

Create Database

If required, replace all Privilege Manager entries with desired name of the database (for example: prod_PrivilegeManager). Database settings and files can be modified but file groups created by this script are required.

```
USE [master]
GO

IF NOT EXISTS (SELECT * FROM [master].[sys].[databases] WHERE name = N'PrivilegeManager')
BEGIN
    PRINT 'Create database: PrivilegeManager'
    CREATE DATABASE [PrivilegeManager] COLLATE SQL_Latin1_General_CP1_CI_AS
END
GO

IF NOT EXISTS (SELECT * FROM [PrivilegeManager].[sys].[objects] WHERE name = N'tableSystemValues' AND type in (
N'U'))
BEGIN
    PRINT 'Set database settings'
    IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
    BEGIN
        EXEC [PrivilegeManager].[dbo].[sp_fulltext_database] @action = 'disable'
    END
    ALTER DATABASE [PrivilegeManager] SET ANSI_NULL_DEFAULT OFF
    ALTER DATABASE [PrivilegeManager] SET ANSI_NULLS OFF
    ALTER DATABASE [PrivilegeManager] SET ANSI_PADDING OFF
    ALTER DATABASE [PrivilegeManager] SET ANSI_WARNINGS OFF
    ALTER DATABASE [PrivilegeManager] SET ARITHABORT OFF
    ALTER DATABASE [PrivilegeManager] SET AUTO_CLOSE OFF
    ALTER DATABASE [PrivilegeManager] SET AUTO_CREATE_STATISTICS ON
    ALTER DATABASE [PrivilegeManager] SET AUTO_SHRINK OFF
    ALTER DATABASE [PrivilegeManager] SET AUTO_UPDATE_STATISTICS ON
    ALTER DATABASE [PrivilegeManager] SET CURSOR_CLOSE_ON_COMMIT OFF
    ALTER DATABASE [PrivilegeManager] SET CURSOR_DEFAULT GLOBAL
    ALTER DATABASE [PrivilegeManager] SET CONCAT_NULL_YIELDS_NULL OFF
    ALTER DATABASE [PrivilegeManager] SET NUMERIC_ROUNDABORT OFF
    ALTER DATABASE [PrivilegeManager] SET QUOTED_IDENTIFIER OFF
    ALTER DATABASE [PrivilegeManager] SET RECURSIVE_TRIGGERS OFF
    ALTER DATABASE [PrivilegeManager] SET DISABLE_BROKER
    ALTER DATABASE [PrivilegeManager] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
    ALTER DATABASE [PrivilegeManager] SET DATE_CORRELATION_OPTIMIZATION OFF
    ALTER DATABASE [PrivilegeManager] SET TRUSTWORTHY OFF
    ALTER DATABASE [PrivilegeManager] SET ALLOW_SNAPSHOT_ISOLATION OFF
    ALTER DATABASE [PrivilegeManager] SET PARAMETERIZATION SIMPLE
    ALTER DATABASE [PrivilegeManager] SET READ_WRITE
    ALTER DATABASE [PrivilegeManager] SET RECOVERY FULL
    ALTER DATABASE [PrivilegeManager] SET MULTI_USER
    ALTER DATABASE [PrivilegeManager] SET PAGE_VERIFY CHECKSUM
    ALTER DATABASE [PrivilegeManager] SET DB_CHAINING OFF
END
GO

IF NOT EXISTS (SELECT * FROM [PrivilegeManager].[sys].[filegroups] WHERE name = N'Logging')
BEGIN
    PRINT 'Create filegroup: Logging'
    ALTER DATABASE [PrivilegeManager] ADD FILEGROUP [Logging]
END
GO

IF NOT EXISTS (SELECT * FROM [PrivilegeManager].[sys].[filegroups] WHERE name = N'History')
BEGIN
    PRINT 'Create filegroup: History'
```

```
ALTER DATABASE [PrivilegeManager] ADD FILEGROUP [History]
```

```
END  
GO
```

```
IF NOT EXISTS (SELECT * FROM [PrivilegeManager].[sys].[filegroups] WHERE name = N'Reports')
```

```
BEGIN  
PRINT 'Create filegroup: Reports'  
ALTER DATABASE [PrivilegeManager] ADD FILEGROUP [Reports]  
END  
GO
```

```
IF EXISTS (SELECT * FROM [PrivilegeManager].[sys].[database_files] WHERE name = N'PrivilegeManager_log' AND size < 3200)
```

```
BEGIN  
PRINT 'Modify file: PrivilegeManager_log'  
ALTER DATABASE [PrivilegeManager] MODIFY FILE ( NAME = N'PrivilegeManager_log', SIZE = 25MB , MAXSIZE = UNLIMITED, FILEGROWTH = 25MB )  
END  
GO
```

```
IF EXISTS (SELECT * FROM [PrivilegeManager].[sys].[database_files] WHERE name = N'PrivilegeManager' AND size < 6400)
```

```
BEGIN  
PRINT 'Modify file: PrivilegeManager'  
ALTER DATABASE [PrivilegeManager] MODIFY FILE ( NAME = N'PrivilegeManager', SIZE = 50MB , MAXSIZE = UNLIMITED , FILEGROWTH = 50MB )  
END  
GO
```

```
DECLARE @stringFolder varchar(500)  
DECLARE @stringAddFile nvarchar(1000)
```

```
SELECT @stringFolder = REVERSE(SUBSTRING(REVERSE(physical_name), CHARINDEX('\', REVERSE(physical_name)), 500)) FROM [PrivilegeManager].[sys].[database_files] WHERE name = N'PrivilegeManager'
```

```
IF NOT EXISTS (SELECT * FROM [PrivilegeManager].[sys].[database_files] WHERE name = N'PrivilegeManager_logging1')
```

```
BEGIN  
PRINT 'Create database file: PrivilegeManager_logging1'  
SET @stringAddFile = N'ALTER DATABASE [PrivilegeManager] ADD FILE ( NAME = N''PrivilegeManager_logging1'', FILENAME = N''' + @stringFolder + 'Carillon_logging1.ndf', SIZE = 25MB , MAXSIZE = UNLIMITED , FILEGROWTH = 25MB ) TO FILEGROUP [Logging]'  
EXEC (@stringAddFile)  
END
```

```
IF NOT EXISTS (SELECT * FROM [PrivilegeManager].[sys].[database_files] WHERE name = N'PrivilegeManager_history1')
```

```
BEGIN  
PRINT 'Create database file: PrivilegeManager_history1'  
SET @stringAddFile = N'ALTER DATABASE [PrivilegeManager] ADD FILE ( NAME = N''PrivilegeManager_history1'', FILENAME = N''' + @stringFolder + 'PrivilegeManager_history1.ndf', SIZE = 100MB , MAXSIZE = UNLIMITED , FILEGROWTH = 50MB ) TO FILEGROUP [History]'  
EXEC (@stringAddFile)  
END
```

```
IF NOT EXISTS (SELECT * FROM [PrivilegeManager].[sys].[database_files] WHERE name = N'PrivilegeManager_reports1')
```

```
BEGIN  
PRINT 'Create database file: PrivilegeManager_reports1'  
SET @stringAddFile = N'ALTER DATABASE [PrivilegeManager] ADD FILE ( NAME = N''PrivilegeManager_reports1'', FILENAME = N''' + @stringFolder + 'PrivilegeManager_reports1.ndf', SIZE = 100MB , MAXSIZE = UNLIMITED , FILEGROWTH = 50MB ) TO FILEGROUP [Reports]'  
EXEC (@stringAddFile)  
END
```

