

Secretaria da Fazenda do Estado  
de São Paulo



# Guia de boas práticas de desenvolvimento – Integration Services (SSIS)

Preparado por: Liliam Leme, Renata Festa, Sergio Fonseca

Data: 10/12/2015

As informações contidas neste documento representam a visão atual da Microsoft Corporation sobre os temas discutidos na data da publicação. Como a Microsoft deve responder às mudanças das condições de mercado, este documento não deve ser interpretado como um compromisso da parte da Microsoft, e a Microsoft não pode assegurar a exatidão de qualquer informação apresentada após a data da publicação.

A MICROSOFT NÃO OFERECE GARANTIAS, EXPRESSAS, IMPLÍCITAS OU ESTATUTÁRIAS, QUANTO ÀS INFORMAÇÕES NESTE DOCUMENTO.

O cumprimento de todas as leis pertinentes de direitos autorais é de responsabilidade do usuário. Sem limitar os direitos sob o direito autoral, nenhuma parte deste documento pode ser reproduzida, armazenada em, ou inserida em um sistema de recuperação de dados, ou transmitida de qualquer forma ou por qualquer meio (eletrônico, mecânico, fotocópia, gravação ou outro), ou para qualquer fim, sem a permissão expressa por escrito da Microsoft Corporation.

A Microsoft pode deter as patentes, as solicitações de patentes, as marcas comerciais, os direitos autorais ou outras propriedades intelectuais pertinentes ao objeto deste documento. Exceto quando houver descrição expressa contida em qualquer contrato de licenciamento da Microsoft, o uso deste documento não concede qualquer licença a patentes, marcas comerciais, direitos autorais ou outro tipo de propriedade intelectual.

As descrições de produtos de outras empresas neste documento, caso haja, são fornecidas somente como uma conveniência para você. Nenhuma dessas referências deve ser considerada como endosso ou apoio da Microsoft. A Microsoft não pode garantir a exatidão e os produtos podem mudar ao longo do tempo. Além disso, as descrições destinam-se a ser destaques breves para auxiliar a compreensão, e não como cobertura completa. Para descrições detalhadas desses produtos, consulte os respectivos fabricantes.

© 2013 Microsoft Corporation. Todos os direitos reservados.

Microsoft, Windows e SQL Server são marcas registradas ou comerciais da Microsoft Corporation nos Estados Unidos e/ou em outros países.

Os nomes de empresas e produtos reais aqui mencionados podem ser marcas comerciais de seus respectivos proprietários.

# Índice

Índice .....	3
1 Sumário Executivo .....	1
2 Melhores práticas .....	2
2.1 Processo de carga.....	2
2.1.1 OLEDB Destination.....	2
2.1.2 Operações minimamente logadas .....	3
2.2 Transferência de dados.....	4
2.2.1 Selecione apenas as colunas necessárias.....	4
2.3 Carga de arquivos.....	5
2.3.1 Configure os tipos de dados.....	6
2.3.2 Configure o Fast Parse .....	6
2.3.3 Problemas comuns por falta de conversão .....	7
2.4 Componentes.....	9
2.4.1 OLEDBComponent x SQL Task .....	9
2.4.2 Full & Partially Blocking operations.....	10
2.4.3 Conditional Split.....	15
2.4.4 Conector para Oracle .....	16
2.5 Use mais T-SQL .....	17
2.6 Try\ Catch e Release connection no .Net .....	18
2.7 Log dos pacotes SSIS.....	19
2.8 Tipos de configurações do SSIS .....	20
2.9 Protection Level.....	21
2.10 Documente o pacote.....	22
2.11 Modelagem.....	23
2.12 Modelo de deploy no SQL Server 2012.....	24
2.13 SSIS e Azure .....	25
3 Conclusão.....	26
4 Anexo .....	27
4.1 Como o SSIS funciona.....	27
4.2 Conector Oracle Attunity.....	27
4.3 Habilitar log do SSIS .....	34
4.4 Como habilitar as configurações do SSIS.....	38
4.5 Deployment model 2012.....	41
5 Referências.....	46

# 1 Sumário Executivo

O objetivo deste documento é buscar, através das melhores práticas de desenvolvimento de pacotes em Integration Services, obter ganhos de desempenho, além de sugerir melhores formas de implementar, desenhar e monitorar os pacotes.

Este guia pode ser usado por qualquer time responsável pelo desenvolvimento de pacotes em SQL Server Integration Services (SSIS).

É recomendado que este sirva de base para o desenvolvimento de pacotes com maior qualidade, evitando o uso de certas práticas que levam a um mau desempenho do mesmo.

Este manual se aplica a todas as atuais versões do Integration Services.

## 2 Melhores práticas

O SQL Server Integration Services (SSIS) é uma ferramenta de ETL: Extract Transformation Load. A função do ETL é extrair informações de uma ou mais fontes de dados. Logo, o Integration Services é uma ferramenta para construção de integração de dados, transformando informação em solução.

O SSIS pode ser usado para resolver problemas complexos de negócio, permitindo a transferência de dados em arquivos, copiar documentos entre diretórios locais ou mapeados na rede, mandar email em resposta a eventos, atualizar banco de dados, tratar as informações e administrar objetos no SQL Server. Os pacotes de SSIS podem funcionar sozinhos ou em conjunto com outros pacotes.

Este guia se divide nas principais áreas de desenvolvimento do SSIS e a seguir iremos falar sobre cada uma dessas áreas.

### 2.1 Processo de carga

Existem algumas configurações que podem otimizar o processo de carga dos dados no banco. A seguir, iremos falar de algumas destas configurações.

#### 2.1.1 OLEDB Destination

Considere o uso do componente OLEDB destination, por exemplo, como mostra a figura 1. O componente OLEDB destination possui algumas configurações que afetam a performance da carga dos dados que são persistidos na base.

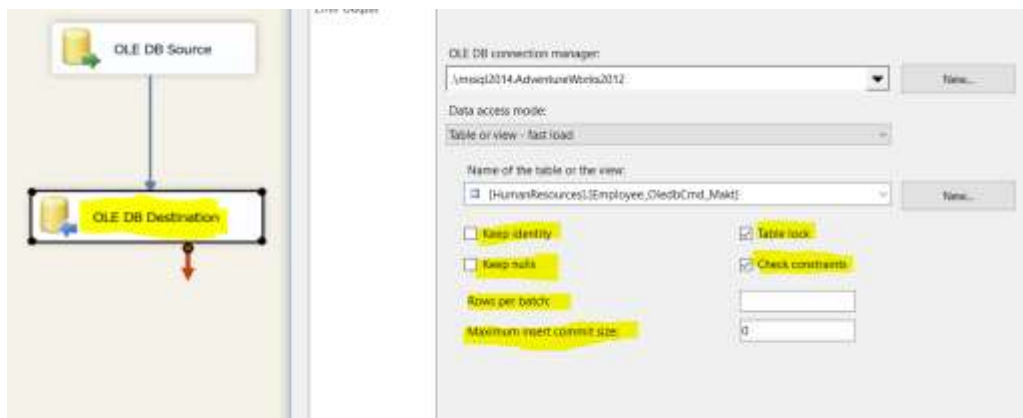


Figura 1: OLEDB destination configuration

**Data Access Mode** – Esta configuração provê a opção *fast load*, a qual internamente usa o *Bulk Insert* para a carga dos dados, ao invés de uma simples INSERT por linha. Essa é a opção default. Somente altere esta configuração caso exista algum motivo específico.

**Keep Identity** – Por padrão, esta configuração aparece como desabilitada. Logo, se a tabela de destino possuir um campo *identity*, este vai ser controlado pela tabela. Todavia, se o objetivo da carga for garantir que os valores *identity* de origem sejam iguais ao de destino, por exemplo, essa configuração

pode ser habilitada e o controle do *identity* passa a ser feito através dos dados sequencias que estiverem sendo transferidos. Com isso, o controle da tabela é desconsiderado.

**Keep Nulls** – Esta configuração permite tratar valores nulos ao inserir na tabela de destino. Caso a opção esteja marcada, valores NULL serão inseridos na tabela, independente do default configurado para aquela coluna na base de dados. No entanto, se a opção estiver desmarcada, o valor default configurado da coluna será respeitado. Caso não haja nenhum valor default configurado para a coluna, será inserido NULL. Esta configuração vem desmarcada por padrão.

**Table Lock** – Por padrão, esta opção vem marcada. Somente devemos alterar se a mesma tabela for usada por outros processos ao mesmo tempo em que a carga estiver ocorrendo. Esta opção significa que durante todo o tempo da carga, um table lock será adquirido na tabela de destino, evitando que outras transações possam efetuar alterações na tabela ao mesmo tempo. Avaliar e testar a utilização do table lock em cenários altamente transacionais.

**Check Constraints** – Esta opção vem marcada por padrão. Com isto, o processo de carga irá validar todas as constraints da tabela de destino. Caso seja possível garantir que os dados da carga não violem nenhuma constraint da tabela de destino, desmarcar esta opção pode promover aumento de performance.

**Rows per batch** – O valor padrão para esta configuração é -1, o que significa que todas linhas serão tratadas como um único batch. Para mudar esse comportamento e dividir todas as linhas em múltiplos batches, basta colocar um valor inteiro positivo que seria o máximo de linhas desejado num batch.

**Maximum insert commit size** – Este valor por padrão é de '2147483647' (o maior valor para o tipo inteiro). O objetivo deste valor é especificar a quantidade de linhas que serão persistidas no banco de dados. O valor padrão faz com que todas as linhas sejam persistidas de uma vez só. Este valor pode ser alterado por um valor positivo, diminuindo o tamanho da transação e aumentando o número de commits no banco de dados. Este tipo de alteração pode causar um gargalo na engine do dataflow, pois serão necessários vários commits para persistir todas as linhas no banco de dados. Por outro lado, isso pode aliviar a pressão no transaction log e tempdb durante a carga de um grande volume de linhas.

**Recomendação:**

- ✓ Use fast load em data access mode quando possível.
- ✓ Use Tablelock, se o cenário for aplicável. É importante lembrar de não utilizar se o pacote SSIS for executado em períodos de muitas transações concorrentes.
- ✓ Avalie a melhor quantidade de linhas em bloco de dados a serem persistidas durante a carga do SSIS. Configure o commit size apropriadamente, assim como o rows por batch.

## 2.1.2 Operações minimamente logadas

Ao se lidar com cenários de grandes quantidades de dados, dê preferência a operações minimamente logadas. As operações minimamente logadas mantém seu rastro no transaction log do SQL, assim como as que são completamente logadas. No entanto, por serem minimamente logadas, essas

operações ocorrem mais rápido do que as que são completamente logadas, além de gerarem menos escrita no arquivo de log e menos I/O.

Exemplo de operações minimamente logadas (isso sem considerar mudar o banco de dados para *bulk-logged*):

- ✓ Use bulk insert mode ao invés do insert linha a linha.
- ✓ Ao invés de delete, use truncate. O delete cria uma entrada para cada linha deletada no log. Já o truncate remove todos os dados da tabela e apenas informa isso no log.
- ✓ Use Switch partition para tabelas particionadas.

**Recomendação:**

- ✓ Dê preferência a operações minimamente logadas sempre que possível

## 2.2 Transferência de dados

Existem algumas ações simples que podem ser tomadas a fim de tornar o processo de carga de dados mais eficiente.

### 2.2.1 *Selecione apenas as colunas necessárias*

Defina quais colunas serão realmente usadas durante a carga dos dados. O SSIS é uma ferramenta de ETL que transfere, carrega e manipula os dados entre origem e destino na memória. Logo, carregue somente os dados que realmente serão necessários, evitando sobrecarga na rede e na memória do servidor.

Quando o OLEDB Source é configurado como no exemplo da figura 2, o comando executado pela SSIS é como um SELECT \*, sendo enviado ao banco de dados.

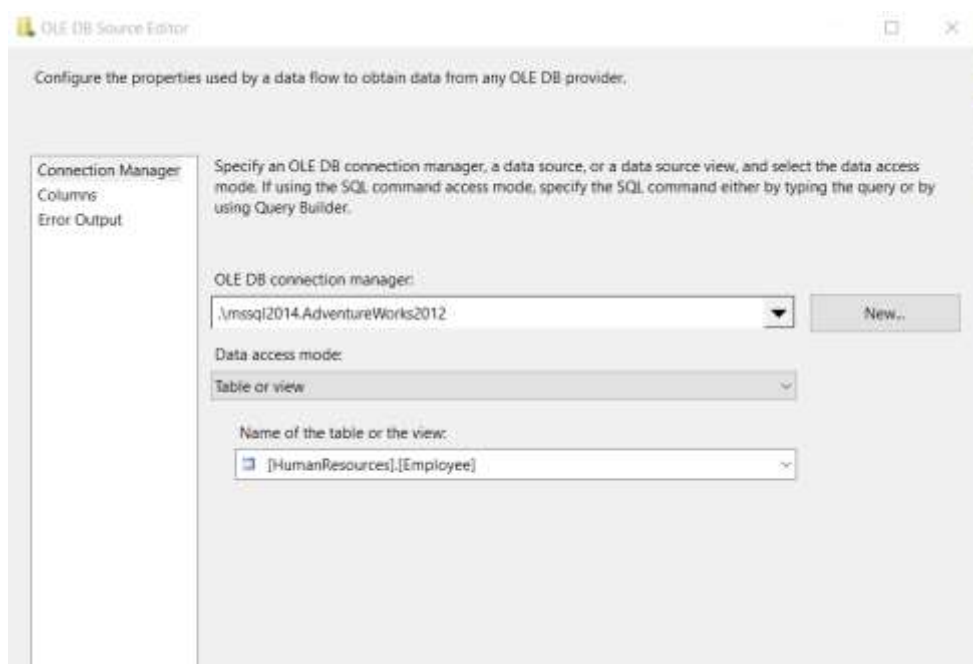


Figura 2 OLEDB Source com Select \*

Desta forma, todas as colunas da tabela *Employee* estão sendo selecionadas para a carga dos dados. Caso não seja necessário o uso de todas as colunas, será trafegado mais dados pela rede do que o necessário, além de utilizar mais memória no servidor.

A seguir, exemplo de uma boa prática de como trazer dados a partir de *oledb source* na figura 3:

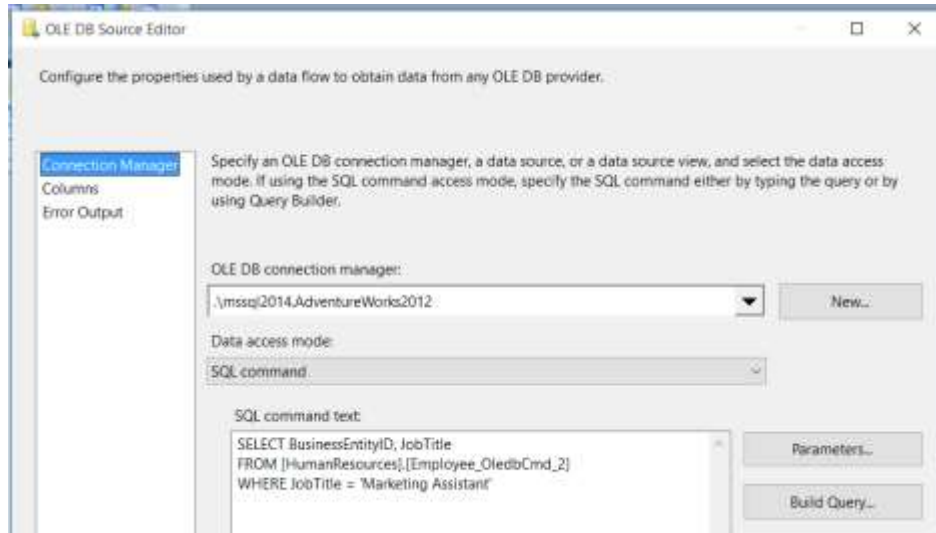


Figura 3 Oledb Source com as colunas

Perceba que neste exemplo, apenas as colunas necessárias para a carga dos dados estão sendo selecionadas dentro do OLEDB Source.

#### Recomendação:

- ✓ Trafegue apenas a quantidade de dados necessária dentro do SSIS. Não traga mais colunas ou mais dados que o necessário.

