

# SSIS Data Integration Lifecycle Management

A technology story by Andy Leonard, Chief Data Engineer, [Enterprise Data & Analytics](#)

# Executive Summary

*This whitepaper is taken from the book [Data Integration Life Cycle Management with SSIS: A Short Introduction by Example](#) (Apress, 2017).*

Microsoft is a software development enterprise. As big as Microsoft is, they cannot possibly respond to *every* request. When I realized this, I began thinking about how I might address gaps I perceived in SSIS lifecycle management. I'd co-founded a consulting company and we (collectively) weren't interested in becoming a software product company. But I was *very* interested in developing products to address gaps in Data Integration Lifecycle Management (DILM).

In 2015 I left the consulting company I co-founded and immediately began developing the software I'd dreamed of building. In my opinion, the most fair answers to the question, "Why?" are:

1. I came to believe the Microsoft SSIS Developer Team would not address items I perceived as "gaps" in the product story; and
2. I came to believe the consulting company I co-founded and I held different visions of how to address DILM issues.

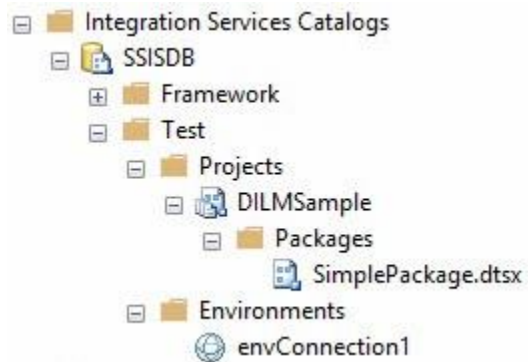
Looking back with years of perspective, I believe focusing on DILM was the best long-term move for me. I started another consulting company, Enterprise Data & Analytics (entdna.com), mostly to fund my coding habit.

In order to practice DevOps with SSIS, enterprises must be able to easily access code configurations and process metadata. SSIS Catalog Browser was designed to surface configurations and metadata stored in the SSIS Catalog. The goal? Present *all* the data on a single surface. Operations professionals need not know "where to dig" to find configurations and process metadata; they need only know how to expand nodes in a tree view control.

The remainder of this whitepaper compares and contrasts the viewer that ships with SSIS and [SSIS Catalog Browser](#), a free utility that is part of the [DILM Suite](#).

# Surfacing the SSIS Catalog

Let's examine the SSIS Catalog surface in the SSMS Object Explorer's Integration Services Catalogs node as shown in Figure 1:



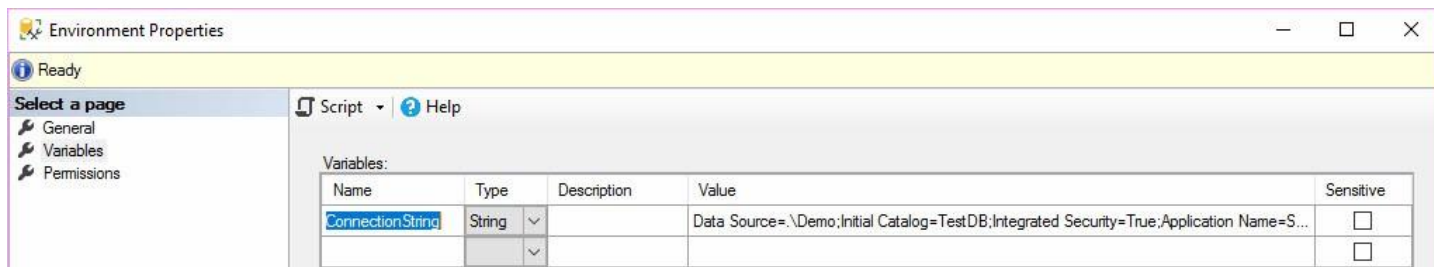
**Figure 1. The SSIS Catalog as Shown in the SSMS Object Explorer Integration Services Catalogs Node**

Beneath the Integration Services Catalogs node we find the SSIS Catalog named SSISDB. Two Catalog Folders are displayed, Framework and Test. The Test Folder contains Projects and Environment virtual folders. The Projects virtual folder contains our SSIS Catalog Project named DILMSample, which in turn contains our SSIS Package named SimplePackage.dtsx. The Environments virtual folder contains our Catalog Environment named envConnection1.

We know – because we've done the work – that there's more there than meets the eye.

