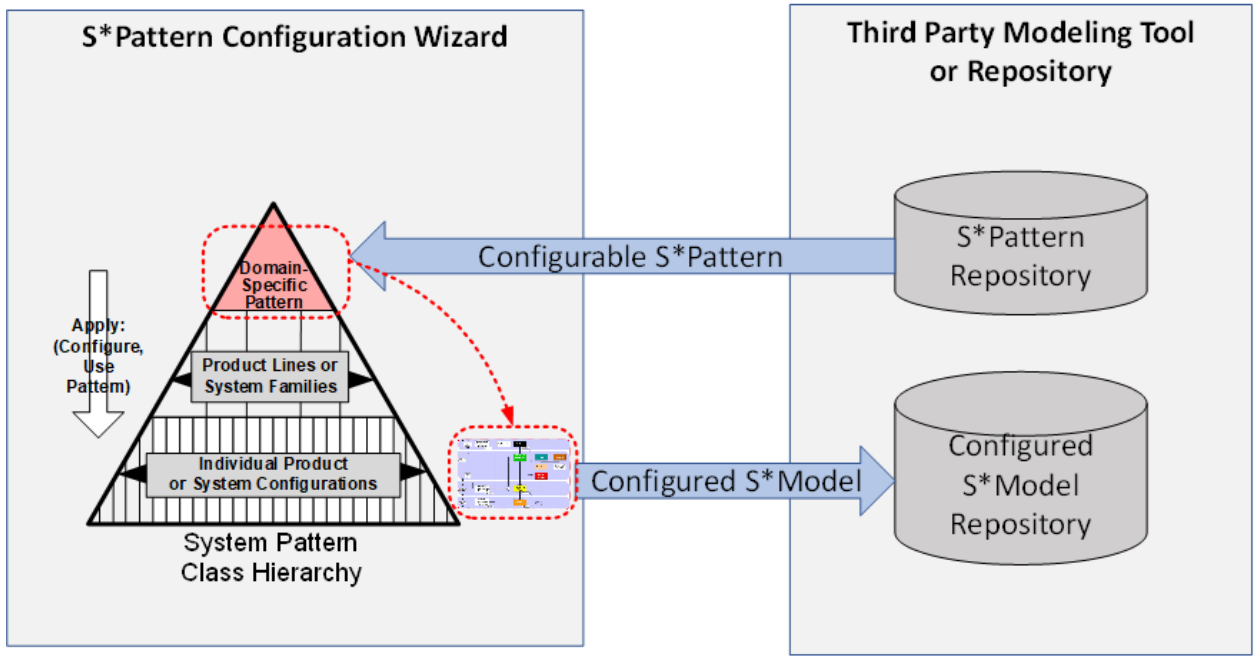
S*Pattern Configuration Wizard: Internals Introduction



CC BY SA

Copyright 2025, System Sciences, LLC

V1.2.4

Licensed under a Creative Commons
Attribution Share Alike-License CC BY SA International 4.0

License Link: https://creativecommons.org/licenses/by-sa/4.0/legalcode

Uses are permitted under this license without further permission from the copyright owner, provided each use (1) is clearly marked to attribute the underlying work to "S*Patterns Community", (2) provides a link to the CC BY SA license, (3) indicates if changes were made, (4) does not suggest the licensor endorses the user or use, (5) does not apply legal terms or technological measures that legally restrict others from doing anything the license permits, and (6) if you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

Permissions beyond the scope of this license are administered through contacting:

Corporate Officer
ICTT System Sciences
378 South Airport Street
Terre Haute, IN 47803
812-232-2208

Systematica is a registered trademark of System Sciences, LLC.

Contents

- Purpose and scope
- Assumed background and roles
- Architecture and technologies
- Documentation resources—built in and external
- S*Pattern Configuration Process: VBA control and M-Code queries
- S*Models Inconsistency Detection Process: VBA control and M-Code queries
- Power Query Queries and Tables; Excel Tabs
- Development environments and aids: M-Code IDE and VBA IDE
- Questions and discussion

- References

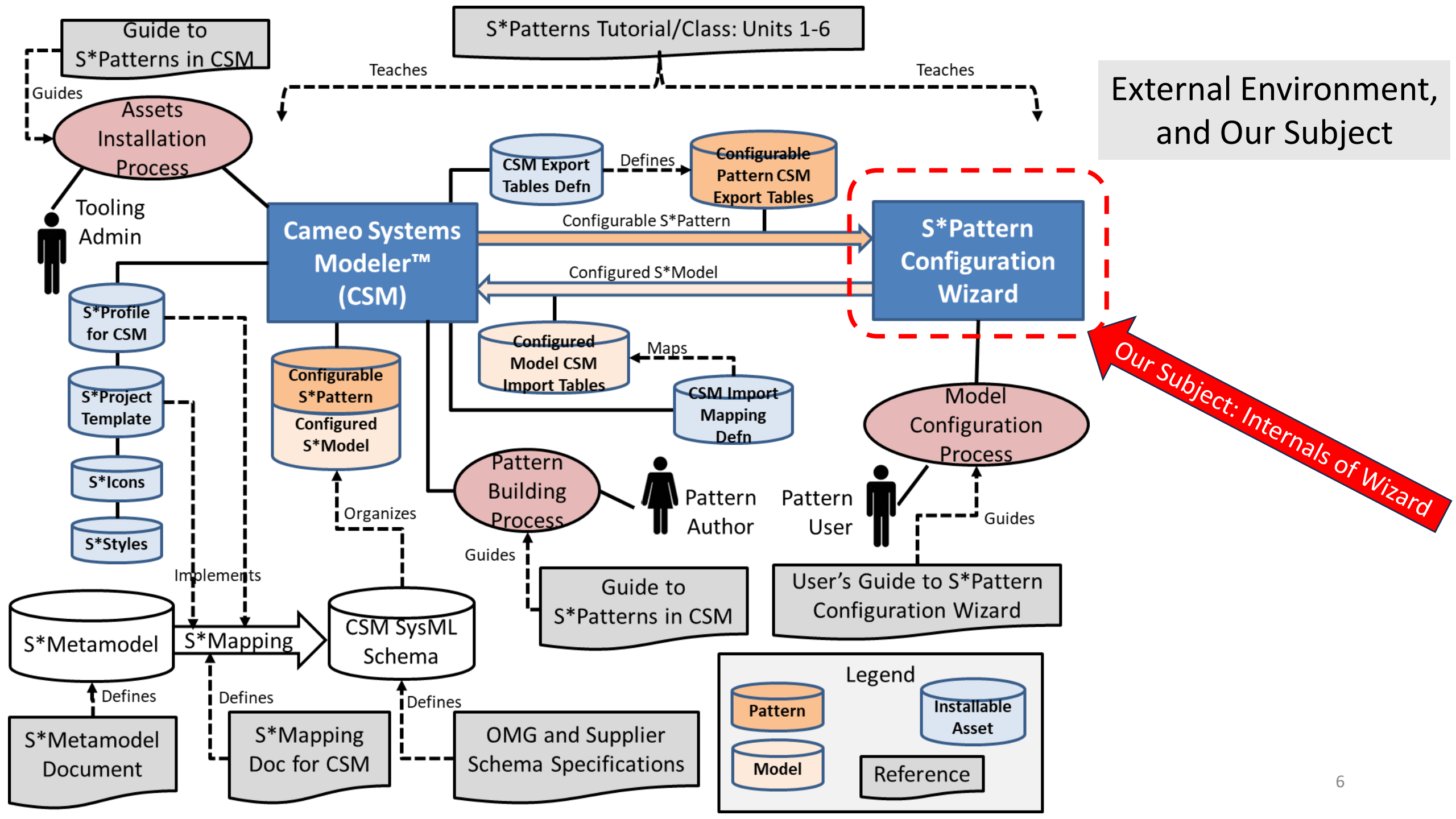
- Attachment I—Level 2 Flow and Queries Architecture Diagrams
- Attachment II—VBA Flow Control Code
- Attachment III—Database Query M-Code

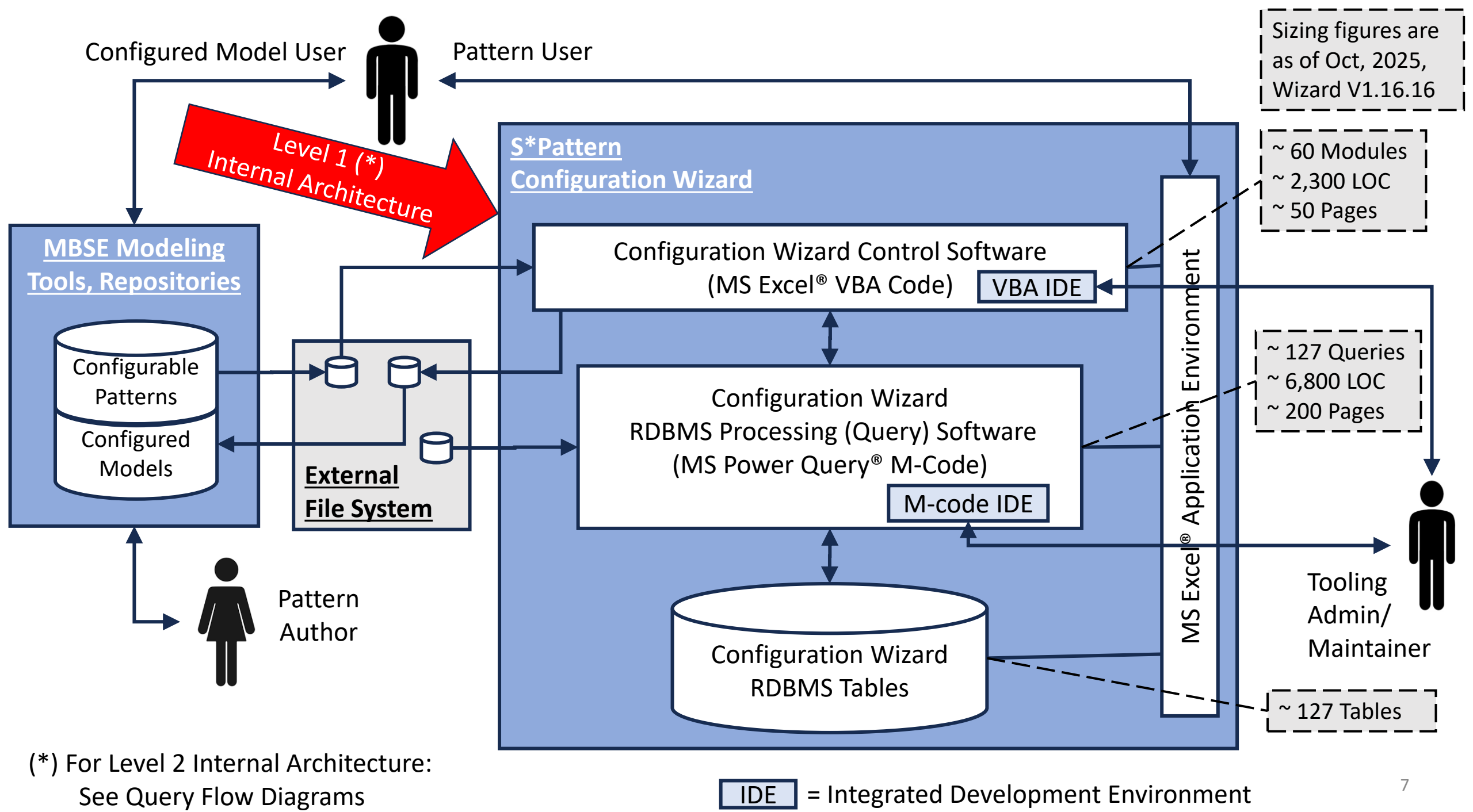
Purpose and scope

- This material is a presentation-oriented short class plus references, introducing the internals of the S*Pattern Configuration Wizard:
  - Internal architecture and technologies
  - Overall query flows
  - Database M-Code samples
  - VBA control code samples
- Including both the Pattern Configuration Process and the Model Consistency Detection Process.
- Listing of detailed documentation resources and references.
- Some diagnostic guidance hints.
- This introductory class is intended to fit into a two-hour time slot; detailed expertise must be gained over a longer period of practice and study.

Assumed background and roles

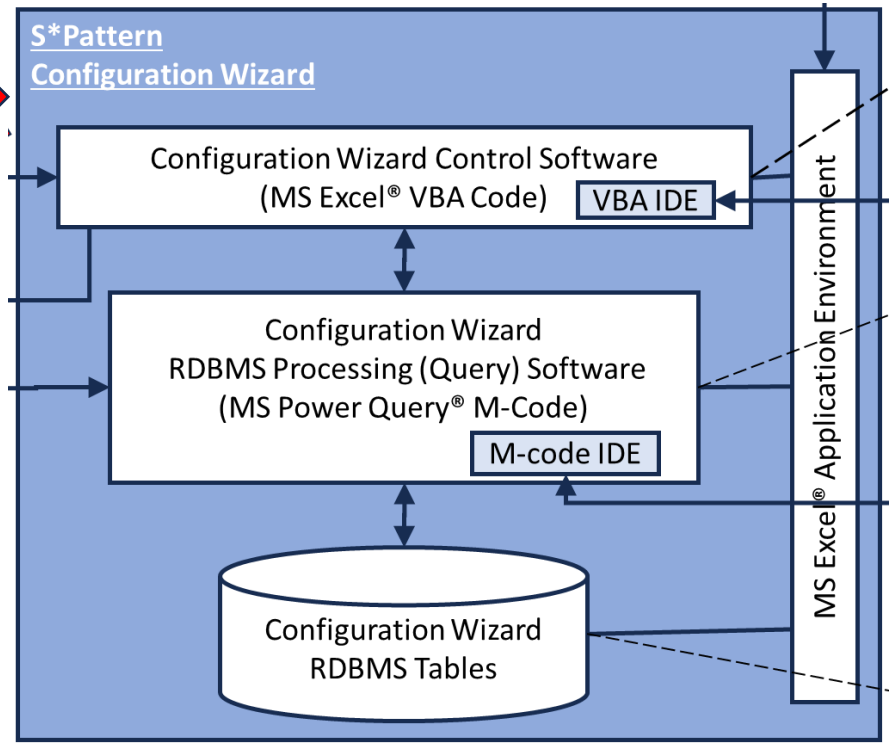
- Referring to the diagram on the next slide, . . .
- This class is aimed primarily for systems engineering “Tooling Admins” (software maintainer specialists and tooling administrators).
- The class may also be audited by individuals who want to obtain general awareness of how internal technologies of the Configuration Wizard are harnessed to produce the capabilities of the Configuration Wizard.
- It is assumed that primary class attendees already have this background:
  - Significant personal experience in the *use and operation of the Configuration Wizard*.
  - General awareness of the internal nature of *RDBMS technologies similar to SQL*.
  - Personal experience with MS M-code and Excel VBA and their development environments are *not required for this class*, but will become mandatory for those who seek to maintain or modify the internal Configuration Wizard software.


Guide to S*Patterns in CSM
S*Patterns Tutorial/Class: Units 1-6
Teaches
Teaches
Guides
Assets Installation Process
Tooling Admin
Cameo Systems Modeler™ (CSM)
CSM Export Tables Defn
Defines
Configurable Pattern CSM Export Tables
Configurable S*Pattern
Configured S*Model
S*Pattern Configuration Wizard
External Environment, and Our Subject
Our Subject: Internals of Wizard
S*Profile for CSM
S*Project Template
S*Icons
S*Styles
Configurable S*Pattern
Configured S*Model
Configured Model CSM Import Tables
Maps
CSM Import Mapping Defn
Model Configuration Process
Pattern Building Process
Pattern Author
Pattern User
Organizes
Guides
Guides
Implements
Guide to S*Patterns in CSM
User's Guide to S*Pattern Configuration Wizard
S*Metamodel
S*Mapping
CSM SysML Schema
Defines
Defines
Defines
S*Metamodel Document
S*Mapping Doc for CSM
OMG and Supplier Schema Specifications
Legend
Pattern
Installable Asset
Model
Reference


Configured Model User
Pattern User
Sizing figures are as of Oct, 2025, Wizard V1.16.16
Level 1 (*) Internal Architecture
S*Pattern Configuration Wizard
~ 60 Modules
~ 2,300 LOC
~ 50 Pages
MBSE Modeling Tools, Repositories
Configuration Wizard Control Software
(MS Excel® VBA Code)
VBA IDE
MS Excel® Application Environment
Configurable Patterns
Configured Models
External File System
~ 127 Queries
~ 6,800 LOC
~ 200 Pages
Configuration Wizard
RDBMS Processing (Query) Software
(MS Power Query® M-Code)
M-code IDE
Pattern Author
Tooling Admin/ Maintainer
Configuration Wizard
RDBMS Tables
~ 127 Tables
(*) For Level 2 Internal Architecture:
See Query Flow Diagrams
IDE = Integrated Development Environment

Configuration Wizard V1.16.16
Microsoft Excel 2021

Level 1
Internal Architecture



Architecture and technologies

- The Level 1 decomposition architecture includes these *underlying information technologies*:
  - MS Excel® VBA overall Wizard flow of control programs, interacting with the Wizard user and invoking M-Code queries.
  - MS Power Query® RDBMS queries, in M-Code, accessing the RDBMS Tables, and providing high performance RDBMS "JOIN" algorithms central to core rule-based Pattern Configuration and difference detection Model Consistency Analysis algorithms.
  - Internal RDBMS tables, managed by M-Code queries.
  - MS Excel® plays a hosting role for the above--the Configuration Wizard is not primarily a "spreadsheet", but is an MS packaging providing Power Query, Tables, and VBA.
  - High productivity Integrated Development Environments (IDEs) for both VBA and M-Code are included, making program-level development / testing / debug efficient.
- At a higher level, the Configuration Wizard is based on pattern-based MBSE technology, including the S*Metamodel.

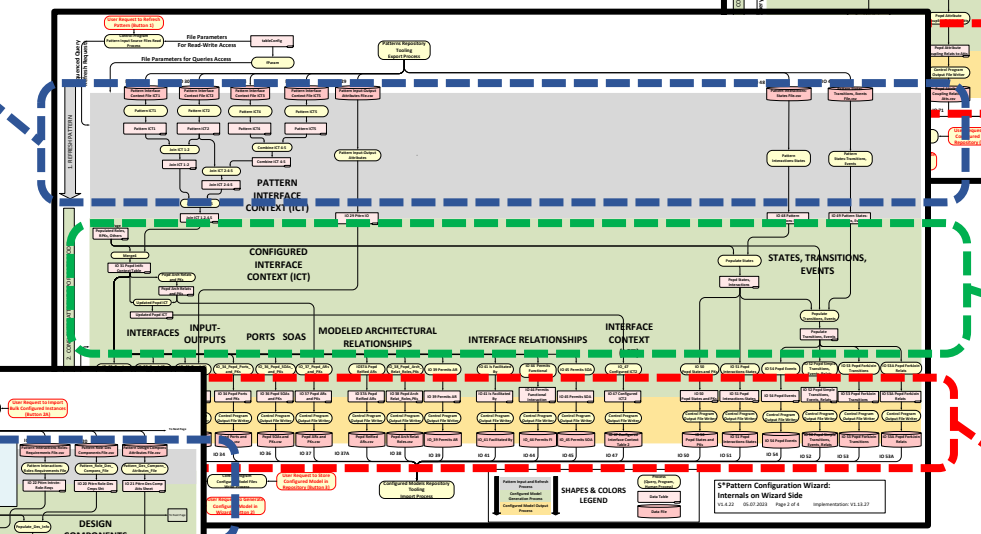
Documentation resources—built in and external

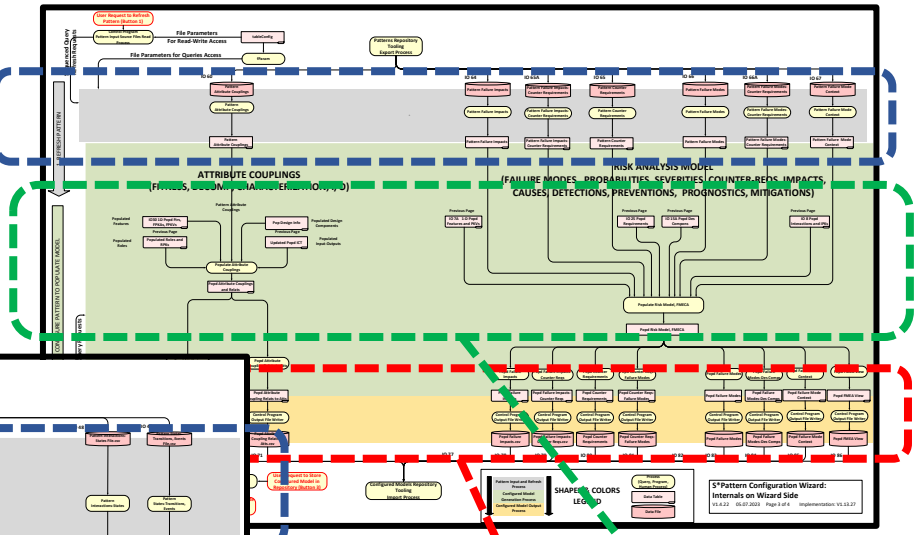
- There are detailed Level 2 flow diagrams.
- The VBA flow control source code is heavily commented and built-in accessible.
- The RDBMS Queries M-Code source is heavily commented and built-in accessible .
- The Wizard has a built-in External Interface Specification
- Excel VBA and its IDE are well documented by MS online and other resources
- MS M-Code and its IDE are well documented by MS online and other resources
- There is a User's Guide to the use of the Configuration Wizard
- The S*Metamodel is documented formally and by educational materials
- There is a formal mapping from S*MTM to SysML, documented.
- There is a user's Guide to Cameo Systems Modeler support for MBSE Patterns

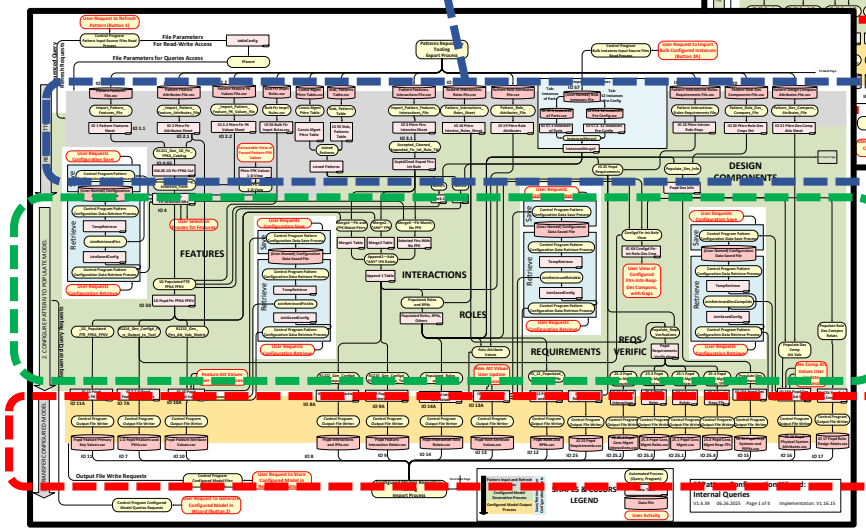
- See the References and Attachments I, II, and III

The S*Pattern Configuration Process: Flow and Code Sample Walk-Through

1: Refresh Configurable Pattern







2: Populate Configured Model

We'll drill in here.