## 2.3 Carga de arquivos.

O processo de carga de arquivos no SSIS passa pelas seguintes fases:

1. **Lê os dados do arquivo:** O SSIS estabelece a conexão com o arquivo e lê os registros. A rapidez dessa leitura depende de fatores externos como: velocidade de disco, contenção de I/O, velocidade de rede e recurso de máquina.
2. **Os dados lidos do arquivo são carregados no pipeline do SSIS:** Neste passo o SSIS carrega os dados em estruturas de memória, ou seja, em buffers. Durante a execução do SSIS, este faz a validação dos dados carregados em memória, de acordo com as regras de metadados pré-estabelecidas como por exemplo, quais campos são inteiros, quais são smallint, e etc.
3. **Agregação de dados:** Depois disso, os dados que foram lidos pelo pipeline do SSIS são agregados e um cache é criado.
4. **Uma transação é aberta no SQL Server.**
5. **Os dados passam para a interface do SQL Server.**
6. **Os dados são carregados na tabela de destino do SQL Server:** Neste ponto as boas práticas de carga na origem de dados (já mencionadas) entram em ação, como: uso de *tablelock*, habilitar o *fast load*, o tamanho do bloco a ser *commitado*, e demais configurações que fazem parte do OLEDB Destination.
7. **A transação é commitada.**



## 2.3.1 Configure os tipos de dados

Quando os dados são lidos do arquivo e carregados no *Data Pipeline*, todos os dados são lidos como *string*, mesmo que existam outros tipos de dados. O objetivo final é carregar os dados na tabela de destino no SQL Server, a qual provavelmente tem colunas com os tipos de dados definidos. Logo, a conversão para os tipos de dados da tabela final precisa ser feita em algum momento da carga do arquivo.

Este tipo de conversão é custoso e prejudica a performance da carga de dados. Por isso, uma consideração importante para reduzir o tamanho dos dados trafegados e aumentar a performance, é configurar os tipos de dados o mais cedo possível durante o processo de carga. Essa configuração pode ser feita na própria definição dos metadados do arquivo para que, uma vez que os dados passem pelo processo de parse (item 2 dos passos de carga de arquivo), estes já sejam carregados com os tipos corretos.

A figura 4 mostra a configuração no qual o tipo de dados foi definido.

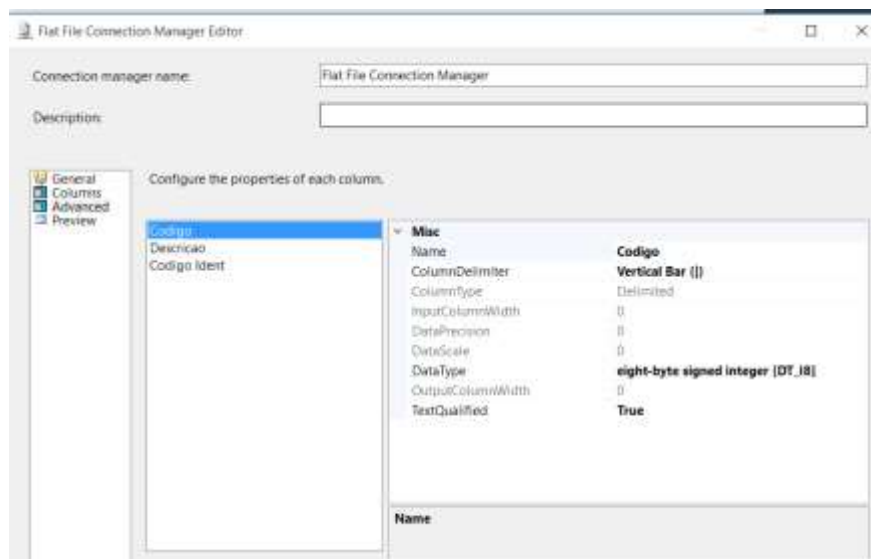


Figura 4 Flat File connection – Configuração de campos.

### Recomendação:

- ✓ Configure adequadamente os tipos de dados do arquivo, não deixando todos os tipos como *string*.

## 2.3.2 Configure o Fast Parse

Quando se está fazendo carga de arquivos e se especifica o tipo de dados certo para cada campo, é possível habilitar o *fast parse*. O *fast parse* aumenta a performance do parsing para tipos de dados: integer, date e time.

Configure o *Fast Parse* através do advanced editor do *Flat File Source* (integer, date and time), como mostra a figura 5.



**Erro: "Invalid character value for cast specification".**

**-Quando ocorre:** Quando carregando dados do componente flat file para o SQL.

**Causa:** Existe um campo datetime dentro da tabela no SQL e o SSIS não está conseguindo fazer a conversão implícita de DT\_STR para DT\_DBTIMESTAMP.

**Resolução:** Use o componente Derived columns para fazer a conversão para DT\_DBTIMESTAMP. Avalie se os dados que estão sendo inseridos na tabela não estão com datas e strings misturadas na mesma coluna.

---

**Erro: *DTS\_E\_INDUCEDTRANSFORMFAILUREONTRUNCATION = The "%1" failed because truncation occurred, and the truncation row disposition on "%2" specifies failure on truncation. A truncation error occurred on the specified object of the specified component.***

**Quando ocorre:** Importando dados do flat file com mais de 50 caracteres para o SQL.

**Causa:** O tamanho default do OutputColumnWidth é 50 para o componente Flat File Connection Manager.

**Resolução:** Aumente o tamanho do OutputColumnWidth do Connection Manager para flat file de maneira adequada.

---

**Erro: DT\_NTEXT data type cannot be converted to DT\_STR using Data Conversion transformation**

**Causa:** Conversão de DT\_NTEXT para DT\_STR não é suportado.

**Resolução:** Converta DT\_NTEXT para DT\_WSTR e então de DT\_WSTR para DT\_STR (Use duas vezes o componente Data Conversion). Ou use o componente Derived Column transformation como: (DT\_STR,10,1252) ((DT\_WSTR,10) ntext\_column)

---

**Erro: "Column "xxxx" cannot convert between unicode and non-unicode string data types".**

**Quando ocorre:** Exportando dados para o excel utilizando o SSIS.

**Causa:** A origem tem colunas "char" e "varchar" na qual para o SSIS é equivalente a DT\_STR, no entanto o EXCEL espera pelo tipo de dados DT\_WSTR.

**Resolução:** Use o componente Data conversion para converter "DT\_STR" para "DT\_WSTR"

---

**Erro: Campos do tipo Decimal estão sendo truncados quando importados Oracle para o SQL Server**

**Causa:** O método GetSchemaTable fora do SSIS no .NET provider para ORACLE Client não está retornando os valores decimais apropriadamente. Como resultado, o DataReaderSrc está setando a precisando para o máximo e arbitrariamente escolhendo a escala de 4.

**Resolução:** Modifique a query de origem para converter o tipo de dados Float para Character. Use Data Conversion para converter de Character para o tipo numérico correto.

---

## 2.4 Componentes

O SSIS tem vários componentes excelentes para manipulação de dados. No entanto, é preciso saber a melhor situação para utilizar cada um deles. Se utilizados de maneira inadequada, poderão comprometer a performance da carga de dados.

### 2.4.1 OLEDBComponent x SQL Task

O SSIS é uma ferramenta de ETL gráfica. Sendo uma ferramenta de ETL ela é excelente para fazer a transferência de dados para grande volume de dados. No entanto existem alguns componentes que fazem o processamento de dados linha a linha. Logo, se o processo de carga de arquivos está concentrando a transformação de um grande número de linhas, usar componentes que processem linha a linha pode ser bem custoso. Considere o exemplo da figura 6.

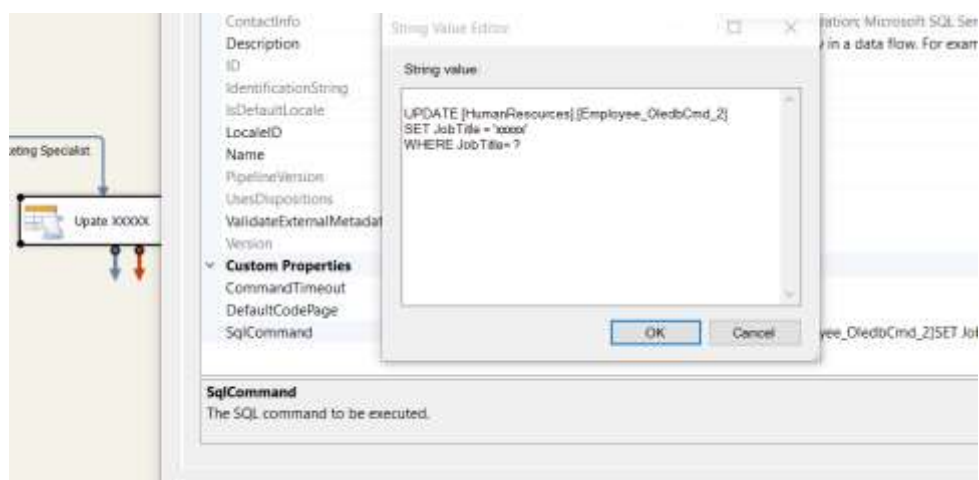


Figura 6 Update linha a linha em um OleDbCommand

Conforme o exemplo, um update no campo JobTitle ocorre na tabela Employee\_OleDbCmd\_2, de acordo com o parâmetro JobTitle passado como filtro. Isso quer dizer que o SSIS vai, linha a linha, percorrer os dados carregados para passar o filtro para o OLEDB Command e este, validar se o filtro traz informações ou não.

Este processo poderia ser adaptado para que o update fosse feito em blocos, ao invés de utilizar o oledb command, evitando que o comando seja realizado linha a linha. Este exemplo é mostrado na figura 7.

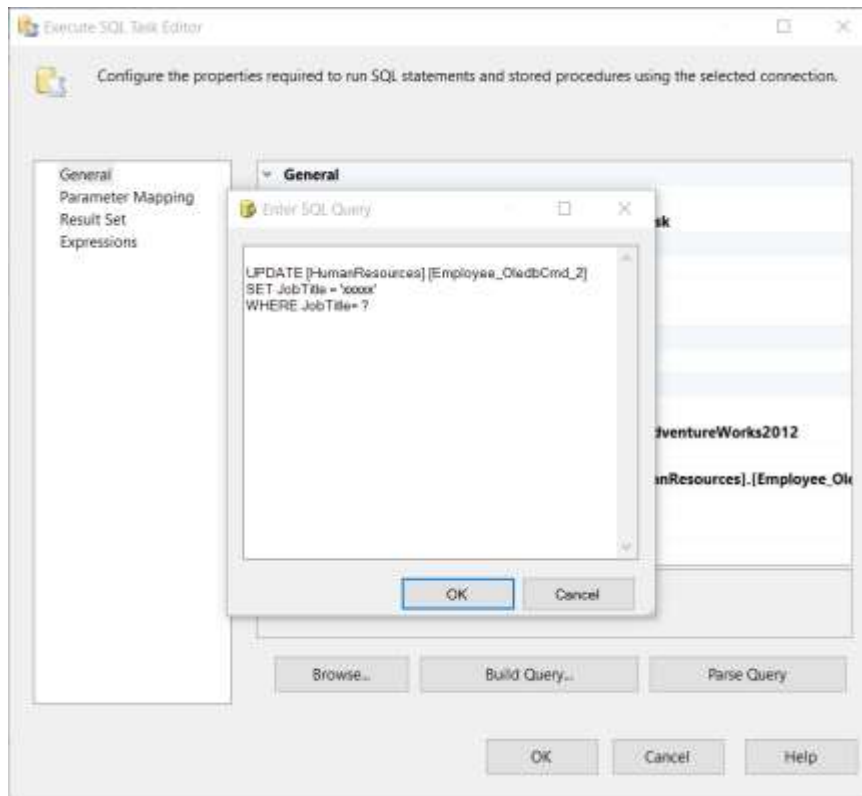


Figura 7 Componente Execute SQL Task

Com uso do componente *Execute SQL Task*, passa-se o parâmetro para a query e o update ocorre em bloco, ao invés de ocorrer linha a linha.

**Recomendação:**

- ✓ Evite transformações linha a linha. Procure fazer o processamento dos dados em blocos, principalmente se for para um grande volume de dados.

## 2.4.2 Full & Partially Blocking operations

A engine do SSIS possui uma arquitetura orientada a buffer. O benefício disto é que não é preciso copiar os dados cada vez que um passo do integration é executado. Logo, uma vez que os dados a serem transferidos forem carregados em memória e o fluxo de dados está dentro do pipeline do SSIS, é possível reusar os buffers, enquanto as operações para manipulação dos dados são executadas. No entanto, como os dados do buffer serão reusados, depende do tipo de operação que está sendo executada.

O reuso pode ser feito através de transformações. No dataflow do integration services existem três tipos de transformação:

1. *Non-blocking*
2. *Semi-blocking*
3. *Fully-blocking*

A seguir mais detalhes sobre esses tipos de transformação.

**Nonblocking ou row transformations** – Este tipo de transformação ou manipula os dados ou cria novos campos usando as informações disponíveis. Exemplos deste tipo transformação são: *Derived columns, Data conversion etc.* Essas transformações não criam novos registros, embora possam criar

novas colunas. Isto quer dizer que a relação entre os dados que passam por essas operações e o resultado da transformação aplicadas a eles é de um relacionamento 1:1. Com isso, o buffer é reaproveitado, sem gerar novas entradas para um novo buffer.

**Partially blocking transformation** – Este tipo de transformação é geralmente usado para combinar vários *datasets*. Elas tendem a ter múltiplas entradas de dados e, como resultado, podem gerar a mesma quantidade, um número menor ou ainda um número maior de linhas. Como o resultado em quantidade de linhas gerados a partir desse tipo de transformação pode não ser a mesma quantidade de linhas a qual foi utilizado na entrada, estas transformações são chamadas de assíncronas. Exemplos de componentes que funcionam desta forma são: *Merge Join*, *Merge* e *Union All*. O resultado deste tipo de transformação faz com que os dados sejam copiados para um novo buffer, e uma nova *thread* pode ser introduzida no dataflow.

**Full Blocking transformations** – Este tipo de transformação precisa ler e processar todos os registros de entrada antes de gerar a saída. Os componentes que se enquadram neste tipo de transformação são os que tem maior impacto em termos de performance. Exemplos deste tipo de transformação são *Aggregate* e *Sort*. Esse tipo de transformação, assim como as *partially blocking*, são assíncrona. Logo, um novo buffer precisa ser criado e uma *nova thread* é introduzida ao processo de carga.

**Importante:** Componentes que reusam o buffer tendem a ser mais rápidos que os componentes que são assíncronos (*partially blocking* e *full blocking*).

Segue resumo na tabela a seguir.

Tabela 1 - Comparação entre tipos de transformação.

	Non-blocking	Semi-blocking	Fully-blocking
Synchronous/asynchronous	Síncrono	Assíncrono	Assíncrono
Número de linhas de saída igual a entrada	Sim	Geralmente não	Geralmente não
Precisa ler todos os dados antes de processar	Não	Não	Sim
Novo buffer é criado?	Não	Sim	Sim
Nova thread se faz necessária?	Não	Geralmente Sim	Sim

Para entender melhor, verifique o exemplo na figura abaixo. Neste exemplo, existem duas fontes de dados OLEDB. Os dados trazidos por essas fontes não foram ordenados na origem e por isso foi necessário o uso do componente *SORT*. Após ordenados, estes dados serão tratados através de um *left join* dentro do SSIS, a ser realizado pelo componente *Merge Join*.

