

## مقدمه

در اکثر موارد در یک Landscape عملیاتی، چنانچه به تجمیع و انتقال داده‌ها از بانک‌های اطلاعاتی مختلف نیاز باشد، از SSIS Package اختصار (SQL Server Integration Service) استفاده می‌شود و معمولاً با تعریف یک Job در سطح SQL Server به اجرای Package در زمانهای مشخص می‌پردازند. چنانچه در موقعیتی لازم باشد که از طریق برنامه کاربردی توسعه یافته، به اجرای Package مبادرت ورزیده شود و البته نخواهیم Job تعریف شده را از طریق کد برنامه، اجرا کنیم و در واقع این امکان را داشته باشیم که همانند یک رویه ذخیره شده تعریف شده در سطح بانک اطلاعاتی به اجرای عمل فوق پردازیم، یک راه حل می‌تواند تعریف یک CLR Stored Procedures باشد. در این مقاله به بررسی این موضوع پرداخته می‌شود، در ابتدا لازم است به بیان تئوری موضوع پرداخته شود (قسمت‌های 1 الی 5) در ادامه به ذکر پیاده سازی روش پیشنهادی پرداخته می‌شود.

## 1- اجرای Integration Service Package جهت اجرای یک Package از ابزارهای زیر می‌توان استفاده کرد:

- command-line ابزار خط فرمان dtexec.exe

- ابزار اجرایی پکیج dtexecui.exe

- استفاده از job SQL Server Agent

**توجه:** همچنین یک Package را در زمان طراحی در BIDS (Business Intelligence Development Studio) می‌توان اجرا نمود.

## 2- استفاده از dtexec جهت اجرای Package با استفاده از ابزار dtexec می‌توان Package‌های ذخیره شده در فایل سیستم، یک SQL Instance و یا Package‌های ذخیره شده در Integration Service را اجرا نمود.

**توجه:** در سیستم عامل‌های 64 بیتی، ابزار dtexec موجود در Integration Service با نسخه 64 بیتی نصب می‌شود. چنانچه بایست Package‌های معینی را در حالت 32 بیتی اجرا کنید، لازم است ابزار dtexec نسخه 32 بیتی نصب شود. ابزار dtexec دستیابی به تمامی ویژگی‌های پیکربندی و اجرای Package از قبیل اتصالات، مشخصات (Properties)، متغیرها، logging و شاخص‌های پردازشی را فراهم می‌کند.

**توجه:** زمانی که از نسخه‌ی ابزار dtexec که با SQL Server 2008 ارائه شده استفاده می‌کنید برای اجرای یک SSIS Package نسخه 2005، Integration Service به صورت موقت Package را به نسخه 2008 ارتقا می‌دهد، اما نمی‌توان از ابزار dtexec برای ذخیره این تغییرات استفاده کرد.

## 2-1- ملاحظات نصب dtexec روی سیستم‌های 64 بیتی به صورت پیش فرض، یک سیستم عامل 64 بیتی که هر دو نسخه 64 بیتی و 32 بیتی ابزار خط فرمان Integration Service را دارد، نسخه 32 بیتی نصب شده را در خط فرمان اجرا خواهد کرد. نسخه 32 بیتی بدین دلیل اجرا می‌شود که در متغیر محیطی (Path environment variable) مسیر directory نسخه 32 بیتی قرار گرفته است. به طور معمول:

```
(<drive>:\Program Files(x86)\Microsoft SQL Server\100\DTS\Binn)
```

**توجه:** اگر از SQL Server Agent برای اجرای Package استفاده می‌کنید، SQL Server Agent به طور خودکار از ابزار نسخه 64 بیتی استفاده می‌کند. SQL Server Agent از Registry و نه از متغیر محیطی Path استفاده می‌کند. برای اطمینان از اینکه نسخه 64 بیتی این ابزار را در خط فرمان اجرا می‌کنید، directory را به directory ای تغییر دهید که شامل نسخه 64 بیتی این ابزار است (>drive:\Program Files\Microsoft SQL Server\100\DTS\Binn) و ابزار را از این مسیر اجرا کنید و یا برای همیشه مسیر قرار گرفته در متغیر محیطی path را با مسیری که نسخه 64 بیتی قرار دارد، جایگزین کنید.