3: Return Configured Model to Repository

Configuration Rules:
Part of an S*Pattern

Feature	FPK	Interaction	IPK Rule

Feature-Interaction Rules Table

Role	RPK	Des Comp	DCPK Rule

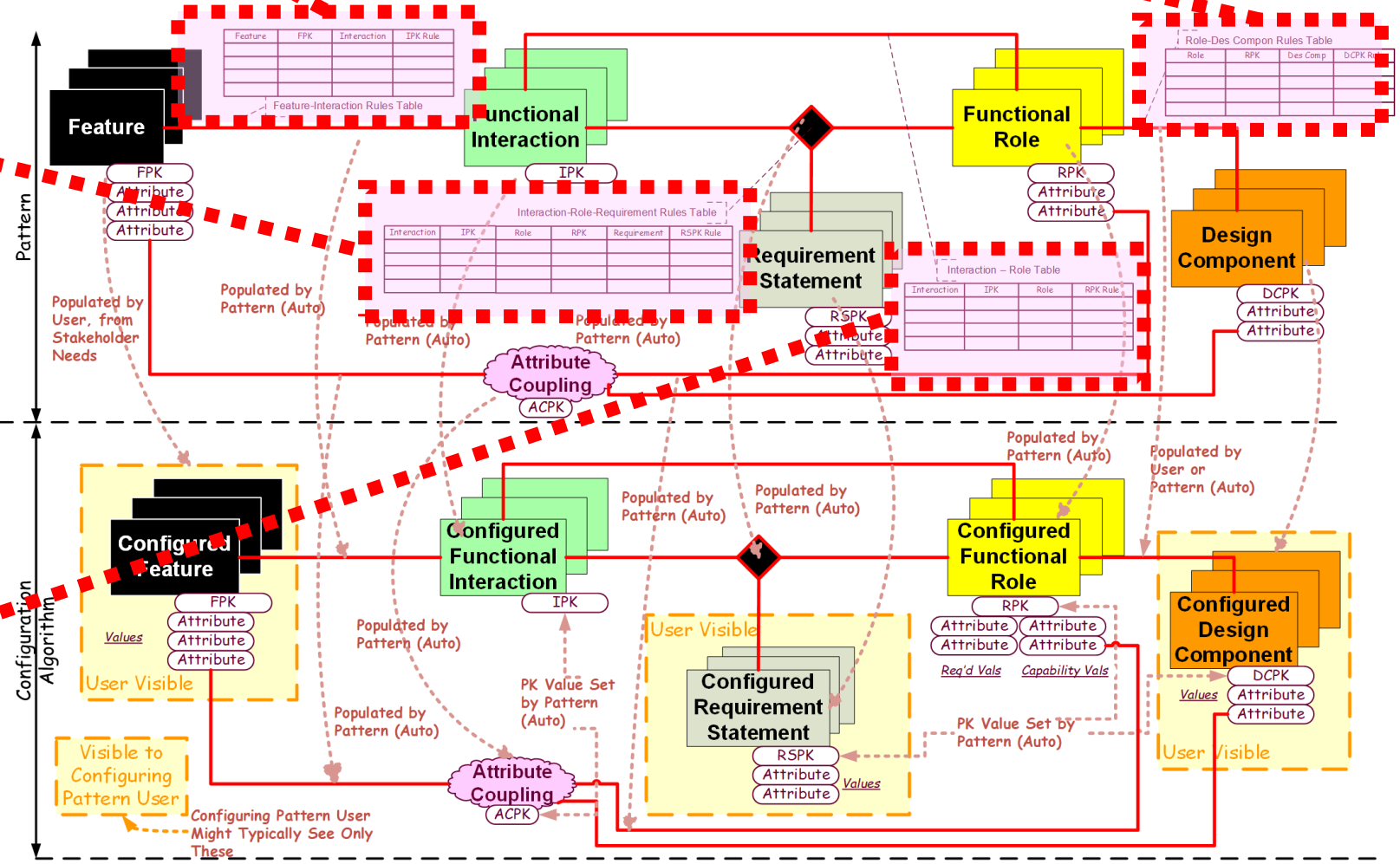
Role-Des Compon Rules Table

Interaction	IPK	Role	RPK	Requirement	RSPK Rule

Interaction-Role-Requirement Rules Table

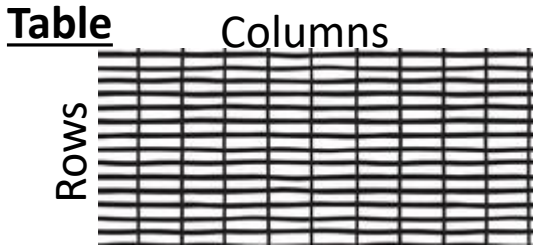
Interaction	IPK	Role	RPK Rule

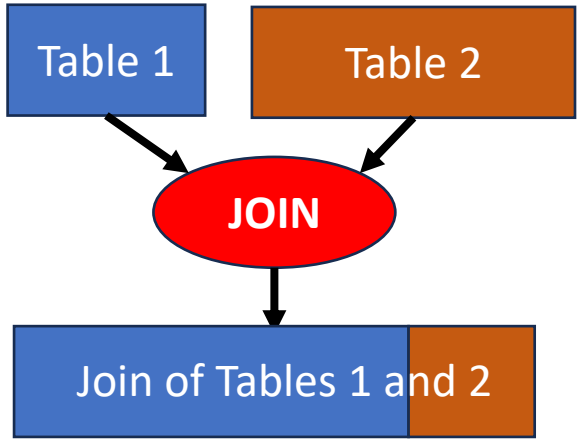
Interaction – Role Table



RDBMS JOINS Directly Fit the Configuration Process

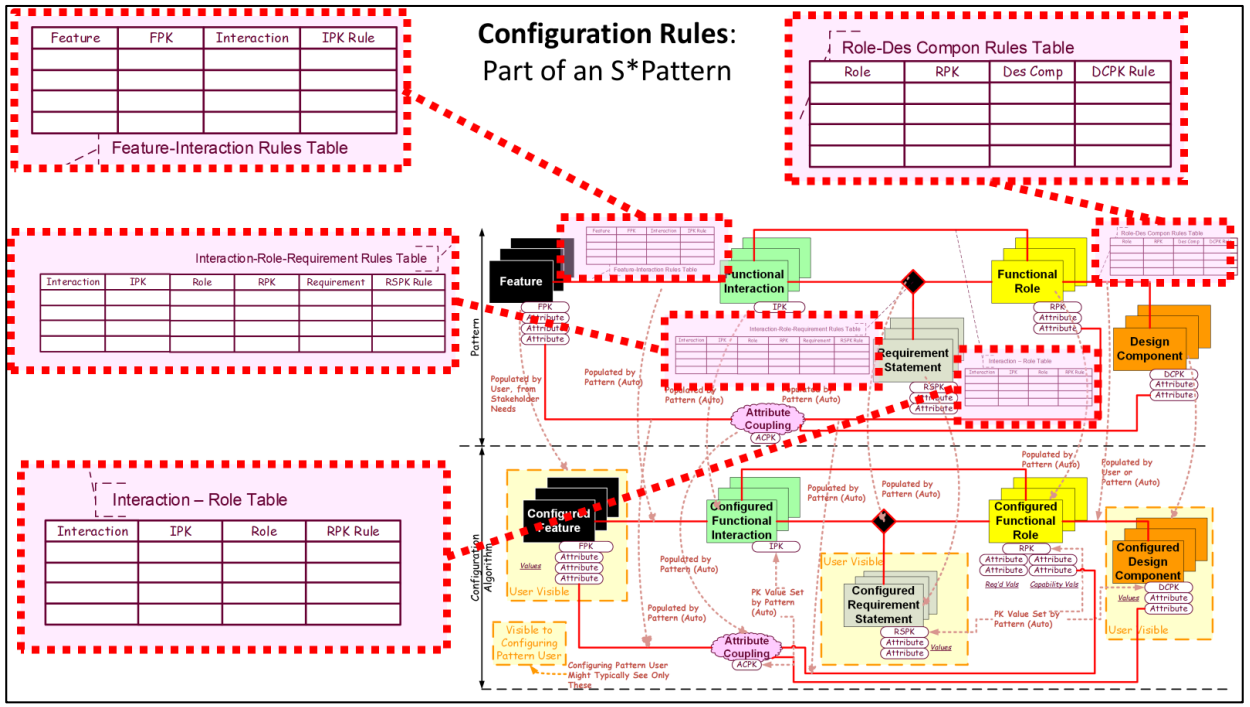
- RDBMS “JOIN” operations create a new table from two existing tables.
- The resulting JOINED table:
  - May have more or fewer rows than the input tables.
  - May have more or fewer columns than the input tables.
  - May be empty.
- There are different types of JOINs:
  - **OUTER JOINs**: Creates a table whose rows come from every combination of rows of both Table 1 and Table 2. (“Pair Permutations”)
  - **INNER JOINs**: Creates a table whose rows comes from only combinations of Table 1 and Table 2 rows that meet some joint criteria, such as equality of some of their column values. (“Projections”)
  - **ANTI JOINs:** Creates a table whose content comes only from rows in one table that are not found in the other table. (“Distinctions”)



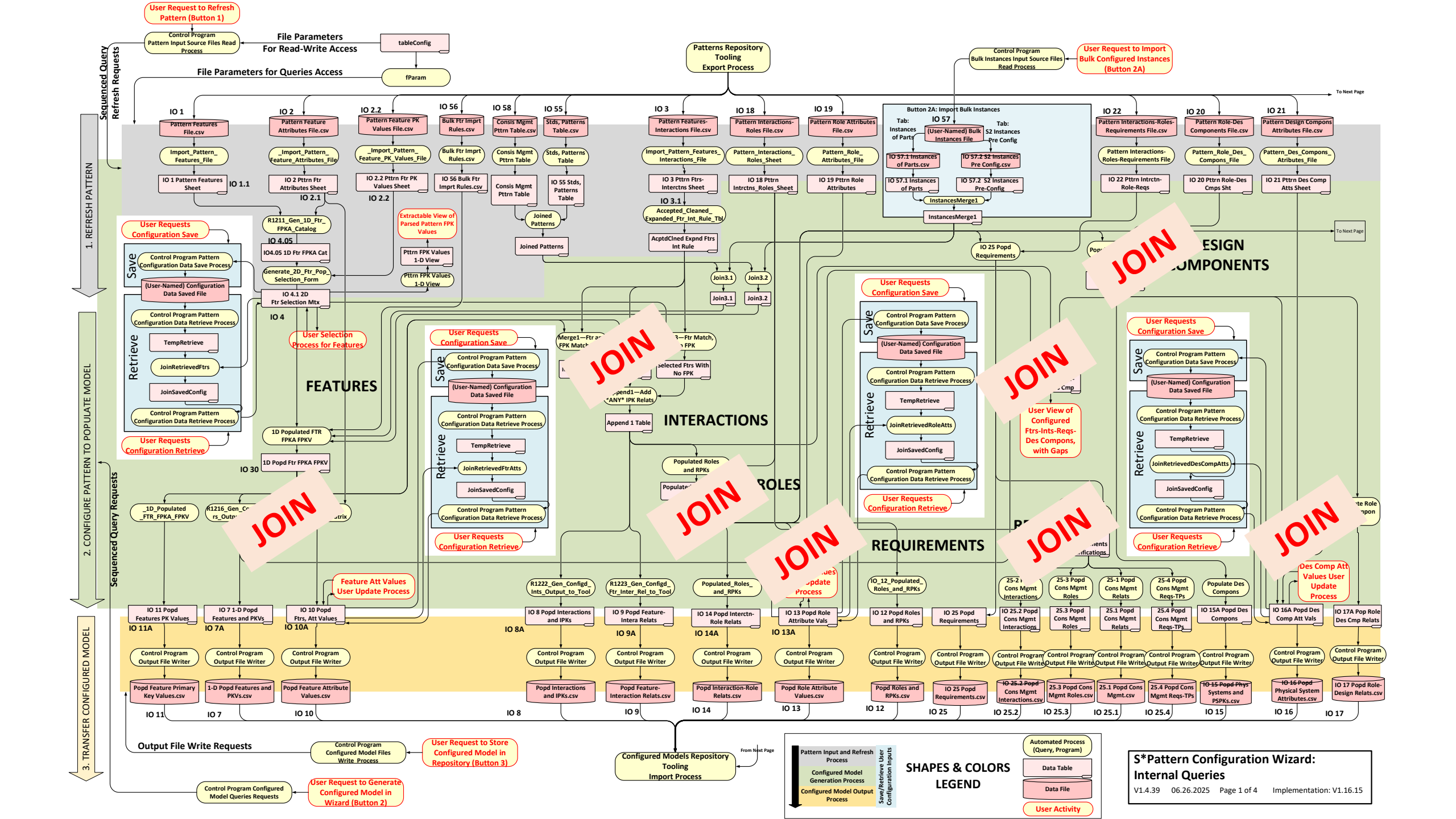


So, INNER JOIN is an excellent implementation of the S*Pattern Configuration Algorithm

- Configured Interactions are the JOIN of [Populated Features & FPKs] with [Feature-Interaction Configuration Rules],
- Configured Roles are the JOIN of [Populated Interactions & IPKs] with [Interaction-Role Configuration Rules],
- Configured Design Components are the JOIN of [Populated Roles] with [Role-Design Component Configuration Rules], etc.



- RDBMS JOIN algorithms “at scale” also have decades of optimization of speed, parallelization, etc., in COTS RDBMS systems (ORACLE, MS SQL SERVER, MS POWER QUERY, etc.)
- Use of JOIN as the central technology of the Configuration Wizard puts it in the mainstream of information technology.


User Request to Refresh Pattern (Button 1)
Control Program Pattern Input Source Files Read Process
File Parameters For Read-Write Access
tableConfig
File Parameters for Queries Access
fParam
Sequenced Query Refresh Requests
1. REFRESH PATTERN
2. CONFIGURE PATTERN TO POPULATE MODEL
3. TRANSFER CONFIGURED MODEL
Sequenced Query Requests
Patterns Repository Tooling Export Process
Control Program Bulk Instances Input Source Files Read Process
User Request to Import Bulk Configured Instances (Button 2A)
To Next Page
IO 1
Pattern Features File.csv
Import_Pattern_ Features_File
IO 1 Pattern Features Sheet
IO 1.1
IO 2
Pattern Feature Attributes File.csv
_Import_Pattern_ Feature_Attributes_File
IO 2 Pttrn Ftr Attributes Sheet
IO 2.1
IO 2.2
Pattern Feature PK Values File.csv
_Import_Pattern_ Feature_PK_Values_File
IO 2.2 Pttrn Ftr PK Values Sheet
IO 56
Bulk Ftr Imprt Rules.csv
IO 56 Bulk Ftr Imprt Rules.csv
IO 58
Consis Mgmt Pttrn Table.csv
Consis Mgmt Pttrn Table
IO 55
Stds, Patterns Table.csv
IO 55 Stds, Patterns Table
Joined Patterns
IO 3
Pattern Features-Interactions File.csv
Import_Pattern_Features_ Interactions_File
IO 3 Pttrn Ftrs-Interctns Sheet
IO 3.1
Accepted_Cleaned_ Expanded_Ftr_Int_Rule_Tbl
AcptdClned Expnd Ftrs Int Rule
IO 18
Pattern Interactions-Roles File.csv
Pattern_Interactions_ Roles_Sheet
IO 18 Pttrn Intrctns_Roles_Sheet
IO 19
Pattern Role Attributes File.csv
Pattern_Role_ Attributes_File
IO 19 Pttrn Role Attributes
Button 2A: Import Bulk Instances
Tab: Instances of Parts
IO 57
(User-Named) Bulk Instances File
Tab: S2 Instances Pre Config
IO 57.1 Instances of Parts.csv
IO 57.2 S2 Instances Pre Config.csv
IO 57.1 Instances of Parts
IO 57.2 S2 Instances Pre-Config
InstancesMerge1
IO 22
Pattern Interactions-Roles-Requirements File.csv
Pattern Interactions-Roles-Requirements File
IO 22 Pttrn Intrctn-Role-Reqs
IO 20
Pattern Role-Des Components File.csv
Pattern_Role_Des_ Compons_File
IO 20 Pttrn Role-Des Cmps Sht
IO 21
Pattern Design Compons Attributes File.csv
Pattern_Des_Compons_ Atributes_File
IO 21 Pttrn Des Comp Atts Sheet
IO 25 Popd Requirements
DESIGN COMPONENTS
User Requests Configuration Save
Save
Control Program Pattern Configuration Data Save Process
(User-Named) Configuration Data Saved File
Retrieve
Control Program Pattern Configuration Data Retrieve Process
TempRetrieve
JoinRetrievedFtrs
JoinSavedConfig
User Requests Configuration Retrieve
R1211_Gen_1D_Ftr_ FPKA_Catalog
IO 4.05
IO4.05 1D Ftr FPKA Cat
Generate_2D_Ftr_Pop_ Selection_Form
IO 4.1 2D Ftr Selection Mtx
IO 4
Extractable View of Parsed Pattern FPK Values
Pttrn FPK Values 1-D View
Pttrn FPK Values 1-D View
User Selection Process for Features
FEATURES
1D Populated FTR FPKA FPKV
IO 30
1D Popd Ftr FPKA FPKV
Join3.1
Join3.2
Merge1—Ftr a… FPK Match
…—Ftr Match, … FPK
…elected Ftrs With No FPK
…pend1—Add *ANY* IPK Relats
Append 1 Table
INTERACTIONS
Populated Roles and RPKs
Populate…
…ROLES
JOIN
User View of Configured Ftrs-Ints-Reqs-Des Compons, with Gaps
REQUIREMENTS
…_1D_Populated _FTR_FPKA_FPKV
R1216_Gen_C… rs_Outp…
…trix
Feature Att Values User Update Process
IO 11 Popd Features PK Values
IO 11A
IO 7 1-D Popd Features and PKVs
IO 7A
IO 10 Popd Ftrs, Att Values
IO 10A
R1222_Gen_Configd_ Ints_Output_to_Tool
IO 8 Popd Interactions and IPKs
IO 8A
R1223_Gen_Configd_ Ftr_Inter_Rel_to_Tool
IO 9 Popd Feature-Intera Relats
IO 9A
Populated_Roles_ and_RPKs
IO 14 Popd Interctn-Role Relats
IO 14A
…Update Process
IO 13 Popd Role Attribute Vals
IO 13A
IO_12_Populated_ Roles_and_RPKs
IO 12 Popd Roles and RPKs
IO 25 Popd Requirements
25-2 Popd Cons Mgmt Interactions
25-3 Popd Cons Mgmt Roles
25-1 Popd Cons Mgmt Relats
25-4 Popd Cons Mgmt Reqs-TPs
Populate Des Compons
IO 15A Popd Des Compons
…Des Comp Att Values User Update Process
IO 16A Popd Des Comp Att Vals
IO 17A Pop Role Des Cmp Relats
Control Program Output File Writer
Popd Feature Primary Key Values.csv
IO 11
1-D Popd Features and PKVs.csv
IO 7
Popd Feature Attribute Values.csv
IO 10
Popd Interactions and IPKs.csv
IO 8
Popd Feature-Interaction Relats.csv
IO 9
Popd Interaction-Role Relats.csv
IO 14
Popd Role Attribute Values.csv
IO 13
Popd Roles and RPKs.csv
IO 12
IO 25 Popd Requirements.csv
IO 25
IO 25.2 Popd Cons Mgmt Interactions.csv
IO 25.2
IO 25.3 Popd Cons Mgmt Roles.csv
IO 25.3
IO 25.1 Popd Cons Mgmt.csv
IO 25.1
IO 25.4 Popd Cons Mgmt Reqs-TPs
IO 25.4
IO 15 Popd Phys Systems and PSPKs.csv
IO 15
IO 16 Popd Physical System Attributes.csv
IO 16
IO 17 Popd Role-Design Relats.csv
IO 17
Output File Write Requests
Control Program Configured Model Files Write Process
User Request to Store Configured Model in Repository (Button 3)
Control Program Configured Model Queries Requests
User Request to Generate Configured Model in Wizard (Button 2)
Configured Models Repository Tooling Import Process
From Next Page
Pattern Input and Refresh Process
Configured Model Generation Process
Configured Model Output Process
Save/Retrieve User Configuration Inputs
SHAPES & COLORS LEGEND
Automated Process (Query, Program)
Data Table
Data File
User Activity
S*Pattern Configuration Wizard: Internal Queries
V1.4.39 06.26.2025 Page 1 of 4 Implementation: V1.16.15

Nature of Joins compared to Vector Spaces

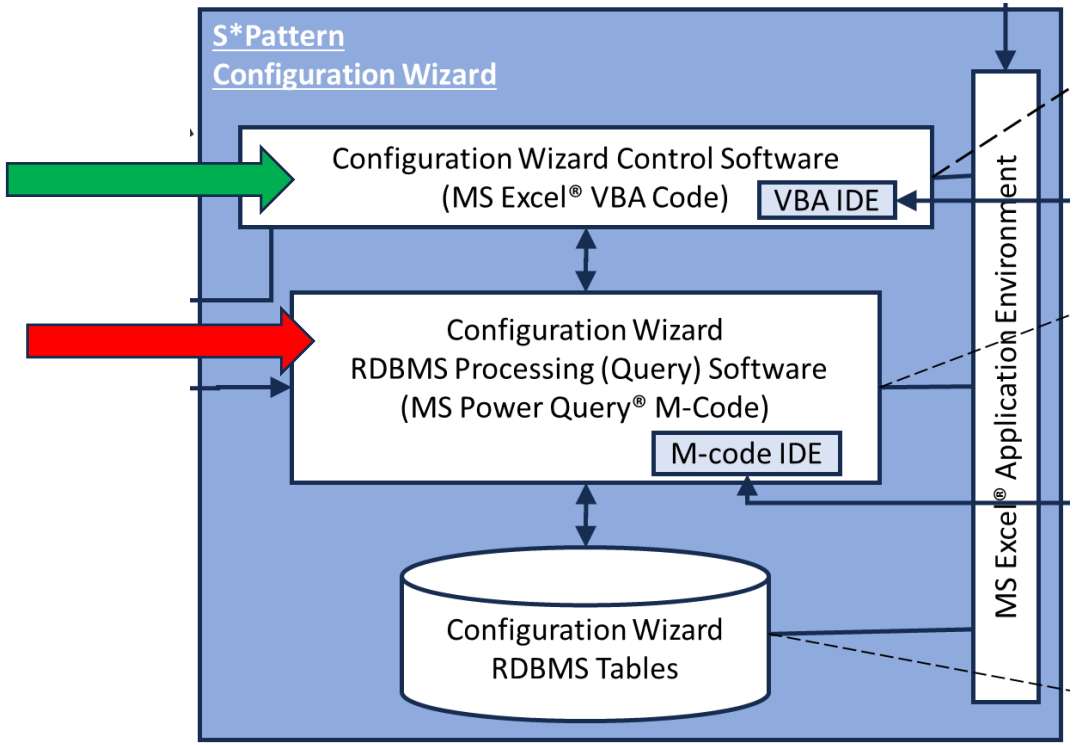
- INNER JOINS can also be understood as vector space projections into subspaces.
- This is a powerful way to understand the S*Pattern Configuration process as it is expressed by INNER JOINS to populate specific models.
- Understanding distance between inconsistent MBSE models in a metric space is also part of this mathematical perspective.
- Although outside the scope of this class, the related ideas are described further in this reference.