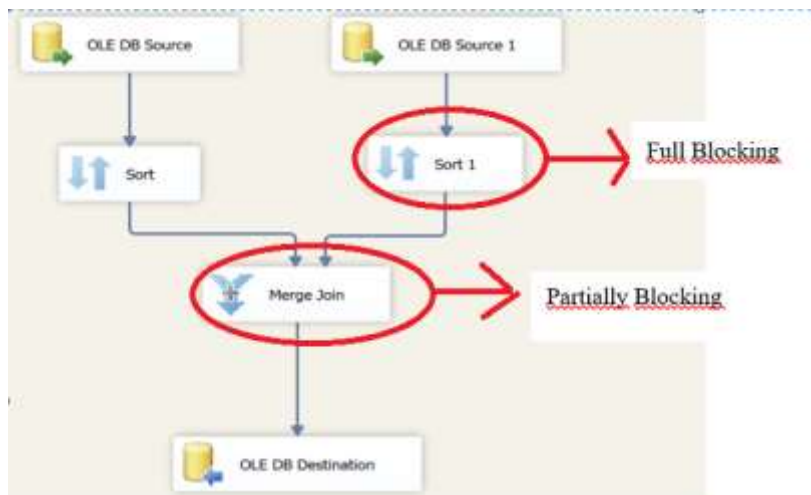


Figura 8 Full e Partially Blocking - merge join

Perceba a existência de uma transformação do tipo *full blocking*, representada pelo componente *sort*, e outra *partially blocking*, representada pelo componente *merge join*. Existem alguns pontos de atenção que podemos observar aqui:

1. Dado que a origem dos dados é um OLE DB, será que os dados não poderiam ter sido ordenados na origem, de forma a evitar o uso de *Sort* e melhorar a performance?
2. Em um cenário onde possuímos dois OLE DBs como sources no mesmo servidor, realmente é necessário fazer o join durante o processo de carga do SSIS, através do *merge join*? Provavelmente, este join poderia ter sido feito na origem também, melhorando a performance.
3. Quantas linhas são trafegadas nessa transformação? Este é um ponto importante que seja avaliado para que, desta forma, escolha-se os melhores componentes para cada situação.

Respondendo à primeira pergunta, neste cenário é possível sim ordenar os dados na origem. Conforme a figura 9 mostra, os dados foram ordenados na fonte.

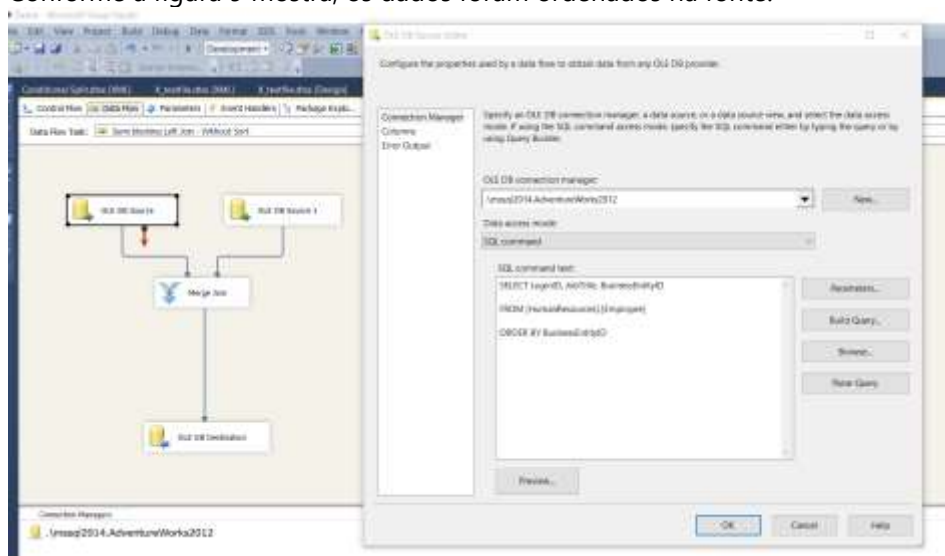


Figura 9 sem sort e ordenando na fonte



Para verificar se realmente os dados estão sendo ordenados dentro do SSIS, basta verificar a configuração *sort key position* do componente OLE DB source.

Esta configuração pode ser vista na figura 10 e pode ser realizada da seguinte forma:

- ✓ Clique com o botão direito do mouse em cima do no componente OLE DB source
- ✓ Selecione a opção show advanced editor
- ✓ Selecione a aba input and output properties
- ✓ Expanda OLE DB Source output, e depois expanda Output columns.
- ✓ Selecione a coluna pela qual os dados foram ordenados (exemplo BusinessEntityID) e em common properties (do lado direito da tela), coloque em *sort key position* o valor 1.

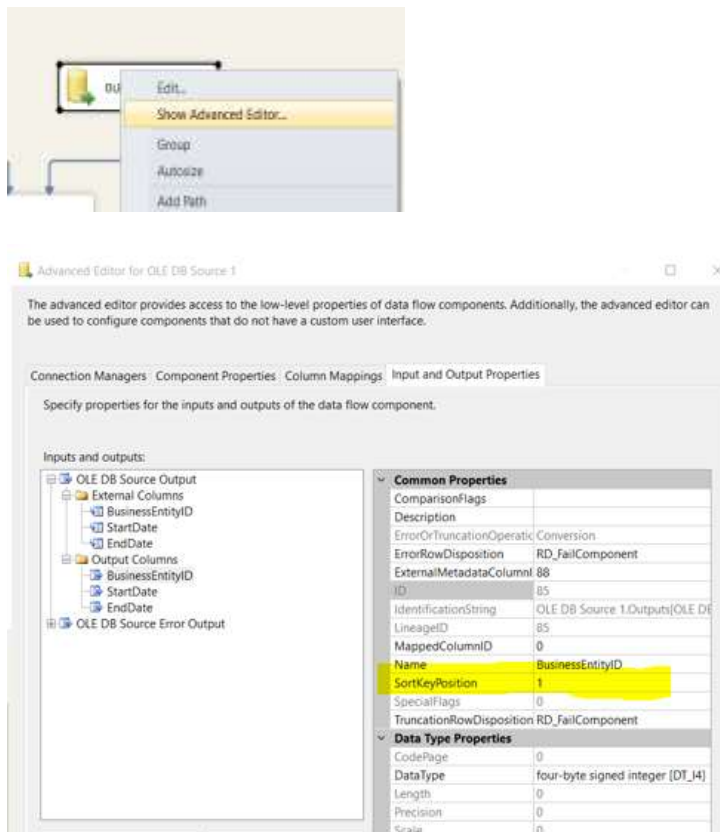


Figura 10 Sort key position configuration informando order by da origem

Outro ponto importante de se observar é a prova do conceito do que é uma *full blocking* transformation ou *partially blocking*. A *full blocking*, representada pelo *Sort*, precisa ler todos os dados antes de ordená-los. Isto é exatamente o que o componente *Sort* faz. O *partially blocking*, representada pelo *merge join*, gera uma saída de dados diferente da entrada (isso em virtude do *left join* realizado entre as duas fontes de dados). Logo, isso irá gerar a necessidade de um novo buffer de dados.

Agora respondendo à segunda pergunta: "Em um cenário onde possuímos dois OLE DBs como sources no mesmo servidor, realmente é necessário fazer o join durante o processo de carga do SSIS, através do *merge join*?" Não é necessário. Isto também pode ser feito na fonte de dados, ao invés de se fazer no SSIS. Uma vez que fosse feito no SQL, não seria necessário o uso do componente *merge join*. Essa mudança traria dois ganhos:



- ✓ Um novo buffer não precisaria ser criado, ocupando menos memória do servidor
- ✓ Com o join sendo realizado na fonte de dados, seria trafegado um número menor de linhas no processo de carga.

A figura abaixo mostra como isto poderia ser implementado.

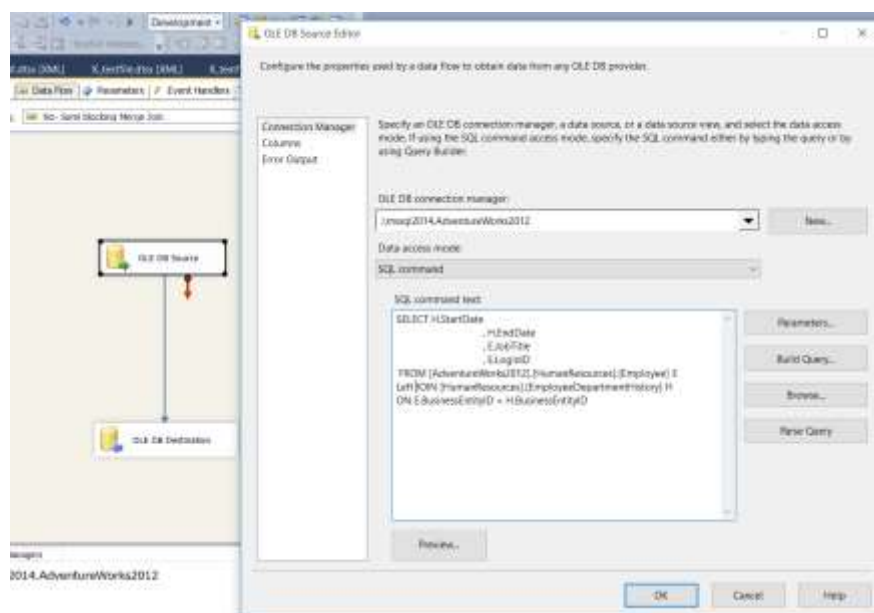


Figura 11 Join entre as fontes de dados sem necessidade do Merge Join

A tabela 2 mostra os componentes do SSIS classificados por tipo de transformação:

Tabela 2 Componentes do SSIS por tipo e transformação

Non-Blocking transformations	Semi-blocking transformations	Blocking transformations
Audit	Data Mining Query	Aggregate
Character Map	Merge	Fuzzy Grouping
Conditional Split	Merge Join	Fuzzy Lookup
Copy Column	Pivot	Row Sampling
Data Conversion	Unpivot	Sort
Derived Column	Term Lookup	Term Extraction
Lookup	Union All	
Multicast		
Percent Sampling		
Row Count		
Script Component		
Export Column		
Import Column		
Slowly Changing Dimension		
OLE DB Command		

**Recomendação:**

- ✓ Evite, se possível, transformações full blocking ou partially blocking.

## 2.4.3 Conditional Split

O *Conditional split* é usado para filtrar dados dentro do SSIS. No entanto, em alguns cenários quando ele é usado próximo a fonte de dados, o uso deste componente torna-se custoso ao pacote, fazendo com que mais dados do que o necessário seja trafegado. Lembrando que os dados no SSIS são carregados em memória para depois serem manipulados. Verifique o seguinte exemplo, ilustrado pela figura 12.

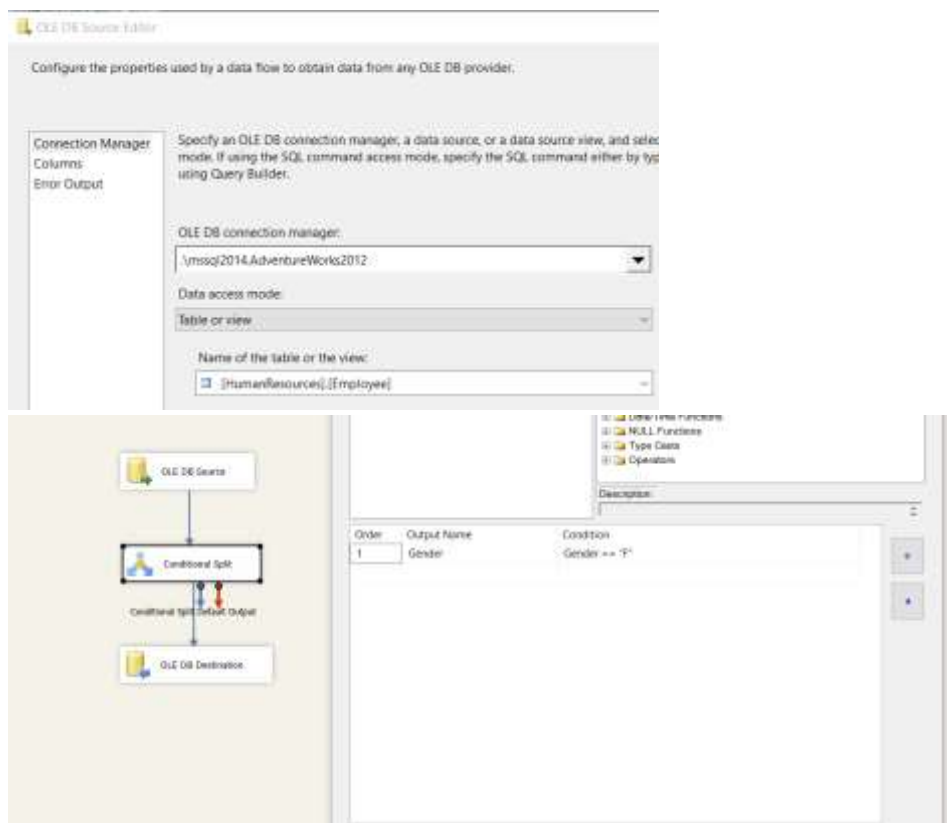


Figura 12 Conditional Split próximo a fonte de dados

Neste exemplo, pode-se perceber que os dados são extraídos da fonte de dados filtrados dentro do *conditional split*, para assim seguir o processamento. Se o *conditional split* está sendo usado para filtro, neste cenário faz mais sentido não usar o *conditional split* e sim, filtrar os dados diretamente na fonte.

Da forma como o exemplo foi construído, todos os dados da tabela de origem (*select \**) estão sendo extraídos até que o *conditional split* os filtre e após isso sejam persistidos no seu destino. Logo, trafega-se mais dados que o necessário dentro do pacote, custando mais para processar.

O exemplo da figura 13, propõe corrigir isso:

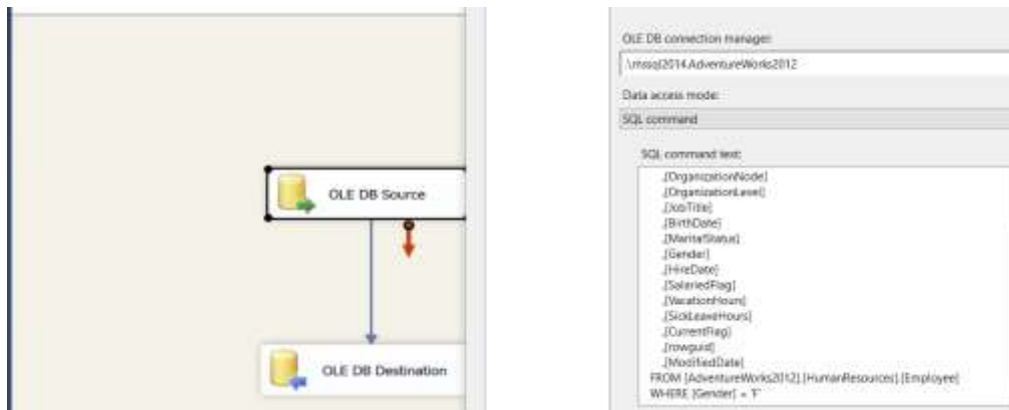


Figura 13 Sem conditional split, filtrando na fonte de dados

Duas mudanças ocorreram neste pacote. A primeira foi que o filtro passou a acontecer na fonte de dados, ao invés de usar o *conditional split* para realizar isso. Logo, a quantidade de dados que estão sendo trafegados pela rede é menor. A segunda mudança foi a definição de quais campos irão ser persistidos na tabela de destino, ao invés de selecionar apenas o nome da tabela dentro da SSIS, trazendo todos os campos da tabela.

**Recomendação:**

- ✓ Evite filtrar os dados com *conditional Split* filtrando, sempre que possível, na origem dos dados.

## 2.4.4 Conector para Oracle

Sendo o SSIS uma ferramenta de ETL, é necessário que o integration consiga obter dados de diferentes fontes com qualidade e performance. Considerando um cenário no qual precise-se extrair dados de um banco de dados Oracle, a Microsoft recomenda o uso do Microsoft Conector para Oracle Attunity. Este conector provê performance otimizada para carga dos dados para Oracle com o uso do SSIS. Este conector foi disponibilizado pela Microsoft sem nenhum custo adicional.

O conector Oracle Attunity tem três componentes principais:

1. Oracle Source, para extrair os dados do Oracle.
2. Oracle Destination, para importar os dados no Oracle.
3. Oracle Connection Manager, para conexão ao banco de dados Oracle.

Para verificar como instalar e disponibilizar este conector na SSIS, verifique o anexo.

Após a instalação, os componentes deste conector estarão disponíveis na SSIS Toolbox, conforme figura a seguir.

