Oracle Configurator Quick Guide

Through the years of working with Oracle Configurator, I've become used to the misconceptions about the Configurator product and the misconceptions about the people who work with Configurator. For anyone thinking about beginning an Oracle Configurator project, thinking about engaging a consultant in working with Oracle Configurator, or just curious about the product itself, read on for my top product and consulting tips.

What is Oracle Configurator?

Oracle Configurator is a frequently misunderstood product. Perhaps because the concept of a Configurator is so fundamental to some implementations, Oracle Configurator is frequently assumed to be the main application for executing an order which uses a configured item. It is not. As is the case with non-configured items, Oracle provides a number of applications, such as Order Management, iStore, Quoting, etc., to manage the ordering of items. Whether the item to be ordered is configured or non-configured, these applications will still be used.

Oracle Configurator is a tool that augments each of these ordering applications at runtime (during the processing of an order) by taking a raw configurable item and helping the end user to configure it.

The behind the scenes story is similar to the end user experience. Oracle Configurator is again merely an adjunct to Inventory and Bills Of Materials. As with a non-configurable item, INV and BOM manage the parts and their structure with the only difference being the inclusion of Optional items in the BOM. Once the setup of items is complete in INV and BOM, they can be used in Oracle Configurator Developer ("OCD") and this is where the Configurator modeling would take place.

Therefore, the Oracle Configurator product is an adjunct product to Inventory and Bills Of Materials for model setup and to Order Management, iStore, Quoting, etc. for runtime.

Configurator Expertise

It is often mistakenly assumed that Configurator expertise is Configurator expertise – that knowing the product is a single skillset and anyone who knows any part of the product knows the whole product. In truth, expertise for Oracle Configurator spans three very different concepts that each requires a different sort of knowledge and technical skillset.

(Note that none of the following descriptions include the typical advanced Oracle Applications knowledge such as Forms or PL/SQL – another reason

why the Oracle Configurator product is unique within the Oracle Applications umbrella.)

Configure To Order Expertise ("Config Expertise Type 1" / "CET1")

As is the case with an Oracle implementation for non-configurable items, an implementation with configurable items should still involve Inventory and BOM expertise. However for configurable items, knowledge of how to architect and manage configurable BOMs is paramount to success. Skipping or skimping on the architecture for a BOM will lead to failure in Configurator. In other words, if the BOM foundation is unsound, you can't build anything reliable upon it.

Once the configurable BOM has been built, it can be "imported" into Configurator. The importing process copies the information from the BOM into Configurator so that the Configurator tools can work with the BOM data. Think of this step as activating the optional use of the Configurator product. (Because even for BOMs with Optional components, use of Configurator product is not necessary.)

A Config Expertise Type 1 resource, therefore, is the person who has the complete vision for the flow of items in your E-Business system. They are the person who needs to be involved first and to oversee all the touch points, including Configurator.

Oracle Configurator Expertise ("Config Expertise Type 2" / "CET2")

With the BOM imported into Configurator, the expertise shifts from that of architecting the BOM and the various optional pieces to using those pieces in the Configurator model. The BOM has provided the raw elements that make up the purchased part and it is up to the Configurator modeler to use these various parts to build a solid Configurator model. This model will include additional model selections (ones more user-friendly than the raw BOM components, for example), the relationships between the BOM elements and the model selections, and the user interface for presenting the Configurator model to the end user.

Skills required for a CET2 resource are knowledge of basic logic modeling skills, an ability to think in terms of the same logic as the Oracle product, and the ability to design a UI that would be easy for the target audience to use. Although some degree of understanding of how the items will be used throughout the rest of the E-Business system is helpful, the focus here is on how the items interact with each other and with the end user.

Configurator Extension Expertise ("Config Expertise Type 3" / "CET3")

Even with all that is available within Configurator, most customers find specific configuration needs that cannot be met by any commercial product out of the box. These needs are frequently related to the specifics of the business or may be connections to legacy systems or advanced calculations. Or even customizations to the user interface experience for the end user. Oracle Configurator is extendable through the use of what is known as Configurator Extensions (CXs, formerly known as Functional Companions).