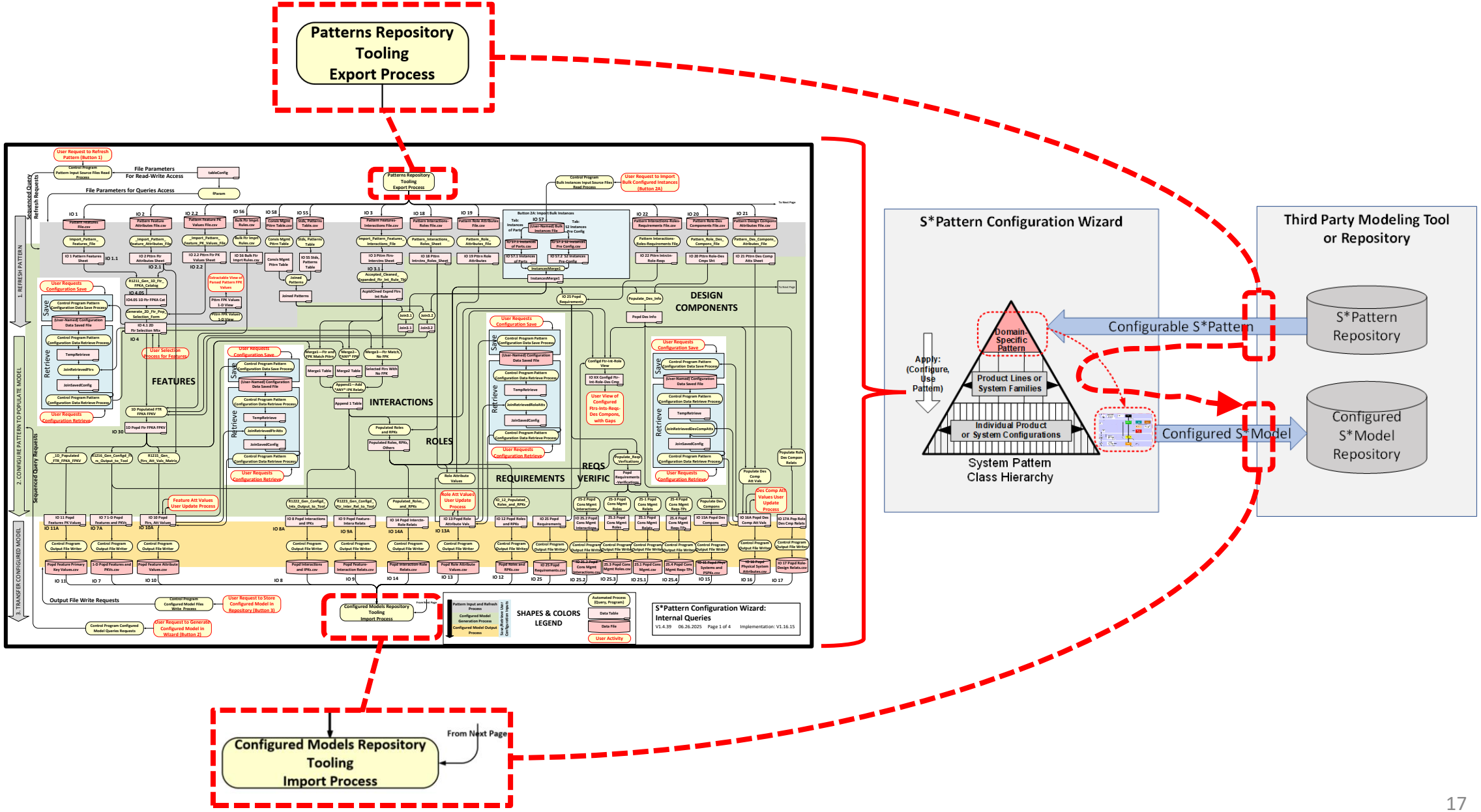
https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:is2025_fellows_27_july_v1.2.1.pdf

Overview of the software implementation

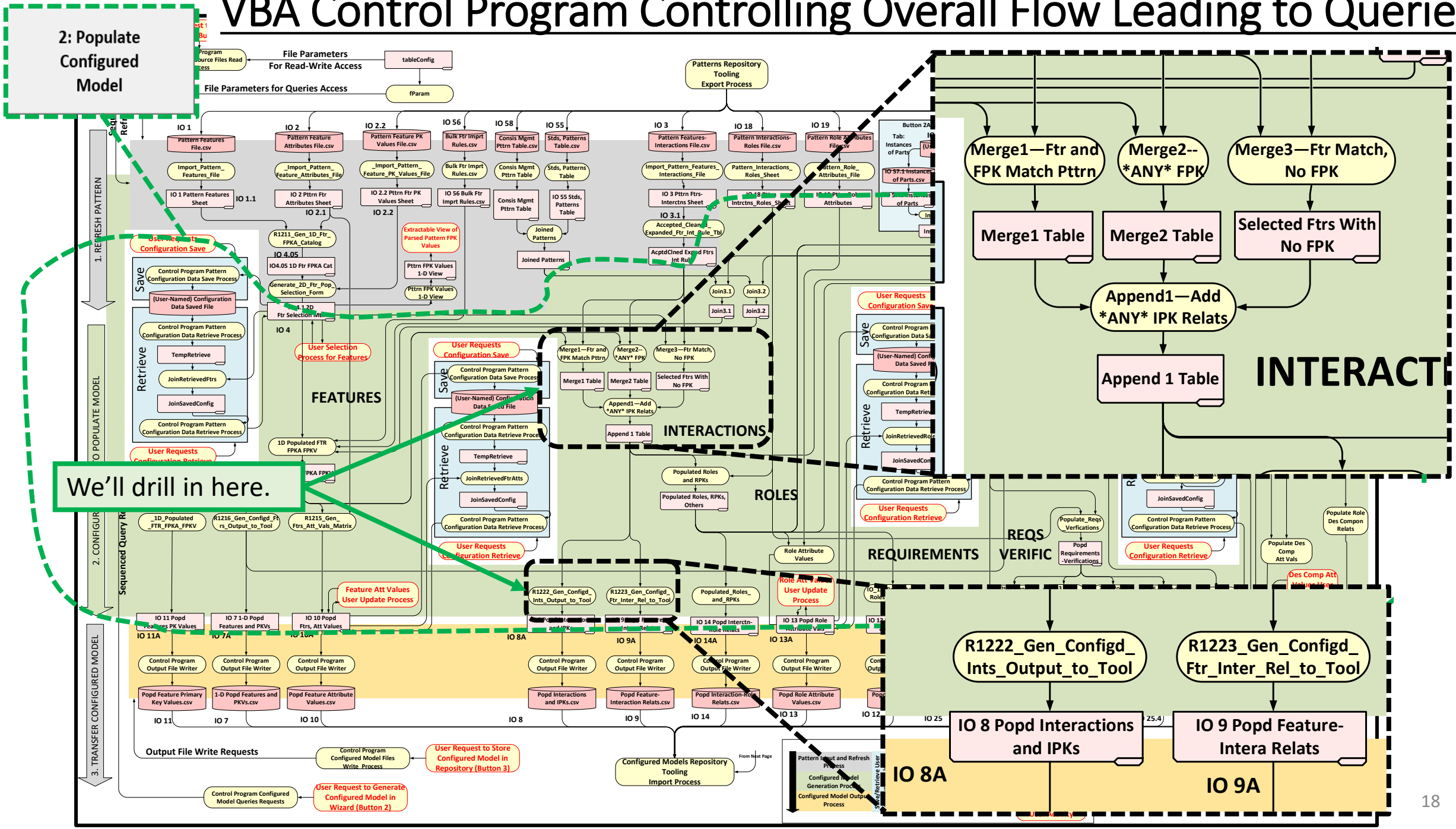
- S*Pattern Configuration Process: VBA control and M-Code queries
- S*Models Inconsistency Detection Process: VBA control and M-Code queries



Conf. Wizard VBA Control Program Controlling Overall Flow Leading to Queries

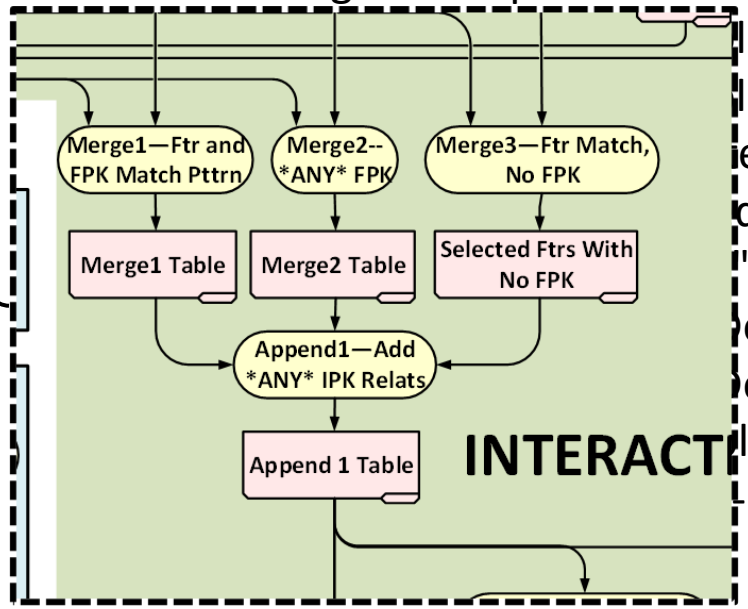


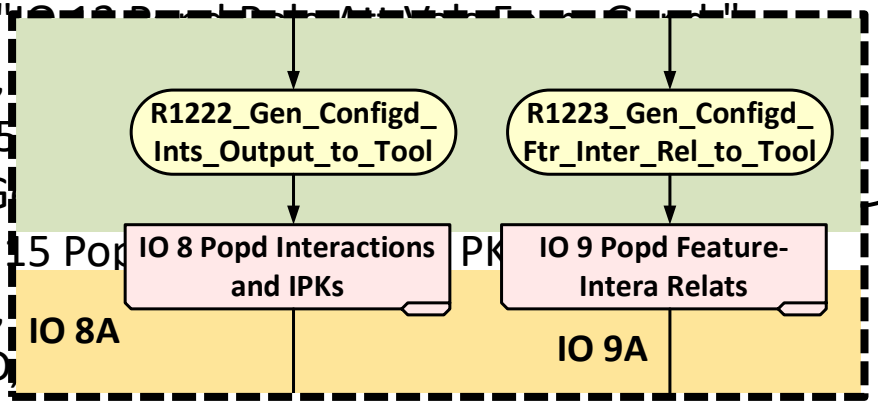
VBA Control Program Controlling Overall Flow Leading to Queries



VBA Control Program Controlling Overall Flow Leading to Queries

```
RefreshAndProgress "1D Popd Ftr FPKA FPKV", RowID, ColID, "1D Confgd Feature Population Gend:"
RefreshAndProgress "Merge 1 Table", RowID, ColID, "Group 1 Interactions Populated:"
RefreshAndProgress "Merge 2 Table", RowID, ColID, "Group 2 Interactions Populated:"
RefreshAndProgress "Selected Ftrs with No FPK", RowID, ColID, "Group 3 Interactions Populated:"          ' Merge 3
RefreshAndProgress "Append 1 Table", RowID, ColID, "Combined Interactions Populated:"
RefreshAndProgress "IO 7 1-D Popd Features and PKVs", RowID, ColID, "Popd Ftrs Sheet Generated:"          ' R1216
RefreshAndProgress "IO 8 Popd Interactions and IPKs", RowID, ColID, "Popd Intrcns Sheet Generated:"       ' R1222
RefreshAndProgress "IO 9 Popd Feature-Intera Relats", RowID, ColID, "Popd Ftr-Ints Sheet Generated:"      ' R1223
RefreshAndProgress "IO 10 Popd Ftrs, Att Values", RowID, ColID, "Popd Ftr Att Values Sheet Generated:"    ' R1215
RefreshAndProgress "IO 11 Popd Feature PK Values", RowID, ColID, "Popd Ftr PK Values Sheet Generated:"  ' 1D Popd
RefreshAndProgress "Populated Roles, RPKs, Others", RowID, ColID, "Popd Roles , RPKs, Others Generated:"
les and RPKs", RowID, ColID, "IO 12 Popd Roles Sheet Generated:"
le Attribute Vals", RowID, ColID, "
erctn-Role Relats", RowID, ColID,
quirements", RowID, ColID, "IO25
", RowID, ColID, "Popd Des Info G
es Compons", RowID, ColID, "IO 15 Pop
es Comp Att Vals", RowID, ColID,
le Des Cmp Relats", RowID, ColID,
-Role View", RowID, ColID, "Confgd Ftr-Int-Role View"
```





VBA Control Program Directly Invoking a Single Query and Logging Progress

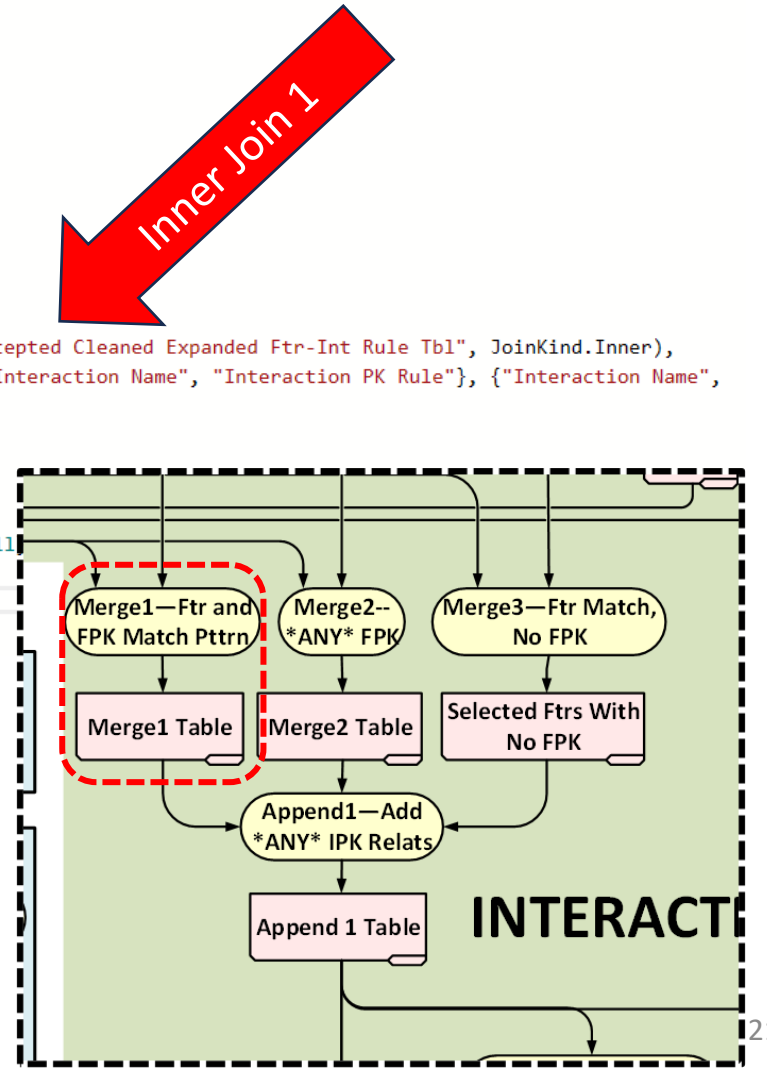
```
'-------------------------------------------
'
'  Routine to refresh a query and log progress on it, including items (rows) counts to log
'
'-------------------------------------------
Sub RefreshAndProgress(SheetName, LogRowID, LogColID, Message)
On Error GoTo QueryError1    ' Detect and trap errors in DB Table queries in what follows
ThisWorkbook.Worksheets(SheetName).ListObjects(1).QueryTable.BackgroundQuery = False
If Not ThisWorkbook.Worksheets(SheetName).ListObjects(1).DataBodyRange Is Nothing Then
  ThisWorkbook.Worksheets(SheetName).ListObjects(1).DataBodyRange.ClearContents 'Clear previous content of table, in case query fails
End If
ThisWorkbook.Worksheets(SheetName).ListObjects(1).QueryTable.Refresh  ' Perform a query; see separate query code
Set LstObj = ThisWorkbook.Sheets(SheetName).ListObjects(1)  '  Retrieve the quantity of objects in the query table
numb = LstObj.ListRows.Count  ' This is the quantity of objects, retrieved for logging
ProgressLog LogRowID, LogColID, Message, numb  ' Generate progress log entry for this query
GoTo ExitRefresh
QueryError1: ProgressLog LogRowID, LogColID, ("***ERROR: " & Message), 0 ' Log error message
ExitRefresh: On Error GoTo 0  ' Clear query error trapping
End Sub
```

M-Code Query “Merge 1”: Inner Join to populate Interactions from Exact Matchup of [Populated Features & FPK Values] versus [Feature-Interaction Rules]

Merge1

Display Option

```
////////////////////////////////////////////////////////////////////////////
//
//  Query to populate interactions for configured model from configured features.
//       This query is for the subset in which the FPK value matches an explicit value in a configuration rule.
//       See other queries for *ANY* wild card case.
//
////////////////////////////////////////////////////////////////////////////
//
//  Perfor inner join of populated interactions of configured model with feature-interations configuration rules from pattern
//        The join match is on the concatenated combination of feature name and FPK value
//
let
    Source = Table.NestedJoin(#"1D Populated FTR, FPKA, FPKV", {"Ftr-FPKV"}, #"Accepted Cleaned Expanded Ftr-Int Rule Tbl", {"Ftr-FPKV"}, "Accepted Cleaned Expanded Ftr-Int Rule Tbl", JoinKind.Inner),
    #"Expanded Accepted Cleaned Expanded Ftr-Int Rule Tbl" = Table.ExpandTableColumn(Source, "Accepted Cleaned Expanded Ftr-Int Rule Tbl", {"Interaction Name", "Interaction PK Rule"}, {"Interaction Name",
        "Interaction PK Rule"}),
//
//   Remove blank and error rows.
//
    #"Removed Errors" = Table.RemoveRowsWithErrors(#"Expanded Accepted Cleaned Expanded Ftr-Int Rule Tbl", {"Feature Name"}),
    #"Removed Blank Rows" = Table.SelectRows(#"Removed Errors", each not List.IsEmpty(List.RemoveMatchingItems(Record.FieldValues(_), {"", null
in
    #"Removed Blank Rows"
```

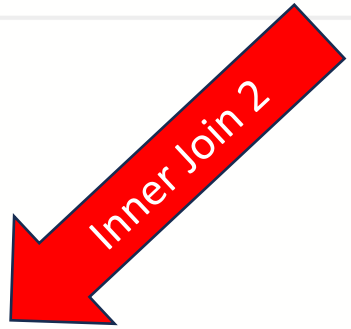


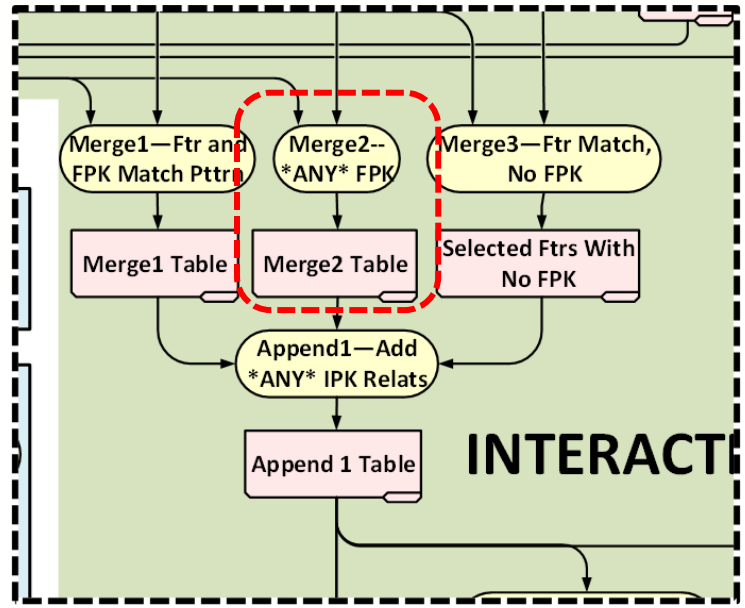
M-Code Query “Merge 2”: Inner Join to populate Interactions from Matchup of [Populated Features & FPK Values] versus [Feature-Interaction Rules with FPK Rule=*ANY*]

Merge2

Display Options

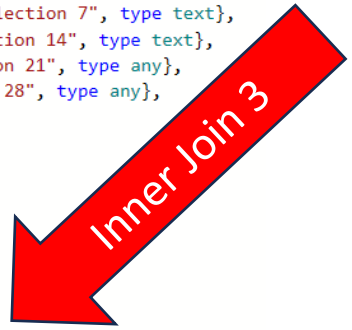
```
////////////////////////////////////////////////////////////////////////////
//
//  Query for second type of interaction population from feature population.
//    This type is for FPK Value confguration rule of *ANY*, for which case
//    the only matching necessary between the populated features and the
//    configuration rules is that the feature name matches--no need to check FPK value.
//
////////////////////////////////////////////////////////////////////////////
//
//  Form inner join of populated features and configuration rules, in which
//    the match criteria is limited to the feature name
//
let
    Source = Table.NestedJoin(#"1D Populated FTR, FPKA, FPKV", {"Feature Name"}, #"Accepted Cleaned Expanded Ftr-Int Rule Tbl", {"Feature Name"}, "Accepted Cleaned Expanded Ftr-Int Rule Tbl", JoinKind.Inner),
    #"Expanded Accepted Cleaned Expanded Ftr-Int Rule Tbl1" = Table.ExpandTableColumn(Source, "Accepted Cleaned Expanded Ftr-Int Rule Tbl", {"FPK Value", "Interaction Name", "Interaction PK Rule"}, {"FPK
        Value.1", "Interaction Name", "Interaction PK Rule"}),
//
//  Filter our all rows except for those with FPK Value configuration rule of *ANY*
//
    #"Filtered Rows" = Table.SelectRows(#"Expanded Accepted Cleaned Expanded Ftr-Int Rule Tbl1", each ([FPK Value.1] = "*ANY*")),
//
//   Remove unneeded output columns.
//
    #"Removed Columns" = Table.RemoveColumns(#"Filtered Rows",{"FPK Value.1"})
in
    #"Removed Columns"
```

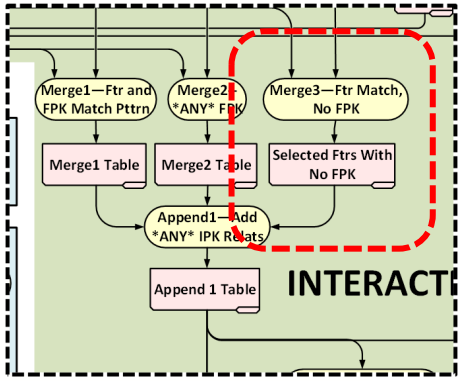




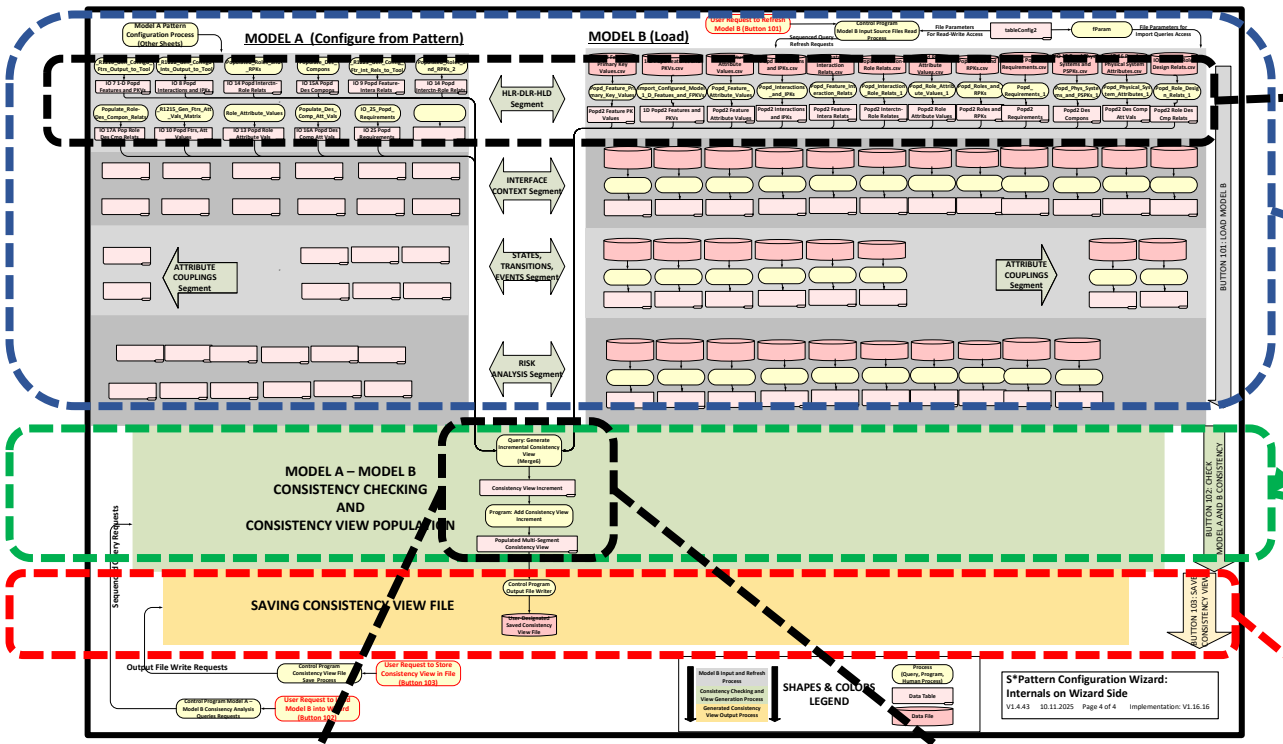
M-Code Query "Merge 2": Inner Join to populate Interactions from Matchup of [Features with no FPK Attribute] versus [Feature-Interaction Rules]