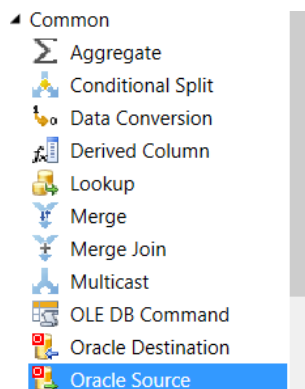


Figura 14 Microsoft Connector Oracle - Data tools Oracle Source/Destination

#### Recomendação:

- ✓ Para extrair dados do Oracle, use os componentes do conector Oracle Attunity.

## 2.5 Use mais T-SQL

Em alguns momentos, o uso excessivo de componentes do SSIS para tarefas que seriam mais simples e mais rápidas dentro do SQL, acaba por prejudicar a performance, ao invés de otimizá-la. Exemplos deste tipo de situação ocorrem com certa frequência em componentes como Derived Column ou Script Task. Veja o exemplo da figura 15.

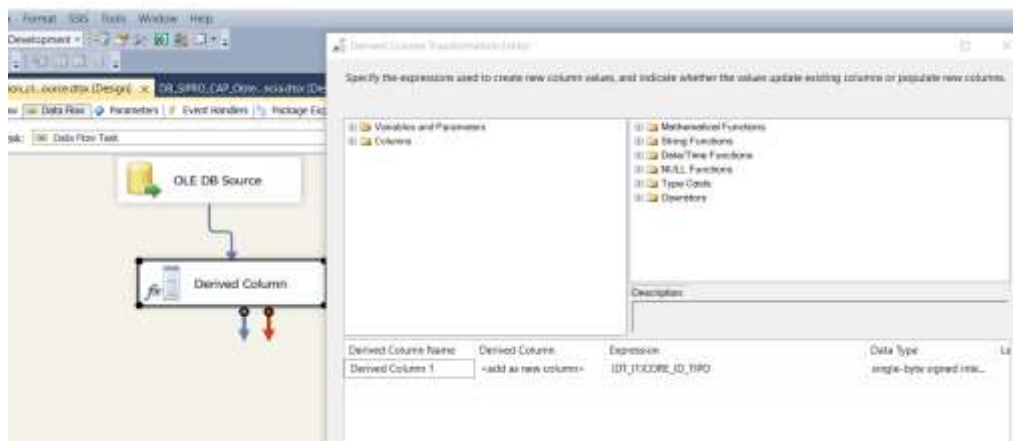


Figura 15 Derived Column Conversion na Source

Neste caso, uma conversão está sendo feita através do componente *Derived Column*. Da mesma forma, poderia ser feito um *Data Conversion*. Este é um bom exemplo de situação na qual, considerando que a fonte de dados é um banco de dados, a conversão poderia ser feita na origem.

Outro exemplo é usar o *script task* ou *script componente* para tarefas que seriam melhor executadas no SQL. Um exemplo deste cenário é usar o script componente como *source* para extração de dados do SQL para conectar no Oracle via script componente, conforme mostra o script a seguir.

Listagem 1 Trecho de código Acquire connection

```
public override void AcquireConnections(object Transaction)
{
    IDTSConnectionManager100 connMgr = this.Connections.con0;
    con = (OracleConnection)connMgr.AcquireConnection(null);

    IDTSConnectionManager100 connMgrSipro = this.Connections.con1;
    Conexao = (SqlConnection)connMgrSipro.AcquireConnection(null);
}

public override void PreExecute()
{
    base.PreExecute();
    try
    {
        obterDadosERGON();
    }
    catch (Exception ex)
    {
        preExecuteResult = ex.Message;
    }
}

private void XYZ()
{
    string dataBaseName = Variables.oracleDatabaseName;
    OracleCommand oracleCmd = new OracleCommand(dataBaseName + ".pck_teste_teste",
con);
    oracleCmd.CommandType = CommandType.StoredProcedure;
    OracleParameter pi_numfunc = oracleCmd.Parameters.Add("codigo ",
OracleType.Number);
    pi_numfunc.Direction = ParameterDirection.Input;
    pi_numfunc.Value = DBNull.Value;
```

Se os dados estão sendo obtidos de uma fonte de dados, sem nenhum tratamento adicional, o qual precisaria ser proporcionado pelo script componente. Esta extração deveria ser feita através dos componentes de extração de dados, os quais foram criados e otimizados para esta finalidade.

**Recomendação:**

- ✓ Avalie com cuidado e use mais T-SQL, nem tudo fica mais rápido no SSIS.

## 2.6 Try\ Catch e Release connection no .Net

Usar *try catch* para controlar os erros dentro dos scripts no SSIS é considerada uma boa prática. Isso permite que o processo de troubleshooting, em caso de alguma falha no processo de carga, se torne mais detalhado e mais claro. Um exemplo de *try catch* no *script task*, pode ser observado a seguir.

Listagem 2 Try catch script task

```
public void Main()
{
    bool fireAgain = true;
    Dts.Events.FireInformation(0, string.Empty, "Começando", string.Empty, 0, ref
fireAgain);

    try
    {
        Dts.Events.FireWarning(0, string.Empty, "Irá gerar um erro, só espera",
string.Empty, 0);
        throw new ArgumentNullException();
    }

    catch (Exception e)
    {
        // Raise event so it can be logged by SSIS
        Dts.Events.FireError(0, string.Empty, "Sabia! Um erro ocorreu!!", "",
0);
        Dts.TaskResult = (int)ScriptResults.Failure;
    }
}
```

Outra boa prática é liberar a conexão para reuso pelo connection pool, uma vez que esta conexão tenha sido adquirida pelo acquire connection através do Script Task ou Script Componente. A listagem 3, mostra um trecho de código com o uso de release connection.

Listagem 3 Release Connection

```
connectionMgr.ReleaseConnection(myConnection);
```

#### Recomendação:

Se estiver usando Script Componente ou Script Task:

- ✓ Libere as conexões através do comando release connection ao adquiri-las no pacote por meio do acquire connection.
- ✓ Use Try catch para tratar erros.

## 2.7 Log dos pacotes SSIS

Geralmente os pacotes SSIS estão em constante mudança. Isto ocorre devido a regras de negócio que mudam, correções que são feitas e assim por diante. Com isso, erros ocorrem mesmo nas cargas mais estáveis e nem sempre é simples de entender os motivos pelos quais esses erros ocorreram ou exatamente qual será o ponto de correção.

Em virtude disso é importante saber que o SSIS provê mecanismos que permitem gravar erros gerados dentro do pacote, assim como auditar *tasks* executadas.

Para auditar as tarefas do pacote SSIS, primeiro é preciso avaliar quais passos são importantes que são auditados. Esta primeira avaliação é necessária uma vez que auditar e gerar log de todas as tarefas da SSIS pode fazer com que a execução do pacote fique lenta. Da mesma forma, um pacote

que não possui log nenhum pode ser executado mais rápido, no entanto haverá uma dificuldade muito maior de se realizar *troubleshooting* do pacote em caso de falha na execução.

Para saber como habilitar e utilizar os diferentes tipos de log do SSIS, consulte o anexo deste manual.

**Recomendação:**

Configure o log de maneira eficiente e eficaz, de acordo com o pacote SSIS:

- ✓ SSIS Logging
- ✓ Event Handlers
- ✓ Logs Customizados

## 2.8 Tipos de configurações do SSIS

Um pacote de SSIS bem arquitetado, precisa responder as mudanças de ambiente ou a outros fatores, sem que seja necessário alterar o código do pacote cada vez que alguma mudança de configuração é realizada.

Para tornar essa flexibilidade possível, existem algumas opções. No SQL Server 2008 e no 2012 existe uma diferença entre essas opções, mas a partir do 2012, o range de opções ficou ainda maior e melhor com a criação de parâmetros e ambientes. Isso será detalhado a seguir.

### Tipos de configuração:

**XML File Configuration:** Nesta configuração a propriedade do path, assim como o valor configurado, são salvos em arquivo XML, conforme figura abaixo.

XML

```
<?xml version="1.0" ?>
- <DTSConfiguration>
- <DTSConfigurationHeading>
  <DTSConfigurationFileInfo GeneratedBy="MARINER\rsalas" GeneratedFromPackageName="Test XML Config - MSDN Article"
  </DTSConfigurationHeading>
- <Configuration ConfiguredType="Property" Path="\Package.Variables[User::ConfigTarget].Properties[Value]" ValueTyp
  <ConfiguredValue>Run time Value from config file - direct Method</ConfiguredValue>
  </Configuration>
-
  </DTSConfiguration>
```

Figura 16 XML configured value

Logo, valores como caminhos de pasta na rede por exemplo, poderiam ficar configurados no arquivo XML. Perceba que desta forma há flexibilidade para mudanças dentro deste arquivo durante deploy em ambientes diferentes. Ao habilitar esta opção de configuração dentro do SSIS, o Package Configuration Wizard cria este arquivo automaticamente.

**Environment Variables:** Nesta opção uma variável de ambiente é criada no sistema operacional e guarda o valor configurado do pacote. Cada propriedade configurada (caminho de pasta, nome de servidor etc) deverá ter sua própria variável de ambiente.

**Registry Entry:** Esta opção permite guardar configurações no registry do Windows, de forma similar a variável de ambiente.

**Parent Package:** Quando um pacote filho, relacionada a outro pacote principal, é executada via Execute Package Task, é possível passar um determinado valor de um pacote a outro via variável Parente Package. Apesar do nome, esta configuração é definida no pacote filho e a referência é passada para o pacote principal. O nome da variável especificado no pacote principal tem que ser igual ao do pacote filho. Este nome não é validado quando a configuração é criada.

Considerações:

- ✓ Por padrão, a configuração da Parent Package ou o pacote principal é aplicada por último.
- ✓ Este tipo de configuração é aplicado apenas quando o pacote filho é executado.
- ✓ Os nomes de variáveis são case-sensitive, ou seja, "Myvariable" e "MyVariable" são diferentes.

Por padrão, todas essas configurações são desabilitadas. Para saber como habilitá-las, consulte o anexo deste manual.

#### Recomendação:

- ✓ Configure os pacotes tornando flexível o processo de deploy em ambientes diferentes.

## 2.9 Protection Level

A configuração de protection level do pacote permite encriptar porções do código mais sensíveis, como por exemplo usuário e senha ou ainda, todo o código do pacote do SSIS. É importante entender como funciona e a melhor forma de usá-la.

Essa configuração pode ser vista nas propriedades do pacote em protection level, conforme a figura a seguir.

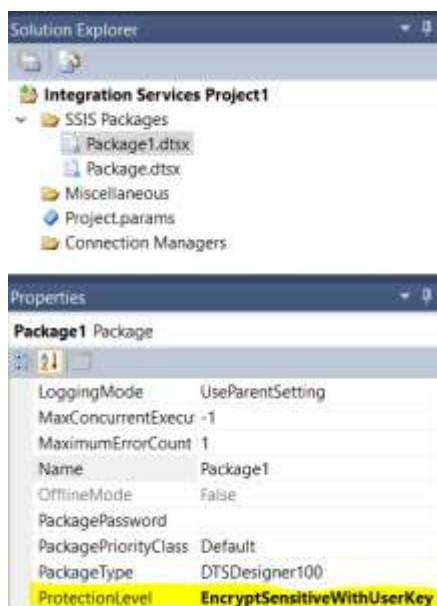


Figura 17 Protection Level



**Dont Save Sensitive:** Quando o pacote é salvo neste protection level, informações sensíveis como senha de usuário não são salvas no XML. Uma vez que o pacote foi salvo nesta configuração, ao abrir o OLE DB Connection Manager a senha estará em branco, mesmo que a opção “save my password” esteja marcada.

**EncryptSensitiveWithUserKey:** Este nível de proteção encripta informações sensíveis baseado nas credenciais do usuário que criou o pacote. Quando o usuário que criou o pacote o abre, as informações sensíveis são descriptadas. Logo, este modo de proteção (padrão do SSIS) não deve ser usado em produção pois dificilmente o usuário que criou o pacote é o mesmo usuário que vai rodar o pacote no ambiente produtivo.

**EncryptSensitiveWithPassword:** Esta configuração requer que uma senha seja especificada no pacote. Esta senha vai encriptar e descriptar as informações sensíveis do pacote.

**EncryptAllWithPassword:** Esta configuração irá encriptar não somente as informações sensíveis, como todo o conteúdo do pacote SSIS, com uma senha.

**EncryptAllWithUserKey** Esta configuração irá encriptar, com as credencias do usuário que criou o pacote, não somente as informações sensíveis, como todo o conteúdo do pacote SSIS.

**ServerStorage** Permite ao pacote reter todas as informações sensíveis quando salvo no SQL Server.

#### Recomendação:

- ✓ Adeque o protection level para o ambiente de produção

## 2.10 Documente o pacote

Criar comentários úteis no código é considerada uma boa prática de programação. Essa prática também se aplica aos pacotes do integration services. Desta forma, para realizar uma alteração no pacote ou mesmo uma correção, não seria necessário tentar entender todo o processo de execução do pacote. O SSIS permite criar comentários dentro dos pacotes. Para criar comentários, basta seguir os passos abaixo.

- 1) Clique com o botão direito do mouse no pacote, conforme mostra a figura a seguir e selecione *Add Annotation*.

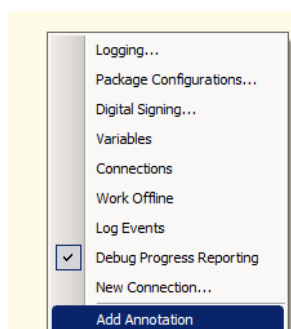


Figura 18 Add Annotation

- 2) Escreva no espaço habilitado para isso, conforme mostra a figura.

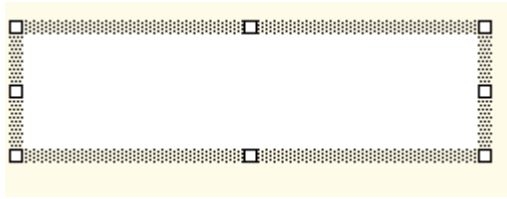


Figura 19 Espaço para documentação

**Recomendação:**

- ✓ Documente o código

## 2.11 Modelagem

Existem alguns métodos que são recomendados para a criação e modelagem de um pacote SSIS, de forma a ficar mais fácil de dar manutenção, acompanhar a execução e analisar os erros.

**Process modularity** – “Dividir para conquistar”. Evite complexos ETLs, divida a lógica do pacote em diferentes passos, segmentando por assunto.

**Package modularity** – Crie sub processos dentro do pacote com uso de containers.

**Componente Modularity** – Crie componentes customizados que possam ser reutilizados com o uso de script task e script componente.

A vantagem desta abordagem é um código mais limpo, elegante, fácil de entender e analisar em caso de erros. Agrupando os pacotes em container por exemplo, permite desabilitar ou habilitar um conjunto de tarefas agrupadas com o botão direito do mouse.

Logo, **EVITE** este tipo de abordagem mostrado na figura abaixo.

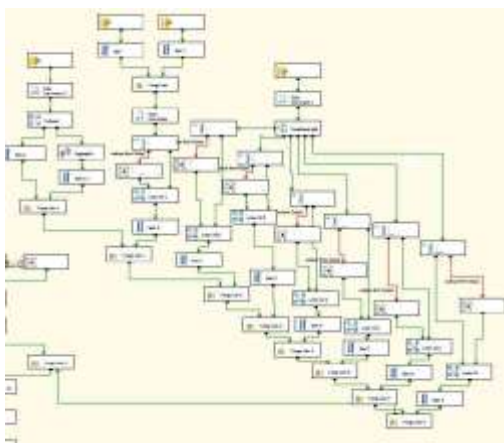


Figura 20 Pacote complexo

Ao invés de pacotes extremamente complexos, simplifique utilizando containers, como no exemplo a seguir.



Figura 21 Pacote simplified

### Recomendação:

- ✓ Simplifique a lógica do pacote.
- ✓ Agrupe as tarefas através do uso de containers

## 2.12 Modelo de deploy no SQL Server 2012

A partir do SQL 2012, existe uma nova opção de deploy para projetos SSIS. Antes desta versão, cada pacote dentro do SSIS era considerado uma unidade única de deploy. Para utilizar o novo modelo de deploy, é necessário que no banco de dados a SSIS Catalog tenha sido criada. Outro detalhe importante é a necessidade de que o pacote SSIS esteja no modo de Project Deployment e o ambiente tenha sido configurado.