## SSIS Catalog Environment Configuration

If we double-click envConnection1 we can see details of our Catalog Environment Variable on the Variables page as shown in Figure 2:

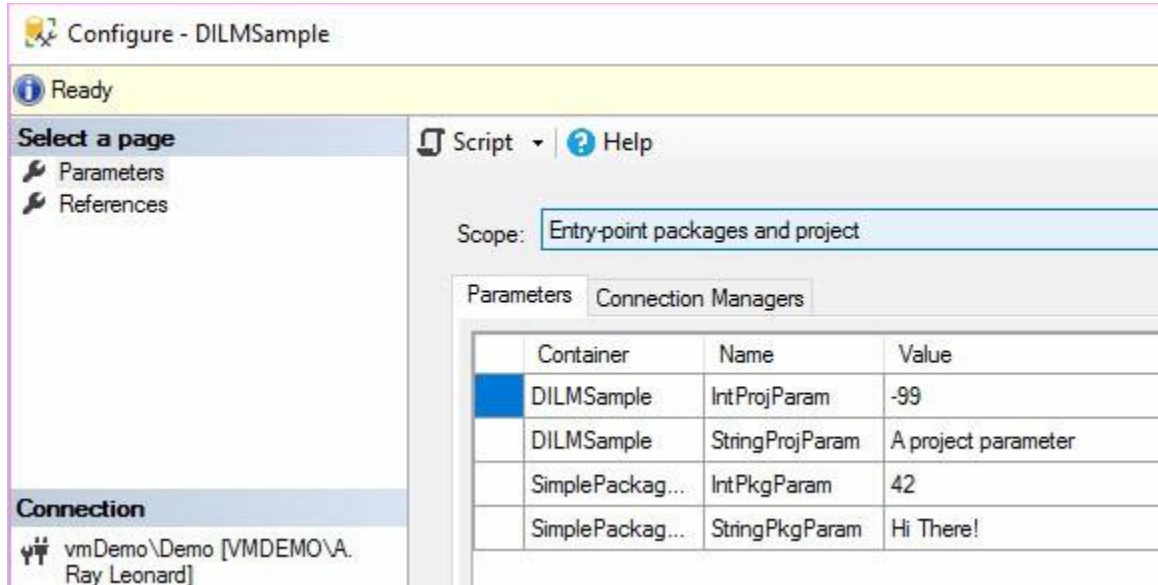


**Figure 2. Viewing the Variables Page of an SSIS Catalog Environment**

The Variables page contains details about SSIS Catalog Environment Variables including name, data type, description, value, and whether the variable is sensitive.