END

```
USE [PrivilegeManager]
GO
```

```
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[tableSystemValues]') AND type in (N'U'))
```

```
BEGIN
```

```
PRINT ' Table: [dbo].[tableSystemValues]'
```

```
CREATE TABLE [dbo].[tableSystemValues](
  [integerSystemValueID] [int] IDENTITY(1,1) NOT NULL,
```

```
  [stringValueTitle] [varchar](50) NOT NULL,
```

```
  [stringValue] [nvarchar](4000) NULL,
```

```
  [integerValue] [bigint] NULL,
```

```
  [binaryValue] [varbinary](256) NULL,
```

```
  [datetimeValue] [datetime] NULL,
```

```
  [modified] [timestamp] NOT NULL,
```

```
  CONSTRAINT [PK_tableSystemValues] PRIMARY KEY CLUSTERED
```

```
(
```

```
  [integerSystemValueID] ASC
```

```
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
```

```
) ON [PRIMARY]
```

```
END
```

```
GO
```

```
IF (SELECT COUNT(*) FROM [dbo].[tableSystemValues] WHERE stringValueTitle = 'Application') = 0
```

```
  INSERT INTO [dbo].[tableSystemValues](stringValueTitle, stringValue) VALUES ('Application', 'PrivilegeManager')
```

```
ELSE
```

```
  UPDATE [dbo].[tableSystemValues] SET stringValue = 'PrivilegeManager' WHERE stringValueTitle = 'Application' AND
  stringValue <> 'PrivilegeManager'
```