Uma das vantagens deste tipo de deploy é que através da configuração dos ambientes, é possível ter flexibilidade na escolha do local no qual a pacote vai rodar. É possível, por exemplo, criar um ambiente de desenvolvimento e outro de produção no SQL e no momento de rodar a pacote escolher em qual ambiente este SSIS vai rodar, sem nenhuma configuração adicional.

Para converter o pacote para o Deployment Model e começar a utilizar esta nova forma de deploy verifique o anexo deste manual.

## Versionamento

Outro ponto interessante é que para se fazer o *rollback* de deploy o processo ficou mais simples.

Ao se clicar com o botão direito em cima da pasta do projeto, existe a opção *Versions*, conforme demonstrado a seguir.



*Figura 22 Versions*

Nesta opção é possível voltar a versões anteriores do deploy do pacote através do botão *Restore to Select Version*, conforme abaixo.

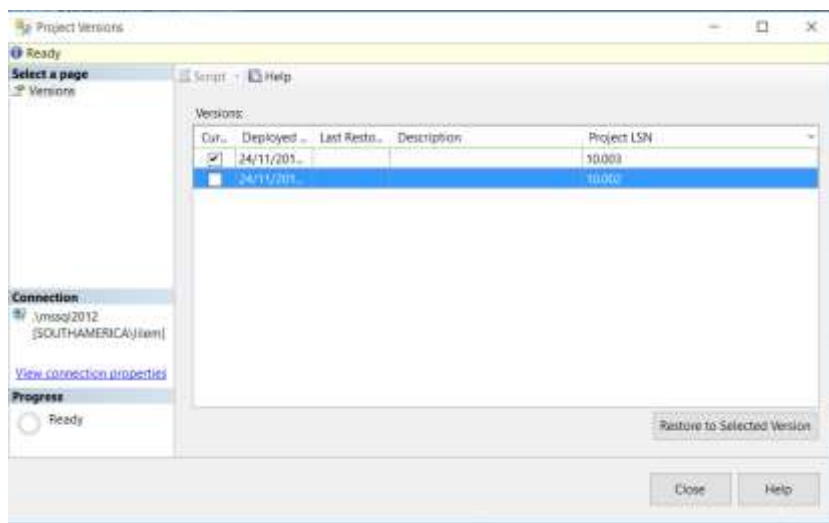


Figura 23 Restore to Select Version

#### Recomendação:

- ✓ Use o deployment model para ter um uso mais flexível da configuração do SSIS.

## 2.13 SSIS e Azure

O SSIS é uma ferramenta estratégica em termos de transferência de dados.

Atualmente existe um feature pack disponível no MSDN para download gratuitamente, a fim disponibilizar um conjunto de novas tarefas no SSIS que permitirão a carga de dados em cenário híbridos. Nestes cenários híbridos seria possível por exemplo carregar dados do SQL Server on premisses para o Azure, do Azure para o SQL Server ou ainda de Azure para Azure.

Este tipo de funcionalidade permite facilitar a integração de soluções mais complexas.

## 3 Conclusão

SSIS é uma ferramenta de ETL com imenso potencial para ajudar no processo estratégico de carga de dados. No entanto, é preciso entender como utilizar e em qual situação aplicar os componentes disponíveis nessa ferramenta.

A combinação de recursos do banco de dados com os recursos do SSIS para transferência de dados pode trazer excelentes benefícios de performance em diversos cenários. Avalie o cenário existente e veja se a SSIS é uma solução possível para o seu caso.

Ao implementar a mesma, analise, teste, revise e siga as melhores práticas deste manual, de modo a obter a melhor performance, configuração e segurança do Integration Services.

## 4 Anexo

### 4.1 Como o SSIS funciona

O processo de fluxo de dados dentro do SSIS cria um buffer – objeto administrado e armazenado com dados em memória – a fim de armazenar as informações recebidas a partir da origem dos dados. Após isto, se aplicável, passará por transformações em memória, até que seja persistido no destino configurado.

A criação, alocação e administração do buffer é feito pelo buffer manager do SSIS, o qual é diferente do buffer manager do SQL. O buffer manager irá gerenciar os dados do dataflow pipeline. Esses dados podem ser compartilhados entre vários componentes dentro do pipeline do SSIS (origem, transformação e destino). Isso faz com que seja desnecessária a realização da cópia física dos dados cada vez que uma transformação é aplicada, por exemplo.

Portanto, os dados serão lidos da origem configurada e em seguida adicionados ao buffer. Uma vez que o buffer esteja cheio, ele será passado para a próxima transformação. Se esta for assíncrona, um novo buffer será criado com os dados lidos e processados, o buffer anterior será descartado para reuso. Se for síncrona os dados serão lidos, processados e passados para a próxima transformação até chegar no destino.

### 4.2 Conector Oracle Attunity

A seguir será tratado passo a passo como instalar e configurar o conector do Oracle Attunity. Antes de iniciar a instalação, lembre-se que é necessário reiniciar o serviço do Integration Services em um determinado momento causando indisponibilidade. Portanto, escolha um horário apropriado para a instalação do conector.

#### **Versões suportadas:**

- ✓ Oracle Attunity 1.0 – SSIS 2008 ( Enterprise ou Development version) / Oracle Client version 9.2.0.4, 10.x ou 11.x
- ✓ Oracle Attunity 2.0 – SSIS 2012 ( Enterprise ou Development version) / Oracle Client version 9.2.0.4, 10.x ou 11.x
- ✓ Oracle Attunity 3.0 – SSIS 2014 ( Enterprise ou Development version) / Oracle Client version 10.x , 11.x ou 12c

#### **Instalação**

- ✓ O Cliente Oracle precisa ser instalado em um computador que esteja usando SQL Server Integration Services (SSIS).
- ✓ Nas plataformas 64 bits (x64 ou I64), é necessário instalar o Oracle Attunity 64 bits para a arquitetura apropriada (x64 ou I64). O Conector Oracle Attunity 32 bits não deve ser instalado junto com as versões 64 bits. Isto ocorre pelo fato do 64 bits instalar tanto o 32 bits quanto o 64 bits. Além disto, o BIDS (SQL Server Business Intelligence Studio) roda em 32 bits, logo a versão do conector a ser instalada refere-se a versão do computador.
- ✓ Se a pacote roda em 32 bits, não é necessário instalar o cliente do Oracle 64 bits.

### **Passos da instalação:**

- ✓ Instale o Oracle Attunity 64 bit connector
- ✓ Instale o client do Oracle 32 bits
- ✓ Instale o cliente do oracle 64 bits
- ✓ Reinicie o computador

### **Onde baixar o Oracle Attunity e instalá-lo:**

- 1) Faça o Download do Attunity do site da Microsoft a partir dos links abaixo, como a figura a seguir exemplifica.

Attunity 2.0 (<http://www.microsoft.com/en-us/download/details.aspx?id=29283>)

Attunity 3.0 (<http://www.microsoft.com/en-us/download/details.aspx?id=44582>)

**Nota:** Verifique as informações referente a compatibilidade do sistema operacional contidas no site para o download.



Figura 24 Download Oracle Attunity

- 2) No exemplo da figura abaixo, está sendo feito o download da versão do Oracle Attunity x64 bits. Selecione a opção a qual se aplica ao computador no qual o conector vai ser instalado.

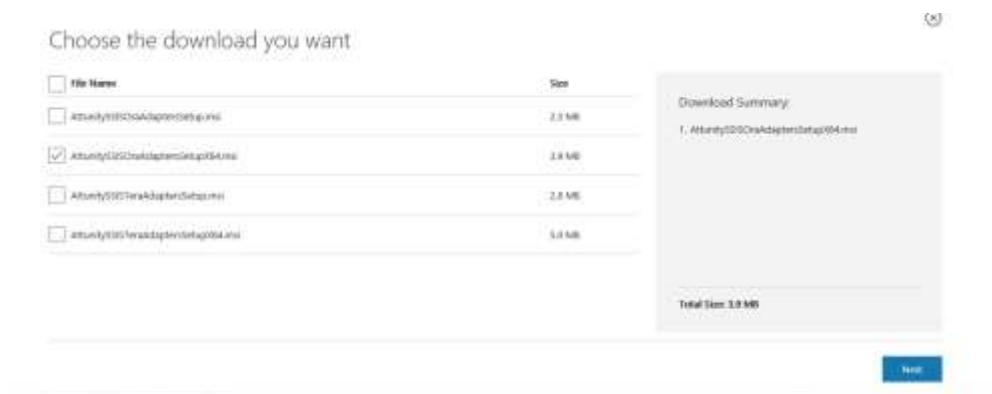


Figura 25 Selecione Oracle Attunity x64

- 3) Após baixar o msi, instale-o clicando duas vezes. No caso, de acordo com o exemplo o arquivo seria: AttunitySSISOraAdaptersSetupX64.msi. O próximo passo consiste em começar a instalação, conforme mostra a figura, clique em next.

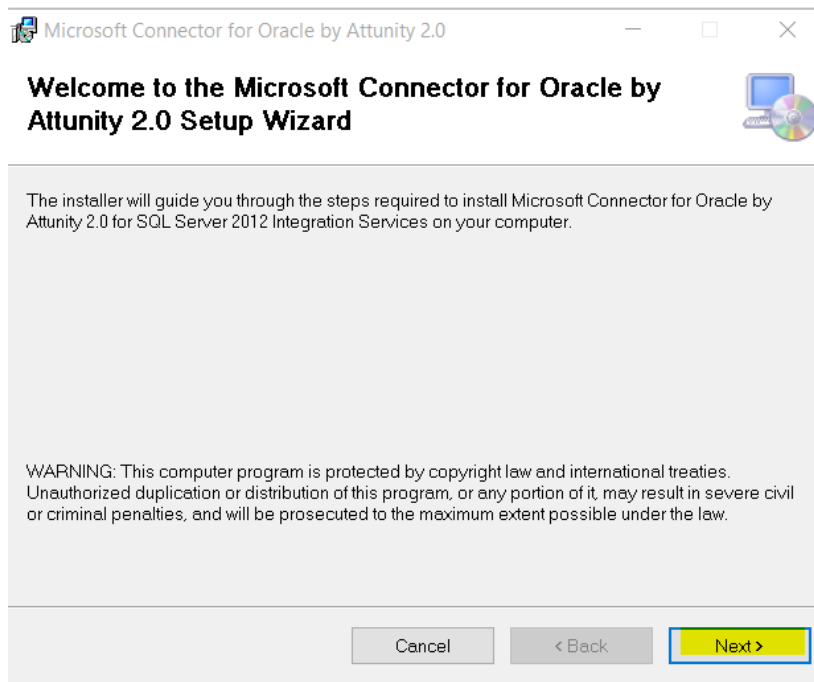


Figura 26 Microsoft Connector Oracle Attunity Tela 1

- 4) E novamente clique em next. Note que o conector deste exemplo é o 2.0 do SQL server 2012, conforme mostra a figura.

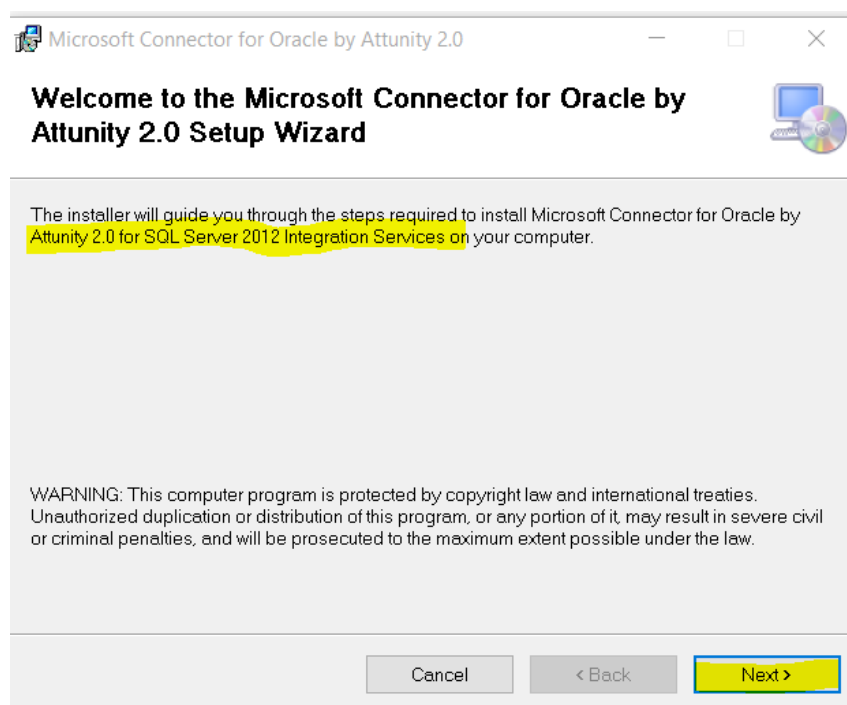


Figura 27 Microsoft Connector Oracle Attunity Tela 2

- 5) Leia e aceite a licença para prosseguir. Conforme mostra a figura, clique em next.



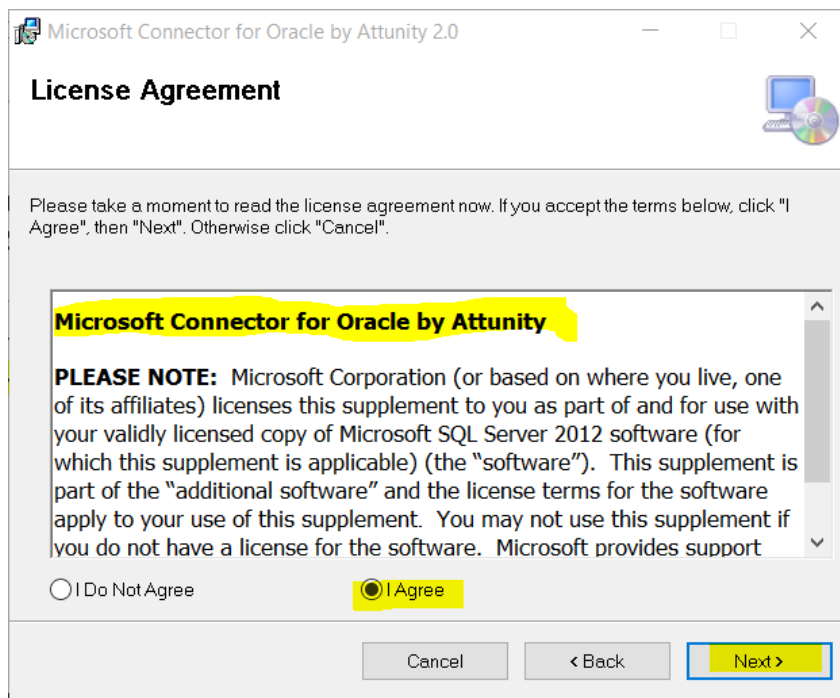


Figura 28 Microsoft Connector Oracle Attunity Tela 3 Licença

- 6) Escolha o lugar em que o conector será instalado, apontando esta informação em folder, a pasta default é c:\program files\... Escolha se apenas o usuário que instalou (Just me) o conector poderá vê-lo ou qualquer usuário (everyone) que logar na máquina. Clique em next conforme mostra a figura.

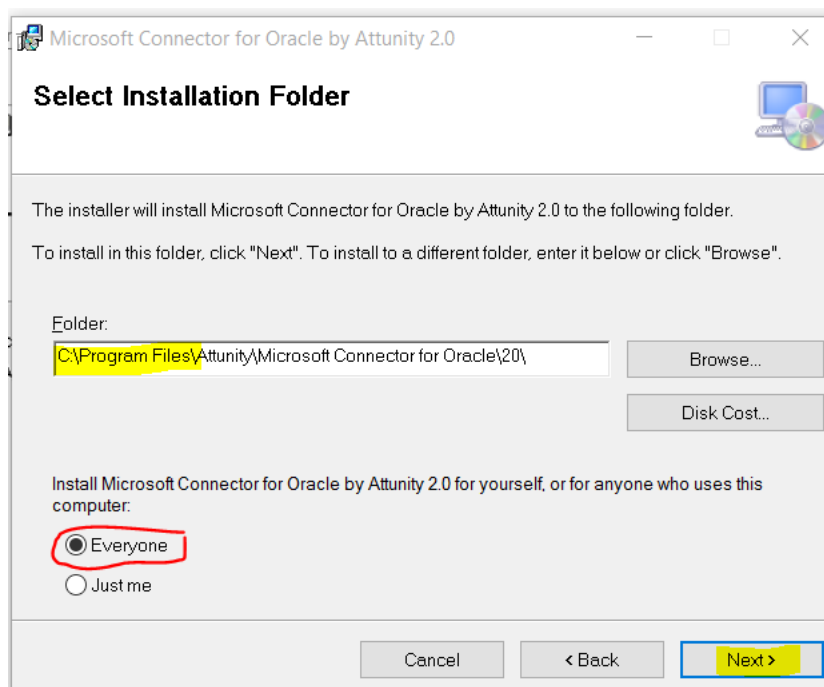


Figura 29 Microsoft Connector Oracle Attunity Tela 4 Local de Instalação

- 7) Confirme a instalação, clicando em Next, conforme mostra a figura.