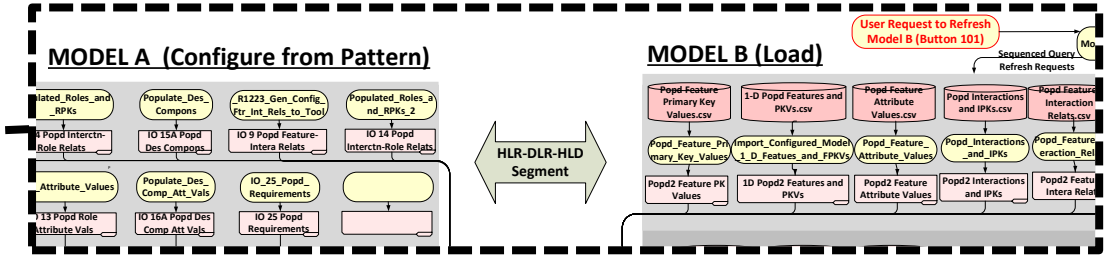
```
//////////
//
//  Query to form third set of populated interactions from populated features and pattern rules.
//    This third set is for features that do not have any FPK attribute, so therefore
//    do not have any conditional rule other than feature-interaction pair appears in
//    the configuration rules. And,the user must have entered "yes" to populate feature.
//
////////////////////////////////////////////////////////////////////////////////////////
//
//  Begin with user-selected entries in 2-D feature population form; assure text type data; remove unneeded cols
//
let
    Source = Excel.CurrentWorkbook(){[Name="Generate_2D_Ftr_Pop_Selection_Form_2"]}[Content],
    #"Changed Type" = Table.TransformColumnTypes(Source,{{"Feature Configuration Rule", type text}, {"Feature Name", type text}, {"Feature Attribute", type text}, {"Feature PK Att Value.1", type text}, {"Feature PK Att Value.2", type text}, {"Feature PK Att Value.3", type text}, {"Feature PK Att Value.4", type text}, {"Feature PK Att Value.5", type text}, {"Feature PK Att Value.6", type text}, {"Feature PK Att Value.7", type text}, {"Feature PK Att Value.8", type text}, {"Feature PK Att Value.9", type text}, {"Feature PK Att Value.10", type text}, {"Feature PK Att Value.11", type text}, {"Feature PK Att Value.12", type text}, {"Feature PK Att Value.13", type text}, {"Feature PK Att Value.14", type text}, {"Feature PK Att Value.15", type text}, {"Feature PK Att Value.16", type text}, {"Feature PK Att Value.17", type text}, {"Feature PK Att Value.18", type any}, {"Feature PK Att Value.19", type any}, {"Feature PK Att Value.20", type any}, {"Feature PK Att Value.21", type any}, {"Feature PK Att Value.22", type any}, {"Feature PK Att Value.23", type any}, {"Feature PK Att Value.24", type any}, {"Feature PK Att Value.25", type any}, {"Feature PK Att Value.26", type any}, {"Feature PK Att Value.27", type any}, {"Feature PK Att Value.28", type any}, {"Feature PK Att Value.29", type any}, {"Feature PK Att Value.30", type any}, {"Feature PK Att Value.31", type any}, {"Populate? Yes/No", type text}, {"Selection 1", type text}, {"Selection 2", type text}, {"Selection 3", type text}, {"Selection 4", type text}, {"Selection 5", type text}, {"Selection 6", type text}, {"Selection 7", type text}, {"Selection 8", type text}, {"Selection 9", type text}, {"Selection 10", type text}, {"Selection 11", type text}, {"Selection 12", type text}, {"Selection 13", type text}, {"Selection 14", type text}, {"Selection 15", type text}, {"Selection 16", type text}, {"Selection 17", type any}, {"Selection 18", type any}, {"Selection 19", type any}, {"Selection 20", type any}, {"Selection 21", type any}, {"Selection 22", type any}, {"Selection 23", type any}, {"Selection 24", type any}, {"Selection 25", type any}, {"Selection 26", type any}, {"Selection 27", type any}, {"Selection 28", type any}, {"Selection 29", type any}, {"Selection 30", type any}, {"Selection 31", type any}}),
    #"Removed Other Columns" = Table.SelectColumns(#"Changed Type",{"Populate? Yes/No", "Feature Name", "Feature Attribute"}),
//
//  Select features which have no FPK attribute and for which pattern user indicated "Yes" to populate
//
    #"Filtered Rows" = Table.SelectRows(#"Removed Other Columns", each ([Feature Attribute] = null)),
    #"Filtered Rows1" = Table.SelectRows(#"Filtered Rows", each [#"Populate? Yes/No"] = "Yes"),
//
//  For remaining rows, use inner join to identify interactions to be populated
//
    #"Merged Queries" = Table.NestedJoin(#"Filtered Rows1", {"Feature Name"}, #"Accepted Cleaned Expanded Ftr-Int Rule Tbl", {"Feature Name"}, "Accepted Cleaned Expanded Ftr-Int Rule Tbl", JoinKind.Inner),
    #"Expanded Accepted Cleaned Expanded Ftr-Int Rule Tbl1" = Table.ExpandTableColumn(#"Merged Queries", "Accepted Cleaned Expanded Ftr-Int Rule Tbl", {"Interaction Name", "Interaction PK Rule"}, {"Interaction Name", "Interaction PK Rule"}),
//
//  Remove columns no longer needed; add cols for FPK Value and for concatenated string Ftr-FPKV
//
    #"Removed Columns" = Table.RemoveColumns(#"Expanded Accepted Cleaned Expanded Ftr-Int Rule Tbl1",{"Populate? Yes/No"}),
    #"Added Custom" = Table.AddColumn(#"Removed Columns", "FPK Value", each null),
    #"Added Custom1" = Table.AddColumn(#"Added Custom", "Ftr-FPKV", each null),
//
//  Reorder columns to align with multiple table combine that follows this query; remove blank and error rows
//
    #"Reordered Columns" = Table.ReorderColumns(#"Added Custom1",{"Feature Name", "Feature Attribute", "FPK Value", "Ftr-FPKV", "Interaction Name", "Interaction PK Rule"}),
    #"Removed Blank Rows" = Table.SelectRows(#"Reordered Columns", each not List.IsEmpty(List.RemoveMatchingItems(Record.FieldValues(_), {"", null}))),
    #"Removed Errors" = Table.RemoveRowsWithErrors(#"Removed Blank Rows", {"Ftr-FPKV"})
in
    #"Removed Errors"
```





The S*Models Inconsistency Detection Process: Flow and Code Sample Walk-Through



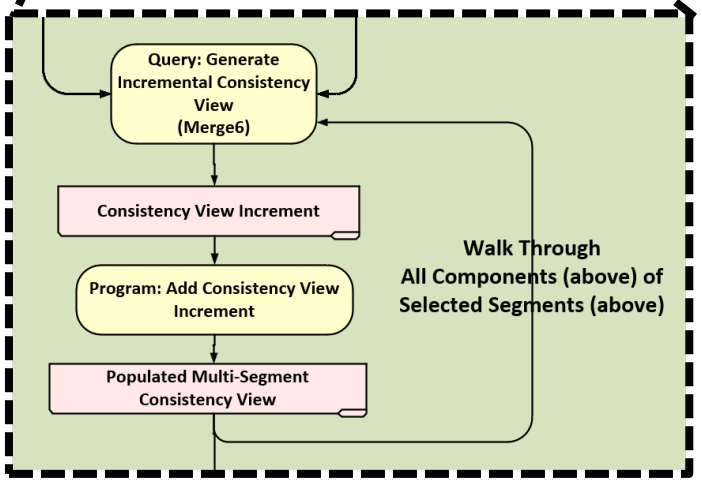


We'll drill in here.

101: Load Model B to Compare

102: Perform Consistency Check

Button 103: Save Consistency View



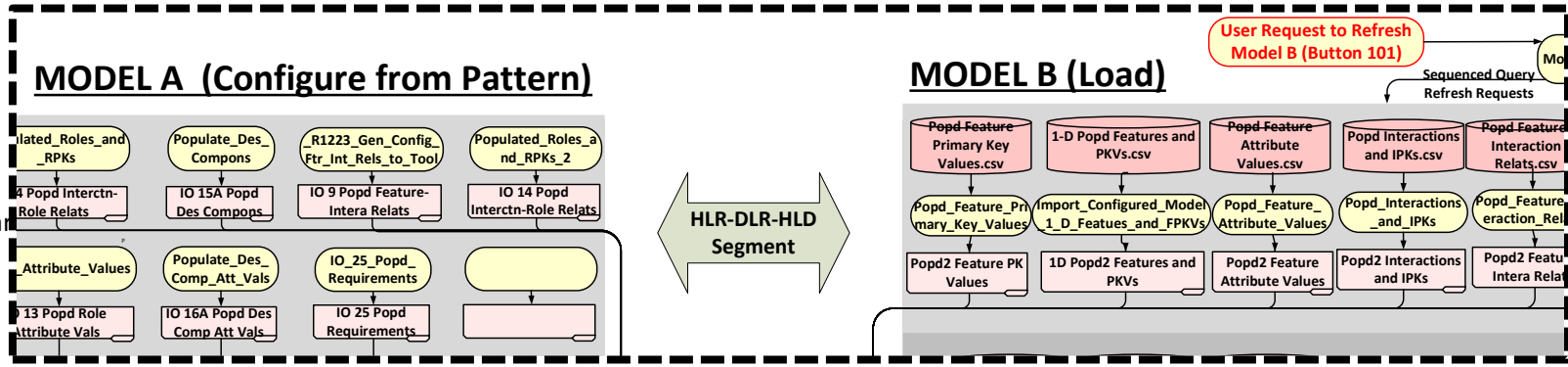
Anti-Joins and the Inconsistency Detection Algorithm

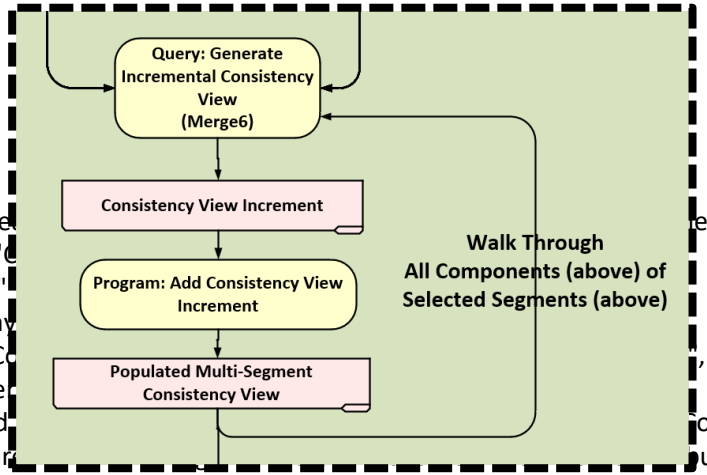
- In comparing two models for their consistency with each other, the Pattern Configuration Wizard uses M-Code Anti-Joins.
- This rapidly detects content of either model that is not in the content of the other model.
- It also provides information needed to calculate a “cosine distance between the two models”.

VBA Control Program Controlling Overall Flow Leading to Queries

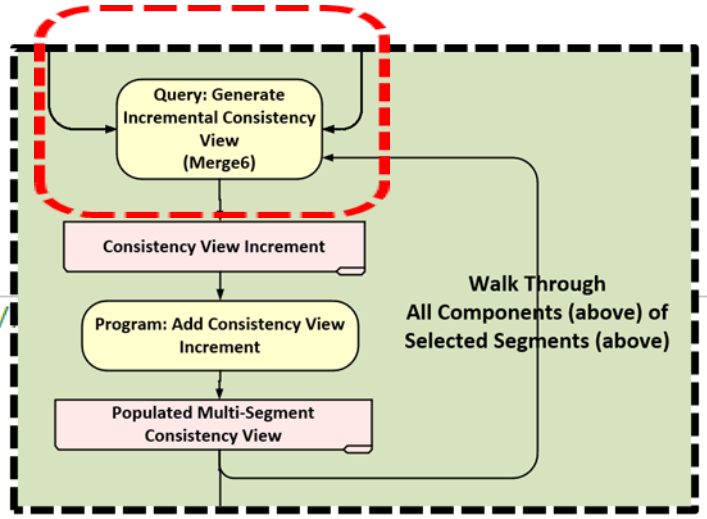
```
'
'-------------------------------------------------
Sub Macro105()
FirstRowID = 8      'Set range of logging to be cleared before process starts.
LastRowID = 134
ColID = "F"
RowID = 8
ClearProgressLog2 FirstRowID, LastRowID, ColID   ' Clear the progress log before star
ProgressLog2 RowID, ColID, "Start Model Compare:", 0   ' Generate first log entry.
'
'  Initialize starting row for incremental build-up of consistency view table rows
'
DestinationStartingRowx = 2
'
'  Clear the consistency view contents before starting the consistency checks.
'
ClearConsistencyView
'
'  Perform series of incremental consistency checks and view generation of segments of consistency view
'
PerformConsistencyCheck "FeaturesConsistencyCheck", "_R1216_Gen_Configd_Ftrs_Output_to_Tool", "Import_Configured_Model_1_D_Fe                                                    eCol"
PerformConsistencyCheck "InteractionsConsistencyCheck", "_R1222_Gen_Configd_Ints_Output_to_Tool", "Popd_Interactions_and_IPKs", "
PerformConsistencyCheck "RolesConsistencyCheck", "Populated_Roles_and_RPKs", "Popd_Roles_and_RPKs", "Configured Role Name", "", "
PerformConsistencyCheck "DesComponsConsisCheck", "Populate_Des_Compons", "Popd_Physical_Systems_and_PSPKs_1", "Configured Phy
PerformConsistencyCheck "Ftr-IntrctnRelatsCheck", "_R1223_Gen_Configd_Ftr_Int_Rels_to_Tool", "Popd2_Feature_Interaction_Relats", "C                                                    , "TwoCols"
PerformConsistencyCheck "Intrctn-RoleRelatsCheck", "Populated_Roles_and_RPKs_2", "Popd_Interaction_Role_Relats_1", "Configured Inte
PerformConsistencyCheck "Role-DesCompRelatsCheck", "Populate_Role_Des_Compon_Relats", "Popd_Role_Design_Relats_1", "Configured                                                    ols"
PerformConsistencyCheck "Ftr-Att-ValsCheck", "_R1215_Gen_Ftrs_Att_Vals_Matrix", "Popd_Feature_Attribute_Values", "Configured Featur                                                    ute Value",
"ThreeCols"
PerformConsistencyCheck "Role-Att-ValsCheck", "Role_Attribute_Values", "Popd_Role_Attribute_Values_1", "Configured Role Name", "Configured Role Attribute Name", "Configured Role Attribute Value", "ThreeCols"
PerformConsistencyCheck "DesComp-Att-ValsCheck", "Populate_Des_Comp_Att_Vals", "Popd_Physical_System_Attributes_1", "Configured Physical System Name", "Configured Physical System Attribute", "Configured
Physical System Attribute Value", "ThreeCols"
PerformConsistencyCheck "Intrctn-RTF-ReqmntCheck", "IO_25_Popd_Requirements", "Popd_Requirements_1", "Owning Interaction", "Configured RTF, Qualified Name", "Requirement Statement", "ThreeCols"
' Case not included above because comparison may be redundant with another:  Popd Feature PK Values

End Sub
```



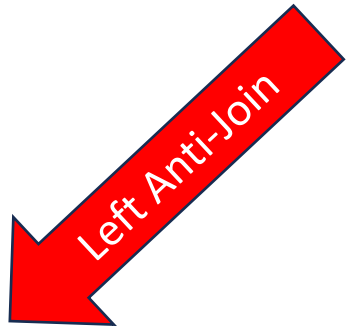


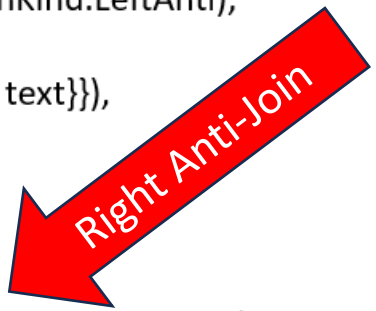
M-Code Query “Merge 6”: Anti-Joins to populate Table of Rows In Model A and not in Model B, or Rows in Model B but not Model A, and Compute Cosine Distance Between Model A and Model B




```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//  Query performed to check consistency of two model segments (of Models A and B, each in a table),
//               using anti-joins to list only inconsistencies.
//
//          Parameters passed to this query are:
//              Parameter0 is the type of consistency check in terms of what model segments are checked.
//              Parameter1 is name of Model A segment table to compare
//              Parameter2 is name of Model B segment table to compare
//              Parameter3 is name of the first table column in (both) Model A and B to check for consistency with each other
//              Parameter4 is name of the second table column (if any) in (both) Model A and B to check for consistency with each other
//              Parameter5 is name of the third table column (if any) in (both) Model A and B to check for consistency with each other
//              Parameter6 is the number of model columns to compare (OneCol, TwoCols, or ThreeCols)
//
//          Output of this query is a table showing any differences between the models, and their inconsistency distance score.
//            Note this comparison disregards (indeed, removes) deep path names to model class owners, but includes model metaclass relationships.
//
//   WARNING: Parameter values are passed to this query by VBA code that inserts values into parameter strings below.
//            Points of parameter insertion are detected by VBA finding locations in m-code below using following convention.
//            Inserts parameter values into first query text string enclosed in quotes (") that follows a preceding
//                substring in quotes which is of the form "Paramx" = "Insert Value Here"; for example,
//                #"Param9" = #"TableName"  or   #"Param9" = "String Value"
//
//            Also requires setting Power Query's Query Options to Ignore Privacy Levels.
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
let
//  Query parameter values get inserted here (right side of =) by VBA parameter passing routine:
```

M-Code Query "Merge 6": Anti-Joins to populate Table of Rows In Model A and not in Model B, or Rows in Model B but not Model A, and Compute Cosine Distance Between Model A and Model B

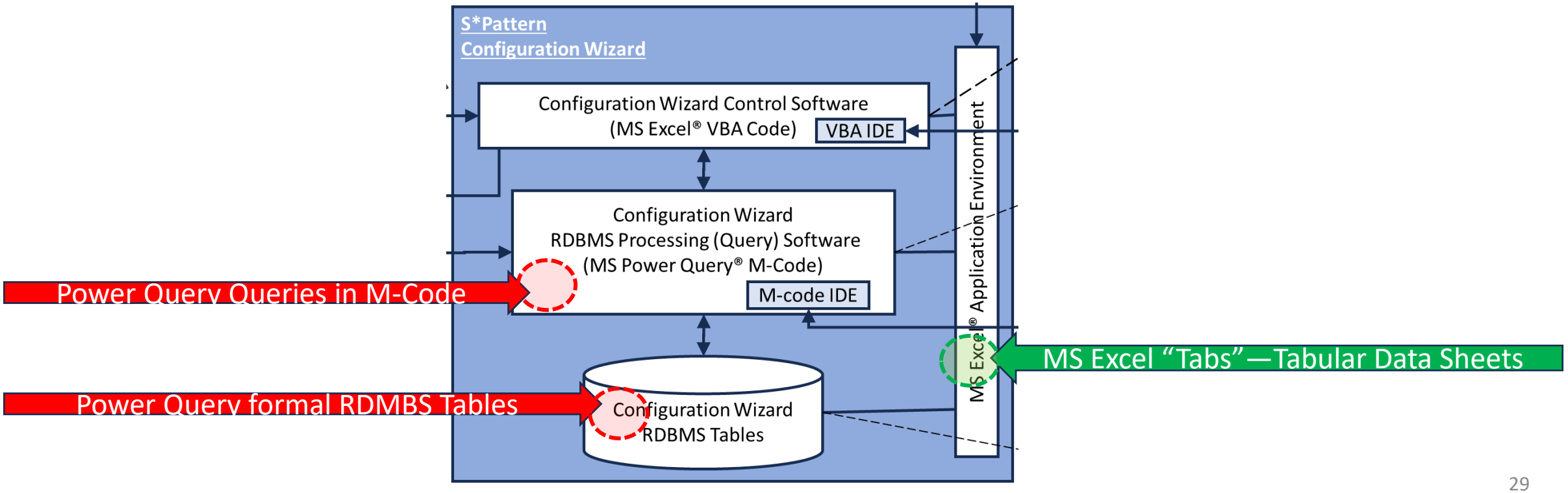





```
//
//  Now for Model A and B together, find rows in Model B that are not in Model A, by anti-join; Count them.
//
   #"Merged Queries1" = Table.NestedJoin(#"Model B", {"Concatenated Name"}, #"Model A", {"Concatenated Name"}, "NewCol", JoinKind.LeftAnti),
   #"In B Not A" = Table.ExpandTableColumn(#"Merged Queries1", "NewCol", {"Concatenated Name"}, {"Concatenated Name.1"}),
   #"Changed Type" = Table.TransformColumnTypes(#"In B Not A",{{"Concatenated Name", type text}, {"Concatenated Name.1", type text}}),
#"In B Not A Row Count" = Table.RowCount(#"Changed Type"),
//
//  Finds rows in Model A that are not in Model B, by anti-join; Count them.
//
   #"Merged Queries2" = Table.NestedJoin(#"Model B", {"Concatenated Name"}, #"Model A", {"Concatenated Name"}, "NewCol", JoinKind.RightAnti),
   #"In A Not B" = Table.ExpandTableColumn(#"Merged Queries2", "NewCol", {"Concatenated Name"}, {"NewCol.Concatenated Name"}),
   #"Changed Type1" = Table.TransformColumnTypes(#"In A Not B",{{"Concatenated Name", type text}, {"NewCol.Concatenated Name", type text}}),
   #"Renamed Columns1" = Table.RenameColumns(#"Changed Type1",{{"NewCol.Concatenated Name", "Concatenated Name.1"}}),
   #"In A Not B Row Count" = Table.RowCount(#"Renamed Columns1"),
//  Calculate consistency score, based on cosine distance between two model vectors (0-1, with 1 indicating complete consistency, 0 for complete
inconsistency)
   #"Cosine Distance Score" = ((#"Model A Row Count"-#"In B Not A Row Count") / Number.Sqrt( #"Model A Row Count"*#"Model B Row Count")),
//  Combine above two cases into one table, with nulls where no equivalent in other column
   #"Combined" = Table.Combine({ #"Renamed Columns1" ,  #"Changed Type" }),
```

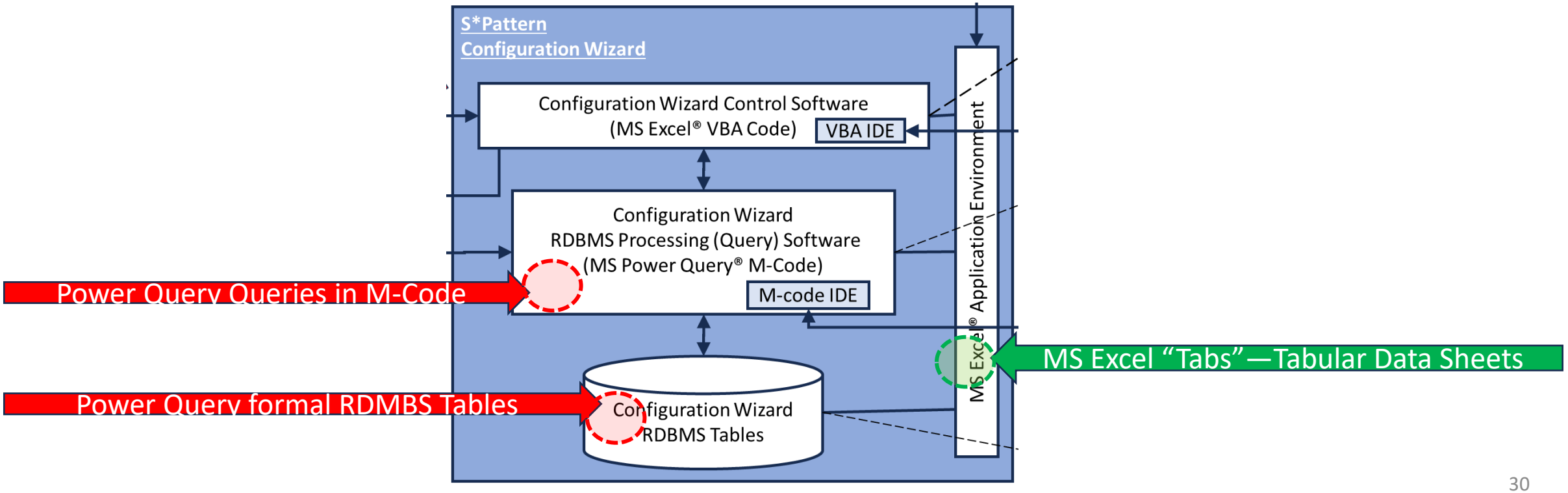
Power Query Queries and Tables; Excel Tabs

- Don’t confuse Power Query RDBMS formal **Tables** with the **Excel “tabs”** (tabular rows/columns spreadsheets).
- A Query produces data in its Power Query Table, and consumes data from other Power Query tables or external sources.
- At the end of a Query, the resulting data in a Power Query Table is “loaded” into a visible Excel spreadsheet.
- The Power Query Tables ***ARE NOT*** the Excel Spreadsheets!

