OmniStudio’s Service Management layer includes data services that read, write, transform, calculate, and track data within and outside of Salesforce. This layer includes processes and tools for users to look up data and configure complex math on top of the Salesforce Platform: OmniStudio Matrices and OmniStudio Calculation Procedures.

But before getting into Calculation Matrices, it’s important you understand what OmniStudio Calculation Procedures are. Calculation Procedures allow complex math to be configured within OmniStudio. Sometimes you need more than a simple mathematical equation, and configuring Calculation Procedures enables you to run just this kind of comprehensive and detailed set of mathematical processing steps.

To handle many factors, values, and calculations, a Calculation Procedure can call up one or more Calculation Matrices. These are simply lookup tables that take a unique input or set of inputs and return an output or set of outputs. Use a Calculation Matrix whenever you need to look up data.

You’ll dig into Calculation Procedures a bit more later. In the meantime, here are a few examples of Calculation Matrices.

* **An insurance premium matrix** that matches characteristics of the insured to policy premiums
* **A weather alert matrix** that sets an alert flag based on weather conditions
* **A demographic matrix** that outputs demographic data based on a location
* **A pricing matrix** that matches a set of product characteristics to a price or prices

A Calculation Matrix adds to the functionality of OmniScripts and OmniStudio Integration Procedures. As you know if you’ve completed other OmniStudio modules, an [OmniScript](https://trailhead.salesforce.com/content/learn/modules/omnistudio-omniscript) is a tool to design and create guided business processes for users, and [OmniStudio Integration Procedures](https://trailhead.salesforce.com/content/learn/modules/omnistudio-integration-procedures) is a tool that retrieves, saves, and manipulates data behind the scenes. You can call a Calculation Matrix from either tool using the Matrix Action.

There are three types of Calculation Matrices, and their uses vary.

* **Standard Calculation Matrix:** Used when your matrix doesn't need to be a part of a group or versioned row by row.
* **Grouped Calculation Matrix:** Used to group similar matrices with the same Input and Output headers together.
* **Row-Versioned Calculation Matrix:** Used if you have a matrix with many rows and may need to change small portions of the data at a time. Each row has its own start date and time

Here are a few things to keep in mind about Calculation Matrices.

* A matrix will only run if the current date and time is between the Start Date/Time and End Date/Time. A matrix with a blank End Date/Time still runs; it is merely set up to run indefinitely.
* If there are multiple versions of a matrix, only one version can be accessed at a time, based on these factors:
  + The date/time range on the matrices. (That is, are they active at the same time?)
  + If they are both active, the highest priority active matrix is the one that runs. (1 is the lowest priority.)
* The previous versions of a matrix will always be available to allow products with old prices to still access the proper price tables.
* If there is a lot of data, matrices can represent ranges. This keeps them small!

OmniStudio Calculation Procedures allow you to perform multiple mathematical operations and transformations at the same time. They take inputs as formatted JSON data and use lookup matrices, algebraic operations, and aggregation operations to calculate new data. They output specified data in formatted JSON and they can have conditional steps.

Calculation Procedures are used to configure matrix lookups, set variables and constants, and perform basic math operations.

Several tools in OmniStudio can perform mathematical operations, including OmniStudio DataRaptor formulas, OmniScript Formula elements, and aggregate functions. So, why should you use Calculation Procedures? Here are some of the advantages:

* Calculation Matrix integration
* Built-in JSON array calculations and aggregation functions
* Server-side execution
* Time-based execution (You can have multiple versions in place that automatically execute based on when the data call is made.)

Calculation Procedures can help guide a customer or agent through a series of questions and pull data from various sources. You can use Calculation Procedures to deliver quotes or determine eligibility for things such as the following.

* An individual health insurance plan to a customer via a website
* A term-life quote via an agent
* A volume discount on an order
* A permit or application fee
* A person applying for benefits or services

When there are a number of factors, values, and calculations, a Calculation Procedure can call one or more Calculation Matrices.

**How to Read a Calculation Procedure**

When reading a Calculation Procedure, it’s important to know its five basic components, outlined here.

Required:

* **Variables and Constants:** Used in calculation steps

Optional:

* **Calculation Steps:** Matrix and Lookup
* **Aggregation Steps:** Allow you to use data from an array input where the calculations are run on each element separately
* **Preprocessor Class:** Apex classes used to manipulate the input of the calculation
* **Postprocessor Class:** Apex classes used to manipulate the output of the calculation

In the calculation step, you can see these components.

1. **Variables in Procedure:** Annual Income and CLV are variables defined in the procedure itself.
2. **Constants in Procedure:** One and the Discounted Rate are constants defined in the procedure itself.
3. **Variables from Lookup:** Retention Cost and Retention Rate are procedure variables from the matrix you created.

## Built-In Simulation

Calculation Procedures have a built-in simulator that takes manual inputs and runs the procedure step by step, showing the results of each step. More importantly, the simulator generates a sample input and output JSON that can be easily used to build DataRaptors. It looks a little something like the following.

Keep in mind versioning for Calculation Procedures is the same as it is for Calculation Matrices:

* A Calculation Procedure version is valid between its start time and end time.
* You can have multiple versions, and they automatically roll over.
* If there are multiple versions that are valid, the highest priority runs. (1 is the lowest priority.)

As you can see, Calculation Matrices and Calculation Procedures work seamlessly together! Now it’s your turn to put them to good use and make complex math simple to run.