Jackson
LUMBER & MILLWORK

Building a Modern Rules-Based Product Configurator Using Synergy .NET

SYNERGY DevPARTNER Conference 2026

Ben Newton, Sr. Programmer
Nick Sarno, Jr. Programmer
October 2026



WHO ARE WE?

ESTABLISHED 1946

Established in 1946 by the Torrisi family and is currently run by the third generation of Torrisis.

BUILDING MATERIALS

We provide lumber, building materials, windows, doors, and hardware to contractors, builders, and homeowners throughout Massachusetts and Southern New Hampshire.

MANUFACTURING

We manufacture custom interior and exterior doors, pre-hung door systems, moldings, and specialty millwork at our state-of-the-art facility in Raymond, NH.



Jackson
LUMBER & MILLWORK

DON PICCIRILLO

- Original architect of Jackson Lumber & Millwork's core business systems.
- Began developing custom software for Jackson in 1983, laying the foundation for the systems still used today.
- Served as the sole developer until 2011, overseeing nearly three decades of system development and maintenance.
- Led our evolution from legacy DIBOL all the way through to modern Synergy .NET, ensuring continued functionality and adaptability.
- Retired at the end of 2025, leaving a legacy of more than four decades of innovation, expertise, and lasting contributions to Jackson's operations.



Jackson
LUMBER & MILLWORK

OUR HISTORY WITH SYNERGEX

CUSTOMERS SINCE ~1983

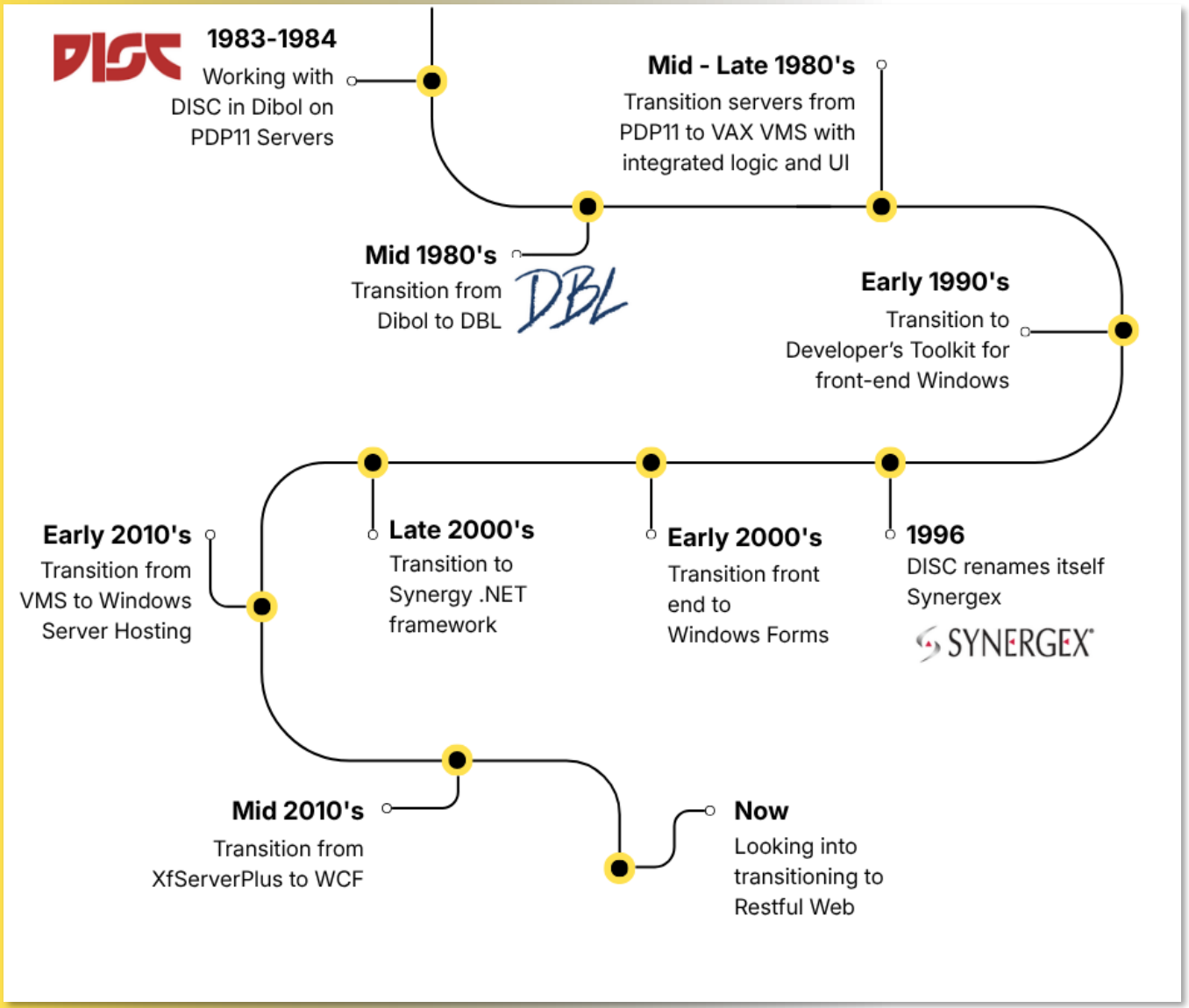
We started working with Bill Mooney when Synergex was still called DISC. At the time, we were working in Dibol on PDP11 servers before transitioning to DBL shortly after.

UPGRADES EVERY ~10 YEARS

Ever since then, we've kept our systems up to date with Synergy's regular upgrades with fully retained databases and no system rewrites. We have data going as far back as 1996 and would have had more if not for file purges to save disk space.

NEXT UP: RESTFUL WEB SERVICE

We're currently looking forward to our next big upgrade to Restful Web Service.



OUR SYSTEM

- **Fully integrated ERP system**, designed, developed, and maintained entirely in-house.
- **Built using C# and Synergy DBL**, providing a flexible foundation for ongoing development.
- Features user interfaces developed in **Windows Forms and XAML-based** environments.
- **Supports both Windows and Android** applications, extending functionality across desktop and mobile devices.



Jackson
LUMBER & MILLWORK

WHAT IS A BOM SYSTEM?

- **A detailed list** of all materials, parts, and components required to manufacture or assemble a product.
- **Identifies each component** by item number, description, and quantity.
- **Organizes materials** into assemblies and subassemblies.
- **Serves as a blueprint** for manufacturing and production.

BILL OF MATERIALS (BOM)

Traditional (Indented) Format

Product Name:	Office Chair
Product Number:	OC-1000
Revision:	A
Effective Date:	2024-01-01
Prepared By:	Engineering
Units:	Each

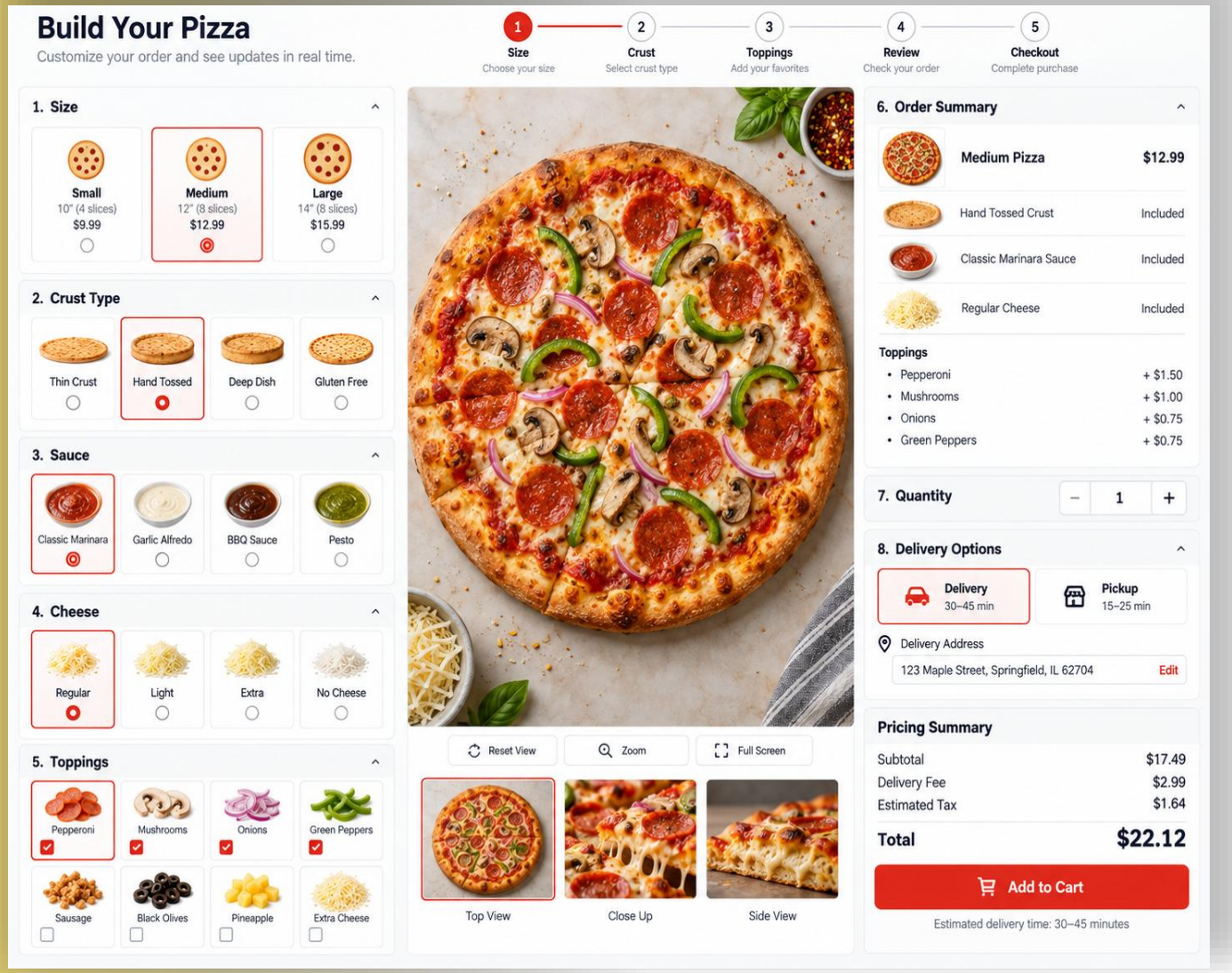


Level	Item Number	Description	Quantity per Assembly	Unit of Measure	Notes
0	OC-1000	Office Chair (Finished Product)	1	Each	
├ 1	CH-100	Chair Assembly	1	Each	
├ 2	BK-100	Back Assembly	1	Each	
├ 3	BK-110	Back Frame	1	Each	Injection Molded
├ 3	BK-120	Mesh Fabric	1	Each	Black
└ 3	BK-130	Back Screws	4	Each	M6 x 20
├ 2	SE-100	Seat Assembly	1	Each	
├ 3	SE-110	Seat Cushion	1	Each	Foam with fabric cover
├ 3	SE-120	Seat Pan	1	Each	Plastic, black
└ 3	SE-130	Seat Screws	4	Each	M6 x 16
├ 2	AR-100	Armrest Assembly (Left)	1	Each	
├ 3	AR-110	Armrest Pad	1	Each	PU
├ 3	AR-120	Armrest Bracket	1	Each	Plastic, black
└ 3	AR-130	Armrest Screws	2	Each	M6 x 20
├ 2	AR-200	Armrest Assembly (Right)	1	Each	
└ 2	BS-100	Base Assembly	1	Each	
├ 3	BS-110	Gas Lift Cylinder	1	Each	
├ 3	BS-120	Base (5-Star)	1	Each	Nylon, black
├ 3	BS-130	Caster Wheel	5	Each	Dual wheel, black
└ 3	BS-140	Base Screws	5	Each	M8 x 25

Jackson LUMBER & MILLWORK

WHAT IS A CONFIGURATOR?