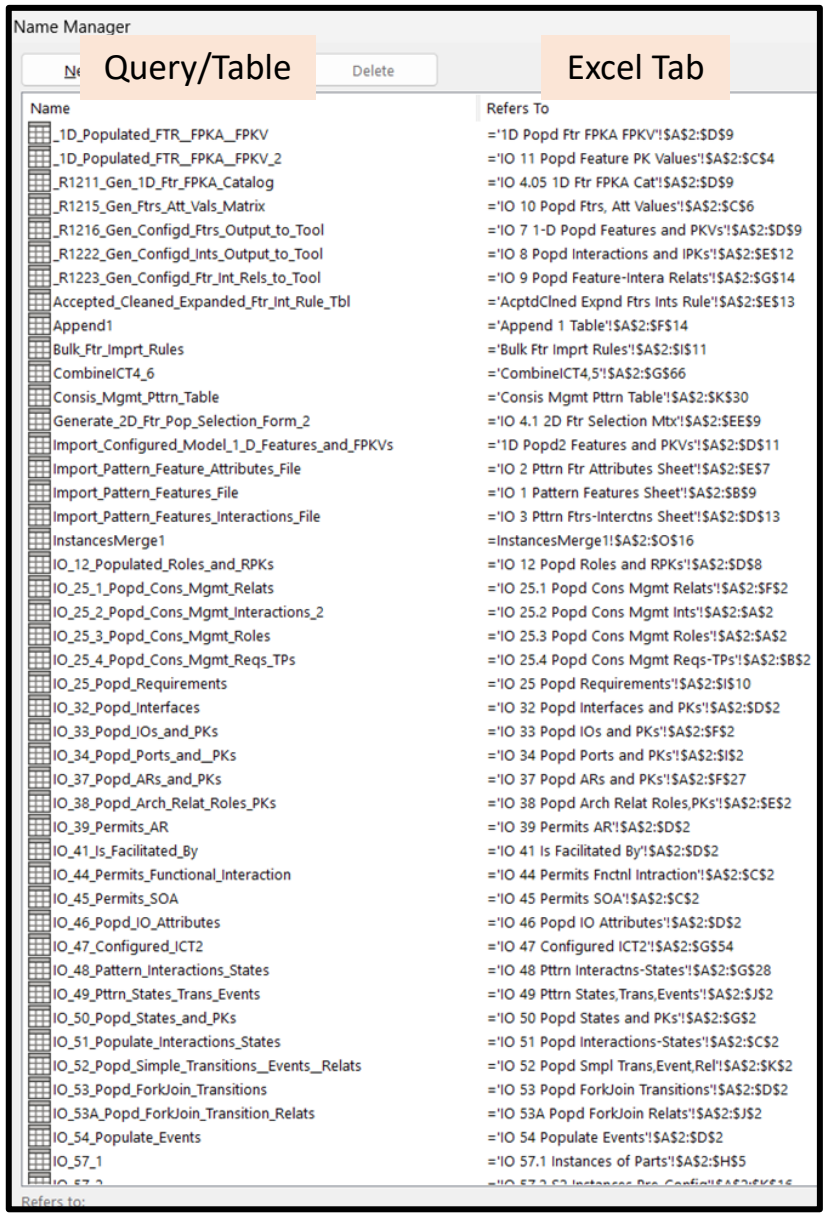


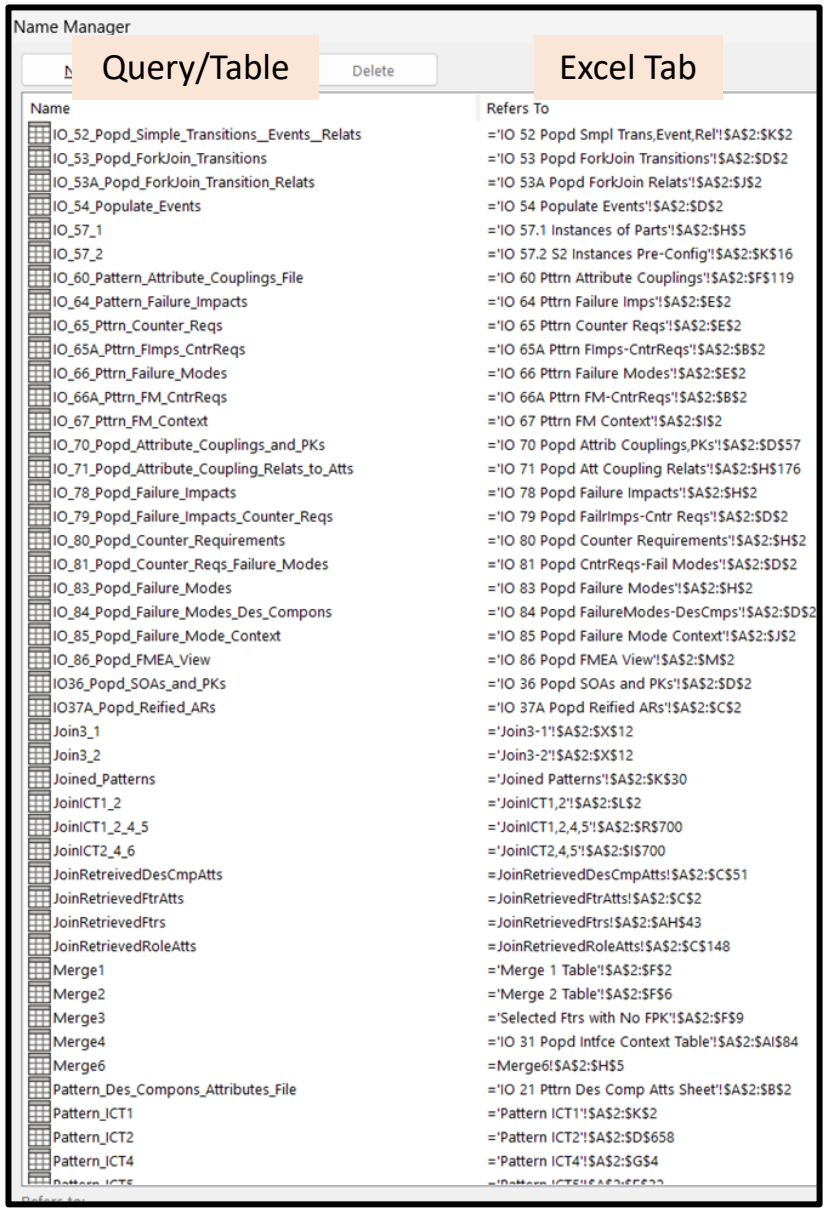
Power Query Queries and Tables; Excel Tabs

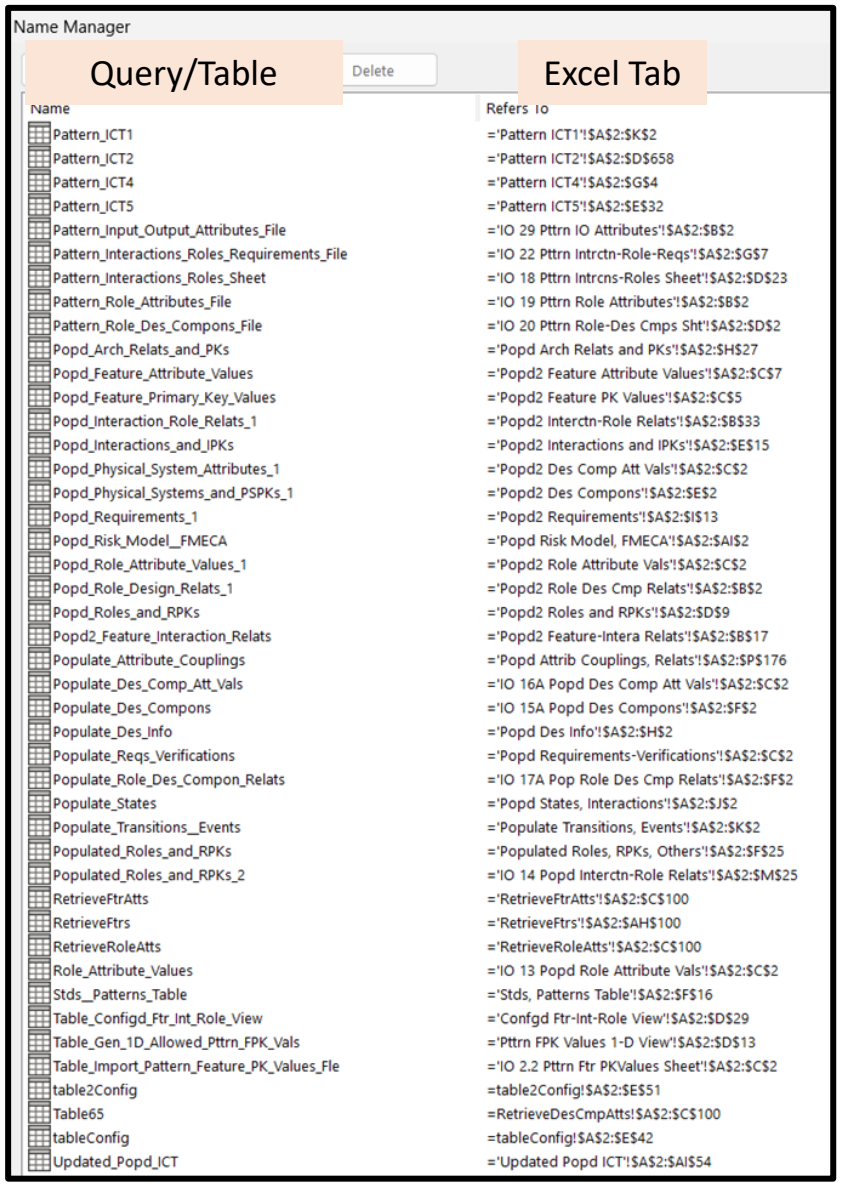
- Each named Query (in M-Code) “goes with” an identically named RDBMS Table.
- Each such RDBMS Table is associated with an MS Excel “tab” spreadsheet that is independently named, and will contain the results of the (high speed) query, after Excel “catches up” with the faster RDBMS tables.
- There is a listing of what named Queries/RDBMS Tables are associated with what Excel “tabs”.



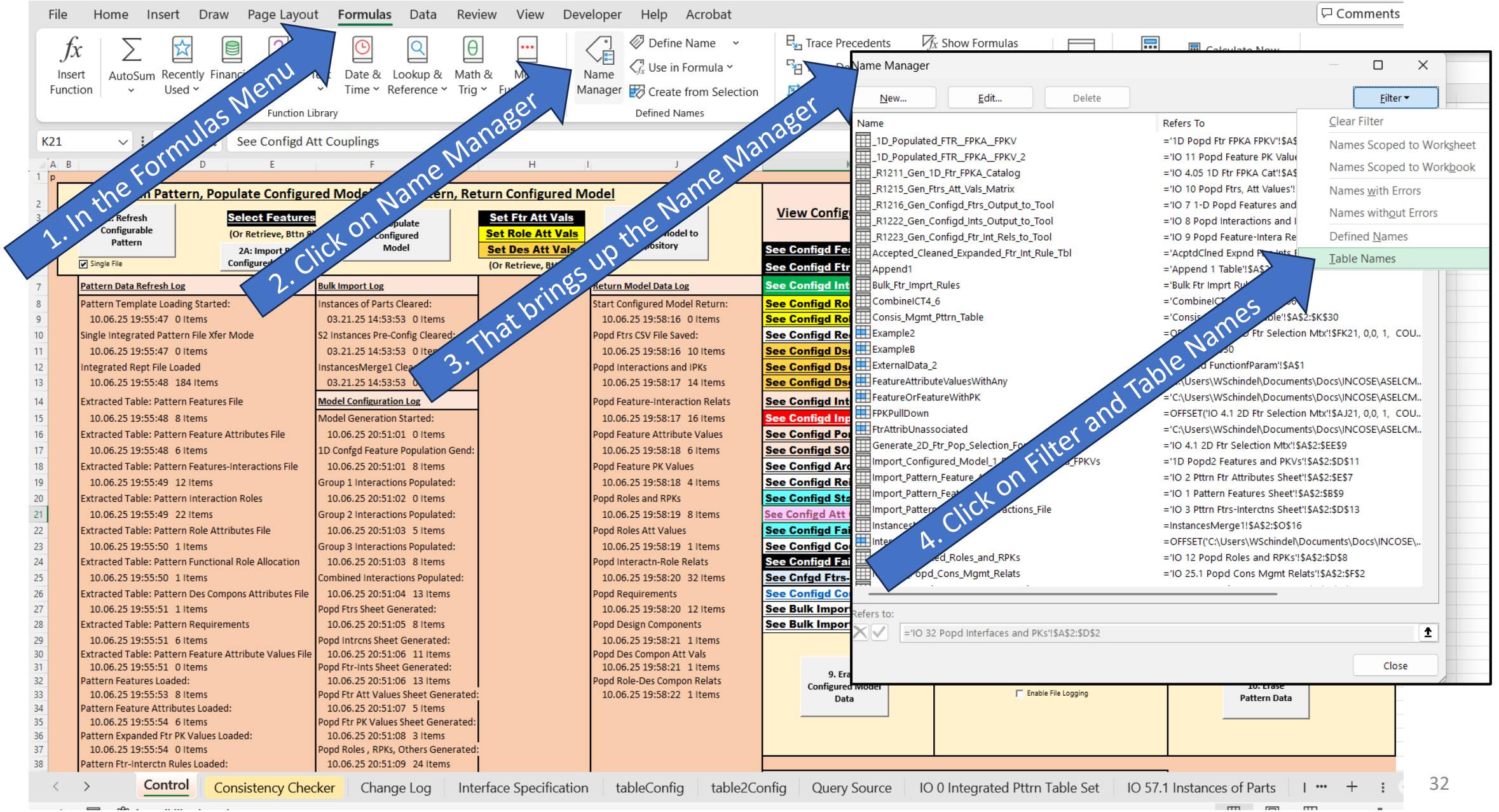
The S*Configuration Wizard's (~127) Queries/Tables and their Associated Excel Tabs







How to generate that list of Query Tables and Tabs Yourself



Development environments and aids: M-Code IDE and VBA IDE

- The MS Excel application development environment includes Integrated Development Environment (IDE) tooling for both the Excel Power Query M-Code query programming environment and for the Excel VBA programming environments.
- The M-Code IDE includes extensive automated M-Code generation as well as tracing and other debugging aids.
- The VBA IDE includes extensive program debugging aids.
- Both languages and the Excel Object Model include large libraries of functions and extensive online documentation.

Filter by title

Power Query M formula language
Quick tour of the Power Query M formula language
Power Query M language specification
Power Query M type system
Expressions, values, and let expression
Comments
Evaluation model
Operators
Types and type conversion
Metadata
Errors
Date and time
Text formatting
Functions
Enumerations
Constants
Dynamic values

Download PDF

Power Query M formula language

Microsoft Power Query provides a powerful data import experience that encompasses many features. Power Query works with desktop Analysis Services, Excel, and Power BI workbooks, in addition to many online services, such as Fabric, Power BI service, Power Apps, Microsoft 365 Customer Insights, and more. A core capability of Power Query is to filter and combine, that is, to mash-up data from one or more of a rich collection of supported data sources. Any such data mashup is expressed using the Power Query M formula language. The M language is a functional, case sensitive language similar to F#.

Get started

QUICKSTART

M quick tour

CONCEPT

Expressions, values, and let expression

Evaluation model

Types and type conversion

How culture affects text formatting

TRAINING

Power Query documentation

Understanding Power Query M functions

Specification

ARCHITECTURE

Power Query M language specification

Power Query M type system

Reference

REFERENCE

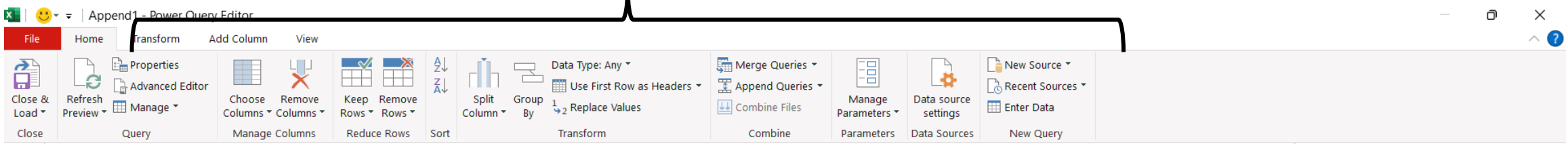
Power Query M function reference

Power Query M enumeration reference

Power Query M constant reference

Power Query M dynamic value reference

M-Code Auto Generation Buttons (Home)



Queries

```
= Table.ReplaceValue(#"Removed Blank Rows1",null,"",Replacer.ReplaceValue,{"IPK Value", "FPK Value"})
```

	Interaction Name	Interaction PK Rule	IPK Value	Configured Interaction Name	Feature Name	FPK Value
1	Purchase Device			Purchase Device	Ease of Purchase	
2	Display Status and Information			Display Status and Information	Ease of Use	
3	Handle Device			Handle Device	Ease of Use	
4	Handle Device			Handle Device	Portability	
5	Consume Electrical Power			Consume Electrical Power	Power Mains Compatibility	
6	Convert Electrical Power	FPK	Power Output 1	Convert Electrical Power [Power Output 1]	Powered Devices Compatibility	Power Output 1
7	Set Configuration	FPK	Power Output 1	Set Configuration [Power Output 1]	Powered Devices Compatibility	Power Output 1
8	Convert Electrical Power	FPK	Power Output 2	Convert Electrical Power [Power Output 2]	Powered Devices Compatibility	Power Output 2
9	Set Configuration	FPK	Power Output 2	Set Configuration [Power Output 2]	Powered Devices Compatibility	Power Output 2
10	Provide Warranty Data			Provide Warranty Data	Reliability and Durability	
11	Resist Contamination			Resist Contamination	Reliability and Durability	
12	Tolerate Moisture			Tolerate Moisture	Reliability and Durability	
13	Handle Device			Handle Device	Safety	Electrical Shock

Live display of (changing) table during query trace

Query Settings

PROPERTIES

Name

Append1

All Properties

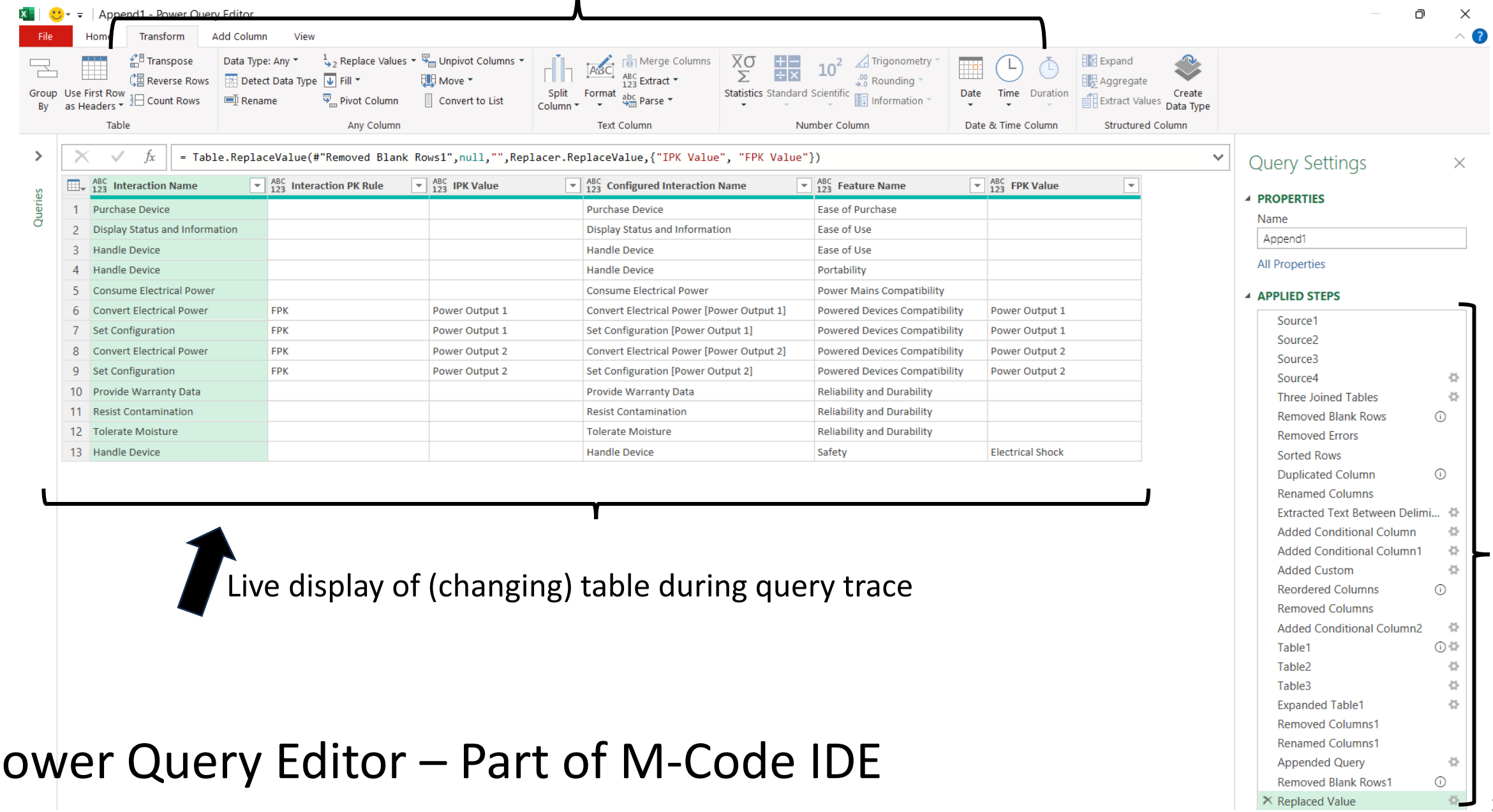
APPLIED STEPS

- Source1
- Source2
- Source3
- Source4
- Three Joined Tables
- Removed Blank Rows
- Removed Errors
- Sorted Rows
- Duplicated Column
- Renamed Columns
- Extracted Text Between Delimi...
- Added Conditional Column
- Added Conditional Column1
- Added Custom
- Reordered Columns
- Removed Columns
- Added Conditional Column2
- Table1
- Table2
- Table3
- Expanded Table1
- Removed Columns1
- Renamed Columns1
- Appended Query
- Removed Blank Rows1
- Replaced Value