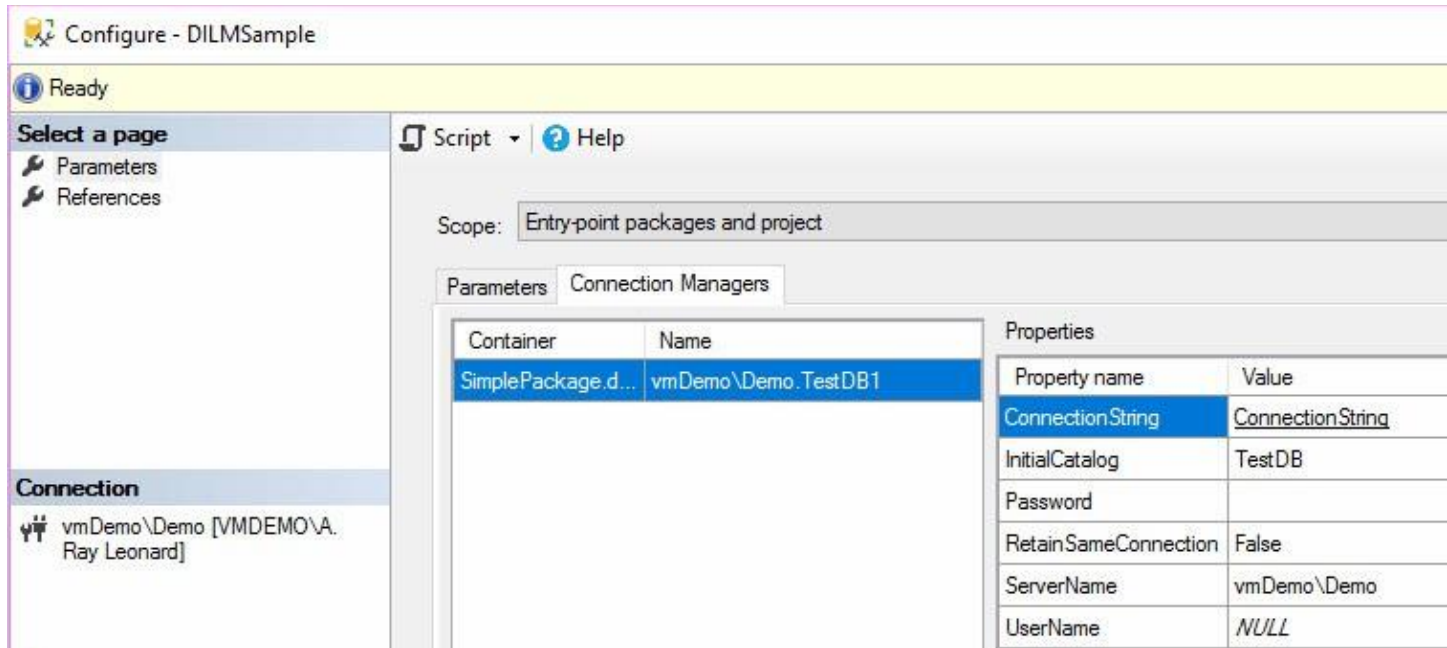
# SSIS Catalog Project Configuration

The Parameters tab on the Parameters page of the SSIS Catalog Project Configuration dialog lists SSIS Project and Package parameters, their container name, and value by default as shown in Figure 3:



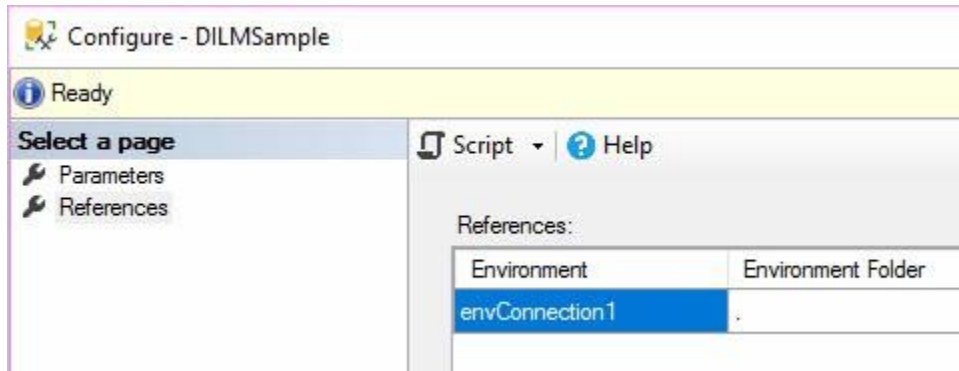
**Figure 3. Viewing Project Parameters and Values for an SSIS Catalog Project**

The Connection Managers tab of the Parameters page contains a list of SSIS Project and Package connection managers and their properties as shown in Figure 4:



**Figure 4. Viewing Connection Manager Parameters and Values for an SSIS Catalog Project**

The References page of the SSIS Catalog Project Configure dialog contains a list of SSIS Catalog Environments the SSIS Catalog Project may reference at runtime as shown in Figure 5:



**Figure 5. Viewing Project References for an SSIS Catalog Project**

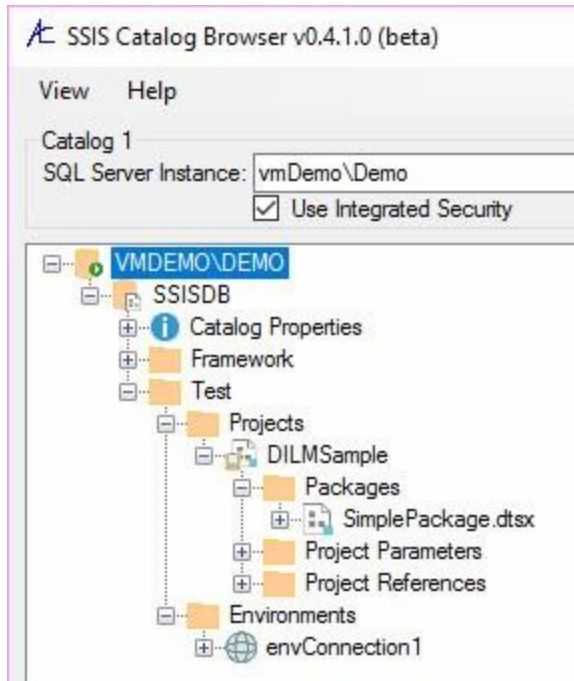
That's a lot of right- and double-clicking just to see what's configured in an SSIS Catalog Project.