- An interactive, rules-based system used to build and customize products.
- Guides users through available product options and selections.
- Automatically generates accurate Bills of Materials (BOMs).
- Calculates pricing dynamically based on selected options.



WHY CONFIGURATORS WIN

TRADITIONAL B.O.M.

- Manually select parts & quantities
- Relies on product knowledge
- Greater potential for errors
- Changes require manual revisions
- Time consuming



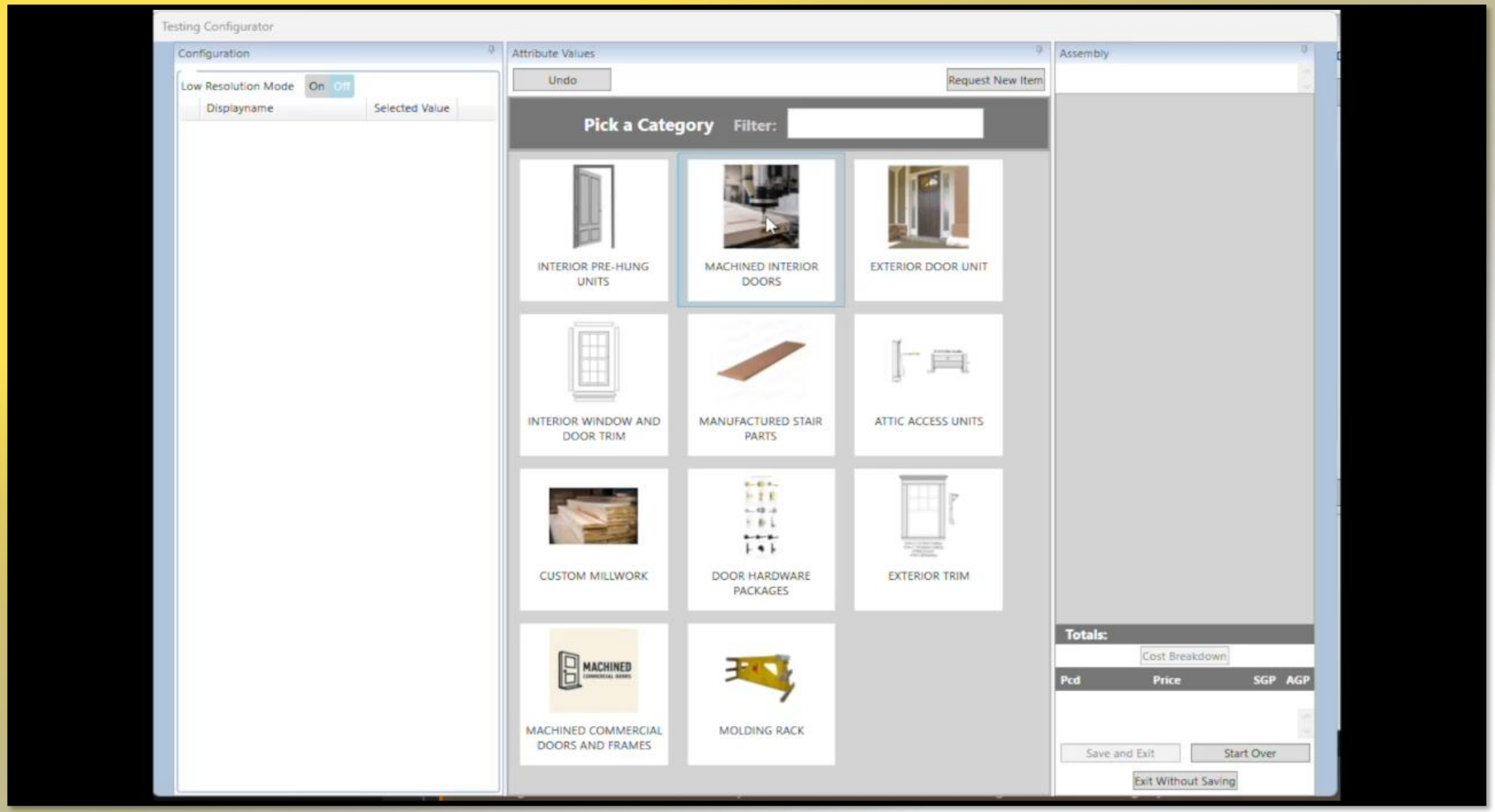
CONFIGURATOR

- Guided Product Selection
- Built In Product Knowledge
- Automatic Rule Validation
- Self Maintaining
- Faster, Consistent Results



Jackson
LUMBER & MILLWORK

INTERIOR DOOR CONFIGURATOR



HOW DOES IT WORK?

ITEM DRIVEN

Every component used in any manufactured product is typed and attributed. The configurator can use this to assist the user in determining which components will be used to manufacture the final product.

USER INPUT

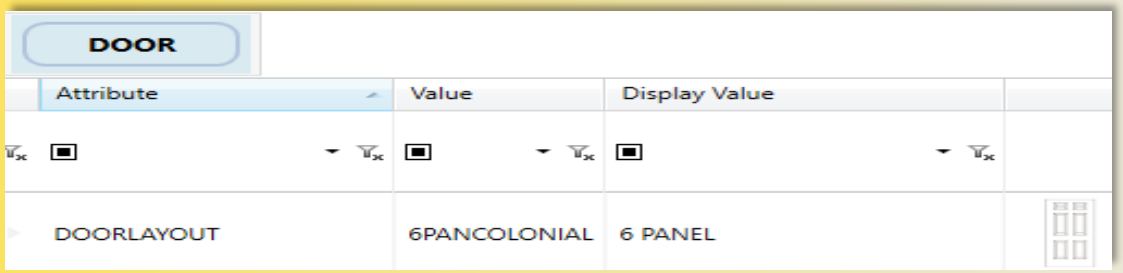
The system presents users with questions to define the properties of the manufactured product they wish to make.

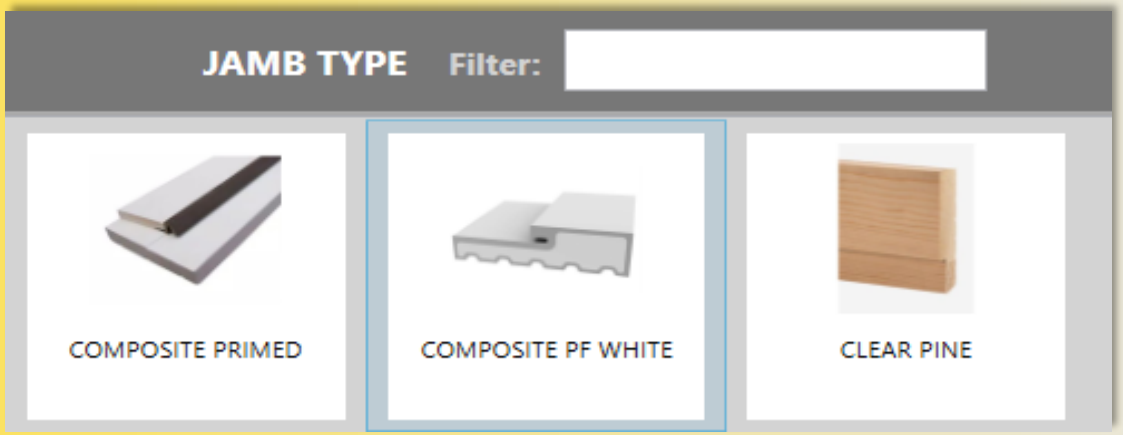
ADAPTIVE QUESTIONS

The system provides the users with curated lists of potential answers based on based on the properties that have already been determined.

PRODUCT RESOLUTION

Thanks to every component in our system being uniquely attributed, the user's answers will eventually resolve to a single manufactured item.