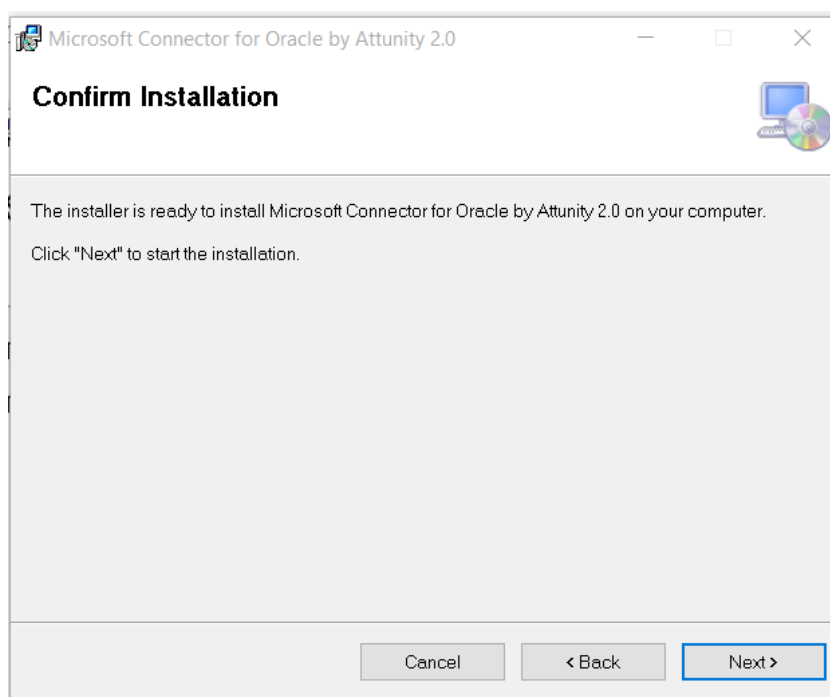


Figura 30 Microsoft Connector Oracle Attunity Tela 5 Confirme a instalação

8) A instalação vai começar, conforme mostra a figura.

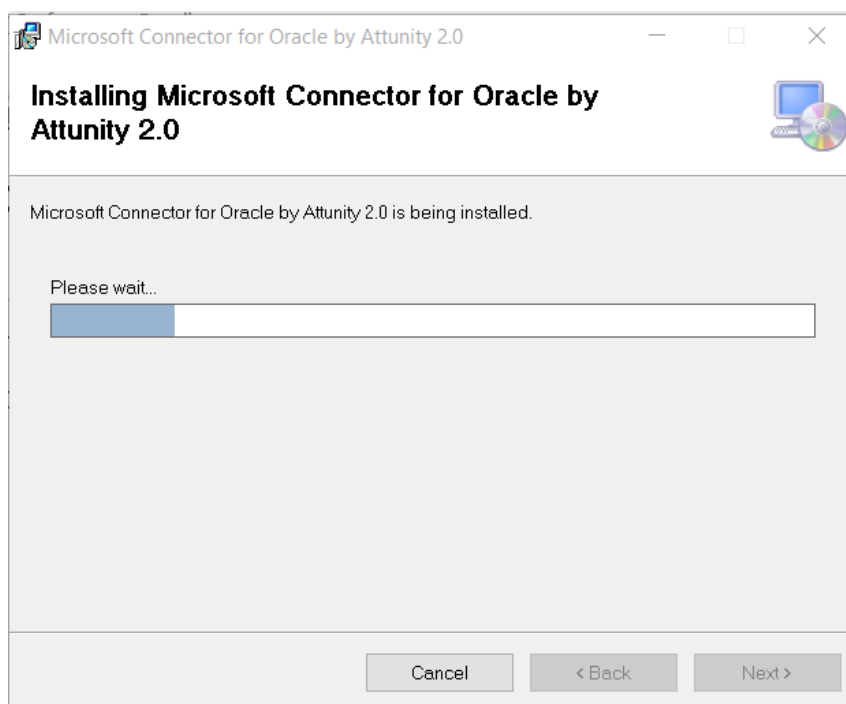


Figura 31 Microsoft Connector Oracle Attunity Tela 6 Instalando

9) Aparecerá uma mensagem informando que o serviço do SQL Service Integration Service será **reiniciado**. Avalie o melhor horário para esta indisponibilidade e clique em Yes, conforme mostra a figura.

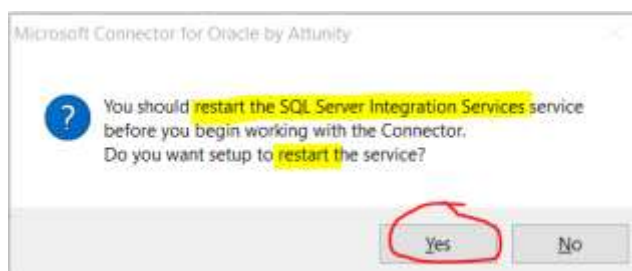


Figura 32 Microsoft Connector Oracle Attunity Tela 7 Restart SSIS Service

10) E a instalação terá sido finalizada, conforme mostra a figura.

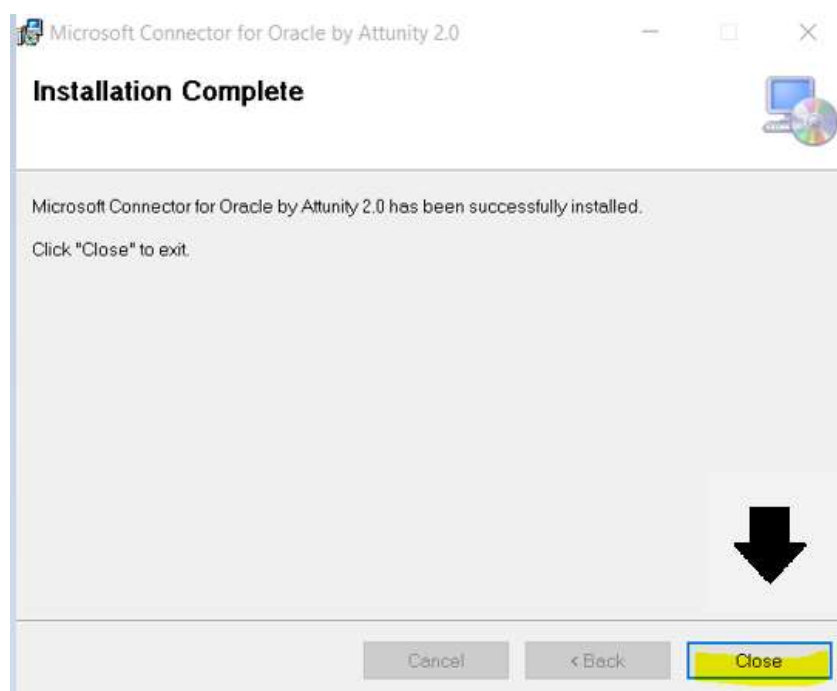


Figura 33 Microsoft Connector Oracle Attunity Tela 8 Finish

11) Após isto, é necessário adicionar o conector ao SSIS. Para adicionar, abra o Data Tools (SQL Server 2012) ou o BIDS (SQL 2008). Clique em New -> Project e selecione Integration Services, conforme mostram as figuras a seguir.

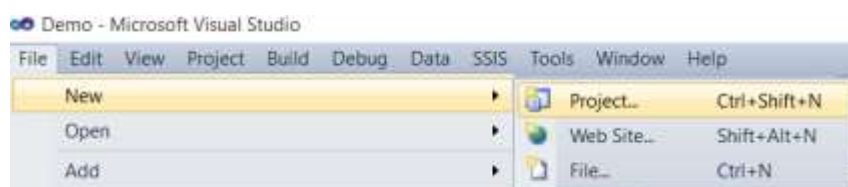


Figura 34 Microsoft Connector Oracle Attunity - Data tools parte 1

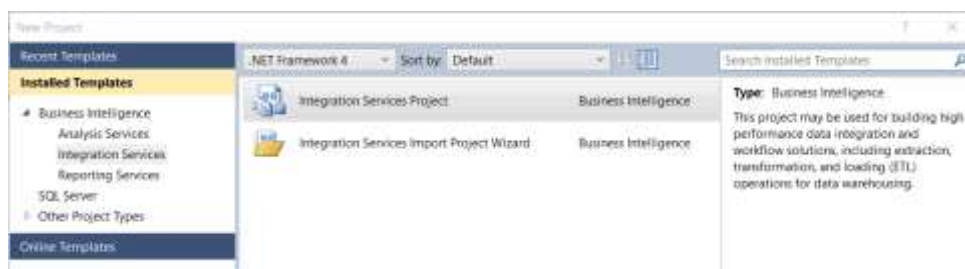


Figura 35 Microsoft Connector Oracle Attunity - Data tools New SSIS

12) Abra a toolbox do SSIS, disponível ao lado esquerdo do projeto.



Figura 36 Microsoft Connector Oracle Attunity - Data tools Toolbox

13) Clique com o botão direito e escolha a opção Choose Items, conforme mostra a figura.

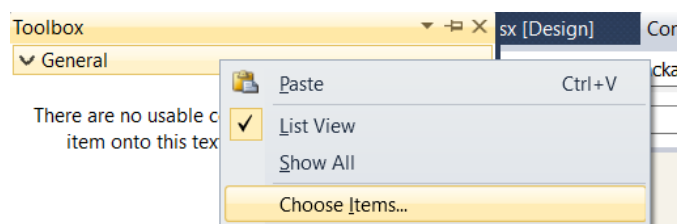


Figura 37 Microsoft Connector Oracle Attunity - Data tools Choose Items

14) Filtre por ORA e adicione os componentes, conforme figura.

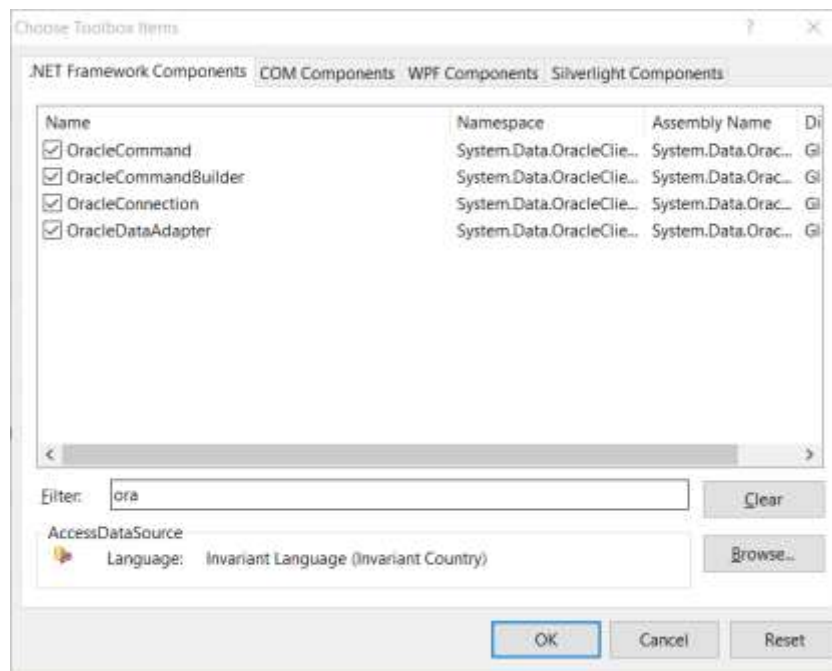


Figura 38 Microsoft Connector Oracle Attunity - Data tools ORA

15) Os componentes estarão disponíveis na SSIS Toolbox

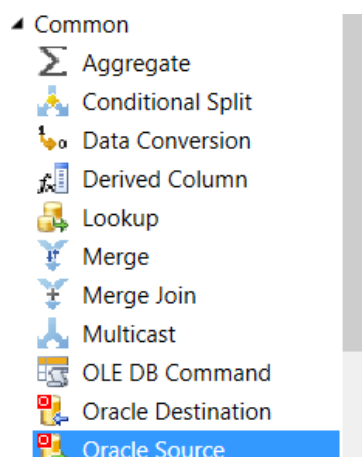


Figura 39 Microsoft Connector Oracle - Data tools Oracle Source/Destination

## 4.3 Habilitar log do SSIS

Existem algumas maneiras de se gravar logs do pacote SSIS. Para habilitar o processo de log do SSIS, basta seguir os passos abaixo:

- 1) Clique com o botão direito do mouse em qualquer área livre do dataflow:

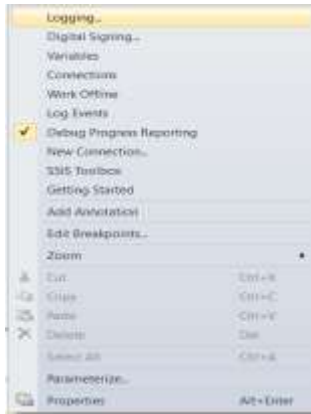


Figura 40 Logging Option

- 2) Selecione a opção Logging.
- 3) Selecione aonde o log será armazenado: Arquivo txt, SQL Server, Profiler, Windows Event Log ou XML.

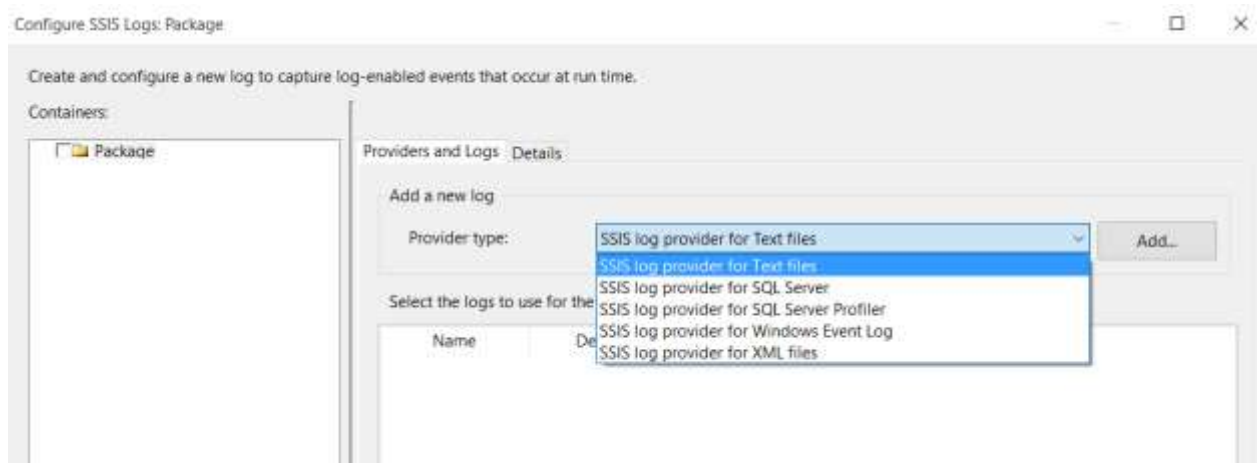


Figura 41 Log Types

- 4) Selecione uma das opções e clique em Add...