## 2-2- تفسیر کدهای خروجی هنگامی که یک Package اجرا می‌شود، dtexec یک کد خروجی (Return Code) بر می‌گرداند:

مقدار	توصیف
0	Package با موفقیت اجرا شده است.
1	Package با خطا مواجه شده است.
3	Package در حال اجرا توسط کاربر لغو شده است.
4	Package پیدا نشده است.
5	Package بارگذاری نشده است.
6	ابزار با یک خطای نحوی یا خطای معنایی در خط فرمان برخورد کرده است.

**2-3-2- قوانین نحوی dtexec** تمامی گزینه‌ها (Options) باید با یک علامت Slash (/) و یا Minus (-) شروع شوند. یک آرگومان باید در یک quotation mark محصور شود چنانچه شامل یک فاصله خالی باشد. گزینه‌ها و آرگومان‌ها بجز رمزعبور حساس به حروف کوچک و بزرگ نیستند.

#### Syntax 2-3-1-

```
dtexec /option [value] [/option [value]]...
```

**2-3-2- Parameters نکته:** در Integration Service، ابزار خط فرمان dtsrun که برای DTS (Data Transformation Service) های نسخه SQL Server 2000 استفاده می‌شد، با ابزار خط فرمان dtexec جایگزین شده است.

- تعدادی از گزینه‌های خط فرمان dtsrun به طور مستقیم در dtexec معادل دارند برای مثال نام Server و نام Package.
- تعدادی از گزینه‌های dtsrun به طور مستقیم در dtexec معادل ندارند.
- تعدادی از گزینه‌های خط فرمان جدید dtexec وجود دارد که در ویژگی‌های جدید Integration Service پشتیبانی می‌شود.

**2-3-3- مثال 1)** به منظور اجرای یک SSIS Package که در SQL Server ذخیره شده است، با استفاده از Windows Authentication :

```
dtexec /sq <Package Name> /ser <Server Name>
```

**(2)** به منظور اجرای یک SSIS Package که در پوشه File System در SSIS Package Store ذخیره شده است :

```
dtexec /dts "\File System\<Package File Name>"
```

**(3)** به منظور اجرای یک SSIS Package که در سیستم فایل ذخیره شده است و مشخص کردن گزینه logging:

```
dtexec /f "c:\<Package File Name>" /l "DTS.LogProviderTextFile; <Log File Name>"
```

**(4)** به منظور اجرای یک SSIS Package که در SQL Server ذخیره شده با استفاده از SQL Server Authentication برای نمونه (user:ssis;pwd:ssis@ssis) و رمز (123):Package

```
dtexec /server "<Server Name>" /sql "<Package Name>" / user "ssis" /Password "ssis@ssis" /De "123"
```

**3- تنظیمات سطح حفاظتی یک Package** به منظور حفاظت از داده‌ها در Package های Integration Service می‌توانید یک سطح

حفاظتی (protection level) را تنظیم کنید که به حفاظت از داده‌های صرفاً حساس یا تمامی داده‌های یک Package کمک نماید. به علاوه می‌توانید این داده‌ها را با یک Password یا یک User Key رمزگذاری نمائید یا به رمزگذاری داده‌ها در بانک اطلاعاتی اعتماد کنید. همچنین سطح حفاظتی که برای یک Package استفاده می‌کنید، الزاماً ایستا (static) نیست و در طول چرخه حیات یک Package می‌تواند تغییر کند. اغلب سطح حفاظتی در طول توسعه یا به محض (deploy) استقرار Package تنظیم می‌شود.

**توجه:** علاوه بر سطوح حفاظتی که توصیف شد، Package‌ها در بانک اطلاعاتی msdb ذخیره می‌شوند که همچنین می‌توانند توسط نقش‌های ثابت در سطح بانک اطلاعاتی (fixed database-level roles) حفاظت شوند. Integration Service شامل 3 نقش ثابت بانک اطلاعاتی برای نسبت دادن مجوزها به Package است که عبارتند از db\_ssisadmin, db\_ssisltduser و db\_ssisoperator

#### 3-1-1- درک سطوح حفاظتی در یک Package اطلاعات زیر به عنوان حساس تعریف می‌شوند:

- بخش password در یک connection string. گرچه، اگر گزینه ای را که همه چیز را رمزگذاری کند، انتخاب کنید تمامی connection string حساس در نظر گرفته می‌شود.
- گره‌های XML task-generated که برچسب (tagged) هایی حساس هستند.
- هر تغییری که به عنوان حساس نشان گذاری شود.

**3-1-1-1 Do not save sensitive** هنگامی که Package ذخیره می‌شود از ذخیره مقادیر ویژگی‌های حساس در Package جلوگیری می‌کند. این سطح حفاظتی رمزگذاری نمی‌کند اما در عوض از ذخیره شدن ویژگی هایی که حساس نشان گذاری شده اند به همراه Package جلوگیری می‌کند.