Displayname	Selected Value	Qty Per
458TTHDPRCOMP		1.00
30IG3		1.00
ARCHSILL		4.00

Jackson LUMBER & MILLWORK

LOGICAL PARSER

- The core of Jackson's configurator
- Solves logical statements and formulas written in a custom string-based language
- Logical statements include placeholder values which are filled in at runtime based on user input
- Allows our rules-based configurator to run dynamically with minimal hard-coded values

Formatting a Formula:

```
Comparison Operators:
Equal To:                  ==
Not Equal To:              !=
Greater Than:              >
Greater Than or Equal To: >=
Less Than:                 <
Less Than or Equal To:     <=

Mathematical Operators:
Addition:       +
Subtraction:    -
Multiplication: *
Division:       /

Logical Operators:
AND
OR

Attributes:
[ATTRIBUTENAME]

Text:
'text'

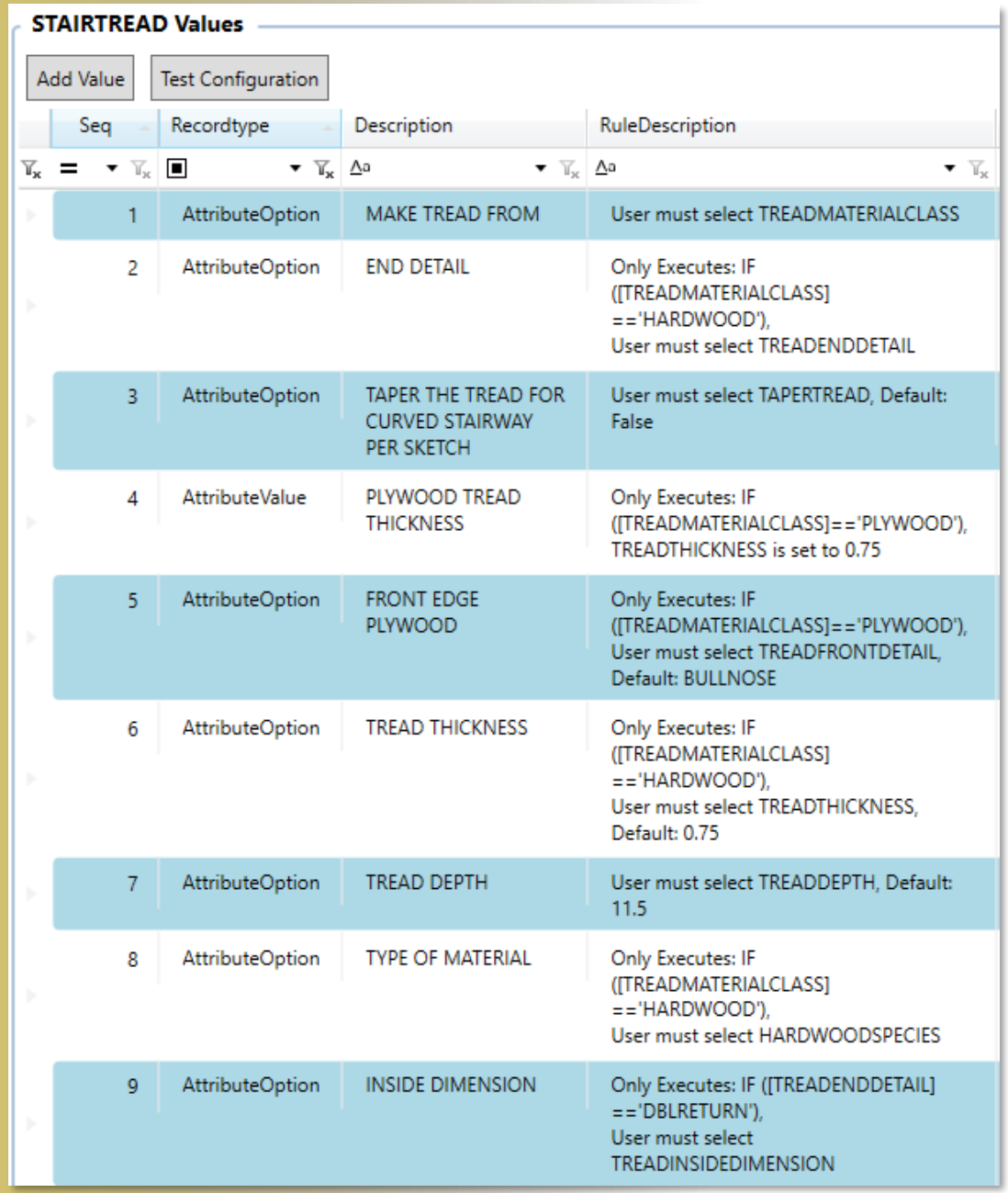
Special Operators:
Force Evaluation:    ^
                     Sample: ^[ATTRIBUTENAME] + 1.5^
                     if the formula is a stringbuilder, any block wrapped in this operator
                     will run as if it were in a value formula, using the Values not Display Values.
Force Actual Value: ~
                     Sample: ^[ATTRIBUTENAME]^
                     if the formula is a stringbuilder, any single attribute wrapped in this operator
                     will display it's Value rather than it's Display Value.

Operations:
Round up to X decimal places:    @ROUNDUP(4/5, 0)
                                          (number/equation to round, decimal places to round to)
Round Down to X decimal places: @ROUNDDOWN(4/5, 0)
                                            (number/equation to round, decimal places to round to)
Round to X decimal places:       @ROUND(4/5, 0)
                                       (number/equation to round, decimal places to round to)
Round to a fraction of a total: @ROUNDBY(38, 4, 96)
                                          (number/equation to round, fractional denominator, total)
Convert Inches to feet and inches: @FEETANDINCHES(18.5)
                                               (Inches to convert)
Convert Decimal to Fraction: @DECIMALTOFRACTION(18.5)
                                             (Decimal to convert)
Insert a line break in results: @Newline()
                                sample: 'words in first line@newline()words in second line'

Logical Statements
IF(a == b)
     Returns true or false based on the evalution of the contents of the IF statement.
IF(a == b){ 2 } ELSE { 1 }
     Returns 2 if the IF statement evaluates to true, otherwise will return 1
IF(a == b) { 2 } ELSEIF(c == d) { 3 } ELSE { 1 }
     Returns 2 if the IF statement evaluates to true, otherwise if the ELSEIF statement evaluates to
true, it will return 3, otherwise will return 1
```

WHAT ARE RULES?

- Rules control the final product configuration and guide users through the selection process.
- Logical formulas determine component properties, descriptions, values, and when rules execute.
- Rules ensure product compatibility, enforce manufacturing requirements, and prevent invalid configurations.
- Rules are driven by user-defined Types and Attributes.



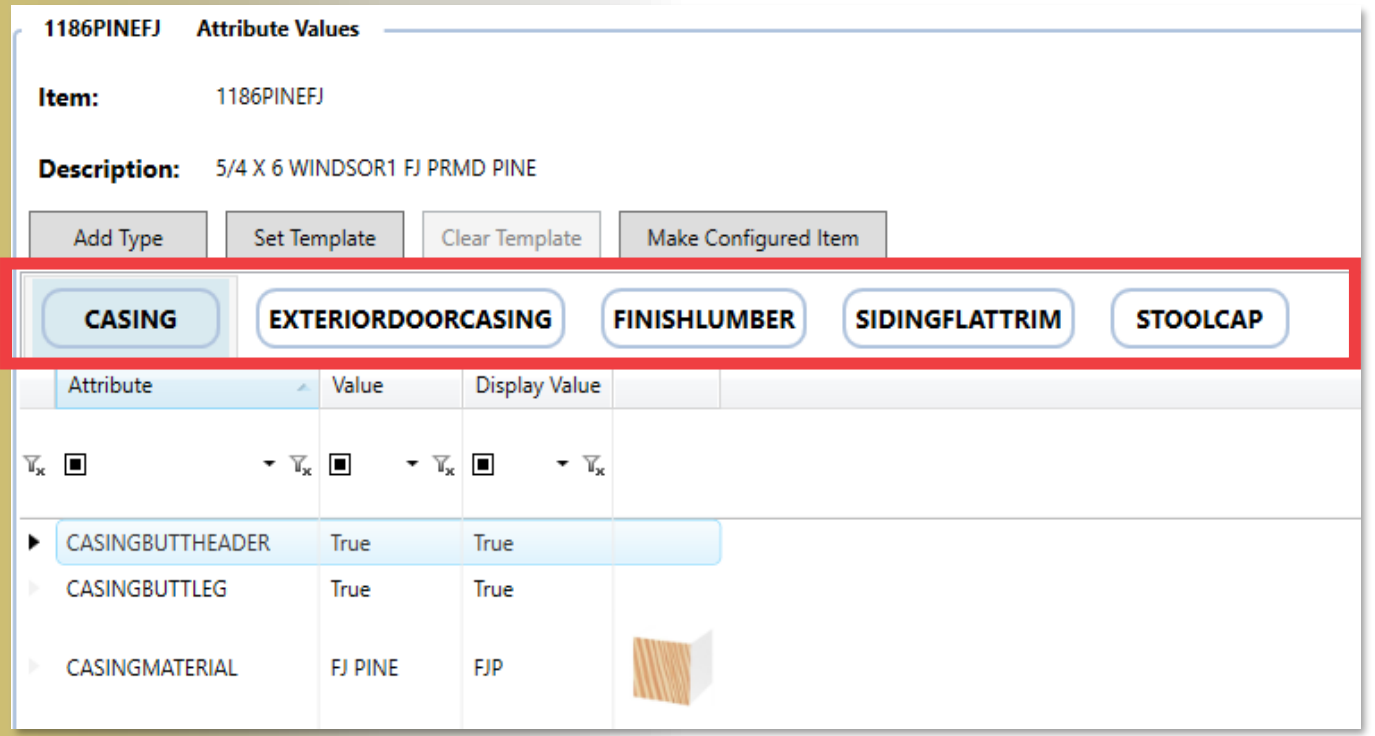
STAIRTREAD Values

Add Value | Test Configuration

Seq	Recordtype	Description	RuleDescription
1	AttributeOption	MAKE TREAD FROM	User must select TREADMATERIALCLASS
2	AttributeOption	END DETAIL	Only Executes: IF ([TREADMATERIALCLASS] =='HARDWOOD'), User must select TREADENDDETAIL
3	AttributeOption	TAPER THE TREAD FOR CURVED STAIRWAY PER SKETCH	User must select TAPERTREAD, Default: False
4	AttributeValue	PLYWOOD TREAD THICKNESS	Only Executes: IF ([TREADMATERIALCLASS]=='PLYWOOD'), TREADTHICKNESS is set to 0.75
5	AttributeOption	FRONT EDGE PLYWOOD	Only Executes: IF ([TREADMATERIALCLASS]=='PLYWOOD'), User must select TREADFRONTDETAIL, Default: BULLNOSE
6	AttributeOption	TREAD THICKNESS	Only Executes: IF ([TREADMATERIALCLASS] =='HARDWOOD'), User must select TREADTHICKNESS, Default: 0.75
7	AttributeOption	TREAD DEPTH	User must select TREADDEPTH, Default: 11.5
8	AttributeOption	TYPE OF MATERIAL	Only Executes: IF ([TREADMATERIALCLASS] =='HARDWOOD'), User must select HARDWOODSPECIES
9	AttributeOption	INSIDE DIMENSION	Only Executes: IF ([TREADENDDETAIL] =='DBLRETURN'), User must select TREADINSIDEDIMENSION

TYPES

- Defines the usage of an item.
- Have a many-to-many relationship with Items.
- A single item can serve multiple purposes, such as a pine board used for casing, trim, or finished lumber.
- Each Type has a unique set of Attributes that define its properties.