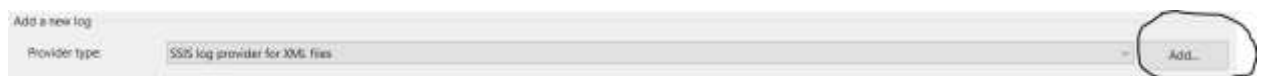


Figura 42 Add..Log Type

- 5) Qualquer uma das opções escolhidas será preciso clicar em Configuration e criar uma conexão para o local aonde os dados serão persistidos, seja este um caminho no disco ou um banco de dados. O único evento que não precisa indicar no Configuration é o Event Log.


Name	Description	Configuration
<input type="checkbox"/> SSIS log provider for Text files	Writes log entries for events to a CSV file	
<input checked="" type="checkbox"/> SSIS log provider for Windows Event Log	Writes log entries for events to the Event Log	 ⇒ não precisa indicar
<input type="checkbox"/> SSIS log provider for SQL Server Profiler	Generates SQL traces that may be captured in SQL Server Profiler	
<input type="checkbox"/> SSIS log provider for SQL Server	Writes log entries for events to a SQL Server database	
<input type="checkbox"/> SSIS log provider for XML files	Writes log entries for events to an XML file	

Figura 43 Provider Type

- 6) Em containers, ao lado esquerdo da tela de logging, é possível configurar quais tarefas serão auditadas. Selecione o nome do pacote (neste exemplo é Package mesmo) ou ainda selecione outras opções como dataflow, execute sql task, etc dependendo do que existir no pacote. A

seleção ocorre clicando duas vezes com o mouse. As opções disponíveis irão variar de acordo com a quantidade de tarefas que existirem no pacote.

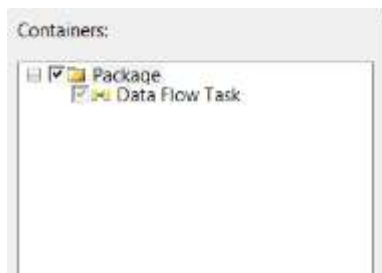


Figura 44 Containers Log

- 7) Ao se selecionar as opções disponíveis em containers que serão logadas, clique em Details (ao lado de provider and logs), e selecione o que será auditado. Por exemplo: se vai se auditar apenas erros ou warnings, a validação de uma task, ou ainda quando uma tarefa é executada.

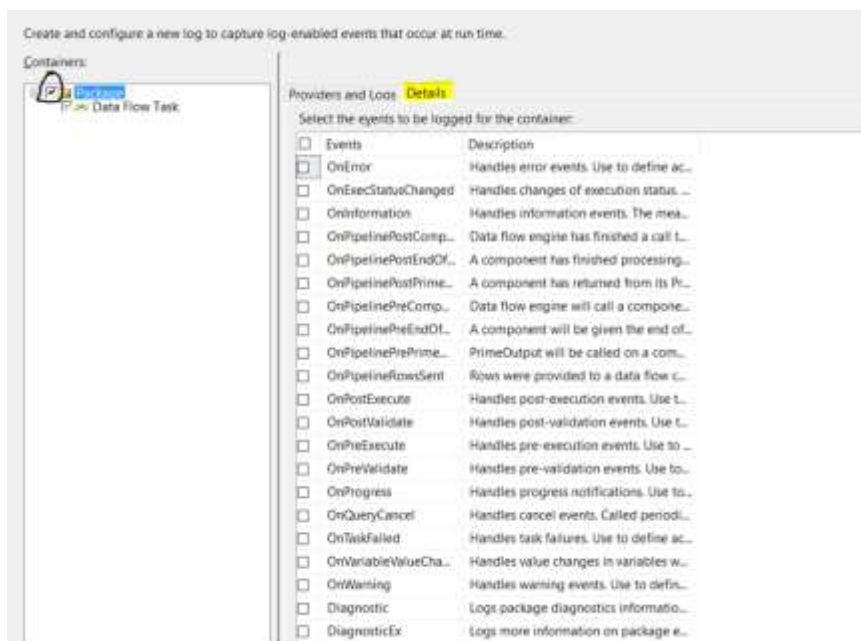


Figura 45 Select events logs

### Logs gravados no SQL Server

Caso a opção escolhida para gravar os dados de log seja o SQL Server, uma tabela chamada `[dbo].[sysssislog]` será criada pelo pacote no banco de dados configurado na opção de logging. Portanto, certifique-se que o usuário que irá rodar a pacote tenha permissão para criar esta tabela. Uma outra alternativa é criar a tabela antes e dar permissão de inserção nesta tabela para o usuário que rodar o pacote.