**3-1-2- Encrypt all with password** به منظور رمزگذاری تمامی Package از یک Password استفاده می‌شود. Package توسط Password ای رمزگذاری می‌شود که کاربر هنگامی که Package را ایجاد یا Export می‌کند، ارائه می‌دهد. به منظور باز کردن Package در SSIS Designer یا اجرای Package توسط ابزار خط فرمان dtexec کاربر بایست رمز Package را ارائه نماید. بدون رمز کاربر قادر به دستیابی و اجرای Package نیست.

**3-1-3- Encrypt all with user key** به منظور رمزگذاری تمامی Package از یک کلید که مبتنی بر Profile کاربر جاری می‌باشد، استفاده می‌شود. تنها کاربری که Package را ایجاد یا Export می‌کند، می‌تواند Package را در SSIS Designer باز کند و یا Package را توسط ابزار خط فرمان dtexec اجرا کند.

**3-1-4- Encrypt sensitive with password** به منظور رمزگذاری تنها مقادیر ویژگی‌های حساس در Package از یک Password استفاده می‌شود. برای رمزگذاری از DPAPI استفاده می‌شود. داده‌های حساس به عنوان بخشی از Package ذخیره می‌شوند اما آن داده‌ها با استفاده از Password رمزگذاری می‌شوند. به منظور باز نمودن Package در SSIS Designer کاربر باید رمز Package را ارائه دهد. اگر رمز ارائه نشود، Package بدون داده‌های حساس باز می‌شود و کاربر باید مقادیر جدیدی برای داده‌های حساس فراهم کند. اگر کاربر سعی نماید Package را بدون ارائه رمز اجرا کند، اجرای Package با خطا مواجه می‌شود.

**3-1-5- Encrypt sensitive with user key** به منظور رمزگذاری تنها مقادیر ویژگی‌های حساس در Package از یک کلید که مبتنی بر Profile کاربر جاری می‌باشد، استفاده می‌شود. تنها کاربری که از همان Profile استفاده می‌کند، Package را می‌تواند بارگذاری (load) کند. اگر کاربر متفاوتی Package را باز نماید، اطلاعات حساس با مقادیر پوچی جایگزین می‌شود و کاربر باید مقادیر جدیدی برای داده‌های حساس فراهم کند. اگر کاربر سعی نماید Package را بدون ارائه رمز اجرا کند، اجرای Package با خطا مواجه می‌شود. برای رمزگذاری از DPAPI استفاده می‌شود.

**3-1-6- Rely on server storage for encryption (ServerStorage)** با استفاده از نقش‌های بانک اطلاعاتی، SQL Server تمامی Package را حفاظت می‌کند. این گزینه تنها زمانی پشتیبانی می‌شود که Package در بانک اطلاعاتی msdb ذخیره شده است.

**4- استفاده از نقش‌های Integration Service** برای کنترل کردن دستیابی به SSIS، Package شامل 3 نقش ثابت در سطح بانک اطلاعاتی است. نقش‌ها می‌توانند تنها روی Package هایی که در بانک اطلاعاتی msdb ذخیره شده اند، بکار روند. با استفاده از SSMS می‌توانید نقش‌ها را به Package نسبت دهید، این انتساب نقش‌ها در بانک اطلاعاتی msdb ذخیره می‌شود.

Role	Read action	Write action
<b>db_ssisadmin or sysadmin</b>	Enumerate own packages Enumerate all packages View own packages View all packages Execute own packages Execute all packages Export own packages Export all packages Execute all packages in SQL Server Agent	Import packages Delete own packages Delete all packages Change own package roles Change all package roles  * به نکته رجوع شود
<b>db_ssisltduser</b>	Enumerate own packages Enumerate all packages View own packages Execute own packages Export own packages	Import packages Delete own packages Change own package roles
<b>db_ssisoperator</b>	Enumerate all packages View all packages Execute all packages Export all packages Execute all packages in SQL Server Agent	None
<b>Windows administrators</b>	View execution details of all running packages	Stop all currently running packages

**\* نکته:** اعضای نقش‌های db\_ssisadmin و dc\_admin ممکن است قادر باشند مجوزهای خودشان را تا سطح sysadmin ارتقا دهند. براساس این ترفیع مجوز امکان اصلاح و اجرای Package‌ها از طریق SQL Server Agent میسر می‌شود. برای محافظت در برابر این ارتقا، با استفاده از یک (account) حساب Proxy با دسترسی محدود، Job هایی که این Package‌ها را اجرا می‌کنند، پیکربندی شوند یا تنها اعضای نقش sysadmin به نقش‌های db\_ssisadmin و dc\_admin افزوده شوند.