## Catalog Browser

The SSIS Catalog is filled with really cool and useful configuration information, but one has to know where to look and – in some cases – *where to look* isn't so obvious.

Enter Catalog Browser, a free utility that is part of the DILM Suite and available at [dilmsuite.com/catalog-browser](http://dilmsuite.com/catalog-browser). Catalog Browser was built to surface the contents of the SSIS Catalog in a single view – a tree that exposes all relevant SSIS Catalog artifacts, properties, and configurations.

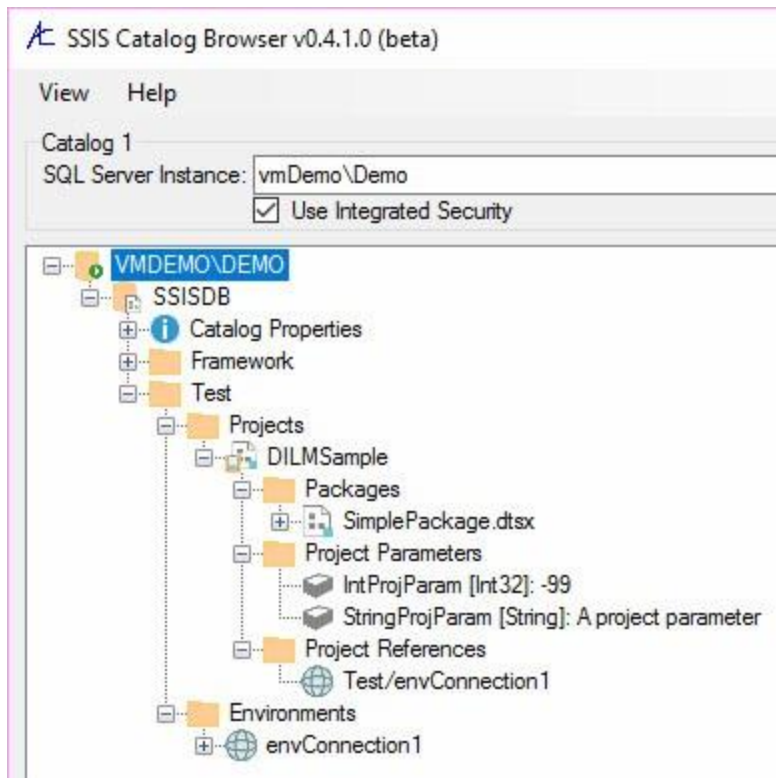
As shown in Figure 6, Catalog Browser surfaces the same metadata as the SSMS Object Explorer Integration Services Catalogs node:



**Figure 6. Catalog Browser Surfacing Part of the SSIS Project and Configurations Metadata**

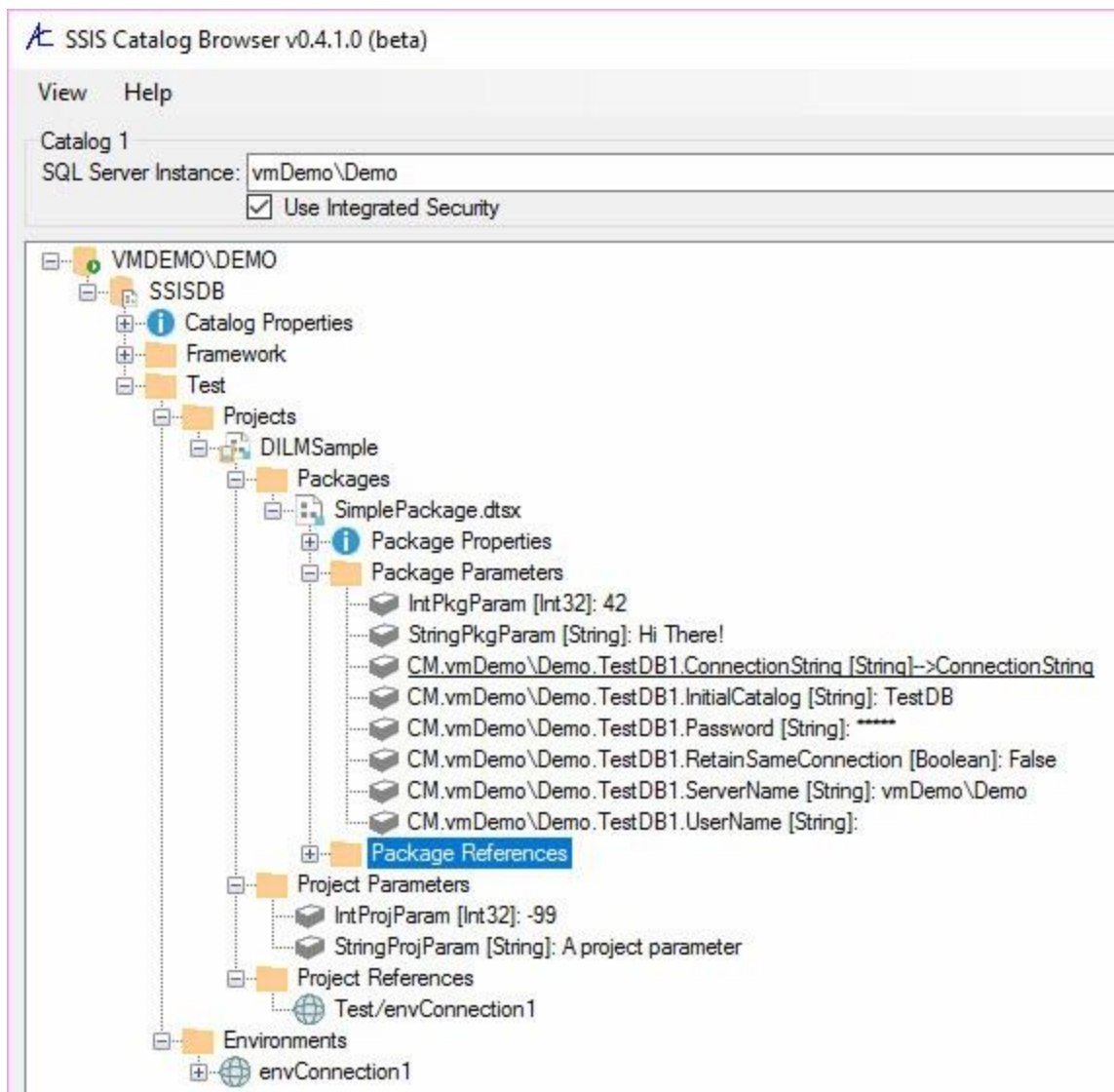
Looking at Figure 6, though, you probably already see some differences between Catalog Browser and the SSMS Object Explorer Integration Services Catalogs node. Please note the Project Parameters and Project References virtual folders present beneath the SSIS Catalog Project, in addition to the Packages virtual folder.

Expanding these virtual folders reveals the SSIS Catalog Project Parameters and Reference as shown in Figure 7:



**Figure 7. SSIS Catalog Project Parameters and References**

Please remember in Figure 3 the SSMS Object Explorer Integration Services Catalogs node surfaced *all* parameters – SSIS Catalog Project Parameters and SSIS Package Parameters. Where are the Package Parameters? They're here in Catalog Browser. To view Package Parameters, expand the SimplePackage.dtsx SSIS Package node as shown in Figure 8:

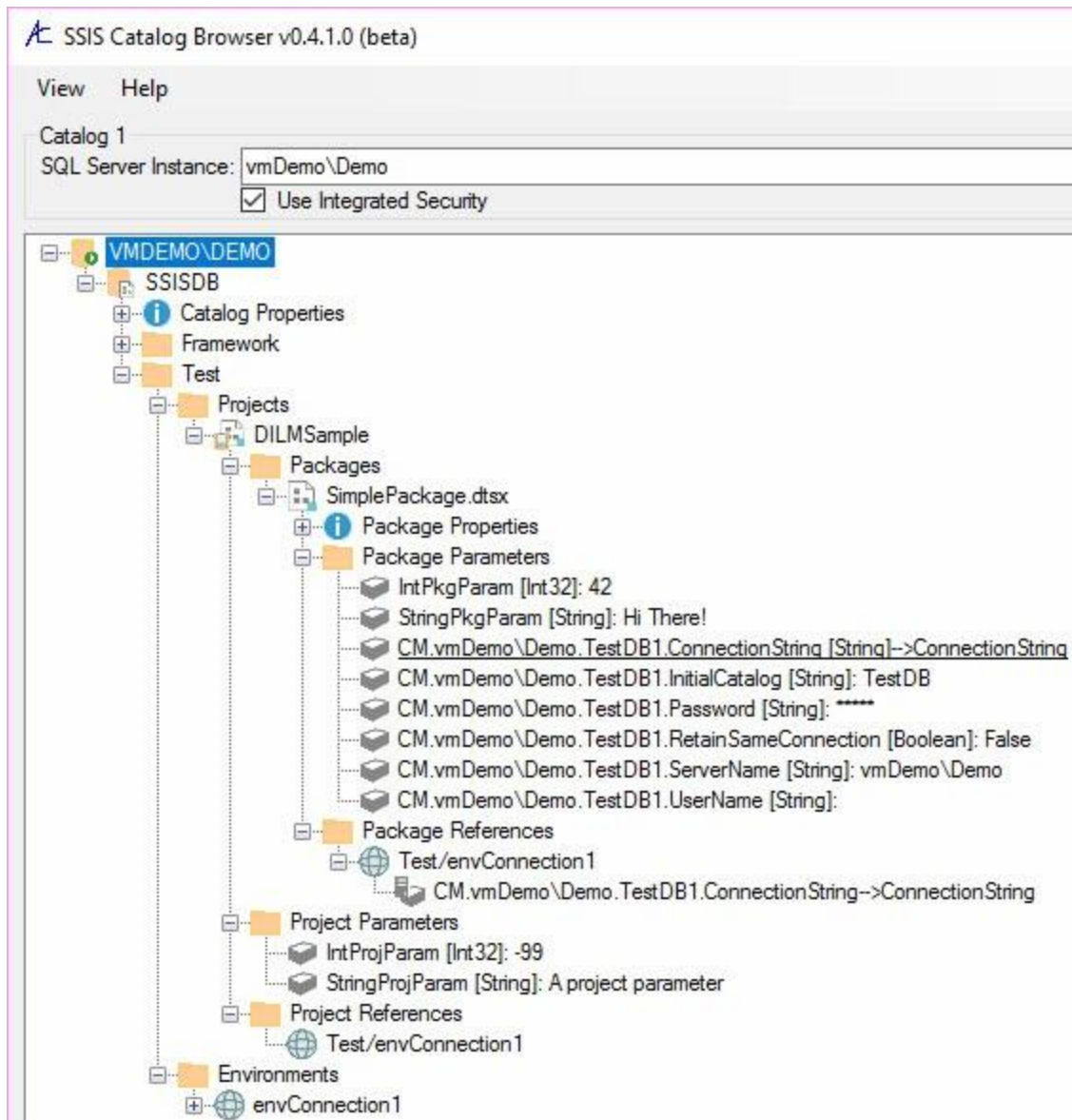


**Figure 8. Viewing SSIS Package Parameters**

Please recall Connection Manager Properties are treated as Parameters in the SSIS Catalog. They are prefixed with “CM.”. We see the SSIS Package Connection Manager vmDemo\Demo.TestDB1 Connection String property is mapped to an SSIS Catalog Environment Variable named ConnectionString.

To surface the Reference used for the Reference Mapping, expand the Package References virtual folder as shown in Figure 9:

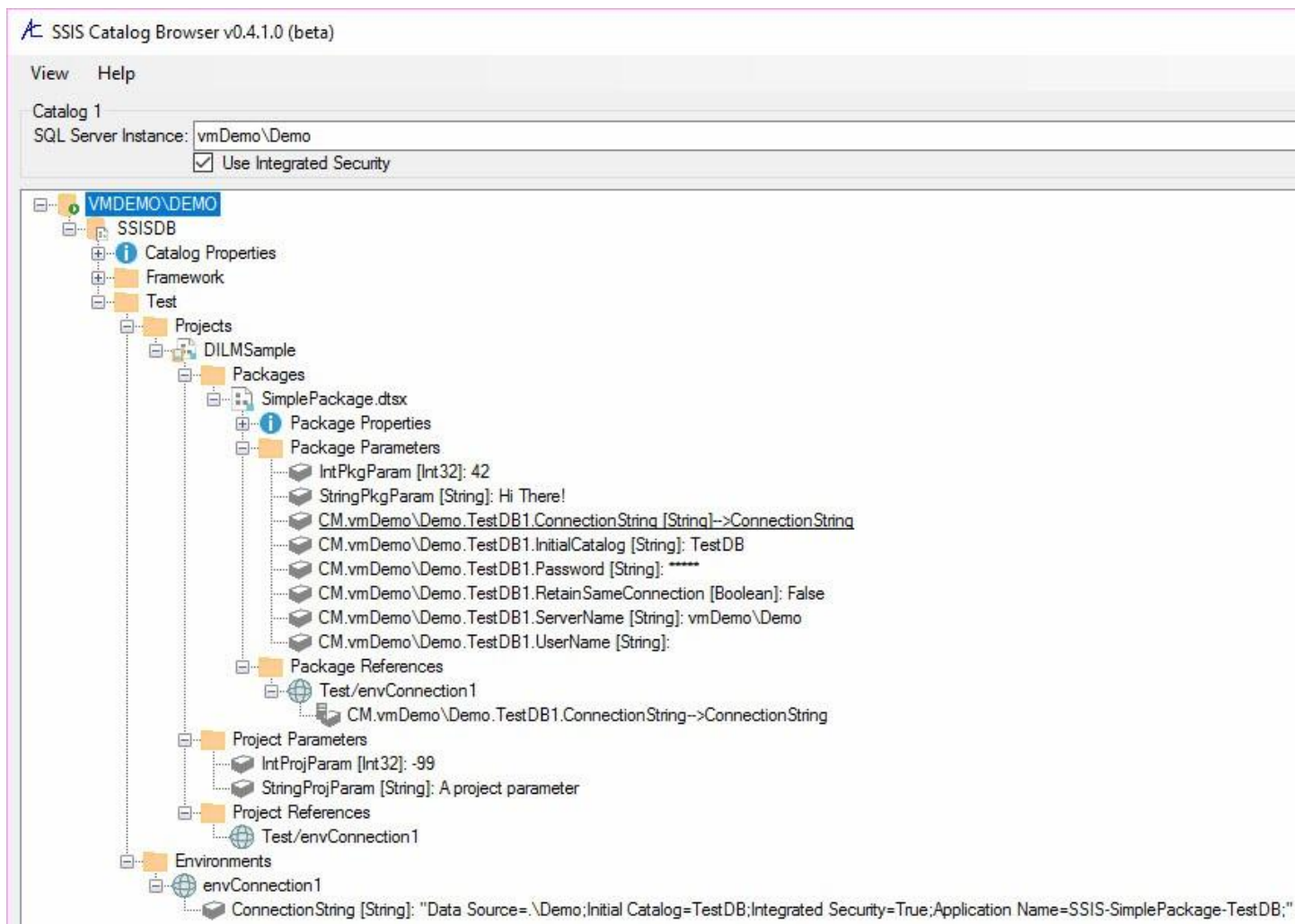




**Figure 9. Viewing the Package Reference**

Expanding the Package Reference virtual folder surfaces the Test/envConection1 Catalog Environment. Expanding the Test/envConection1 Catalog Environment reveals the Catalog Environment Variable named ConnectionString is mapped to the vmDemo\Demo.TestDB1 Connection String property.

But what's the value of the ConnectionString Catalog Environment Variable? Expand the envConection1 Catalog Environment in the Environments virtual Folder to view the collection of Catalog Environment Variables, their data types, and their values as shown in Figure 10:

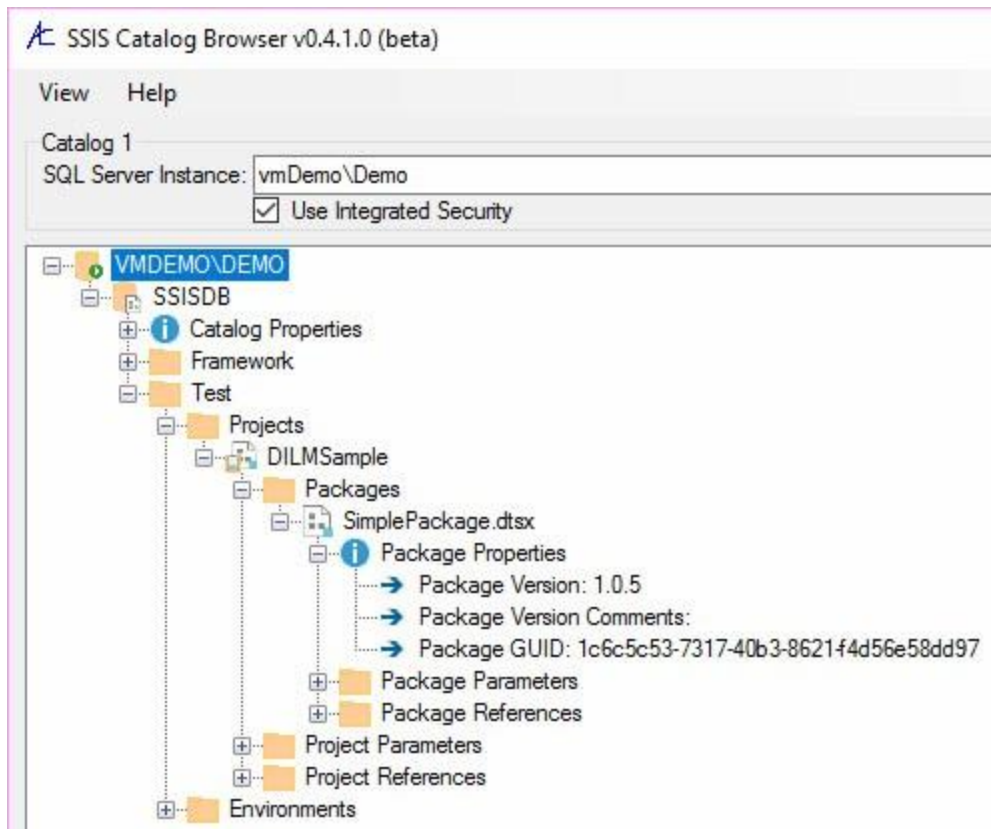


**Figure 10. Catalog Environment Variables, Data Types, and Values**

SSIS Package Properties includes a Package Version property constructed from the Version Major, Version Minor, and Version Build properties of the SSIS package. Every time a developer saves an SSIS package, the Version Build property increments. It's possible to revise an SSIS package and "trick" the Version Build property by manually setting it. I have not yet found a valid use case for doing so to SSIS Catalog-deployed SSIS packages.

The Package Version property can be used to detect different versions of SSIS packages deployed to an SSIS Catalog. Because SSIS developers can manually set the Version Build property, Package Version is not a reliable indication.

The Package Properties virtual folder surfaces SSIS Package metadata as shown in Figure 11:

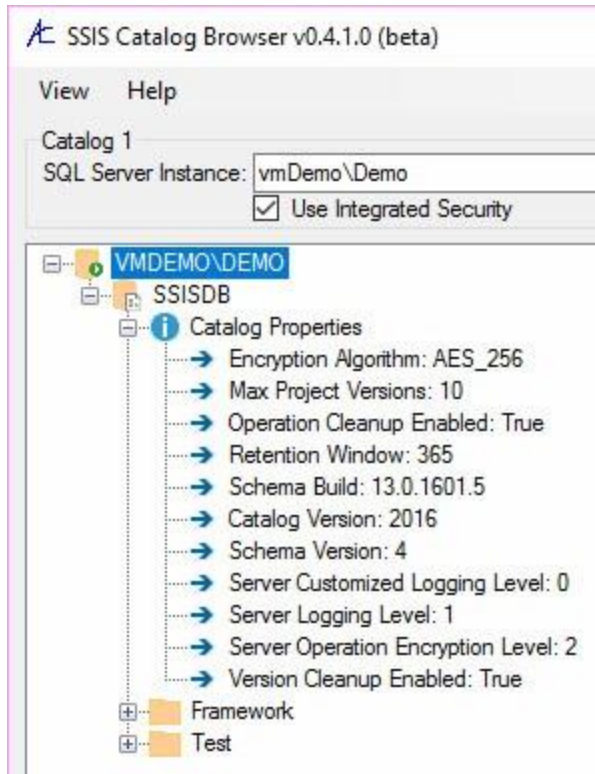


**Figure 11. SSIS Package Properties**

Catalog Properties are handy for detecting differences in patch levels (via the Schema Build property). Catalog Version is a property exposed by Catalog Base – the custom Catalog object that lies beneath Catalog Browser.

Catalog Base works with SSIS 2012, 2014, 2016, 2017, and Azure Data Factory version 2 Integration Runtime (“SSIS in the Cloud”) Catalogs.

The Catalog Properties virtual folder surfaces SSIS Catalog metadata as shown in Figure 12:



**Figure 12. SSIS Catalog Properties**

## Conclusion

Catalog Browser surfaces SSIS Catalog artifacts, configurations metadata, and artifact properties in a single view.