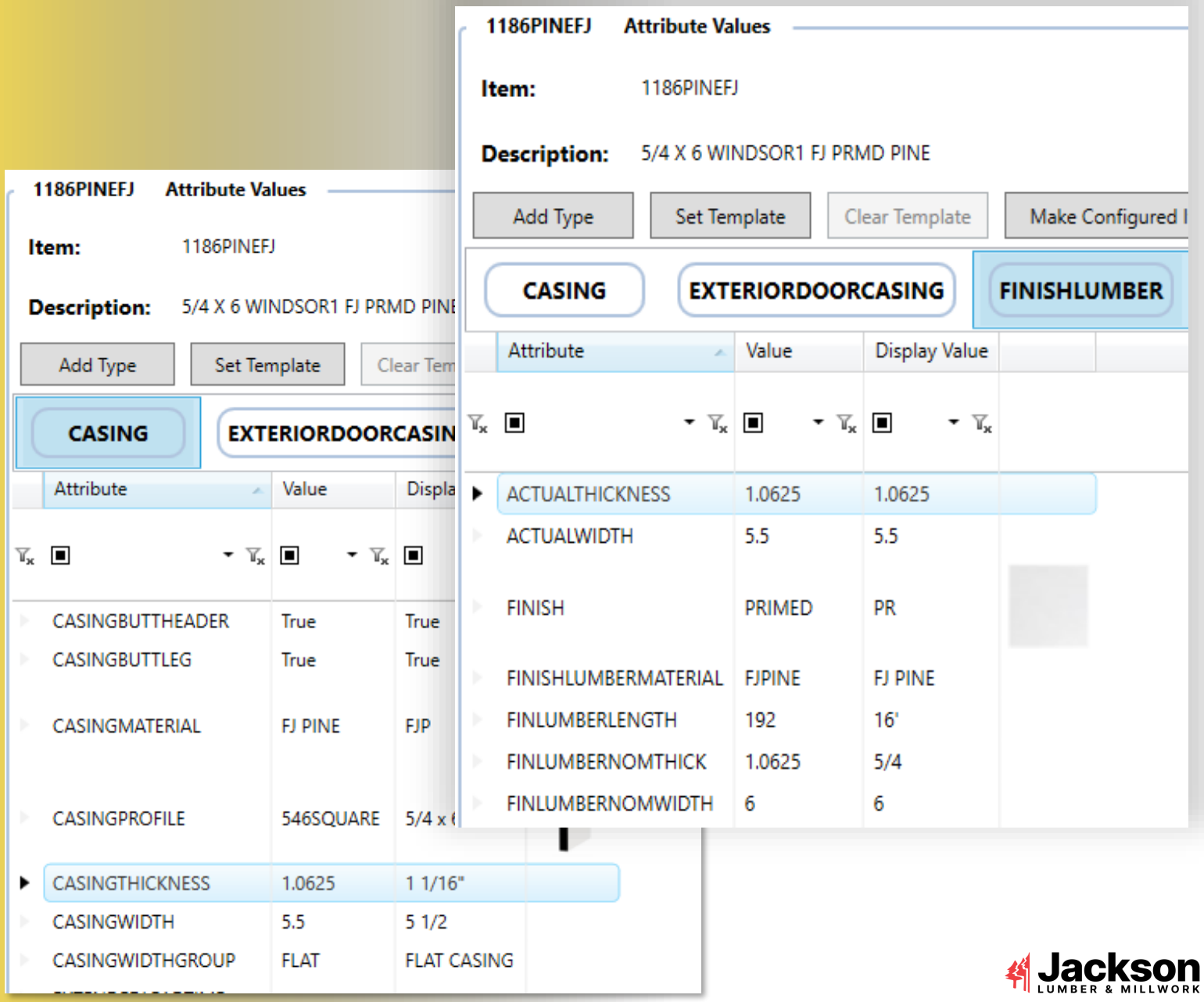




Jackson LUMBER & MILLWORK

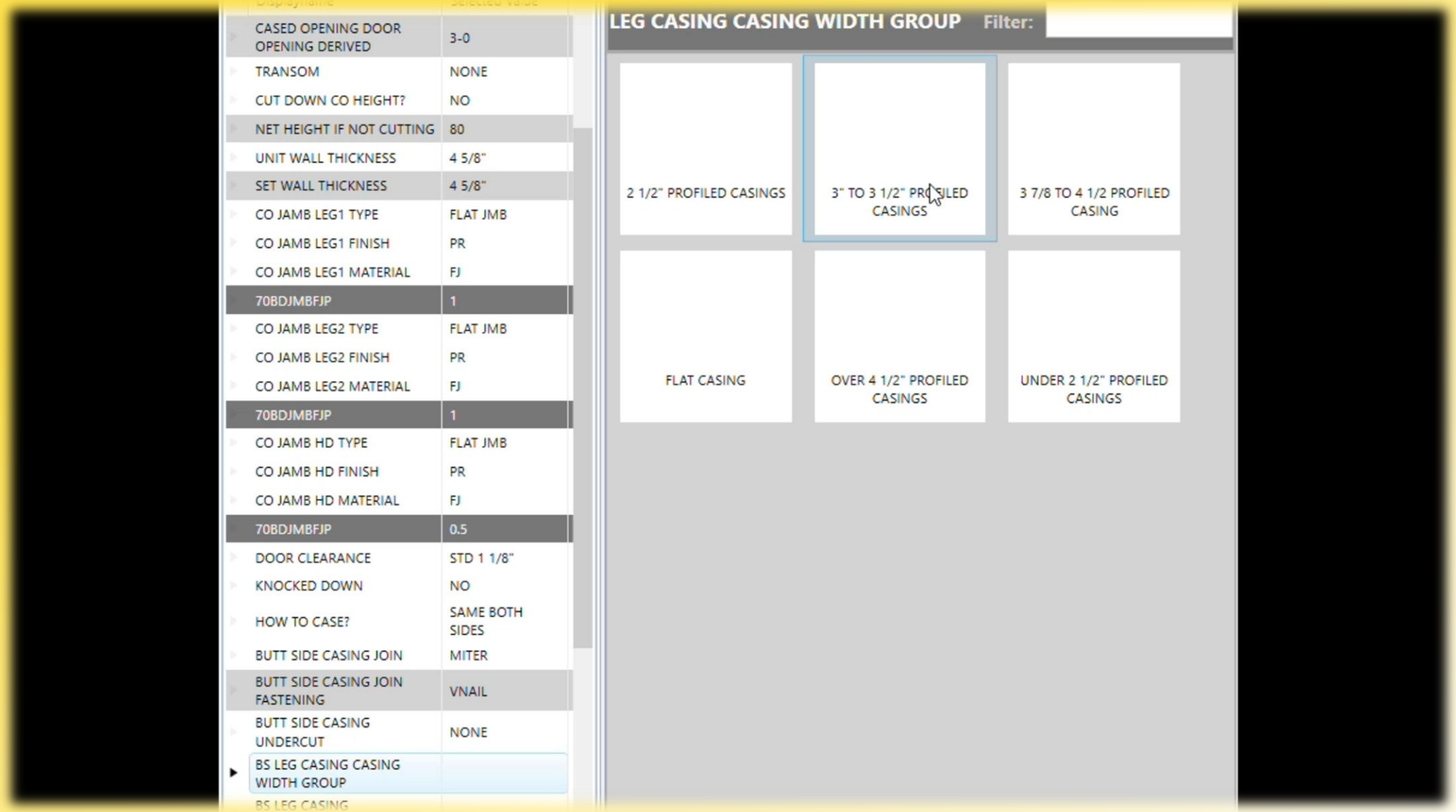
ATTRIBUTES

- Define the properties of a specific Type.
- Reflect manufacturing specifications (e.g., actual vs. nominal width).
- Distinguish items within the configurator.
- Accept predefined values or free-text input.



Jackson LUMBER & MILLWORK

USER-FRIENDLY INTERFACE



RUNNING RULES

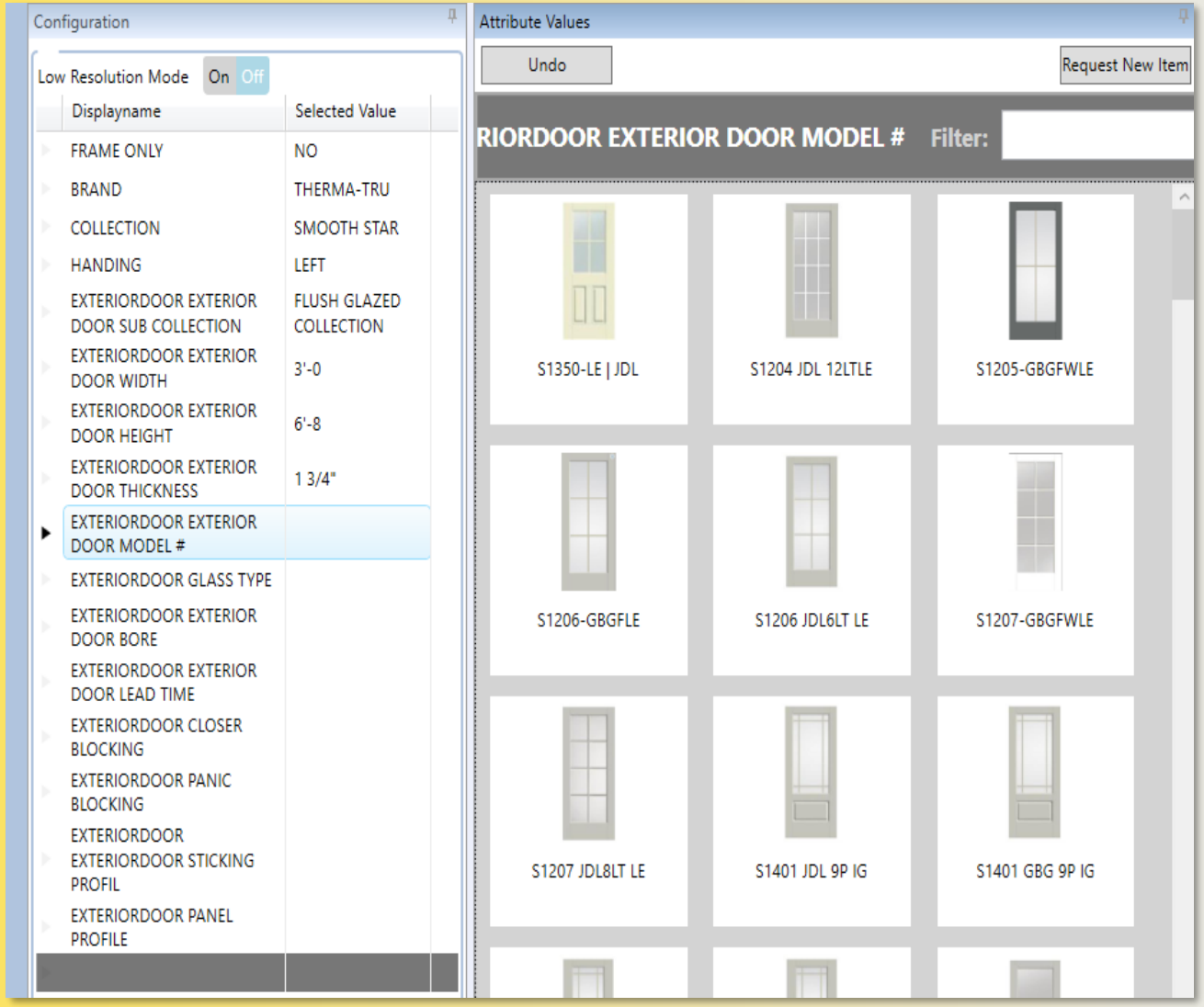
Rules execute sequentially

Each rule attempts to resolve using known values from the rules above it in sequence