همچنین جدول sysssispackages در بانک اطلاعاتی msdb شامل Package هایی است که در SQL Server ذخیره می‌شوند. این جدول شامل ستون هایی که اطلاعاتی درباره نقش هایی که به Package‌ها نسبت داده شده است، می‌باشد. به صورت پیش فرض، مجوزهای نقش‌های ثابت بانک اطلاعاتی db\_ssisadmin و db\_ssisoperator و شناسه منحصر به فرد کاربری (unique security identifier) که Package را ایجاد کرده برای خواندن Package بکار می‌رود، و مجوزهای نقش db\_ssisadmin و شناسه منحصر به فرد کاربری که Package را ایجاد کرده برای نوشتن Package به کار می‌رود. یک User باید عضو نقش db\_ssisadmin و db\_ssisltduser یا db\_ssisoperator برای داشتن دسترسی خواندن Package باشد. یک User باید عضو نقش db\_ssisadmin برای داشتن دسترسی نوشتن Package باشد.

**5- اتصال به صورت Remote به Integration Service** زمانی که یک کاربر بدون داشتن دسترسی کافی تلاش کند به یک Integration Service به صورت Remote متصل شود، با پیغام خطای "Access is denied" مواجه می‌شود. برای اجتناب از این پیغام خطا می‌توان تضمین کرد که کاربر مجوز مورد نیاز DCOM را دارد. به منظور پیکربندی کردن دسترسی کاربر به صورت Remote به سرویس Integration مراحل زیر را دنبال کنید:

- Component Service را باز نمایید ( در Run عبارت dcomcnfg را تایپ کنید).

- گره Component Service را باز کنید، گره Computer و سپس My Computer را باز نمایید و روی DCOM Config کلیک نمایید.
- گره DCOM Config را باز کنید و از لیست برنامه هایی که می‌توانند پیکربندی شوند MsDtsServer را انتخاب کنید.
- روی Properties برنامه MsDtsServer رفته و قسمت Security را انتخاب کنید.
- در قسمت Lunch and Activation Permissions، مورد Customize را انتخاب و سپس روی Edit کلیک نمایید تا پنجره Lunch Permission باز شود.
- در پنجره Lunch Permission، کاربران را اضافه و یا حذف کنید و مجوزهای مناسب را به کاربران یا گروه‌های مناسب نسبت دهید. مجوزهای موجود عبارتند از Local Lunch، Remote Lunch، Local Activation و Remote Activation.
- در قسمت Access Permission مراحل فوق را به منظور نسبت دادن مجوزهای مناسب به کاربران یا گروه‌های مناسب انجام دهید.
- سرویس Integration را Restart کنید.
- مجوز دسترسی Lunch به منظور شروع و خاتمه سرویس، اعطا یا رد می‌شود و مجوز دسترسی Activation به منظور متصل شدن به سرویس، اعطا (grant) یا رد (deny) می‌شود.

**6- پیاده سازی در ابتدا به ایجاد یک CLR Stored Procedures پرداخته می‌شود نام اسمبلی ساخته شده به این نام RunningPackage.dll می‌باشد و حاوی کد زیر است:**

```
Partial Public Class StoredProcedures
    '-----
    'exec dbo.Spc_NtDtexec 'Package','ssis','ssis@ssis','1234512345'
    '-----
    <Microsoft.SqlServer.Server.SqlProcedure>
    Public Shared Sub Spc_NtDtexec(ByVal PackageName As String, _
                                   ByVal UserName As String, _
                                   ByVal Password As String, _
                                   ByVal Decrypt As String)
        Dim p As New System.Diagnostics.Process()
        p.StartInfo.FileName = "C:\Program Files\Microsoft SQL Server\100\DTS\Binn\DTExec.exe"
        p.StartInfo.RedirectStandardOutput = True
        p.StartInfo.Arguments = "/sql " & PackageName & " /User " & UserName & " /Password " & Password
        & " /De " & Decrypt
        p.StartInfo.UseShellExecute = False
        p.Start()
        p.WaitForExit()
        Dim output As String
        output = p.StandardOutput.ReadToEnd()
        Microsoft.SqlServer.Server.SqlContext.Pipe.Send(output)
    End Sub
End Class
```

در حقیقت توسط این رویه به اجرای برنامه dtexec.exe و ارسال پارامترهای مورد نیاز جهت اجرا پرداخته می‌شود. با توجه به توضیحات تئوری بیان شده، سطح حفاظتی Package ایجاد شده Encrypt all with password توصیه می‌شود که رمز مذکور در قالب یکی از پارامتر ارسالی به رویه ساخته شده موسوم به Spc\_NtDtexec ارسال می‌گردد.