CXs are generally written in Java and since Java can serve as a bridge to many other technologies, CXs are not limited to the Java world. CXs can merely send the results of a configuration to another system or they can affect the configuration session itself.

In addition to Java programming (as well as any additional technologies that the CX would be expected to work with), a CET3 resource would also need to be cover the range of CET2 to work with the primary part of the CZ API, the Configurator Interface Object (CIO). Knowledge of the CZ subschema would be helpful too since frequently Configurator CX involve some level of database work.

Combinations

Of course, the delineations are rarely as absolute as described above and you frequently find consultants that can bridge two worlds. I consider myself to be CET2 & CET3 and I know many good consultants that are CET1 and CET2. (CET1 & CET3 skills are rarely found in the same individual due to the disparate nature of the required skillsets.)

Validator vs. Configurator

As the name implies, Oracle Configurator is a configurator, not a validator. A validator is more passive than a configurator. A validator lets the end user make selections without providing any feedback until the end when the user says they are done. That's when the validator would check the selections and provide feedback. Most people are familiar with validators from shopping on the Internet and most people have experienced the situation where the validator tells them there is something invalid and gotten frustrated trying to satisfy the conditions.

A configurator is more active than a validator. A configurator will actively make selections based on those the user makes. The goal is to keep the configuration in a valid state throughout the user interaction process. This helps the user complete the configuration quickly and avoid becoming trapped by the selections they've already made.

Take, for example, making selections to build a car and a requirement that the automatic transmission go with a 6-cylinder engine. With a validator, if the user selects the automatic transmission, they would be able to select a 4-cylinder engine and then be told later that the selections are invalid. But with a configurator, the act of selecting the automatic transmission causes the 6-cylinder engine to be selected. Then the user can see the consequences of that selection and the configurator can further propagate from that selection.

And this is where the Oracle Configurator expertise (CET2 described above) is important. The Oracle Configurator model needs to be constructed such that there are no contradictions and so that the set of requirements is distilled to the most basic providing the end user with a clean experience.

(Note that Oracle Configurator can function as a validator when being used in batch mode. Configurator will load and check that there are no logic contradictions during the later stages of order processing to confirm that the configuration previously created by the user is still valid. Since there is no interactive aspect to this process, a validator is all that is required.)

The Logic Net

There are three parts to Configurator modeling: Structure, Rules, and UI. The first two parts are combined to collectively build the Logic Net: the network of nodes in the structure and the rules that relate them. Whenever a change is made to either the Structure or the Rules in Config Developer, the modeler must click the button to "Generate Logic" before testing. This button is *compiling* the set of nodes and rules in the Logic Net and storing the result in the database. When the model is run, the appropriate Logic Net is loaded into memory allowing the quick interaction for the end user. It's a lot faster to load and interact with the Logic Net than it would be if the Configurator needed to find all the appropriate nodes in the database each time the Configurator was run.

When using Component Sets (Components that have other than a static quantity of 1), the various Component Instances are added to the Logic Net on the fly at run time.

Fusion Configuration Engine

As users of Configurator know all too well, the Configurator engine, the core part of the Configurator that handles logic propagation, has some limitations. In particular, the notion of Defaulting was always an afterthought and it acts like one. To this day, most people know that Default rules can add unexpected behavior to the model (a reason for a CET2 resource to be on the case). Also, there are some things that you can't do in logic that many customers wish that they could like creating instances of a component set through a rule consequences.

Years ago, work began on a new Configurator engine to solve these problems as well as others. For one reason or another (Siebel acquisition, for example), the new engine was delayed. But as of R12.1, the new engine is now released. Since this new engine is the Configurator engine that will be used in the forthcoming "Fusion" product from Oracle (which incorporates all of their various recent E-Business acquisitions), the name Fusion Configuration Engine (FCE) is used.

Since the FCE is markedly different in a number of ways, use of the FCE is optional. In fact, in R12.1, new models are created with the original Engine by default. Any model can be converted to a FCE model, however, to take advantage of the new functionality.