Listagem 4 Criação da tabela `[dbo].[sysssislog]`

```
CREATE TABLE [dbo].[sysssislog](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [event] [sysname] NOT NULL,
    [computer] [nvarchar](128) NOT NULL,
```

```
[operator] [nvarchar](128) NOT NULL,
[source] [nvarchar](1024) NOT NULL,
[sourceid] [uniqueidentifier] NOT NULL,
[executionid] [uniqueidentifier] NOT NULL,
[starttime] [datetime] NOT NULL,
[endtime] [datetime] NOT NULL,
[datacode] [int] NOT NULL,
[databytes] [image] NULL,
[message] [nvarchar](2048) NOT NULL,
PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

### **Logs gerados através do Event Handlers**

Outra opção que pode ser usada para Log são os Event Handlers. Os Event Handlers são eventos configurados e disparados por executáveis dentro do control flow. Para acessar essa opção, clique em event handler ao lado de parameters, conforme mostra a figura. Expanda em Executables a tarefa que será logada.

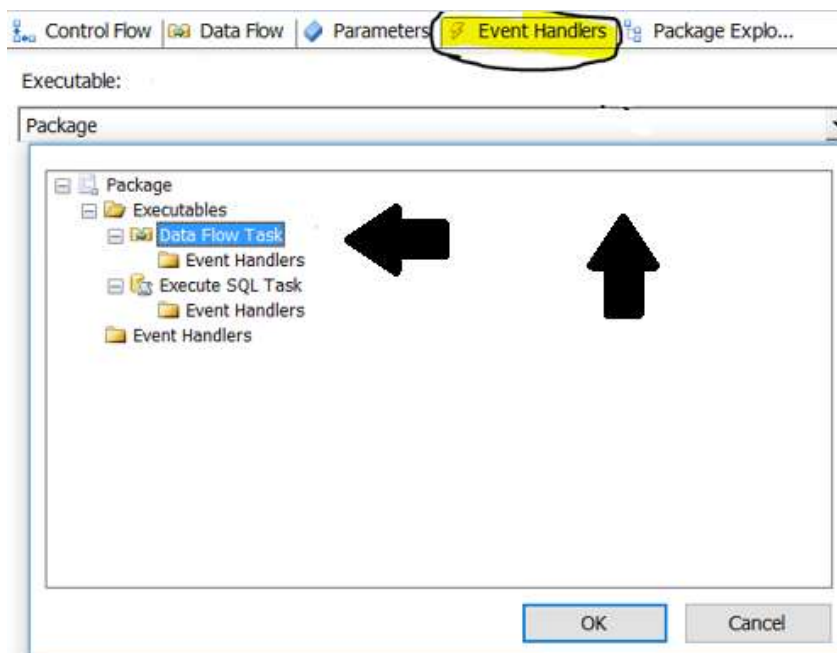


Figura 46 Task logada event handler

Perceba que as tarefas dentro do Data Flow não aparecem disponíveis. Isso por que elas não são consideradas executáveis. Apenas o data flow é considerado executável e as tasks dentro dele fazem parte do executável data flow. Logo, os eventos só podem ser configurados até o nível do control flow e não do data flow.



Uma vez selecionado o executável que será monitorado, seleciona-se o evento que vai causar o event handler disparar. Este evento, pode ser um erro, uma validação, um cancelamento etc.

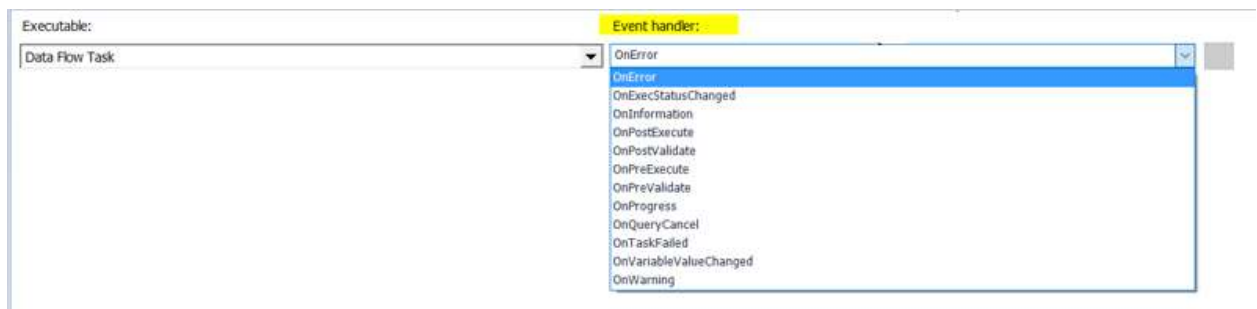


Figura 47 Eventos do event handler

A partir do disparo do event handler, configura-se uma ação que poderia ser um novo fluxo ou a customização da gravação de um evento de log no Execute SQL Task, por exemplo.



Figura 48 Event handler custom

A principal diferença entre a opção de logging do SSIS e o Event Handler é que o event handler permite ações adicionais baseado em eventos, como por exemplo, um determinado evento pode criar um fluxo de processos dentro de um dataflow.

Existem outras formas customizadas de logar o processamento do pacote. É relevante que exista sempre um controle de log do pacote e que este controle seja o suficiente para entender aonde ocorreu uma determinada falha no pacote, mas ao mesmo tempo não impacte a performance do pacote.

## 4.4 Como habilitar as configurações do SSIS

### Habilitar configuração no SQL 2008

Clique no menu SSIS e selecione package configurations.

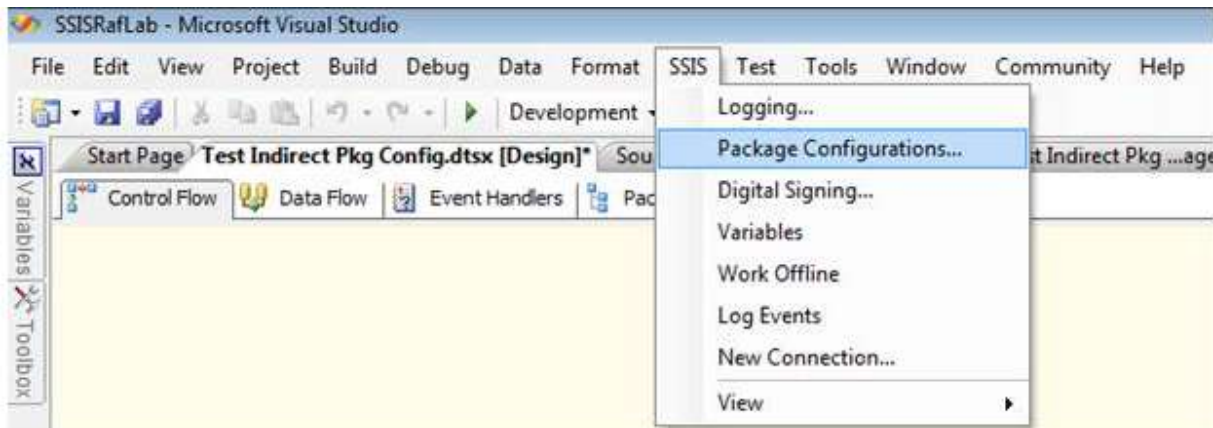


Figura 49 package configurations SSIS 2008

## Habilitar configuração no SQL 2012

É possível habilitar a configuração através do menu SSIS, como no SQL 2008, mas também da seguinte forma

- 1) Clique nas propriedades do pacote e em Configurations selecione (Collection). Uma janela para criação de configuração será aberta.

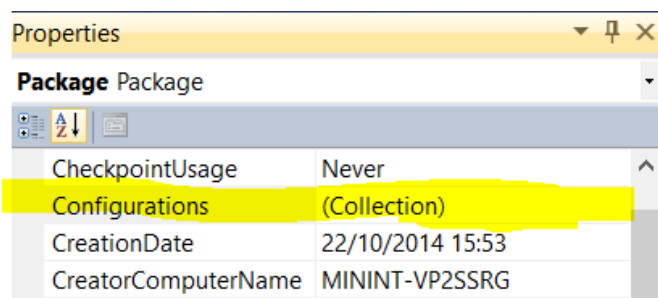


Figura 50 Configurations SQL 2012

- 2) Clique no checkbox enable configuration e em seguida no botão Add..

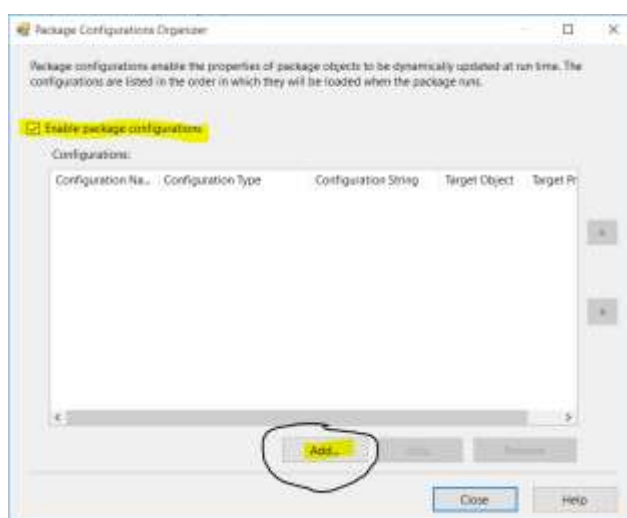


Figura 51 Enable package configuration

## Habilitar configuração no SQL 2008 ou 2012

A partir deste ponto os passos para realizar as configurações são as mesmas tanto para o SQL 2008 e 2012.

- 1) Selecione a configuração que será aplicada através do combo *configuration type*.

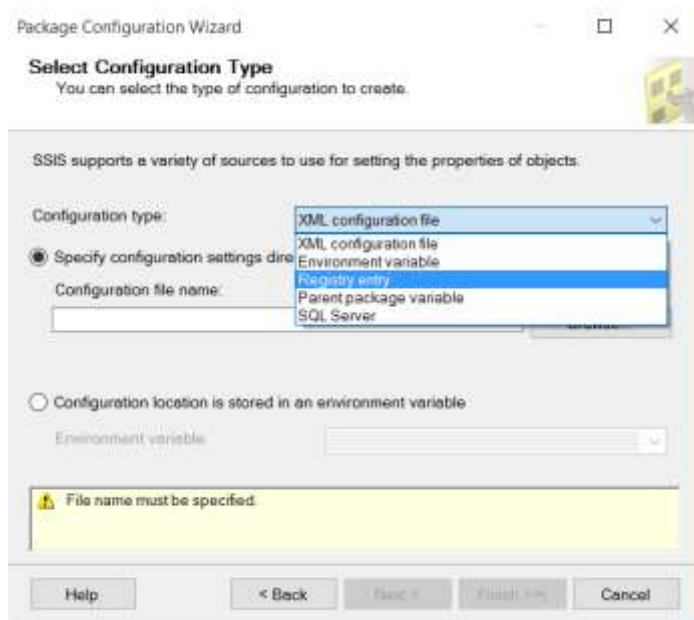


Figura 52 Configuration type Sql 2012

- 2) Próximo passo consiste em escolher qual configuração vai ser aplicada. No exemplo a seguir, foi escolhido SQL Server, ou seja, as configurações serão salvas numa tabela no banco de dados.

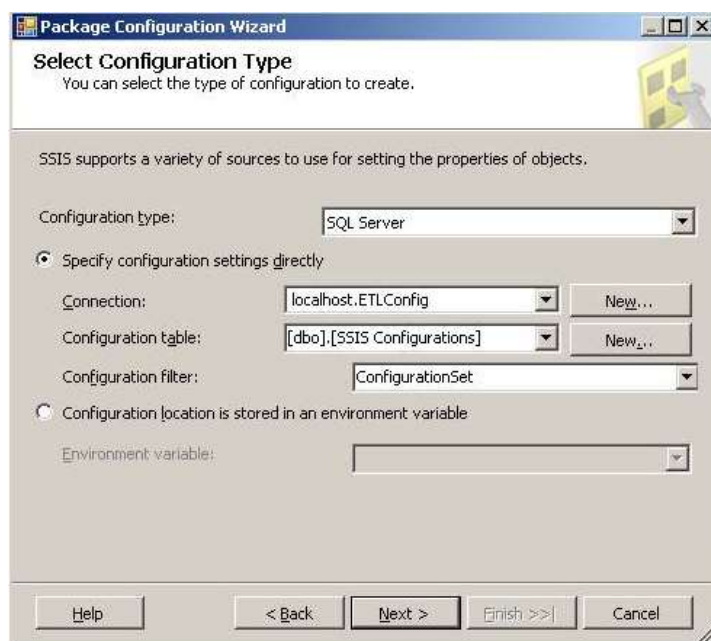


Figura 53 Select configuration type

3) Selecione a configuração da pacote e o valor que você vai armazenar.



Figura 54 Select properties to export

Basicamente, as propriedades do pacote são expostas para que possam ser alteradas com novos valores através da configuração criada, seja isso em arquivo, tabela, variável de ambiente, registry ou parent package variable. Por padrão, estas configurações vêm desabilitadas e podem ser habilitadas conforme já mencionado.

Note que:

- ✓ As configurações do SQL são muito flexíveis. Múltiplas configurações podem ser armazenadas em arquivos ou tabelas.
- ✓ Múltiplos pacotes podem compartilhar a mesma configuração.
- ✓ Podem existir filtros diferentes para uma mesma tabela de configuração (opção de *configuration type* SQL Server) sendo compartilhadas por diferentes pacotes.
- ✓ As políticas de segurança do banco de dados e backup, podem ser reaproveitadas se a configuração estiver configurada numa tabela no banco de dados.
- ✓ O pacote cria a tabela de configuração no banco de dados selecionado.

## 4.5 Deployment model 2012

Para converter a pacote para o Deployment Model, é preciso dentro do Data Tools selecionar o projeto através do Solution Explorer, conforme figura a seguir, clicar com o botão de mouse em cima dele e convertê-lo para Deployment Model, caso ainda não esteja configurado desta forma.

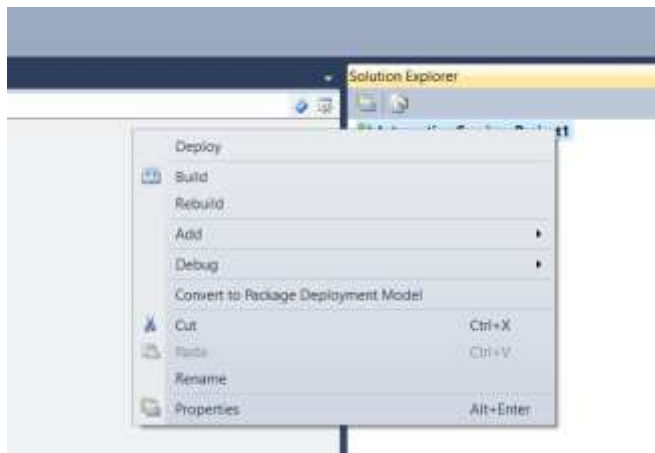


Figura 55 Converte to package deployment model

Configure dentro do pacote um parâmetro para o banco de dados, por exemplo com o nome de Databasename. Este parâmetro vai servir de referência para a configuração de ambiente, fazendo com que de acordo com o que estiver configurado nos ambientes do servidor, o pacote saiba em que banco de dados executar a carga dos dados. Para isso, basta abrir o pacote no Data Tools e na aba *Parameters*, conforme figura abaixo.



Figura 56 Parameters Database

Após isso faça o deploy do pacote. Para isso, novamente clique com o botão direito em cima do projeto no solution explorer conforme figura – dentro do data tools – e escolha a opção deploy.



Figura 57 Data tool deploy

Durante o deploy, escolha o servidor em que será feito a publicação e o caminho no qual o projeto ficará armazenado no SQL, dentro da na SSIS Catalog (SSISDB), conforme exemplificado na figura a seguir.

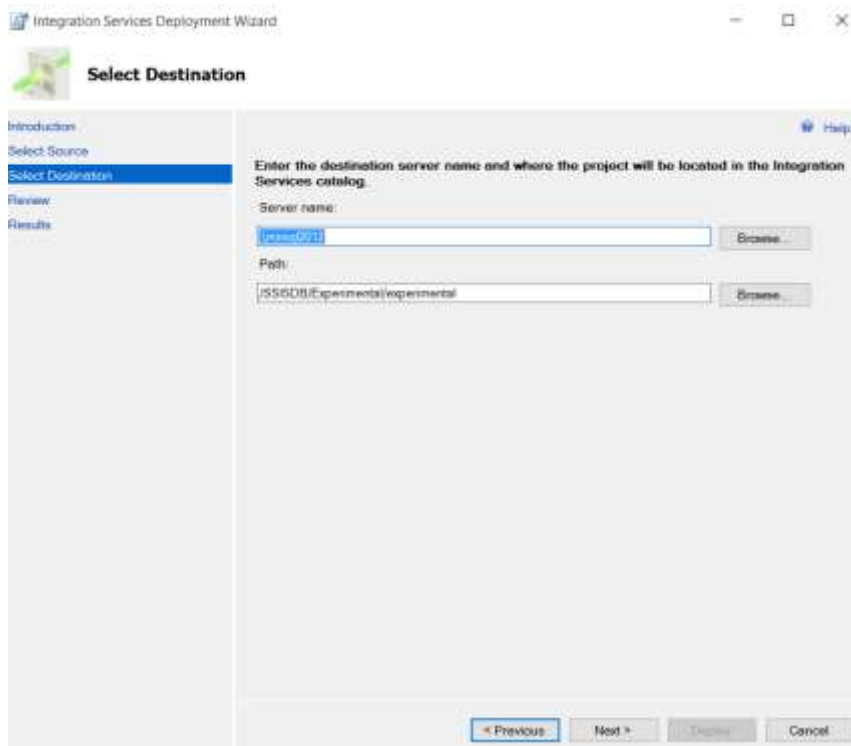


Figura 58 Deploy destination

Para configurar os ambientes dentro da SSIS Catalog, basta clicar com o botão direito do mouse em environments, disponível dentro da SSISDB, expandindo a pasta na qual o projeto foi publicado, como mostra a figura.

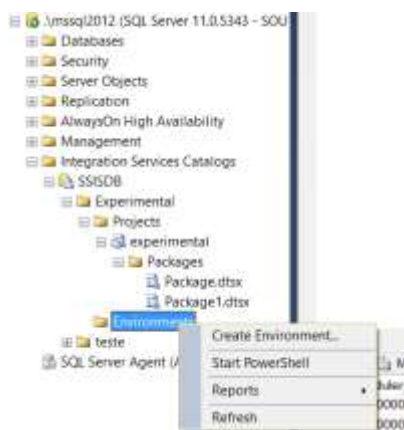


Figura 59 SSIS Catalog – Environment

Depois de criada a variável Environment, basta clicar com botão direito do mouse em cima dele e selecionar as propriedades. Procure pela aba Variables e crie as variáveis de configuração de ambiente. Por exemplo, para desenvolvimento crie uma variável chamada Databasename (como o nome do parâmetro criado dentro da pacote) que aponte para o banco de dados de desenvolvimento que se chama Dev\_DB, conforme a figura. Para o ambiente de produção, teria-se a mesma variável chamada Databasename com o valor de banco de dados de dados de produção, que seria o Prod\_DB.

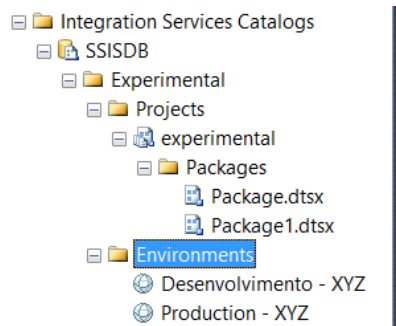


Figura 60 Environments

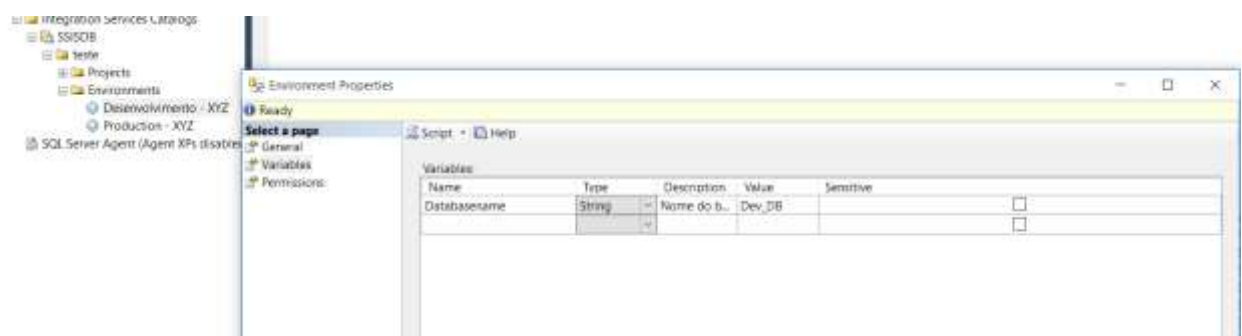


Figura 61 SSIS Catalog Variable environment

Uma vez configurado o ambiente é preciso associá-lo aos pacotes que foram publicadas na SSIS Catalog, conforme mostra a figura abaixo. Para isso selecione a pacote com o botão direito do mouse, dentro do projeto publicado e escolha a opção configure.

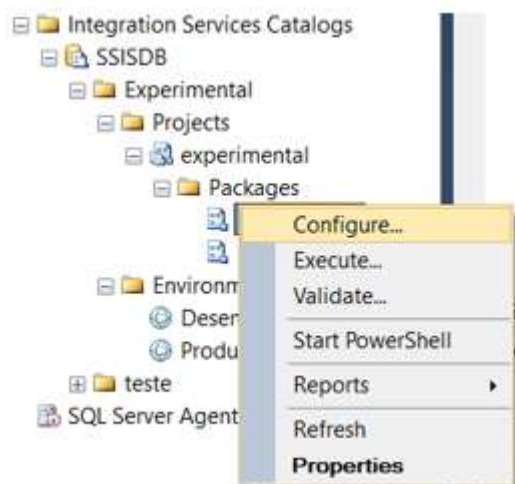


Figura 62 SSIs Configure reference environment

Selecione References, clique em Add... e referencie o ambiente, por exemplo de Desenvolvimento e clique em OK.

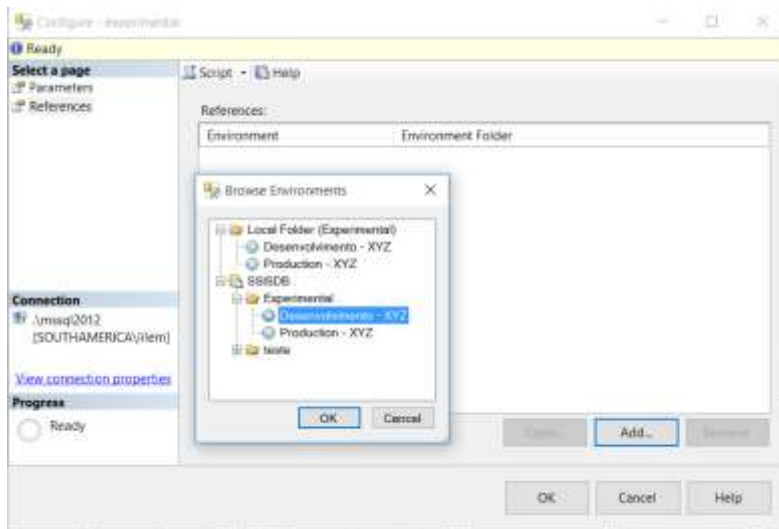


Figura 63 Dev SSISDB

Depois disso, clique novamente com o botão direito em cima da pacote e escolha execute. Conforme figura.

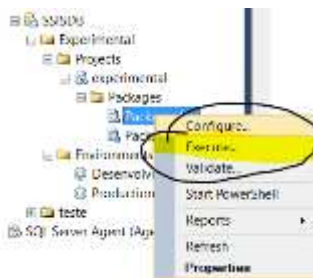


Figura 64 Configure Reference

Perceba que apenas os ambientes que foram configurados em *Reference* (opção *Configure*) estarão disponíveis para a execução, ao se habilitar Environment, conforme mostra a figura a seguir.

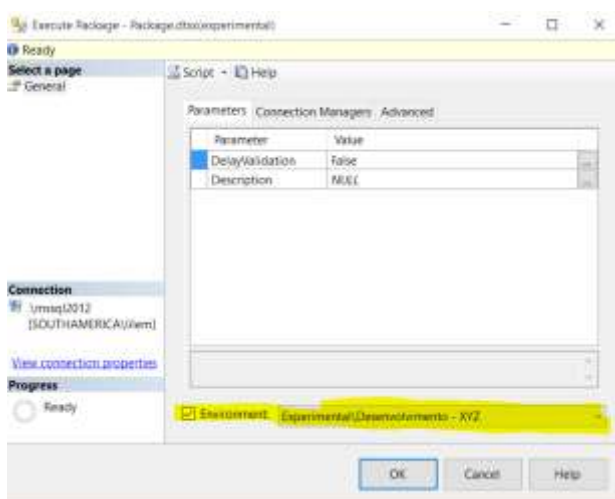


Figura 65 Enviroment execute

Portanto, a partir do SQL 2012, a questão de configuração do banco de dados em diferentes ambientes passou a ser consideravelmente mais fácil e flexível.



# 5 Referências

## SSIS Best Practices

[http://download.microsoft.com/download/F/A/C/FACAEBFA-A22F-4990-8DA5-80EC2F088B1A/02b\\_Ssis\\_Best\\_Practices.pdf](http://download.microsoft.com/download/F/A/C/FACAEBFA-A22F-4990-8DA5-80EC2F088B1A/02b_Ssis_Best_Practices.pdf)

## Using the Microsoft Connector for Oracle by Attunity with SQL Server 2008 Integration Services

[https://technet.microsoft.com/en-us/library/ee470675\(v=sql.100\).aspx](https://technet.microsoft.com/en-us/library/ee470675(v=sql.100).aspx)

## OLE DB Command Transformation

<https://technet.microsoft.com/en-us/library/ms141138.aspx>

## SQL Server 2005 Integration Services: A Strategy for Performance

<https://technet.microsoft.com/library/Cc966530>

## Using the Microsoft Connector for Oracle by Attunity with SQL Server 2008 Integration Services

[https://technet.microsoft.com/en-us/library/ee470675\(v=sql.100\).aspx](https://technet.microsoft.com/en-us/library/ee470675(v=sql.100).aspx)

## Attunity connector for SSIS Oracle

<http://www.attunity.com/products/attunity-connect/ssis-connectors-for-oracle-and-teradata>

## Microsoft Connectors v2.0 for Oracle and Teradata

<http://www.microsoft.com/en-us/download/details.aspx?id=29283>

## Data Loading Performance Guide

<https://msdn.microsoft.com/en-us/library/dd425070.aspx>

## Considerations for High Volume ETL using SQL Server

<https://technet.microsoft.com/en-us/library/cc671624.aspx>

## OLE DB Command Transformation - TechNet - Microsoft

<https://technet.microsoft.com/en-us/library/ms141138.aspx>

## Using the Microsoft Connector for Oracle by ... - TechNet

[https://technet.microsoft.com/en-us/library/ee470675\(v=sql.100\).aspx](https://technet.microsoft.com/en-us/library/ee470675(v=sql.100).aspx)

## ConnectionManager.ReleaseConnection Method (Object)

<http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.dts.runtime.connectionmanager.releaseconnection.aspx>

## Integration Services (SSIS) Event Handlers

<https://msdn.microsoft.com/en-us/library/ms140223.aspx>

## Understanding Integration Services Package Configurations

<https://msdn.microsoft.com/en-us/library/cc895212.aspx>

Azure Feature Pack

[https://msdn.microsoft.com/en-us/library/mt146770\(v=sql.120\).aspx](https://msdn.microsoft.com/en-us/library/mt146770(v=sql.120).aspx)

<https://msdn.microsoft.com/en-us/library/dd425070.aspx>

<https://technet.microsoft.com/en-us/library/cc671624.aspx>