# Why Use Configurations?

Integration Services packages are used primarily to manage the movement of data from one place to another. To do that, the package needs to know where it is moving data from and where it is moving data to. (I call these places the data sinks.)

Typically Integration Services packages are built on a different environment to where they are intended to be executed in production. For that reason alone, it is important that the package developer has a way of informing the package where those data sinks are located.

# What Is an Integration Services Configuration?

People often refer to the .dtsConfig file that the configuration wizard within the Business Intelligence Development Studio produces as an Integration Services configuration, but that is not the case. A .dtsConfig file is actually a collection of Integration Services configurations, and that distinction is important.

A configuration is a name-value pair. The name is a property within a package that needs to be changed, and the value is the value to be assigned to that property

**Use Configurations Sparingly**

Just because it is possible to set a configuration on a property, it may not be sensible to do so.Be prudent when planning your configurations, and only apply them where necessary. It is rare that you need more configurations than there are connection managers in your package.

**Configure Only the ConnectionString Property for Connection Managers**

After reading that statement, you may ask: *“But what if I want to set the username and password for each connection manager? That’s two configurations already.”*

That’s true.If you want to put configurations on a connection manager, then you should apply a configuration to the ConnectionString property only. There is no need to apply a configuration for each of the ServerName, UserName, Password, and InitialCatalogproperties. Doing so would be cumbersome, and many connection strings require more information than those four properties can provide. So, in the interests of flexibility and ease of administration, ignore those four properties in favor of the ConnectionString property.

**Feature Comparison of Configuration Types**

This table provides a high-level feature comparison of each configuration type.

|  |  |  |  |
| --- | --- | --- | --- |
| Configuration Type | Where is the property path stored? | Where is the value stored? | Indirect configurations possible? |
| XML Configuration File | External to the package | External to the package | Yes |
| Environment variable | In the package | External to the package | No |
| Parent package | In the package | External to the package | No |
| Registry | In the package | External to the package | No |
| SQL Server | External to the package | External to the package | Yes |