Query steps (trace)

Power Query Editor – Part of M-Code IDE

M-Code Auto Generation Buttons (Transform)

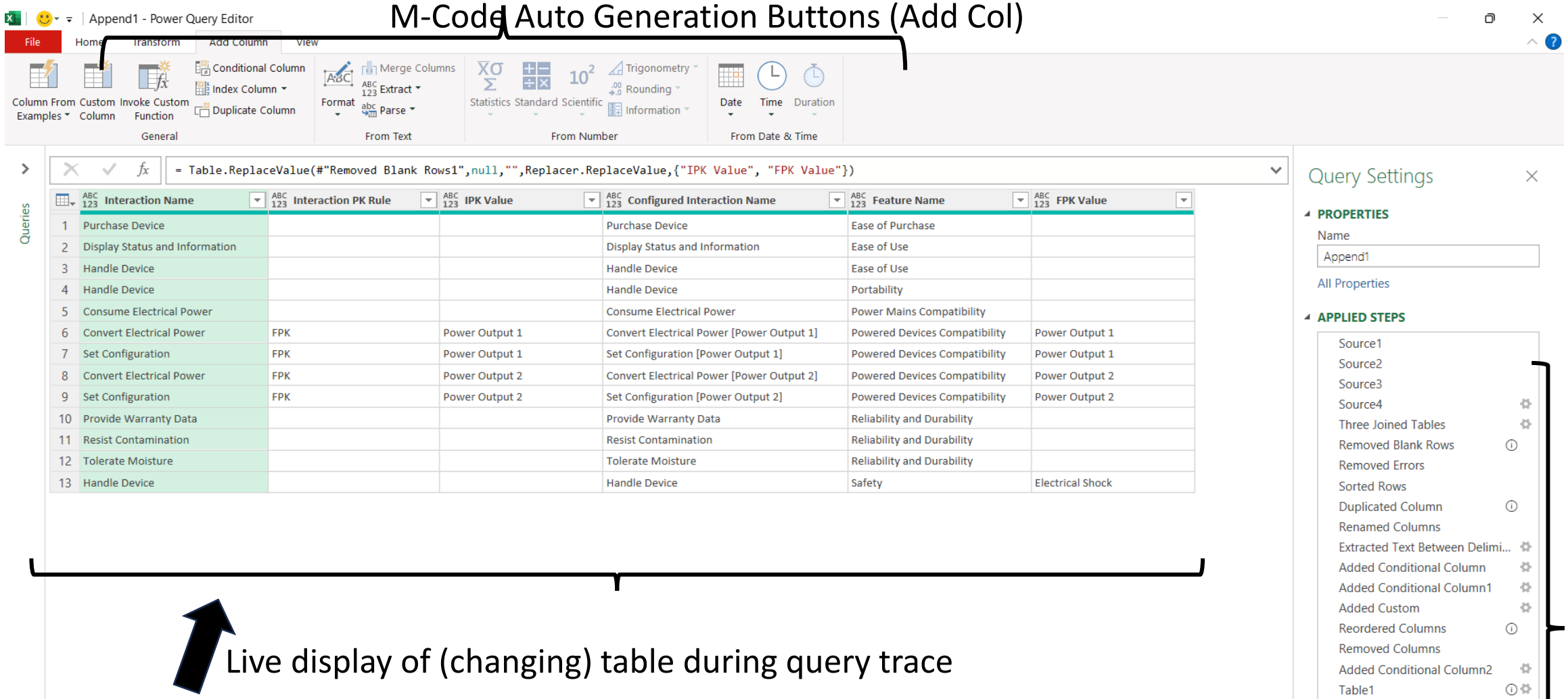


	Interaction Name	Interaction PK Rule	IPK Value	Configured Interaction Name	Feature Name	FPK Value
1	Purchase Device			Purchase Device	Ease of Purchase	
2	Display Status and Information			Display Status and Information	Ease of Use	
3	Handle Device			Handle Device	Ease of Use	
4	Handle Device			Handle Device	Portability	
5	Consume Electrical Power			Consume Electrical Power	Power Mains Compatibility	
6	Convert Electrical Power	FPK	Power Output 1	Convert Electrical Power [Power Output 1]	Powered Devices Compatibility	Power Output 1
7	Set Configuration	FPK	Power Output 1	Set Configuration [Power Output 1]	Powered Devices Compatibility	Power Output 1
8	Convert Electrical Power	FPK	Power Output 2	Convert Electrical Power [Power Output 2]	Powered Devices Compatibility	Power Output 2
9	Set Configuration	FPK	Power Output 2	Set Configuration [Power Output 2]	Powered Devices Compatibility	Power Output 2
10	Provide Warranty Data			Provide Warranty Data	Reliability and Durability	
11	Resist Contamination			Resist Contamination	Reliability and Durability	
12	Tolerate Moisture			Tolerate Moisture	Reliability and Durability	
13	Handle Device			Handle Device	Safety	Electrical Shock

Live display of (changing) table during query trace

Query steps (trace)

Power Query Editor – Part of M-Code IDE



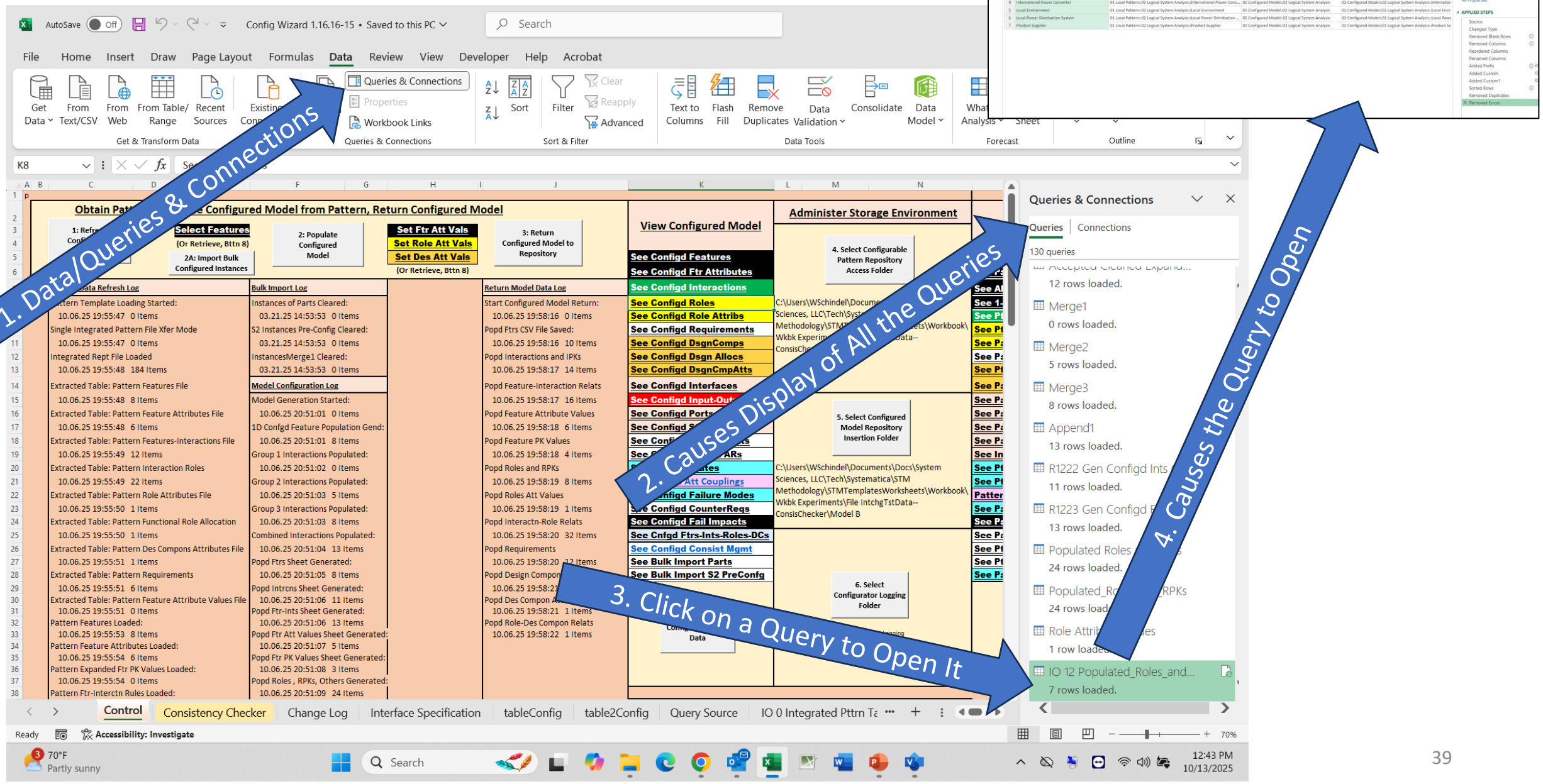
Power Query Editor – Part of M-Code IDE

Advanced M-Code Editor for Direct Source Code Entry, Display, Editing

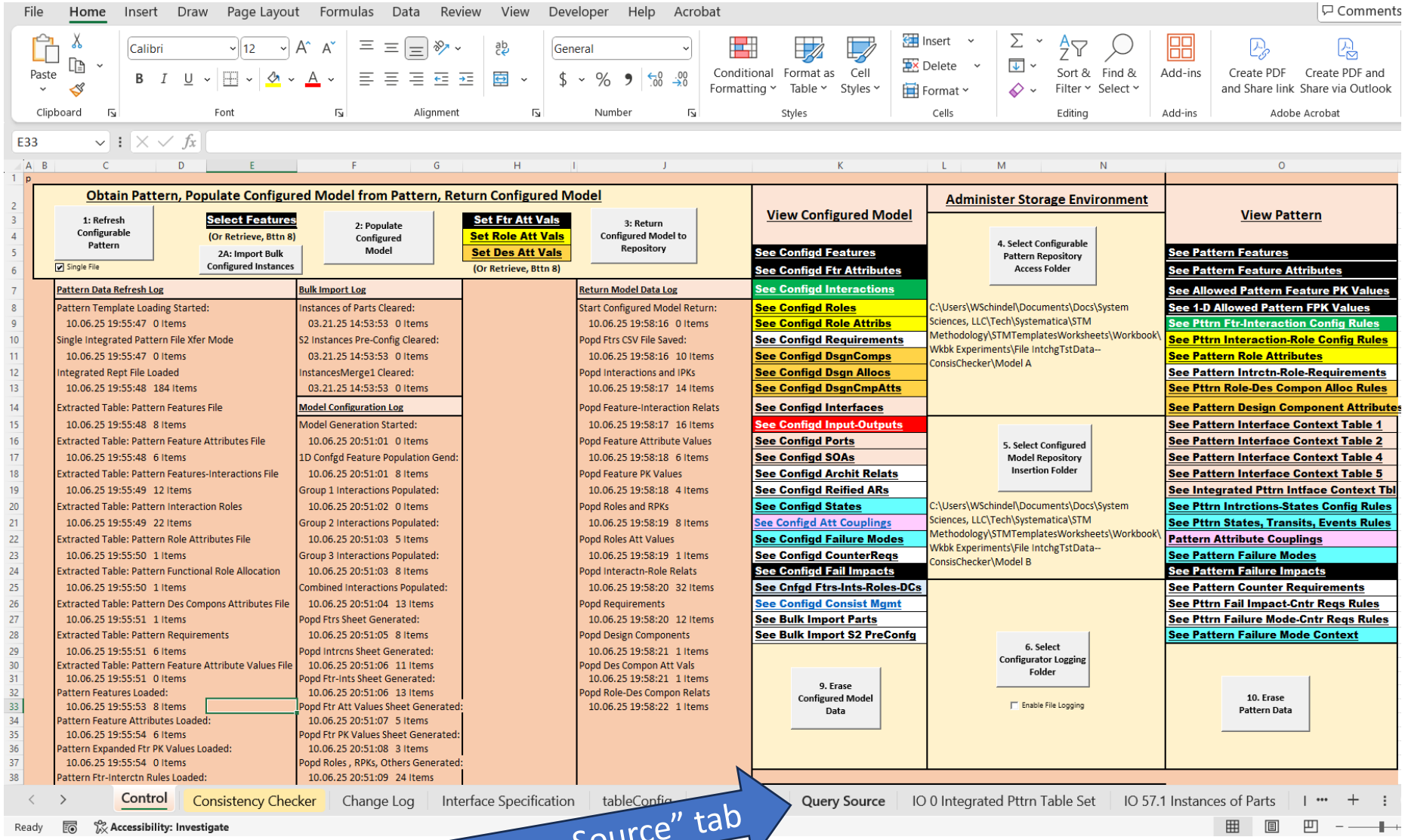


Power Query Editor – Part of M-Code IDE

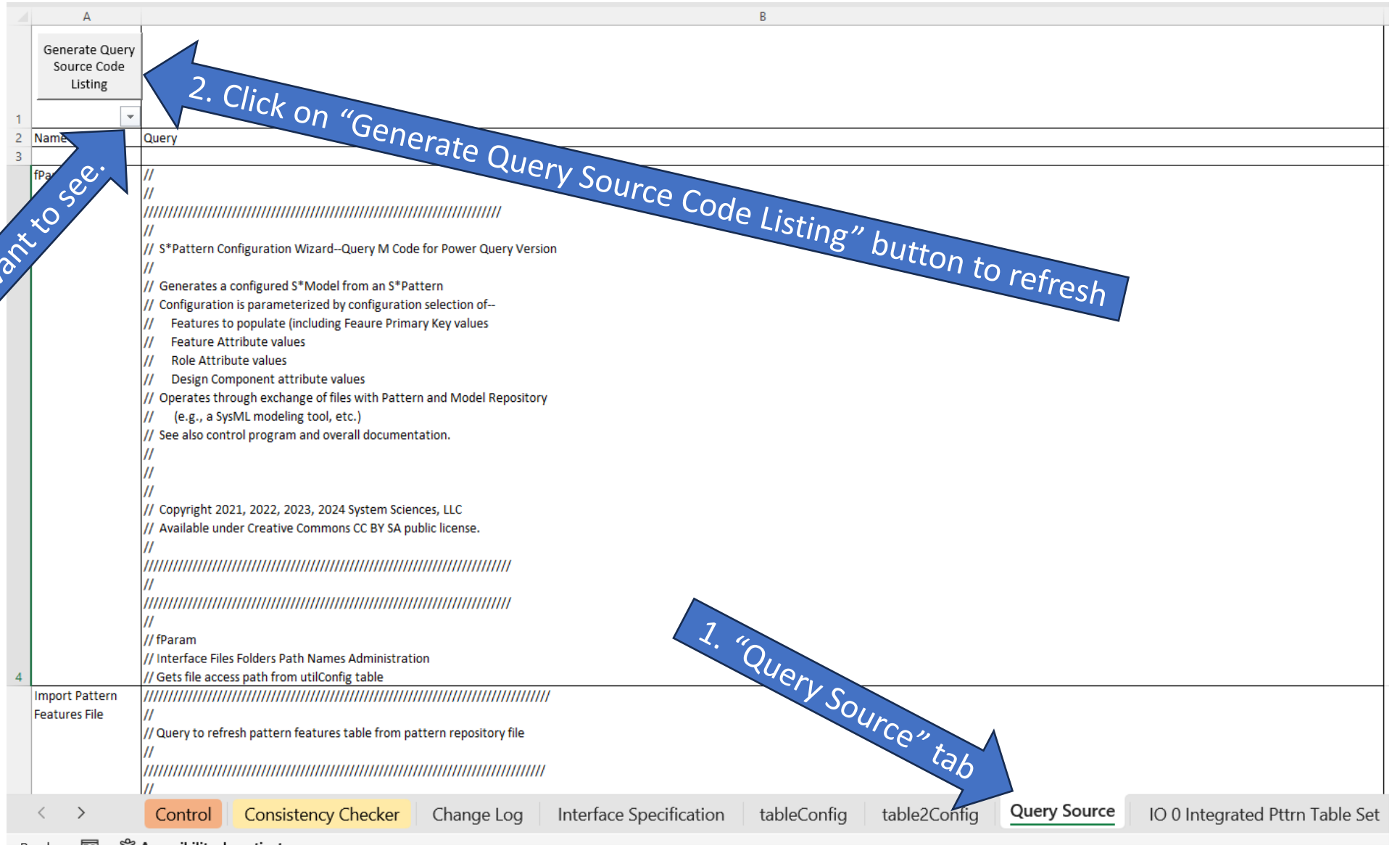
Entering the Power Query Editor



The short-cut way to see the M-Code Queries—but not edit.



The short-cut way to see the M-Code Queries—but not edit.



learn.microsoft.com/en-us/office/vba/api/overview/excel

Microsoft Ignite

November 17–21, 2025

Register now

Filter by title

Office VBA Reference
- Access
- Excel
  - Overview
  - Concepts
  - Object model
  - Excel Graph Visual Basic reference
- Office for Mac
- Outlook
- PowerPoint
- Project
- Publisher
- Visio
- Word
- Language reference
- Library reference

Excel VBA reference

07/11/2022

This reference contains conceptual overviews, programming tasks, samples, and references to help you develop Excel solutions.

> **Note**
>
> Interested in developing solutions that extend the Office experience across multiple platforms? Check out the new Office Add-ins model.

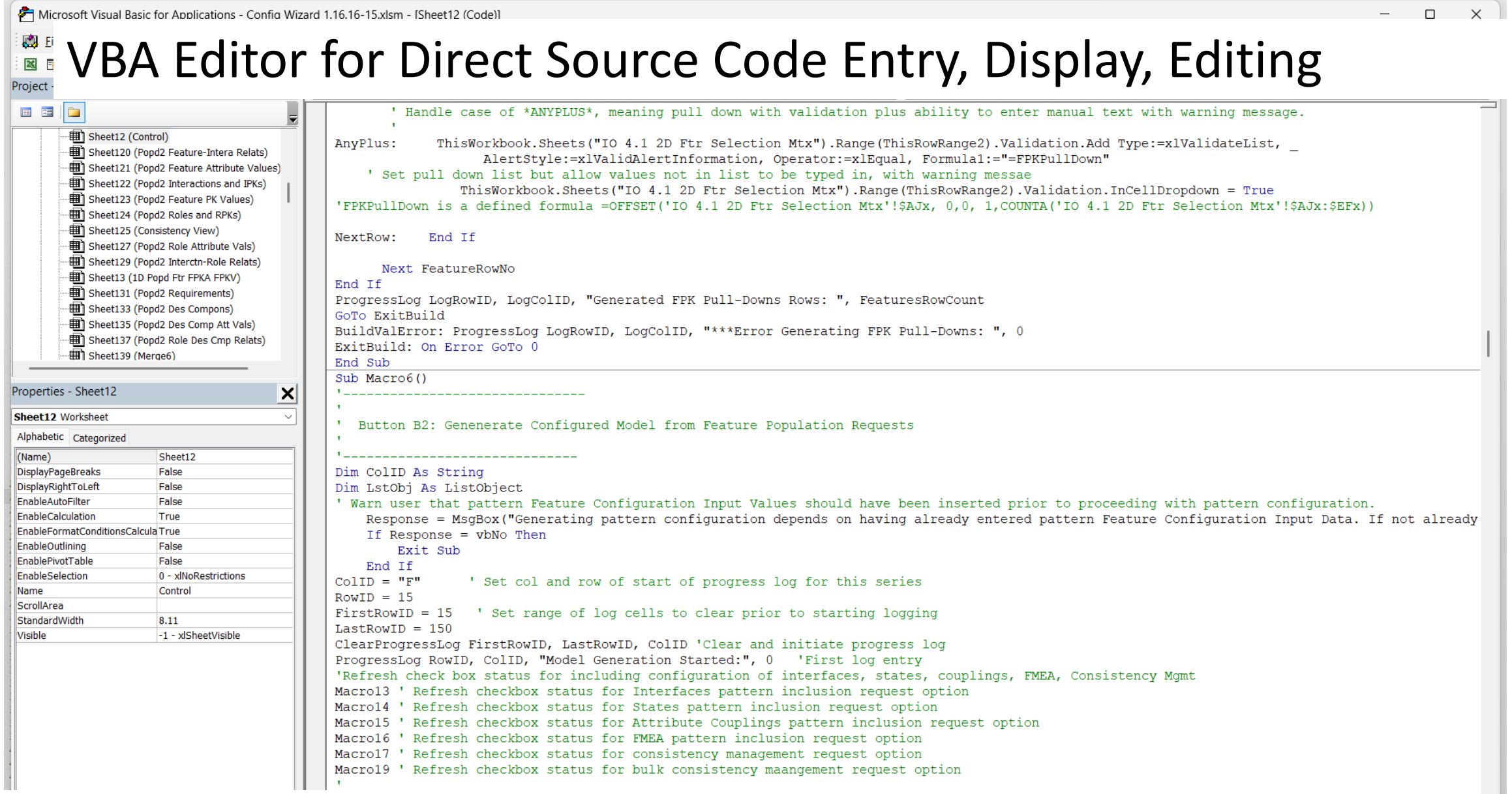
Use the table of contents in the navigation on the left to view the topics in the following sections:

- Concepts: Provides important concepts for developing custom Excel solutions.
- Object model reference: Provides reference materials for the Excel object model.
- Graph Visual Basic reference

Was this page helpful?