Unresolved rules prompt the user for input

Users are presented with a curated list of possible answers to pick from

Once a rule is resolved, the next rule executes



Jackson LUMBER & MILLWORK

TYPES OF RULES

ATTRIBUTE OPTION

Allows users to select attribute values

AttributeOption	DOOR UNIT TYPE	User must select INTERIORDOORUNITTYPE

ATTRIBUTE VALUE

Sets a fixed, predetermined value

AttributeValue	LABEL THE DOOR	Only Executes: IF ([DOORCORE] =='20MINUTE'), LABELTHEDOOR is set to True

ATTRIBUTE RULE

Logical statements that set values

AttributeRule	SET WALL THICKNESS	Only Executes: IF ([INTDOORWALL]!=0)

LABOR

Assigns labor hours to appropriate cost centers

Labor	LABOR FOR ANGLE TOP DOOR	Only Executes: IF ([ANGLETOPDOOR] ==TRUE), Hours on Workcenter '4331': 3

MATERIAL RULE

Assigns specific material to be used

Material	LIFT HANDLES	Only Executes: IF ([DOORCORE] =='20MINUTE' AND [DOORHEIGHT]>80), Allocates 2 of LIFTHANDLE

MATERIAL SELECTION

Resolves components, prompts for unknown values

MaterialSelection	BIFOLD TRACK	Only Executes: IF ([INTERIORDOORUNITTYPE]=='BIFOLD')

MATERIAL SELECTION RULES

CONTINUOUS PROMPTING

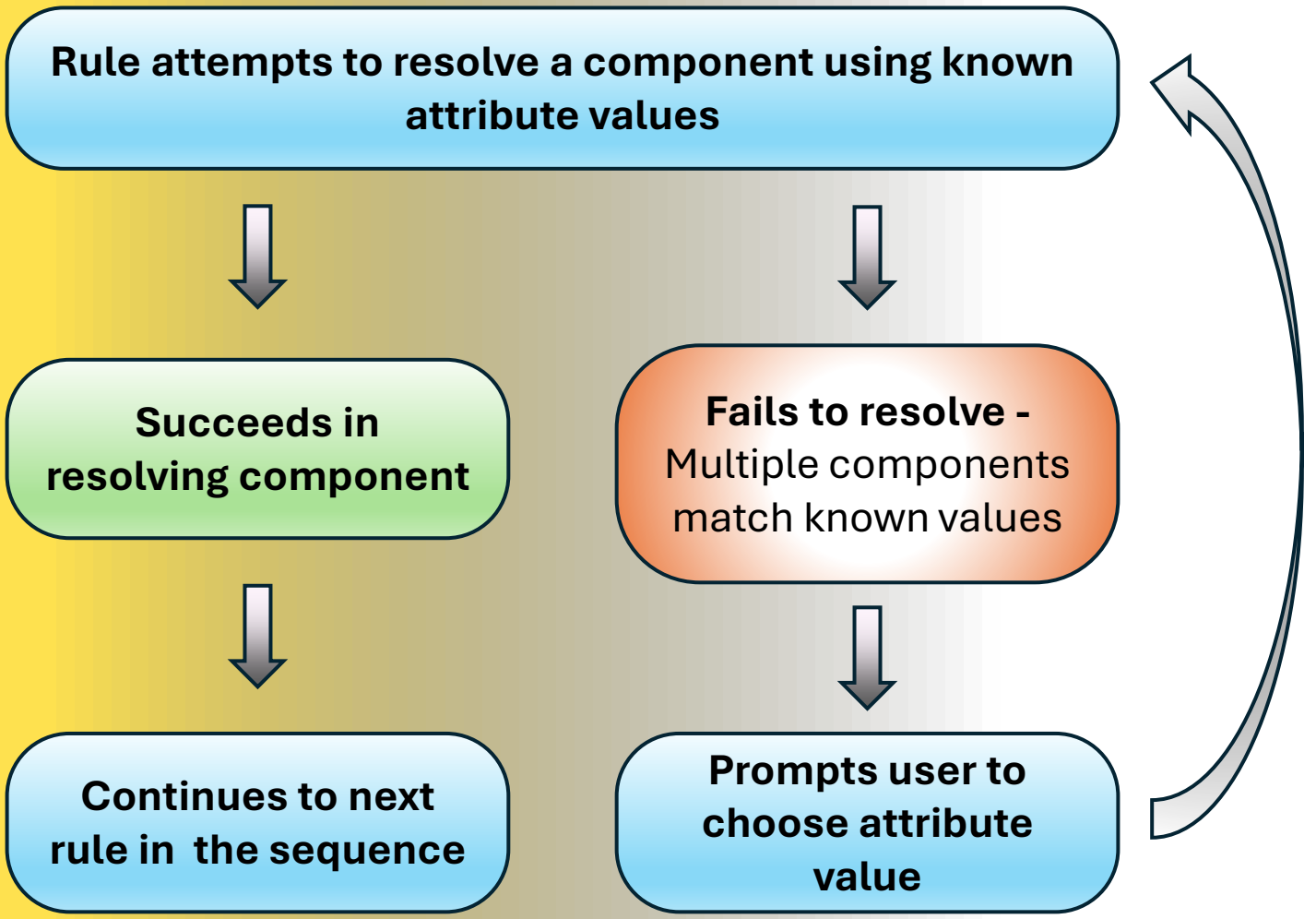
Automatically prompt users to fill in unknown values until an item can be resolved.

ALLOWS FLEXIBILITY

These rules allow our configurator to be extremely flexible and accommodate new attributes and attribute values with no changes.

MINIMAL MAINTENANCE

When a new attribute or attribute value is introduced to the system, it does not need to be manually included in any ruleset in order for the system to prompt the user for it.



Jackson LUMBER & MILLWORK

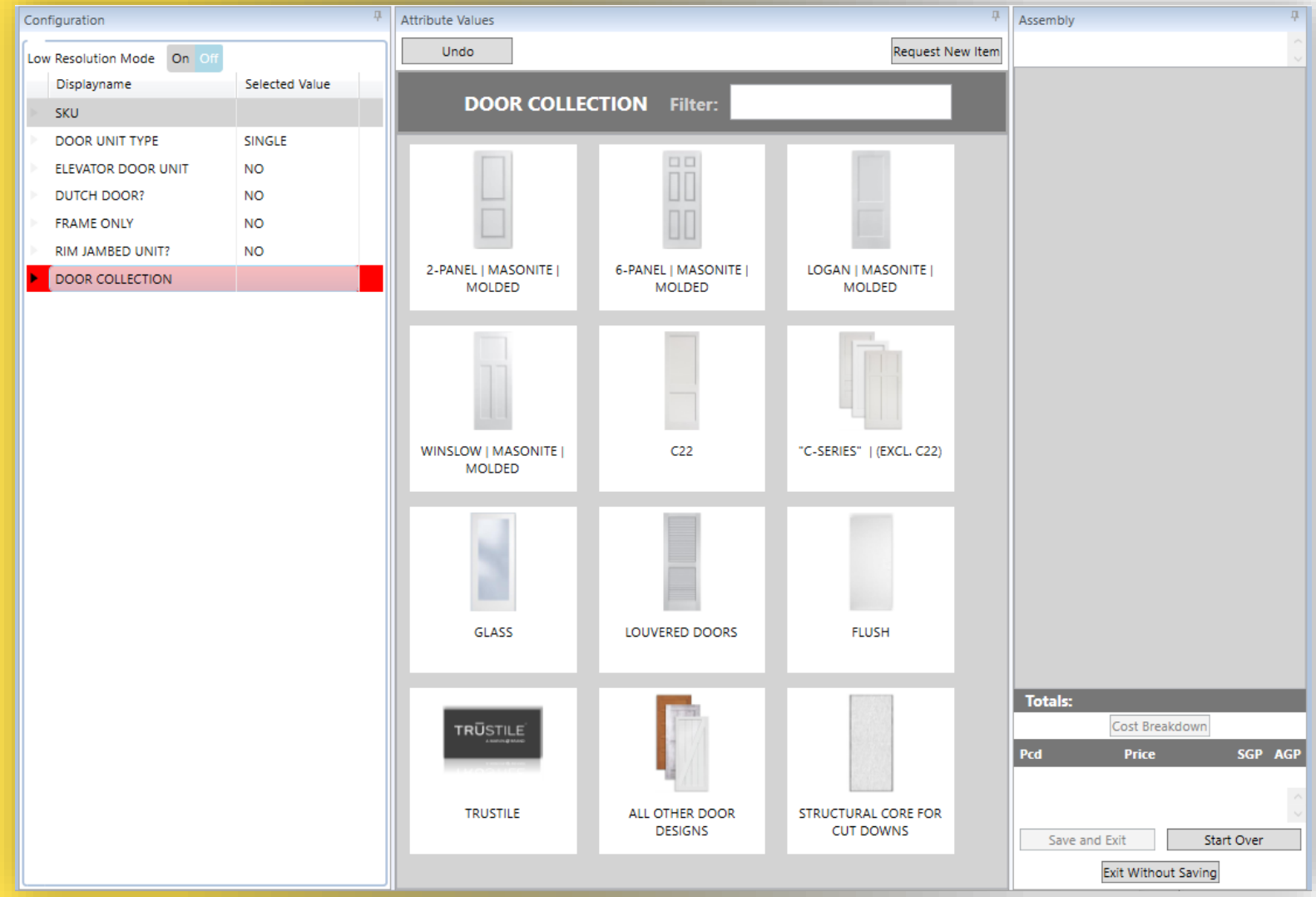
WHY NOT JUST USE MATERIAL SELECTIONS?

