1&web=1&e=qtYz4j)

**Usage Scenarios and Examples**

1. **Show data without inserting (WhatIf by default)**:

EXEC PerfStats.dbo.CaptureDBQueryPlans @WhatIf = 1;

* + Displays captured query plan information without storing it in the QueryPlanHistory table.

1. **Show data without inserting (Verbose output)**:

EXEC PerfStats.dbo.CaptureDBQueryPlans @WhatIf = 1, @Verbose = 1;

* + Displays captured query plans and outputs the code being executed.

1. **Capture all plans for a specific stored procedure with the default threshold**:

EXEC PerfStats.dbo.CaptureDBQueryPlans @StoredProcName = 'getAuditLogin', @Verbose = 1;

* + Captures and displays plans for the getAuditLogin stored procedure where execution exceeds the default threshold.

1. **Capture all plans for a specific stored procedure with no threshold**:

EXEC PerfStats.dbo.CaptureDBQueryPlans @StoredProcName = 'getAuditLogin', @ElapsedTimeThreshold = 0;

* + Captures and displays all plans for the getAuditLogin procedure regardless of execution time.
  + For those databases in the ADMIN realm (Perfstats and DBA) then @DatabaseType='ADMIN' variable will need to be added to the request.

1. **Capture all plans for a specific database**:

EXEC PerfStats.dbo.CaptureDBQueryPlans @Database = 'DBA';

* + Captures and displays plans for all queries executed in the specified database.
  + For those databases in the ADMIN realm (Perfstats and DBA) then @DatabaseType='ADMIN' variable will need to be added to the request.

1. **Capture all plans for a specific database with no threshold**:

EXEC PerfStats.dbo.CaptureDBQueryPlans @Database = 'DBA', @ElapsedTimeThreshold = 0;

* + Captures and displays all query plans in the DBA database regardless of execution time.
  + For those databases in the ADMIN realm (Perfstats and DBA) then @DatabaseType='ADMIN' variable will need to be added to the request.

1. **Insert data without showing code and capturing all plans**:

EXEC PerfStats.dbo.CaptureDBQueryPlans @WhatIf = 0, @Verbose = 0, @ElapsedTimeThreshold = 0;

* + Stores captured plans in QueryPlanHistory without outputting the code and with no time threshold.

1. **Insert data and display the code being executed**:

EXEC PerfStats.dbo.CaptureDBQueryPlans @WhatIf = 0, @Verbose = 1;

* + Stores capture plans and display the code for troubleshooting or review.

1. **Insert data without displaying code (default settings)**:

EXEC PerfStats.dbo.CaptureDBQueryPlans @WhatIf = 0, @Verbose = 0;

* + Stores captured plans with the default execution threshold and minimal output.

1. **Clear query plan for a specific stored procedure**:

EXEC PerfStats.dbo.CaptureDBQueryPlans @StoredProcName = 'storedproc ', @Admin = 0;

* + Clears the query plan cache for the storedproc procedure, useful for refreshing outdated plans.
  + If there are no plans to clear, then no work will be done.