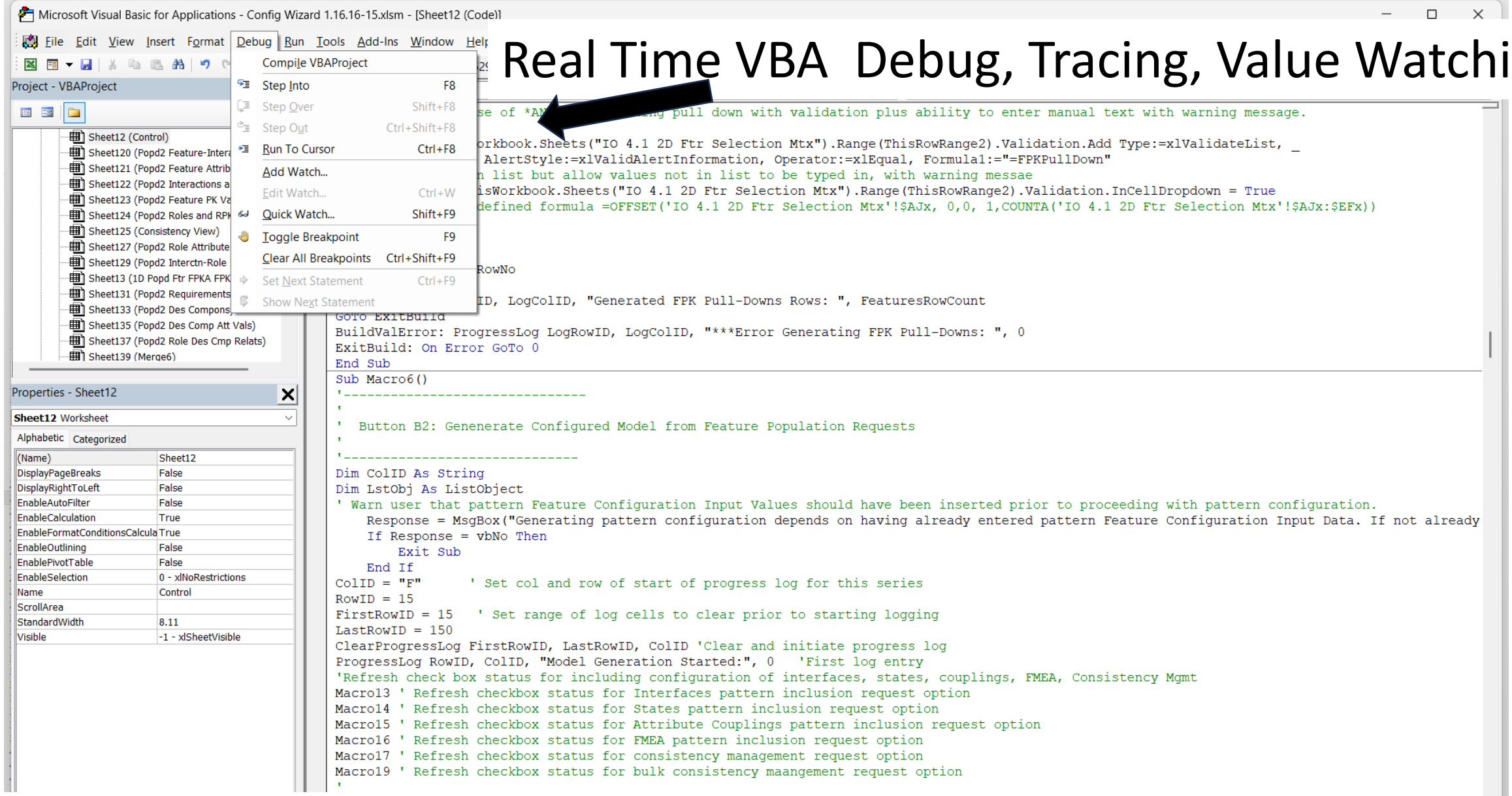
Yes | No

VBA Editor for Direct Source Code Entry, Display, Editing



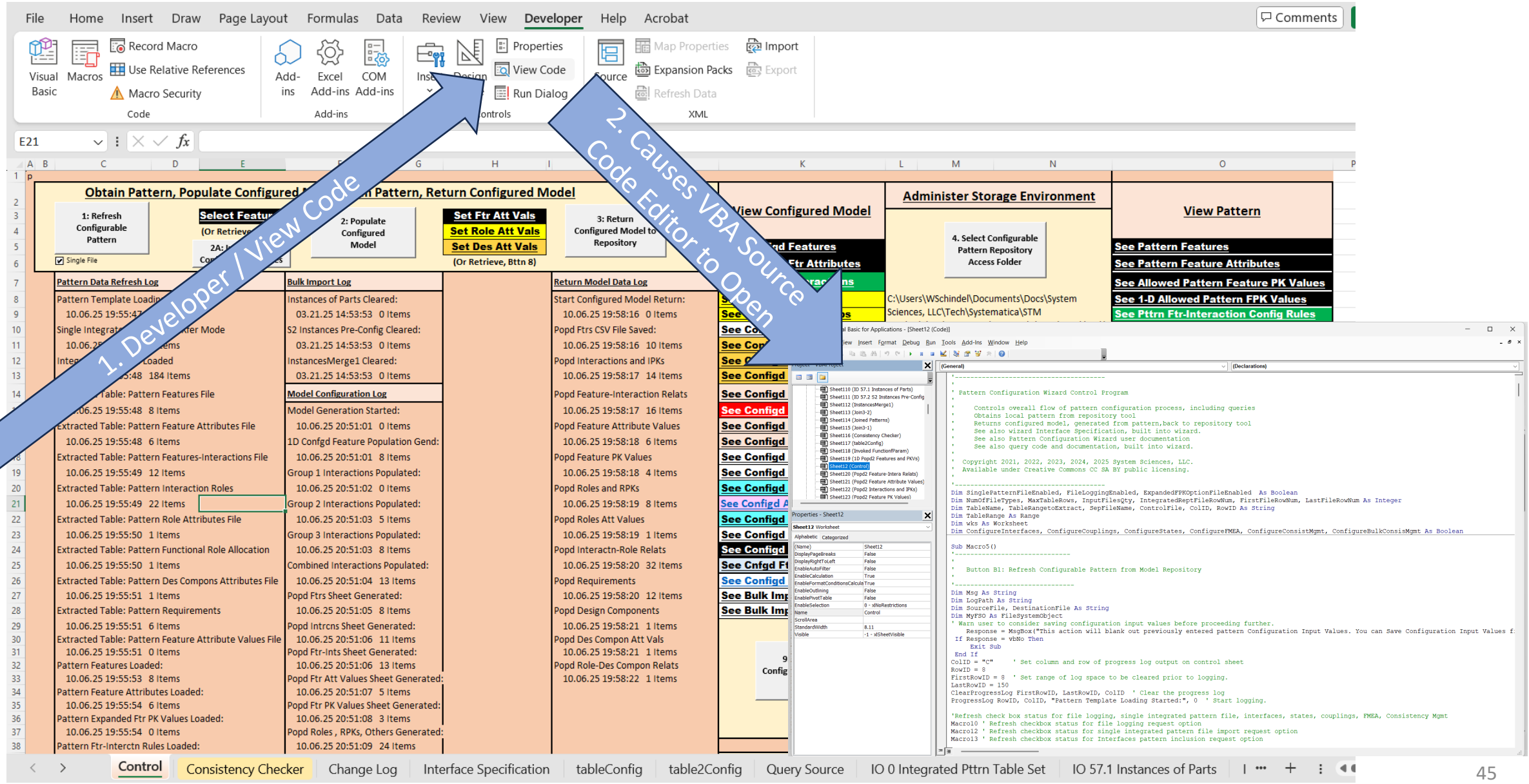
VBA Editor – Part of VBA IDE

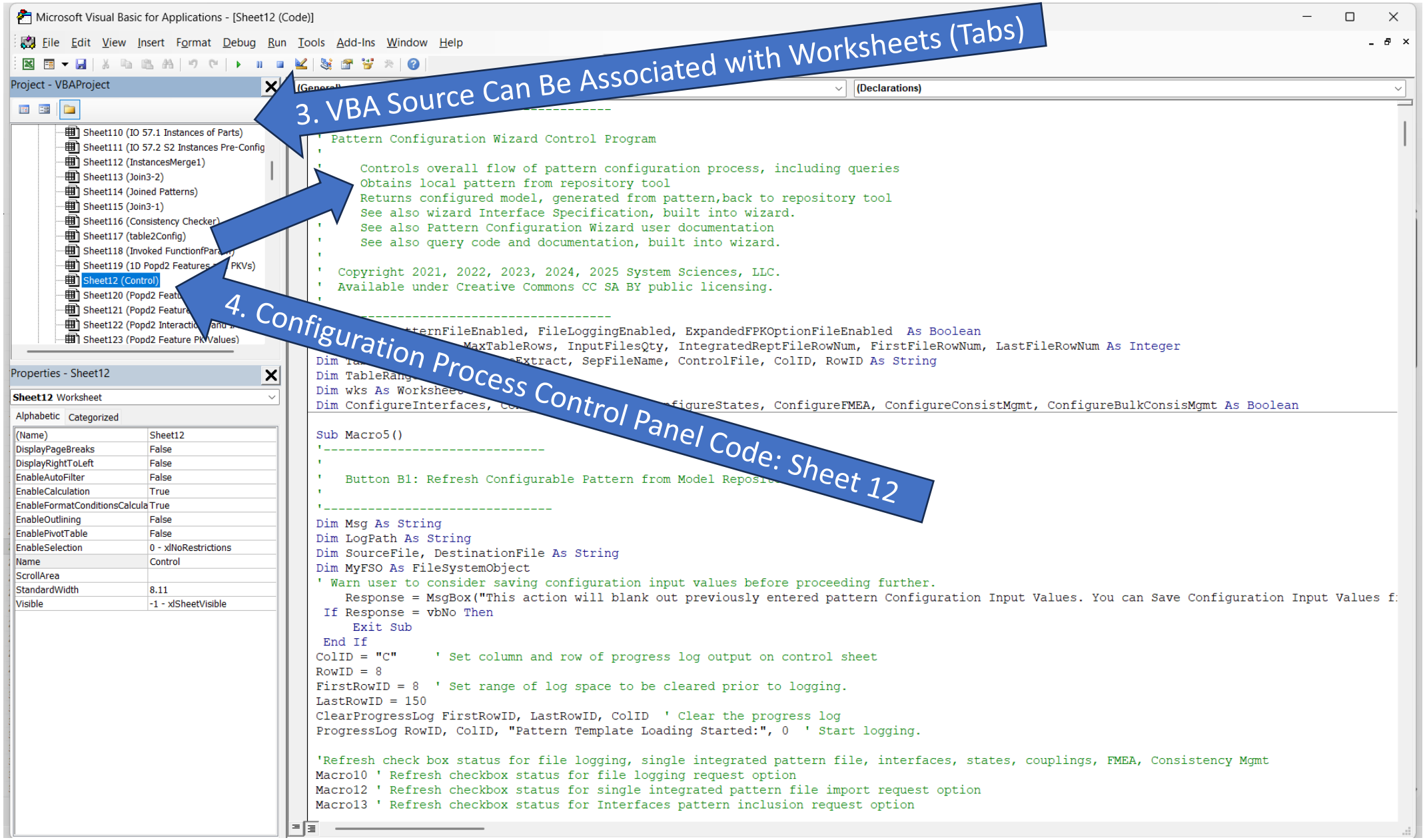
Real Time VBA Debug, Tracing, Value Watching

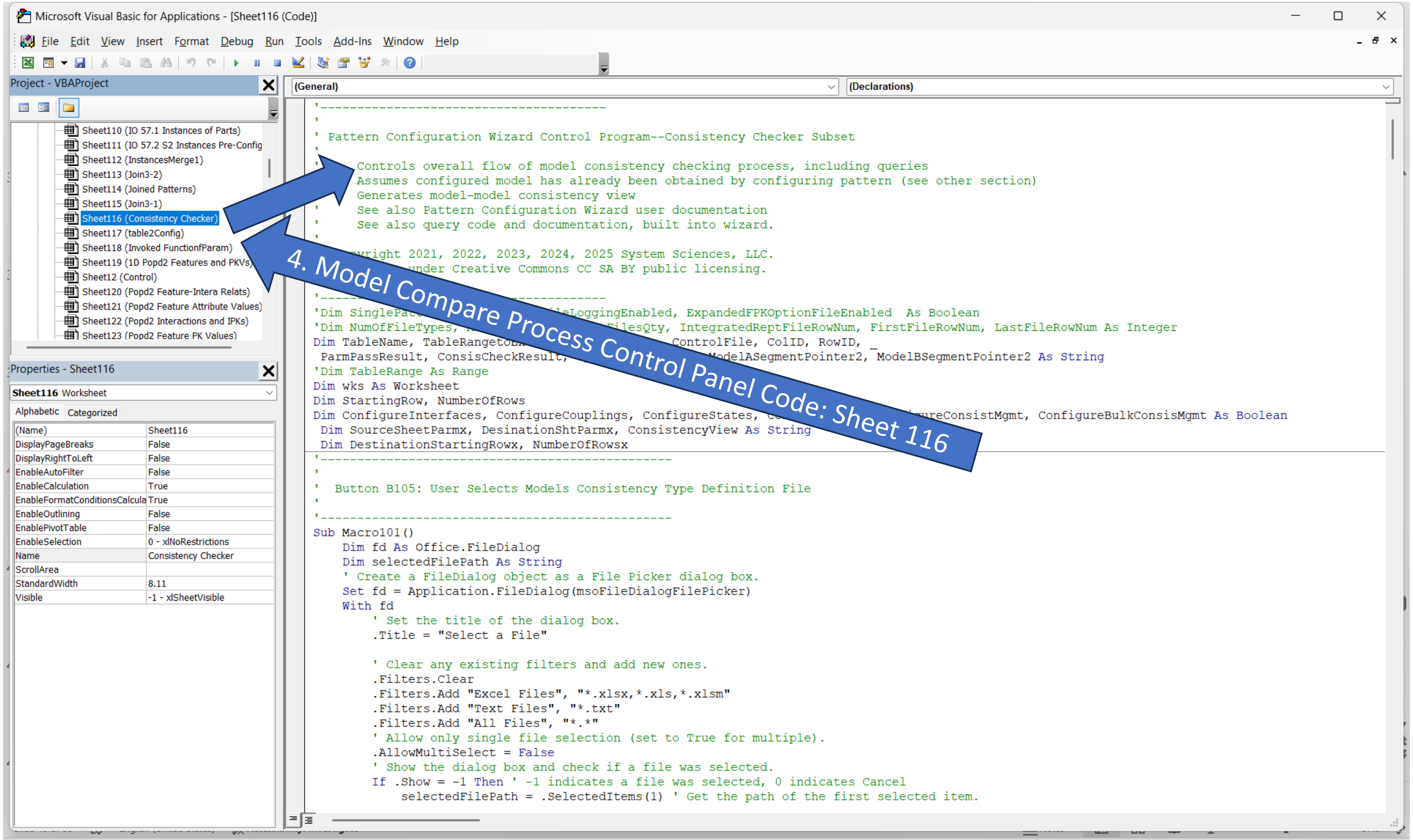


VBA Editor – Part of VBA IDE

Entering the Excel VBA Code Access IDE




3. VBA Source Can Be Associated with Worksheets (Tabs)
4. Configuration Process Control Panel Code: Sheet 12
Microsoft Visual Basic for Applications - [Sheet12 (Code)]
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Project - VBAProject
Sheet110 (IO 57.1 Instances of Parts)
Sheet111 (IO 57.2 S2 Instances Pre-Config
Sheet112 (InstancesMerge1)
Sheet113 (Join3-2)
Sheet114 (Joined Patterns)
Sheet115 (Join3-1)
Sheet116 (Consistency Checker)
Sheet117 (table2Config)
Sheet118 (Invoked FunctionfPara
Sheet119 (1D Popd2 Features
PKVs)
Sheet12 (Control)
Sheet120 (Popd2 Featu
Sheet121 (Popd2 Feature
Sheet122 (Popd2 Interacti
Sheet123 (Popd2 Feature PK Values)
Properties - Sheet12
Sheet12 Worksheet
Alphabetic Categorized
(Name) Sheet12
DisplayPageBreaks False
DisplayRightToLeft False
EnableAutoFilter False
EnableCalculation True
EnableFormatConditionsCalcula True
EnableOutlining False
EnablePivotTable False
EnableSelection 0 - xlNoRestrictions
Name Control
ScrollArea
StandardWidth 8.11
Visible -1 - xlSheetVisible
(General)
(Declarations)
' Pattern Configuration Wizard Control Program
'
' Controls overall flow of pattern configuration process, including queries
' Obtains local pattern from repository tool
' Returns configured model, generated from pattern,back to repository tool
' See also wizard Interface Specification, built into wizard.
' See also Pattern Configuration Wizard user documentation
' See also query code and documentation, built into wizard.
'
' Copyright 2021, 2022, 2023, 2024, 2025 System Sciences, LLC.
' Available under Creative Commons CC SA BY public licensing.
'
Dim ConfigureInterfaces,
Sub Macro5()
'
' Button B1: Refresh Configurable Pattern from Model
'
Dim Msg As String
Dim LogPath As String
Dim SourceFile, DestinationFile As String
Dim MyFSO As FileSystemObject
' Warn user to consider saving configuration input values before proceeding further.
Response = MsgBox("This action will blank out previously entered pattern Configuration Input Values. You can Save Configuration Input Values f
If Response = vbNo Then
Exit Sub
End If
ColID = "C" ' Set column and row of progress log output on control sheet
RowID = 8
FirstRowID = 8 ' Set range of log space to be cleared prior to logging.
LastRowID = 150
ClearProgressLog FirstRowID, LastRowID, ColID ' Clear the progress log
ProgressLog RowID, ColID, "Pattern Template Loading Started:", 0 ' Start logging.
'Refresh check box status for file logging, single integrated pattern file, interfaces, states, couplings, FMEA, Consistency Mgmt
Macro10 ' Refresh checkbox status for file logging request option
Macro12 ' Refresh checkbox status for single integrated pattern file import request option
Macro13 ' Refresh checkbox status for Interfaces pattern inclusion request option


Microsoft Visual Basic for Applications - [Sheet116 (Code)]
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Project - VBAProject
Sheet110 (IO 57.1 Instances of Parts)
Sheet111 (IO 57.2 S2 Instances Pre-Config
Sheet112 (InstancesMerge1)
Sheet113 (Join3-2)
Sheet114 (Joined Patterns)
Sheet115 (Join3-1)
Sheet116 (Consistency Checker)
Sheet117 (table2Config)
Sheet118 (Invoked FunctionfParam)
Sheet119 (1D Popd2 Features and PKVs)
Sheet12 (Control)
Sheet120 (Popd2 Feature-Intera Relats)
Sheet121 (Popd2 Feature Attribute Values)
Sheet122 (Popd2 Interactions and IPKs)
Sheet123 (Popd2 Feature PK Values)
Properties - Sheet116
Sheet116 Worksheet
Alphabetic Categorized
(Name) Sheet116
DisplayPageBreaks False
DisplayRightToLeft False
EnableAutoFilter False
EnableCalculation True
EnableFormatConditionsCalcula True
EnableOutlining False
EnablePivotTable False
EnableSelection 0 - xlNoRestrictions
Name Consistency Checker
ScrollArea
StandardWidth 8.11
Visible -1 - xlSheetVisible
(General)
(Declarations)
' Pattern Configuration Wizard Control Program--Consistency Checker Subset
' Controls overall flow of model consistency checking process, including queries
' Assumes configured model has already been obtained by configuring pattern (see other section)
' Generates model-model consistency view
' See also Pattern Configuration Wizard user documentation
' See also query code and documentation, built into wizard.
' Button B105: User Selects Models Consistency Type Definition File
Sub Macro101()
Dim fd As Office.FileDialog
Dim selectedFilePath As String
' Create a FileDialog object as a File Picker dialog box.
Set fd = Application.FileDialog(msoFileDialogFilePicker)
With fd
' Set the title of the dialog box.
.Title = "Select a File"
' Clear any existing filters and add new ones.
.Filters.Clear
.Filters.Add "Excel Files", "*.xlsx,*.xls,*.xlsm"
.Filters.Add "Text Files", "*.txt"
.Filters.Add "All Files", "*.*"
' Allow only single file selection (set to True for multiple).
.AllowMultiSelect = False
' Show the dialog box and check if a file was selected.
If .Show = -1 Then ' -1 indicates a file was selected, 0 indicates Cancel
selectedFilePath = .SelectedItems(1) ' Get the path of the first selected item.
4. Model Compare Process Control Panel Code: Sheet 116