EFFICIENCY

While material selection rules are capable of walking a user all the way through the selection of a component, manually asking for answers to certain key attributes ahead of time can dramatically improve the speed with which the configurator runs.

CURATED QUESTIONS

There are 4800 different door slabs in our system. By asking the user a few simple questions ahead of time, we can reduce the number of items that the material selection has to consider to a fraction of that total.



Jackson LUMBER & MILLWORK

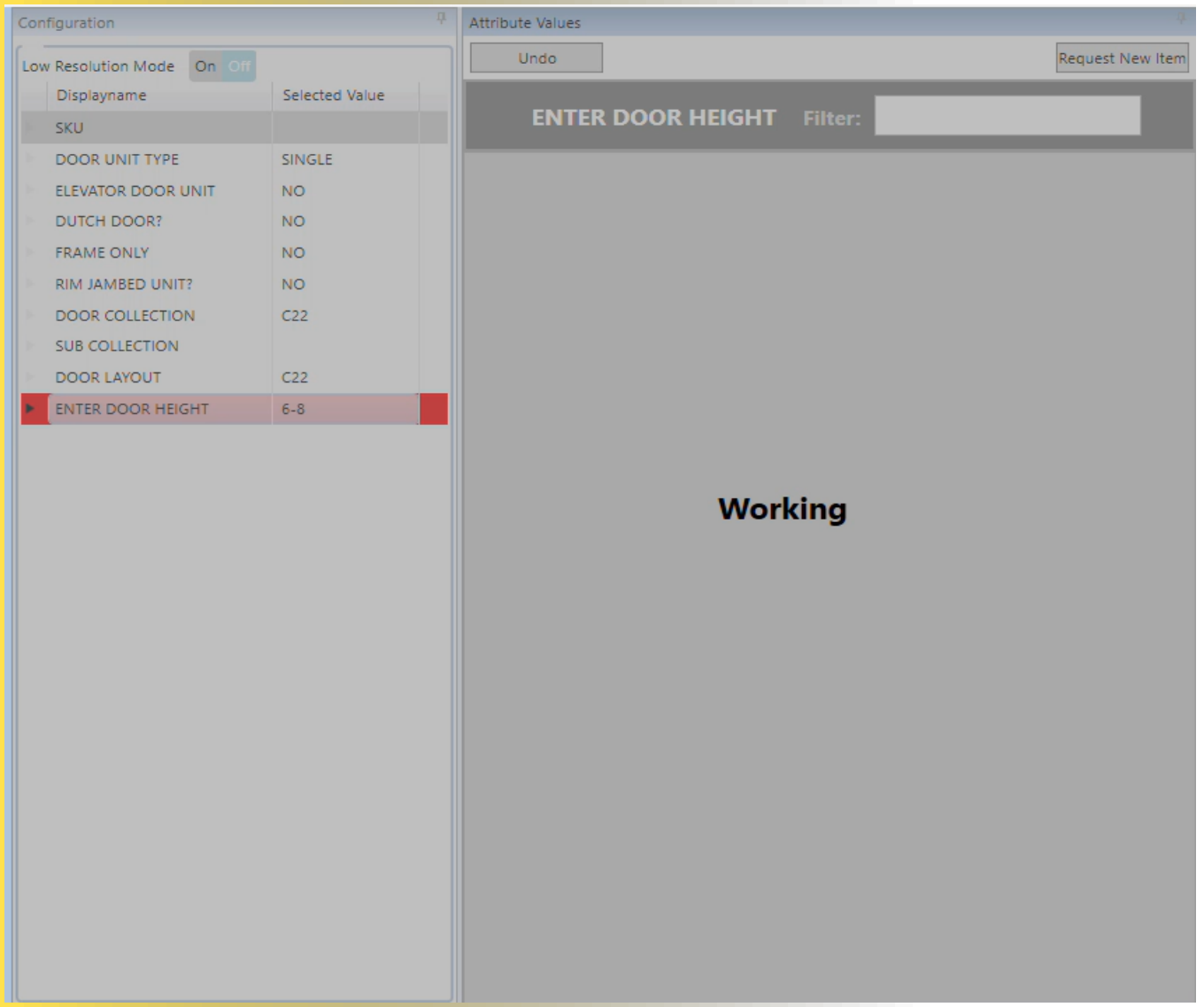
OUR MAIN SYNERGY DBL METHOD

- Runs every time a user answers a question.
- Takes in and returns a list of 'Configurations', objects which are created whenever a rule executes.
- Every time a rule runs, a configuration is generated, added to the list, and an attempt is made to resolve it automatically using currently known values.
- Whenever the method reaches a rule that it can't resolve, it passes it back to the user along with a filtered down list of possible values based on previous answers

```
public method BuildConfigurationAndPrimeNext,   ConfigurationBuilderResult
    required in aCompany,           String
    required in aConfigId,          String
    required in aLocation,          String
    required in configChanged,      @Configurations
    required in originalValue,      String
    required in aConfigurations,    @List<@Configurations>
    required in aOrderNo,           int
    required in aLastGoodItemAtts,      @List<@Itemattributes>
    required in aInDebugMode,       boolean
    required out aConfigurationsOut,    @List<@Configurations>
    required out aMessage,          String
    required out ranChangeMode,     boolean
    required out aDescription,      String
    required out aGoodItemAtts,     @List<@Itemattributes>
    required out aNextAttChoices,   @List<@Productattributevalues>
    required out aConfigsToShow,    @List<int>
    required out aDebugConfigsToShow,   @List<int>
    required out aConfigToSelect,   int
    required out aDebugConfigToSelect,  int
    required out aAttsForItems,     @List<@Itemattributes>
    required inOut aChangedAttributes,  @List<string>
    endparams
```

THE END USER EXPERIENCE

- Any rules that have been resolved appear on the sidebar for the user to see.
- The user may return to edit previously answered questions at any time.
- If there are any incomplete configurations, the user will be prompted to answer the next question in the sequence.
- Once the user provides an answer, the process repeats until all rules have been processed and the manufactured item is complete.



THE LOGIC BEHIND MATERIAL SELECTION

- When a material selection rule is reached, the server retrieves a list of all relevant product type attributes.
- The server compares that list against the current list of known values.
- An incomplete Configurations record is generated for each attribute that does not have a known value.
- An attempt is made to resolve the final component for this rule.

```
if (aLastRuleAnswered!=0)
begin
    tRule = CSharpRoutines.LinqMethods.getConfiguratorRule(aConfigRules, aLastRuleAnswered)
    if(tRule != ^null)
    begin
        if (tRule.Recordtype == ConfigRecordType.MaterialSelection.ToString())
        begin
            startTime = DateTime.Now

            data itemResolved,  Boolean,    false

            result = GetConfigurationMaterialResults(tRule,aConfigurationsOut, aLocation,itmattCh,attvalCh,it
```

```
data _pta,  @Producttypeattributes
foreach _pta  in ptaList
begin
    data okToAdd,   boolean,    true
    data confExists,    @Configurations

    if(!aRule.Optionalmaterial)
    begin
        foreach confExists in aConfigurations
        begin
            if (confExists.Attribute.Trim()==_pta.Attribute.Trim())
            begin
                if(confExists.Producttype.Trim() == aRule.Producttype.Trim() && confExists.Productsubtype.Trim() == aRule.Productsubtype.Trim())
                begin
                    okToAdd = false
                    exitloop
                end
            end
        end
    end

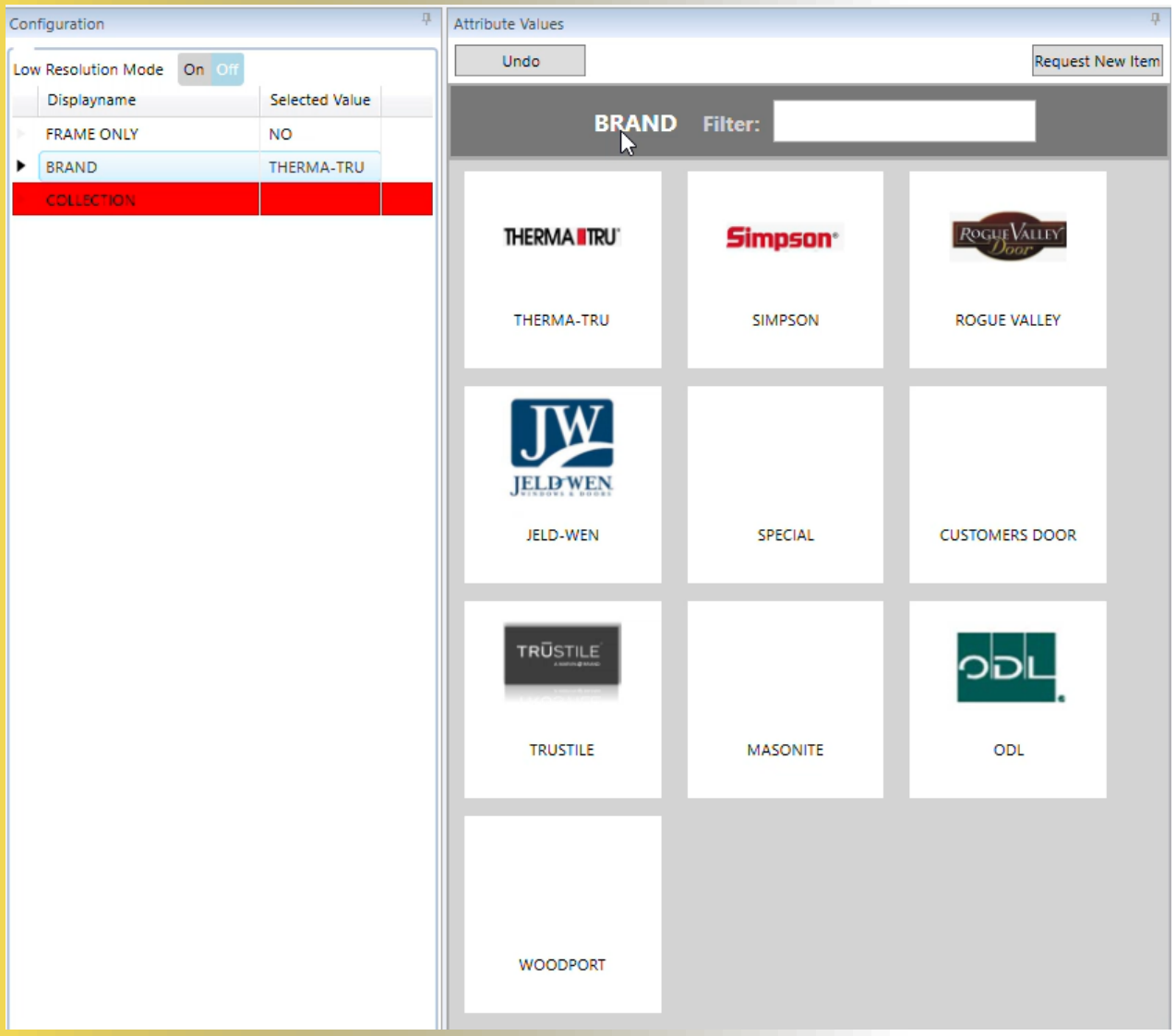
    if (!okToAdd)
    begin
        nextloop
    end
    newConfig = new Configurations()

    newConfig.Answerstatus = ConfigurationAnsweredStatus.UnAnswered.ToString()
    newConfig.Attribute = %atrimtostring(_pta.attribute)
    newConfig.Displayname = %atrimtostring(_pta.attribute)
```

