





Creating Natively Compiled Stored Procedures

SQL Server 2016 and later

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Natively compiled stored procedures do not implement the full Transact-SQL programmability and query surface area. There are certain Transact-SQL constructs that cannot be used inside natively compiled stored procedures. For more information, see [Supported Features for Natively Compiled T-SQL Modules](#).

There are, however, several Transact-SQL features that are only supported for natively compiled stored procedures:

- Atomic blocks. For more information, see [Atomic Blocks](#).
- **NOT NULL** constraints on parameters of and variables in natively compiled stored procedures. You cannot assign **NULL** values to parameters or variables declared as **NOT NULL**. For more information, see [DECLARE @local_variable \(Transact-SQL\)](#).
 - CREATE PROCEDURE dbo.myproc (@myVarchar varchar(32) **not null**) ...
 - DECLARE @myVarchar varchar(32) **not null** = "Hello"; -- (Must initialize to a value.)
 - SET @myVarchar = **null**; -- (Compiles, but fails during run time.)
- Schema binding of natively compiled stored procedures.

Natively compiled stored procedures are created using [CREATE PROCEDURE \(Transact-SQL\)](#). The following example shows a memory-optimized table and a natively compiled stored procedure used for inserting rows into the table.

Transact-SQL

```
create table dbo.Ord
(OrdNo integer not null primary key nonclustered,
 OrdDate datetime not null,
 CustCode nvarchar(5) not null)
with (memory_optimized=on)
go

create procedure dbo.OrderInsert(@OrdNo integer, @CustCode nvarchar(5))
with native_compilation, schemabinding
as
begin atomic with
(transaction isolation level = snapshot,
 language = N'English')
```

```
declare @OrdDate datetime = getdate();
insert into dbo.Ord (OrdNo, CustCode, OrdDate) values (@OrdNo, @CustCode, @OrdDate);
end
go
```

In the code sample, **NATIVE_COMPILATION** indicates that this Transact-SQL stored procedure is a natively compiled stored procedure. The following options are required:

Option	Description
SCHEMABINDING	A natively compiled stored procedure must be bound to the schema of the objects it references. This means that tables referenced by the procedure cannot be dropped. Tables referenced in the procedure must include their schema name, and wildcards (*) are not allowed in queries (meaning no <code>SELECT * from . . .</code>). SCHEMABINDING is only supported for natively compiled stored procedures in this version of SQL Server.
BEGIN ATOMIC	<p>The natively compiled stored procedure body must consist of exactly one atomic block. Atomic blocks guarantee atomic execution of the stored procedure. If the procedure is invoked outside the context of an active transaction, it will start a new transaction, which commits at the end of the atomic block. Atomic blocks in natively compiled stored procedures have two required options:</p> <p>TRANSACTION ISOLATION LEVEL. See Transaction Isolation Levels for Memory-Optimized Tables for supported isolation levels.</p> <p>LANGUAGE. The language for the stored procedure must be set to one of the available languages or language aliases.</p>

See Also

[Natively Compiled Stored Procedures](#)

Community Additions

© 2017 Microsoft