در قدم بعدی نیاز به Register کردن dll ساخته شده در سطح بانک اطلاعاتی SQL Server است، این گام‌ها پس از اتصال به SQL Server Management Studio به شرح زیر است:

#### 1- فعال کردن CLR در سرویس SQL Server

```
SP_CONFIGURE 'clr enabled',1
GO
RECONFIGURE
```

#### 2- فعال کردن ویژگی TRUSTWORTHY در بانک اطلاعاتی مورد نظر

```
ALTER DATABASE <Database Name> SET TRUSTWORTHY ON
GO
RECONFIGURE
```

## 3- ایجاد Assembly و Stored Procedure در بانک اطلاعاتی مورد نظر

Assembly ساخته شده با نام RunningPacakge.dll در ریشه C: کپی شود. بعد از ثبت نمودن این Assembly لزومی به وجود آن نمی‌باشد.

```
USE <Database Name>
GO
CREATE ASSEMBLY [RunningPackage]
AUTHORIZATION [dbo]
FROM 'C:\RunningPackage.dll'
WITH PERMISSION_SET = UNSAFE
Go
CREATE PROCEDURE [dbo].[Spc_NtDtexec]
@PackageName [nvarchar](50),
@UserName [nvarchar](50),
@Password [nvarchar](50),
@Decrypt [nvarchar](50)
WITH EXECUTE AS CALLER
AS
EXTERNAL NAME [RunningPackage].[RunningPackage.StoredProcedures].[Spc_NtDtexec]
GO
```

**توجه:** Application User برنامه بایست دسترسی اجرای رویه ذخیره شده Spc\_NtDtexec را در بانک اطلاعاتی مورد نظر داشته باشد همچنین بایست عضو نقش db\_ssisoperator در بانک اطلاعاتی msdb باشد. (منظور از Application User، لاگین است که در Connection string برنامه قرار داده اید.)

در برنامه کاربردی تان کافی است متدی به شکل زیر ایجاد و با توجه به نیازتان در برنامه به فراخوانی آن و اجرای Package بپردازید.

```
Private Sub ExecutePackage()
    Dim oSqlConnection As SqlClient.SqlConnection
    Dim oSqlCommand As SqlClient.SqlCommand
    Dim strCnt As String = String.Empty
    strCnt = "Data Source=" & txtServer.Text & ";User ID=" & txtUsername.Text & ";Password=" &
    txtPassword.Text & ";Initial Catalog=" & cmbDatabaseName.SelectedValue.ToString() & ";"
    Try
        oSqlConnection = New SqlClient.SqlConnection(strCnt)
        oSqlCommand = New SqlClient.SqlCommand
        With oSqlCommand
            .Connection = oSqlConnection
            .CommandType = System.Data.CommandType.StoredProcedure
            .CommandText = "dbo.Spc_NtDtexec"
            .Parameters.Clear()
            .Parameters.Add("@PackageName", System.Data.SqlDbType.VarChar, 50)
            .Parameters.Add("@UserName", System.Data.SqlDbType.VarChar, 50)
            .Parameters.Add("@Password", System.Data.SqlDbType.VarChar, 50)
            .Parameters.Add("@Decrypt", System.Data.SqlDbType.VarChar, 50)
            .Parameters("@PackageName").Value = txtPackageName.Text.Trim()
            .Parameters("@UserName").Value = txtUsername.Text.Trim()
            .Parameters("@Password").Value = txtPassword.Text.Trim()
            .Parameters("@Decrypt").Value = txtDecrypt.Text.Trim()
        End With
        If (oSqlCommand.Connection.State <> System.Data.ConnectionState.Open) Then
            oSqlCommand.Connection.Open()
            oSqlCommand.ExecuteNonQuery()
            System.Windows.Forms.MessageBox.Show("Success")
        End If
        If (oSqlCommand.Connection.State = System.Data.ConnectionState.Open) Then
            oSqlCommand.Connection.Close()
        End If
    Catch ex As Exception
        MessageBox.Show(ex.Message, "Error")
    End Try
End Sub 'ExecutePackage
```