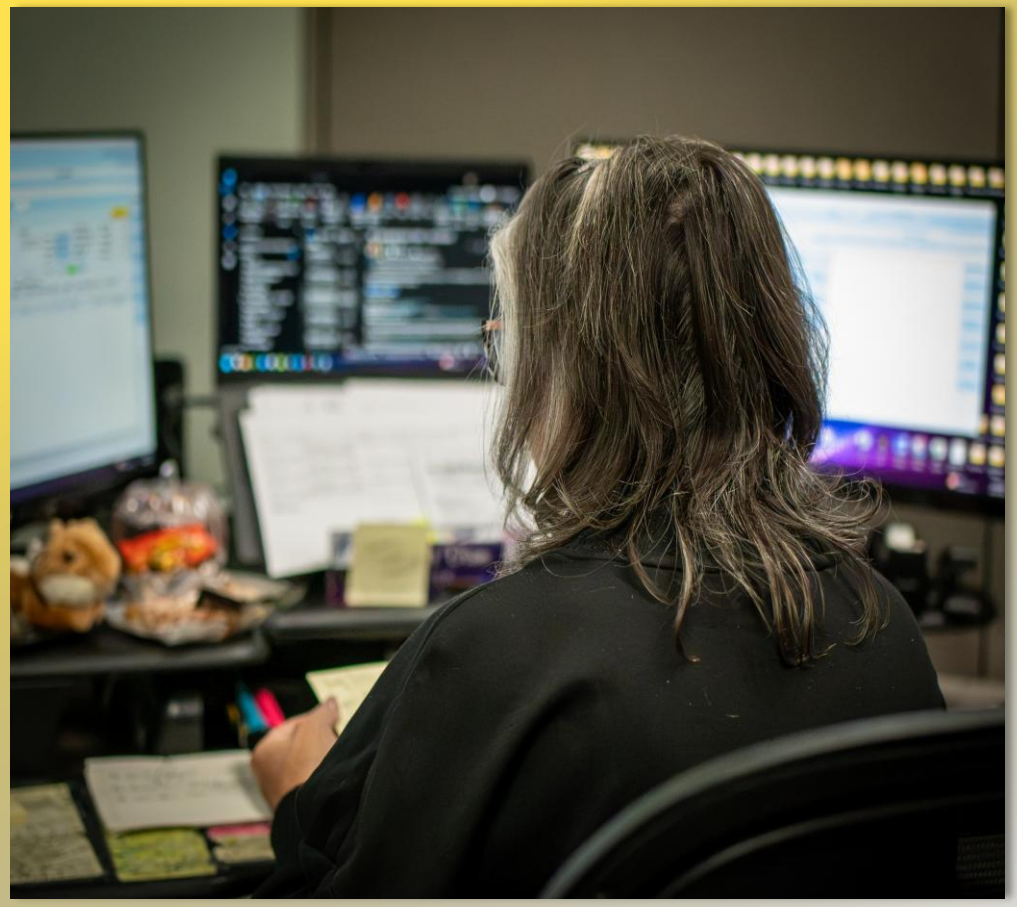
Jackson LUMBER & MILLWORK

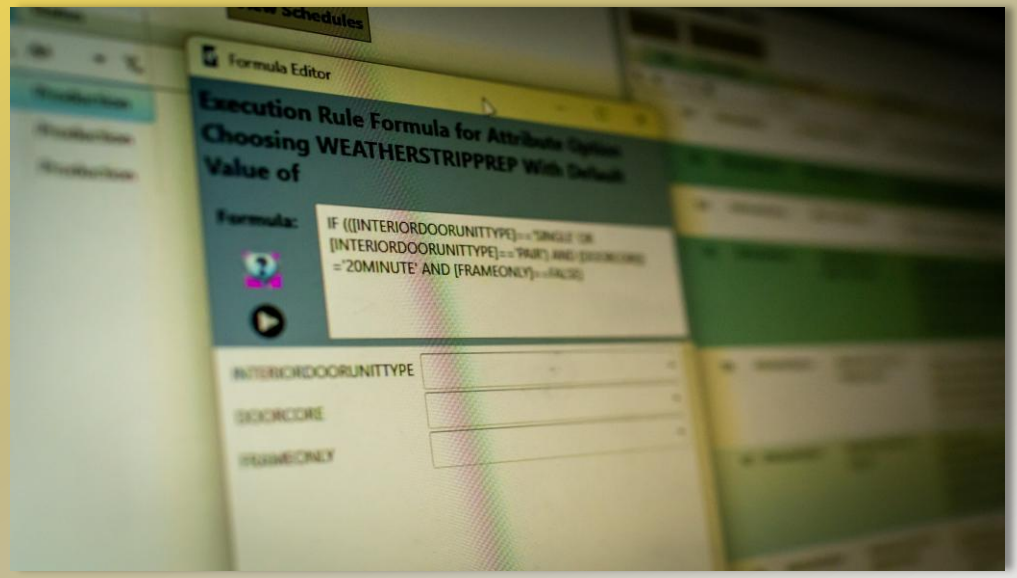
THE LOGIC BEHIND MATERIAL SELECTION

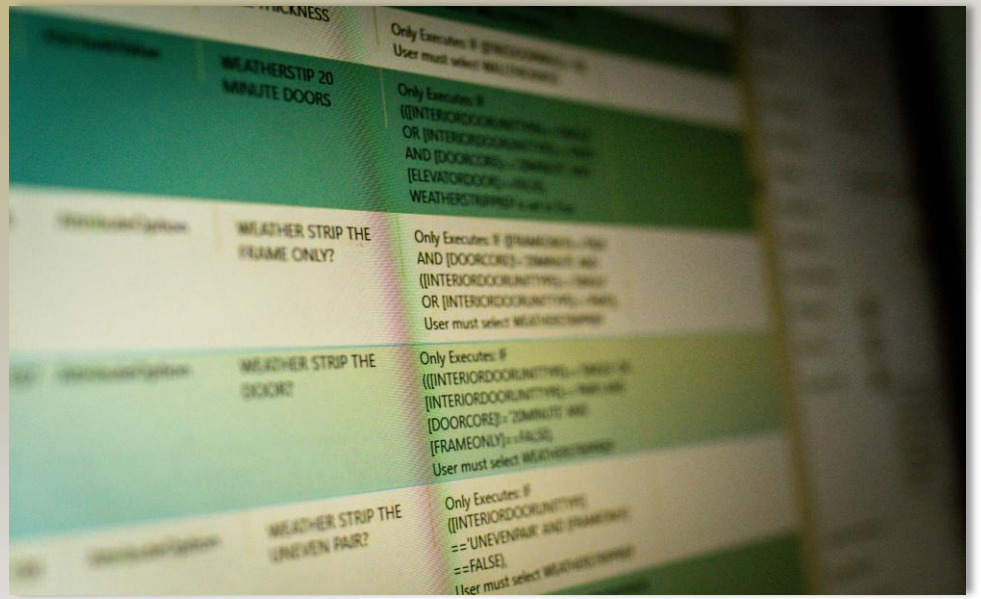
- If a unique component is resolved, all unanswered values are filled in from that component's attributes.
- Otherwise, for each unanswered attribute, the method looks at that attribute's value across all possible matching components. If the attribute has a single unique value, it gets applied to that Configuration.
- Once all attributes that can be resolved have been, the list of Configurations is handed back to the user to prompt the next question.



DESIGNED FOR NON-PROGRAMMERS







Jackson LUMBER & MILLWORK

COMPLEX RULES

- Allows users with no background in traditional code or Synergy to write formulas which execute at runtime
- We allow them to build what we call "Option Sets", grouping many formulas into a bundle and then assigning them to Configurator Rules.

DOORCASINGISW Rules

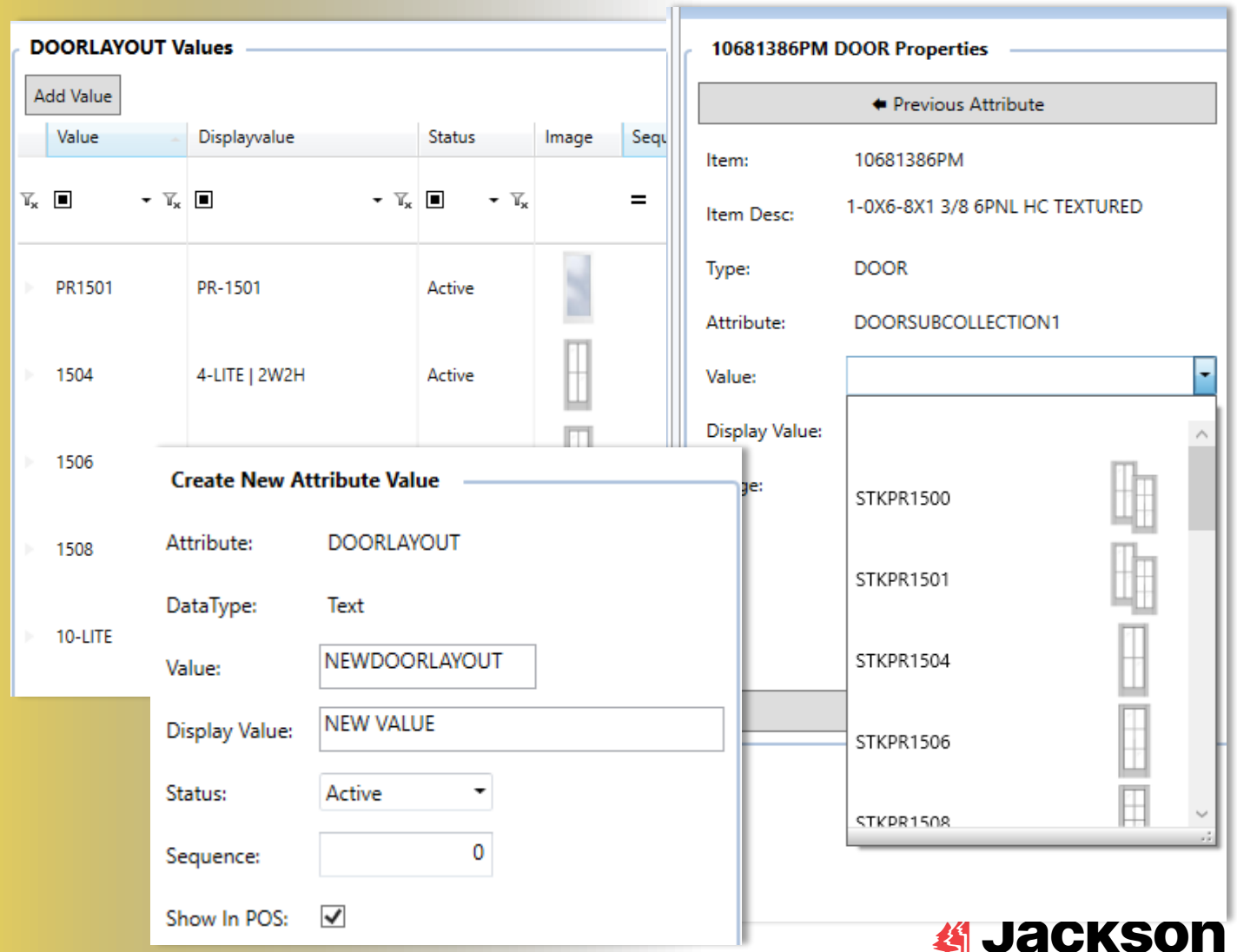
Add Rule

Sequence	Description	Productsubtype	Conditionformula	Attribute	Evaluation	Value
26		OBS LEG CASING	IF ([INTERIORDOORUNITTYPE]=='PAIR' AND [JAMBTYPE]=='SGLRABBET' AND [ASTRAGAL] ==TRUE)			@DECIMALTOFRACTION(^(2*[DOORNETWIDTH])-1 +.5+.5^) + '"'
27		BS LEG CASING	IF ([INTERIORDOORUNITTYPE]=='PAIR' AND [ASTRAGAL]==TRUE)			@DECIMALTOFRACTION(^(2*[DOORNETWIDTH]) +.5+.5^) + '"'
28			IF ([INTERIORDOORUNITTYPE]=='PAIR' AND [CONFIGURATIONSKU]=='MI' AND [ASTRAGAL] ==FALSE)			@DECIMALTOFRACTION(^ ([DOOROPENINGWIDTH])+.5^) + '"'
29			IF ([INTERIORDOORUNITTYPE]=='PAIR' AND [CONFIGURATIONSKU]=='MI' AND [ASTRAGAL] ==TRUE)			@DECIMALTOFRACTION(^ ([DOOROPENINGWIDTH])+1^) + '"'
30		OBS LEG CASING	IF ([INTERIORDOORUNITTYPE]=='UNEVENPAIR' AND [JAMBTYPE]!='SGLRABBET' AND [ASTRAGAL] ==FALSE)			@DECIMALTOFRACTION(^[DOORWIDTH] +[DOORWIDTH2]+.5^) + '"'
31		OBS LEG CASING	IF ([INTERIORDOORUNITTYPE]=='UNEVENPAIR' AND [JAMBTYPE]=='SGLRABBET' AND [ASTRAGAL] ==FALSE)			@DECIMALTOFRACTION(^[DOORWIDTH] +[DOORWIDTH2]-1+.5^) + '"'
32		BS LEG CASING	IF ([INTERIORDOORUNITTYPE]=='UNEVENPAIR' AND [ASTRAGAL]==FALSE)			@DECIMALTOFRACTION(^[DOORWIDTH] +[DOORWIDTH2]+.5^) + '"'
35		OBS LEG CASING	IF ([INTERIORDOORUNITTYPE]=='UNEVENPAIR' AND [JAMBTYPE]!='SGLRABBET' AND [ASTRAGAL] ==TRUE)			@DECIMALTOFRACTION(^[DOORWIDTH] +[DOORWIDTH2]+.5+.5^) + '"'
36		OBS LEG CASING	IF ([INTERIORDOORUNITTYPE]=='UNEVENPAIR' AND [JAMBTYPE]=='SGLRABBET' AND [ASTRAGAL] ==TRUE)			@DECIMALTOFRACTION(^[DOORWIDTH] +[DOORWIDTH2]-1+.5+.5^) + '"'
37		BS LEG CASING	IF ([INTERIORDOORUNITTYPE]=='UNEVENPAIR' AND [ASTRAGAL]==TRUE)			@DECIMALTOFRACTION(^[DOORWIDTH] +[DOORWIDTH2]+.5+.5^) + '"'
50			IF ([EXTDOORCASEHOW]=='INTCASINGBS')			@DECIMALTOFRACTION(^[EXTDOORWIDTH] + .1875+.5^) + '"'
51			IF ([EXTDOORCASEHOW]=='INTCASING')			@DECIMALTOFRACTION(^[EXTDOORWIDTH] + .1875^) + '"'

Jackson LUMBER & MILLWORK

MAINTAINED BY PRODUCT SPECIALISTS

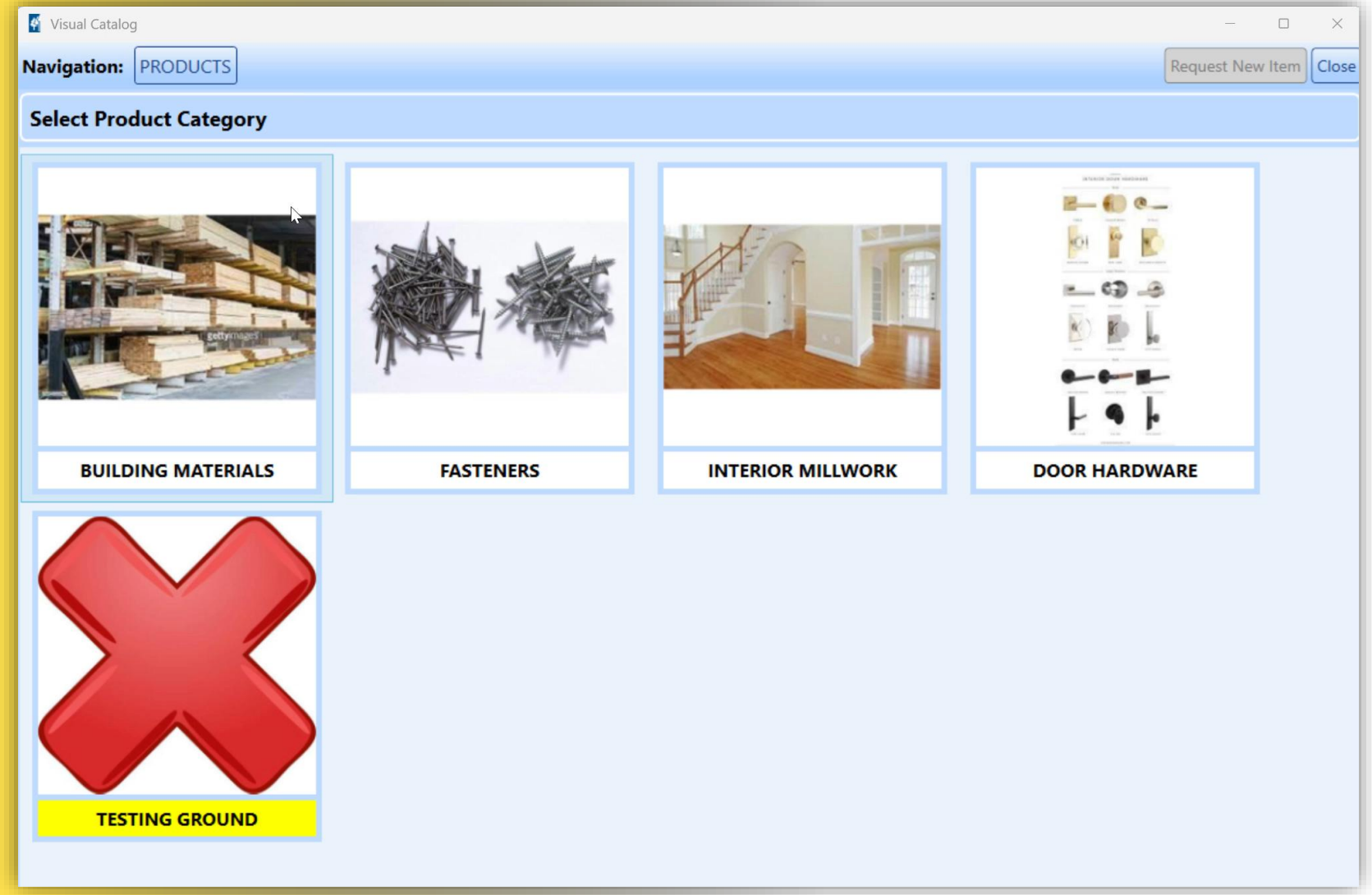
- Allows non-programmers to maintain the system and create new configurations.
- Enables product specialists to directly manage Items, Attributes, and Rules.
- Simplifies updates and reduces reliance on programmers or IT support.



Jackson LUMBER & MILLWORK

VISUAL CATALOG

- The Visual Catalog demonstrates the benefits of assigning Attributes to all Items and components.
- Enables salespeople with limited product knowledge to confidently guide customers to the right products.



Jackson LUMBER & MILLWORK

Jackson
LUMBER & MILLWORK

Questions?

Thank you.

SYNERGY DevPARTNER Conference 2026

Ben Newton, Sr. Programmer
Nick Sarno, Jr. Programmer
October 2026