Questions and discussion

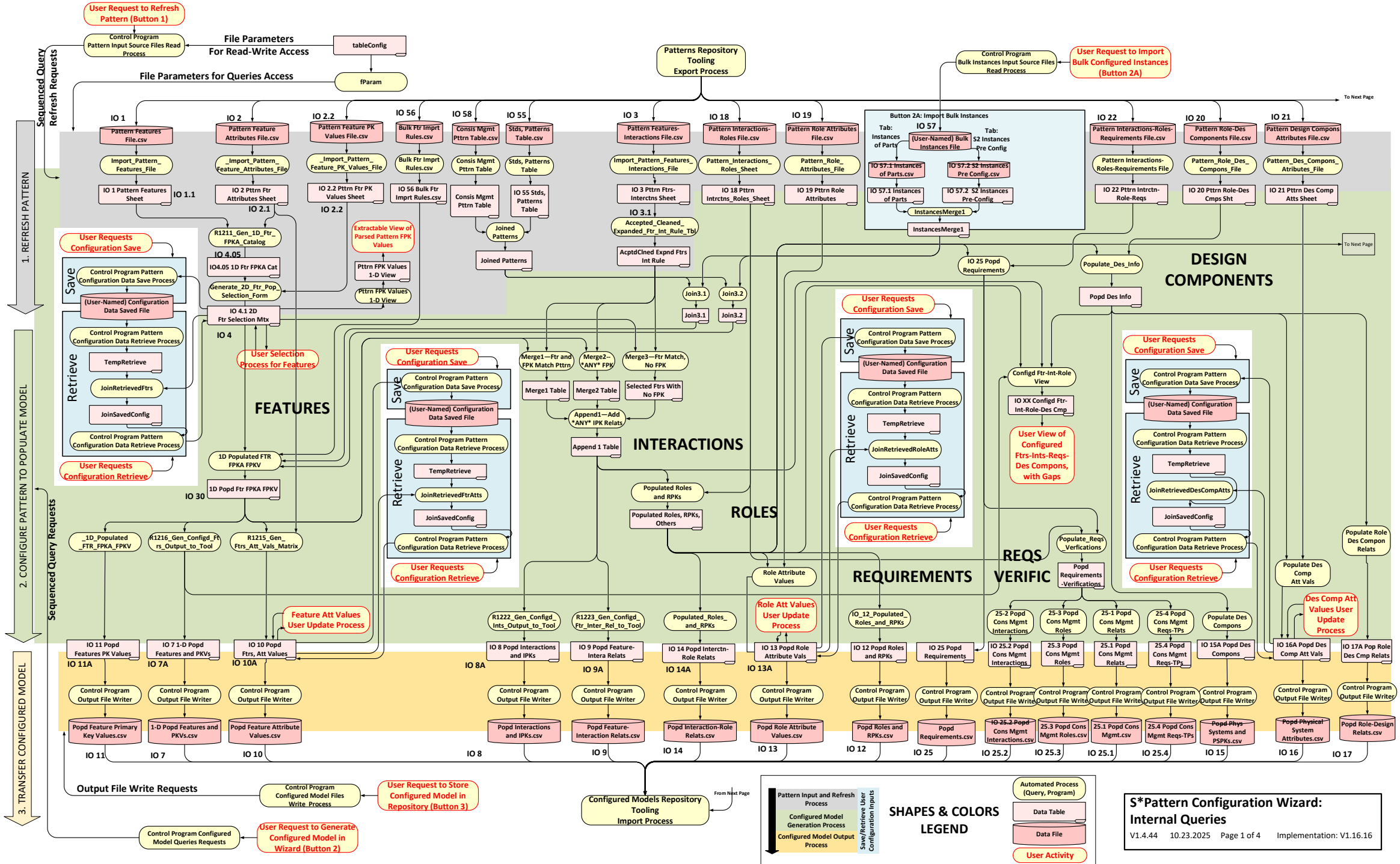
-
-
-
-
-
-
-

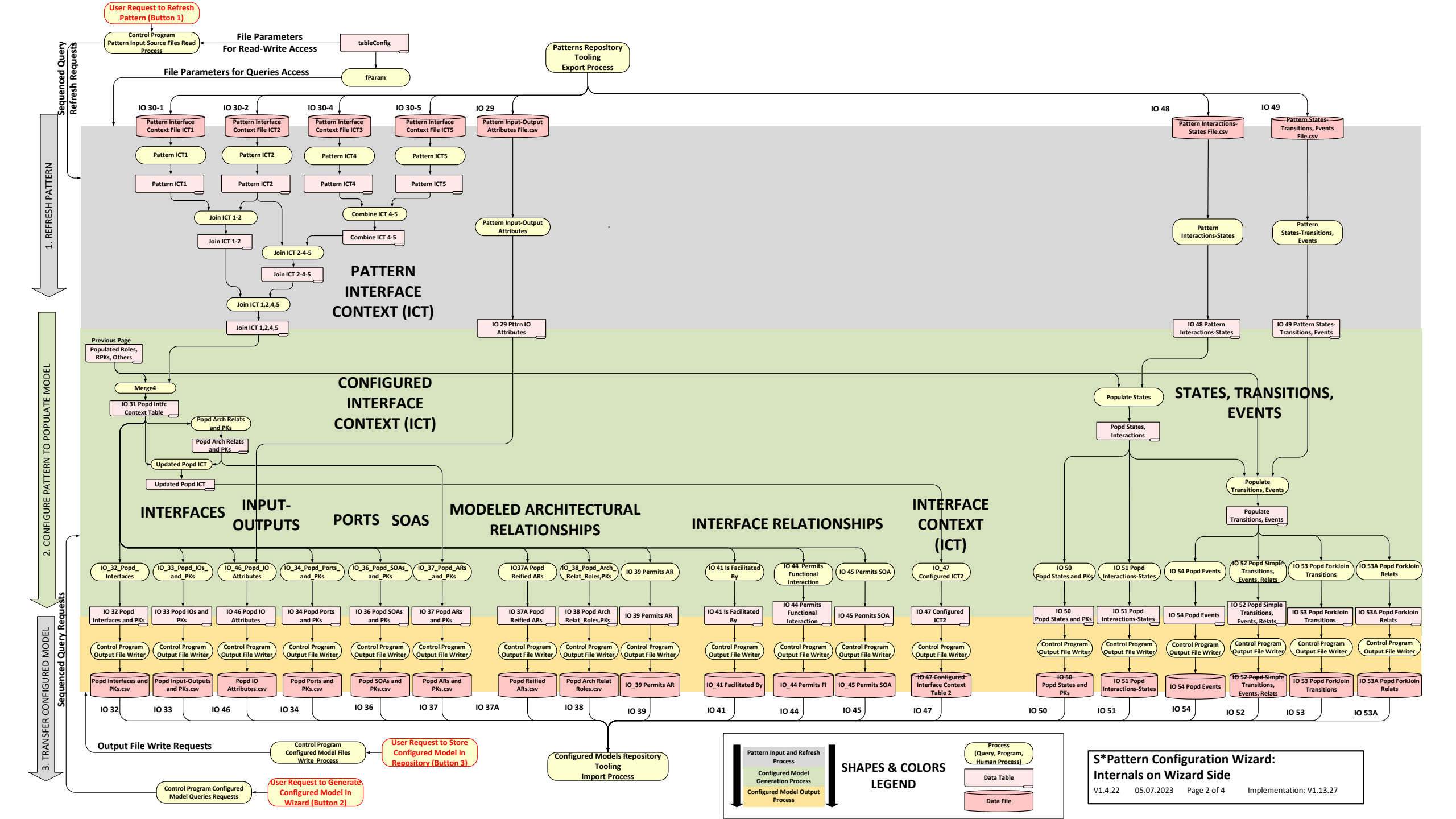
References

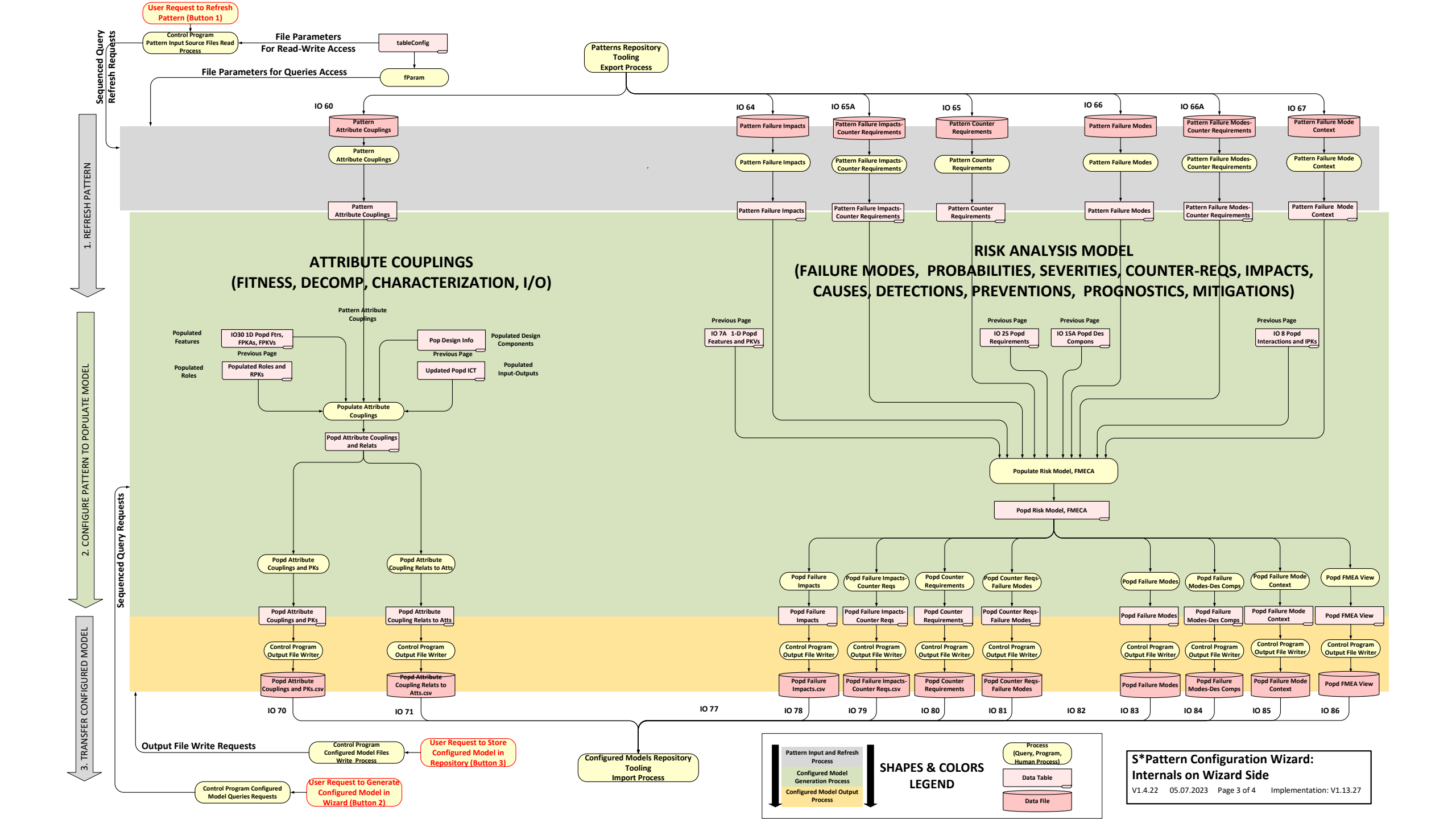
1. “Power Query M Formula Language”. Microsoft Corp. 2025. Accessed from -- https://learn.microsoft.com/en-us/powerquery-m/
2. “Excel VBA Reference”. Microsoft Corp. 2025. Accessed from -- https://learn.microsoft.com/en-us/office/vba/api/overview/excel
3. INCOSE/OMG MBSE Patterns Working Group Web Site: https://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns
4. “Guide to the S*Pattern Configuration Wizard ”, V1.5.3, INCOSE MBSE Patterns Working Group. Aug 08, 2024. Download from -- https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:guide_to_the_pattern_configuration_wizard_v1.5.3.pdf
5. “Guide to Tool-Specific Support for S* Patterns in Dassault Cameo Systems Modeler™”, V1.4.6. INCOSE MBSE Patterns Working Group. 2025. Download from -- https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:guide_to_systematica_patterns_in_cameo_systems_modeler_v1.4.6.pdf
6. “Systematica Metamodel”, Version 8.0. INCOSE MBSE Patterns Working Group. May 01, 2025. Download from -- https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:systematica_8_metamodel_v8.0.61.pdf
7. “S*Metamodel Mapping for Cameo Systems Modeler Versions 2019-2024xR2”, V1.12.1, May 12, 2025. Download from -- https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:systematica_mapping_for_magicdraw_csm_v1.12.1.pdf

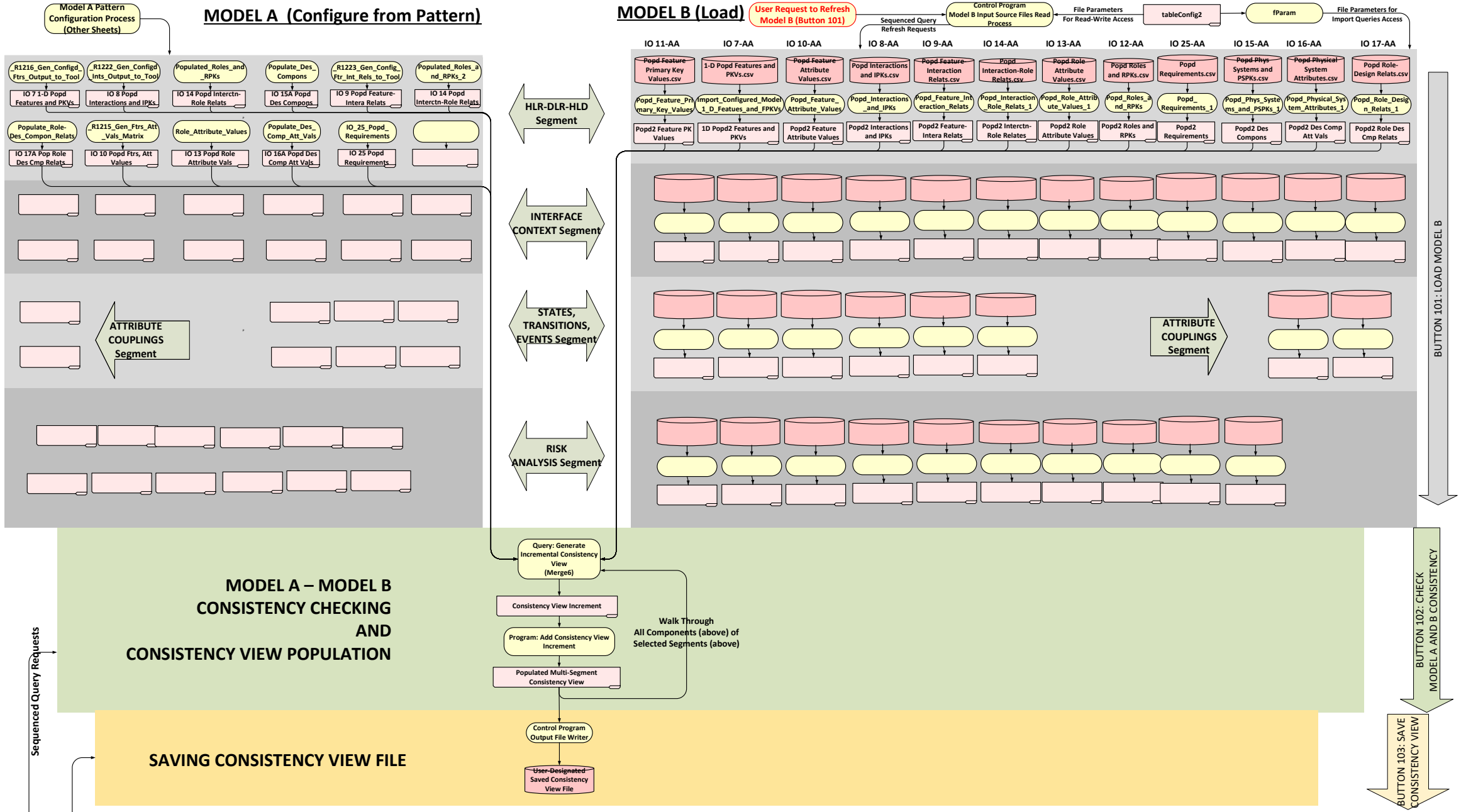
Attachment—Level 2 Flow and Queries Architecture Diagrams

Attachment I—Level 2 Flow and Queries Architecture Diagrams


User Request to Refresh Pattern (Button 1)
Control Program Pattern Input Source Files Read Process
File Parameters For Read-Write Access
tableConfig
File Parameters for Queries Access
fParam
Sequenced Query Refresh Requests
Patterns Repository Tooling Export Process
Control Program Bulk Instances Input Source Files Read Process
User Request to Import Bulk Configured Instances (Button 2A)
To Next Page
1. REFRESH PATTERN
2. CONFIGURE PATTERN TO POPULATE MODEL
3. TRANSFER CONFIGURED MODEL
Sequenced Query Requests
IO 1
Pattern Features File.csv
Import_Pattern_ Features_File
IO 1 Pattern Features Sheet
IO 1.1
IO 2
Pattern Feature Attributes File.csv
_Import_Pattern_ Feature_Attributes_File
IO 2 Pttrn Ftr Attributes Sheet
IO 2.1
IO 2.2
Pattern Feature PK Values File.csv
_Import_Pattern_ Feature_PK_Values_File
IO 2.2 Pttrn Ftr PK Values Sheet
IO 56
Bulk Ftr Imprt Rules.csv
Bulk Ftr Imprt Rules.csv
IO 56 Bulk Ftr Imprt Rules.csv
IO 58
Consis Mgmt Pttrn Table.csv
Consis Mgmt Pttrn Table
Consis Mgmt Pttrn Table
IO 55
Stds, Patterns Table.csv
Stds, Patterns Table
IO 55 Stds, Patterns Table
Joined Patterns
Joined Patterns
IO 3
Pattern Features-Interactions File.csv
Import_Pattern_Features_ Interactions_File
IO 3 Pttrn Ftrs-Interctns Sheet
IO 3.1
Accepted_Cleaned_ Expanded_Ftr_Int_Rule_Tbl
AcptdClned Expnd Ftrs Int Rule
IO 18
Pattern Interactions-Roles File.csv
Pattern_Interactions_ Roles_Sheet
IO 18 Pttrn Intrctns_Roles_Sheet
IO 19
Pattern Role Attributes File.csv
Pattern_Role_ Attributes_File
IO 19 Pttrn Role Attributes
Button 2A: Import Bulk Instances
IO 57
Tab: Instances of Parts
(User-Named) Bulk Instances File
Tab: S2 Instances Pre Config
IO 57.1 Instances of Parts.csv
IO 57.2 S2 Instances Pre Config.csv
IO 57.1 Instances of Parts
IO 57.2 S2 Instances Pre-Config
InstancesMerge1
InstancesMerge1
IO 22
Pattern Interactions-Roles-Requirements File.csv
Pattern Interactions-Roles-Requirements File
IO 22 Pttrn Intrctn-Role-Reqs
IO 20
Pattern Role-Des Components File.csv
Pattern_Role_Des_ Compons_File
IO 20 Pttrn Role-Des Cmps Sht
IO 21
Pattern Design Compons Attributes File.csv
Pattern_Des_Compons_ Atributes_File
IO 21 Pttrn Des Comp Atts Sheet
IO 25 Popd Requirements
Populate_Des_Info
Popd Des Info
DESIGN COMPONENTS
To Next Page
User Requests Configuration Save
Save
Control Program Pattern Configuration Data Save Process
(User-Named) Configuration Data Saved File
Retrieve
Control Program Pattern Configuration Data Retrieve Process
TempRetrieve
JoinRetrievedFtrs
JoinSavedConfig
Control Program Pattern Configuration Data Retrieve Process
User Requests Configuration Retrieve
R1211_Gen_1D_Ftr_ FPKA_Catalog
IO 4.05
IO4.05 1D Ftr FPKA Cat
Generate_2D_Ftr_Pop_ Selection_Form
IO 4.1 2D Ftr Selection Mtx
IO 4
Extractable View of Parsed Pattern FPK Values
Pttrn FPK Values 1-D View
Pttrn FPK Values 1-D View
User Selection Process for Features
FEATURES
1D Populated FTR FPKA FPKV
1D Popd Ftr FPKA FPKV
IO 30
_1D_Populated _FTR_FPKA_FPKV
R1216_Gen_Configd_Ftrs_Output_to_Tool
R1215_Gen_ Ftrs_Att_Vals_Matrix
Join3.1
Join3.2
Join3.1
Join3.2
Merge1—Ftr and FPK Match Pttrn
Merge2--*ANY* FPK
Merge3—Ftr Match, No FPK
Merge1 Table
Merge2 Table
Selected Ftrs With No FPK
Append1—Add *ANY* IPK Relats
Append 1 Table
INTERACTIONS
Populated Roles and RPKs
Populated Roles, RPKs, Others
ROLES
Role Attribute Values
Role Att Values User Update Process
IO 13 Popd Role Attribute Vals
IO 13A
Configd Ftr-Int-Role View
IO XX Configd Ftr-Int-Role-Des Cmp
User View of Configured Ftrs-Ints-Reqs-Des Compons, with Gaps
Populate_Reqs_Verfications
Popd Requirements-Verifications
REQS VERIFIC
REQUIREMENTS
IO_12_Populated_ Roles_and_RPKs
IO 12 Popd Roles and RPKs
IO 25 Popd Requirements
25-2 Popd Cons Mgmt Interactions
25-3 Popd Cons Mgmt Roles
25-1 Popd Cons Mgmt Relats
25-4 Popd Cons Mgmt Reqs-TPs
Populate Des Compons
IO 25.2 Popd Cons Mgmt Interactions
IO 25.3 Popd Cons Mgmt Roles
IO 25.1 Popd Cons Mgmt Relats
IO 25.4 Popd Cons Mgmt Reqs-TPs
IO 15A Popd Des Compons
Populate Des Comp Att Vals
Des Comp Att Values User Update Process
IO 16A Popd Des Comp Att Vals
IO 17A Pop Role Des Cmp Relats
Populate Role Des Compon Relats
IO 11 Popd Features PK Values
IO 11A
IO 7 1-D Popd Features and PKVs
IO 7A
IO 10 Popd Ftrs, Att Values
IO 10A
Feature Att Values User Update Process
IO 8 Popd Interactions and IPKs
IO 8A
IO 9 Popd Feature-Intera Relats
IO 9A
IO 14 Popd Interctn-Role Relats
IO 14A
R1222_Gen_Configd_ Ints_Output_to_Tool
R1223_Gen_Configd_ Ftr_Inter_Rel_to_Tool
Populated_Roles_ and_RPKs
Control Program Output File Writer
Popd Feature Primary Key Values.csv
IO 11
1-D Popd Features and PKVs.csv
IO 7
Popd Feature Attribute Values.csv
IO 10
Popd Interactions and IPKs.csv
IO 8
Popd Feature-Interaction Relats.csv
IO 9
Popd Interaction-Role Relats.csv
IO 14
Popd Role Attribute Values.csv
IO 13
Popd Roles and RPKs.csv
IO 12
Popd Requirements.csv
IO 25
IO 25.2 Popd Cons Mgmt Interactions.csv
IO 25.2
IO 25.3 Popd Cons Mgmt Roles.csv
IO 25.3
IO 25.1 Popd Cons Mgmt.csv
IO 25.1
IO 25.4 Popd Cons Mgmt Reqs-TPs
IO 25.4
Popd Phys Systems and PSPKs.csv
IO 15
Popd Physical System Attributes.csv
IO 16
Popd Role-Design Relats.csv
IO 17
Output File Write Requests
Control Program Configured Model Files Write_Process
User Request to Store Configured Model in Repository (Button 3)
Control Program Configured Model Queries Requests
User Request to Generate Configured Model in Wizard (Button 2)
Configured Models Repository Tooling Import Process
From Next Page
Pattern Input and Refresh Process
Configured Model Generation Process
Configured Model Output Process
Save/Retrieve User Configuration Inputs
SHAPES & COLORS LEGEND
Automated Process (Query, Program)
Data Table
Data File
User Activity
S*Pattern Configuration Wizard: Internal Queries
V1.4.44 10.23.2025 Page 1 of 4 Implementation: V1.16.16


User Request to Refresh Pattern (Button 1)
Control Program Pattern Input Source Files Read Process
File Parameters For Read-Write Access
tableConfig
File Parameters for Queries Access
fParam
Sequenced Query Refresh Requests
Patterns Repository Tooling Export Process
1. REFRESH PATTERN
IO 30-1
IO 30-2
IO 30-4
IO 30-5
IO 29
Pattern Interface Context File ICT1
Pattern Interface Context File ICT2
Pattern Interface Context File ICT3
Pattern Interface Context File ICT5
Pattern Input-Output Attributes File.csv
Pattern ICT1
Pattern ICT2
Pattern ICT4
Pattern ICT5
Join ICT 1-2
Combine ICT 4-5
Join ICT 2-4-5
Join ICT 1,2,4,5
PATTERN INTERFACE CONTEXT (ICT)
Pattern Input-Output Attributes
IO 29 Pttrn IO Attributes
IO 48
IO 49
Pattern Interactions-States File.csv
Pattern States-Transitions, Events File.csv
Pattern Interactions-States
Pattern States-Transitions, Events
IO 48 Pattern Interactions-States
IO 49 Pattern States-Transitions, Events
2. CONFIGURE PATTERN TO POPULATE MODEL
Previous Page Populated Roles, RPKs, Others
Merge4
IO 31 Popd Intfc Context Table
Popd Arch Relats and PKs
Updated Popd ICT
CONFIGURED INTERFACE CONTEXT (ICT)
STATES, TRANSITIONS, EVENTS
Populate States
Popd States, Interactions
Populate Transitions, Events
INTERFACES
INPUT-OUTPUTS
PORTS
SOAS
MODELED ARCHITECTURAL RELATIONSHIPS
INTERFACE RELATIONSHIPS
INTERFACE CONTEXT (ICT)
IO_32_Popd_ Interfaces
IO_33_Popd_IOs_ and_PKs
IO_46_Popd_IO Attributes
IO_34_Popd_Ports_ and_PKs
IO_36_Popd_SOAs_ and_PKs
IO_37_Popd_ARs _and_PKs
IO37A Popd Reified ARs
IO_38_Popd_Arch_ Relat_Roles,PKs
IO 39 Permits AR
IO 41 Is Facilitated By
IO 44 Permits Functional Interaction
IO 45 Permits SOA
IO_47 Configured ICT2
IO 50 Popd States and PKs
IO 51 Popd Interactions-States
IO 54 Popd Events
IO 52 Popd Simple Transitions, Events, Relats
IO 53 Popd ForkJoin Transitions
IO 53A Popd ForkJoin Relats
IO 32 Popd Interfaces and PKs
IO 33 Popd IOs and PKs
IO 46 Popd IO Attributes
IO 34 Popd Ports and PKs
IO 36 Popd SOAs and PKs
IO 37 Popd ARs and PKs
IO 37A Popd Reified ARs
IO 38 Popd Arch Relat_Roles,PKs
IO 39 Permits AR
IO 41 Is Facilitated By
IO 44 Permits Functional Interaction
IO 45 Permits SOA
IO 47 Configured ICT2
IO 50 Popd States and PKs
IO 51 Popd Interactions-States
IO 54 Popd Events
IO 52 Popd Simple Transitions, Events, Relats
IO 53 Popd ForkJoin Transitions
IO 53A Popd ForkJoin Relats
Control Program Output File Writer
Popd Interfaces and PKs.csv
Popd Input-Outputs and PKs.csv
Popd IO Attributes.csv
Popd Ports and PKs.csv
Popd SOAs and PKs.csv
Popd ARs and PKs.csv
Popd Reified ARs.csv
Popd Arch Relat Roles.csv
IO_39 Permits AR
IO_41 Facilitated By
IO_44 Permits FI
IO_45 Permits SOA
IO 47 Configured Interface Context Table 2
IO 50 Popd States and PKs
IO 51 Popd Interactions-States
IO 54 Popd Events
IO 52 Popd Simple Transitions, Events, Relats
IO 53 Popd ForkJoin Transitions
IO 53A Popd ForkJoin Relats
IO 32
IO 33
IO 46
IO 34
IO 36
IO 37
IO 37A
IO 38
IO 39
IO 41
IO 44
IO 45
IO 47
IO 50
IO 51
IO 54
IO 52
IO 53
IO 53A
3. TRANSFER CONFIGURED MODEL
Sequenced Query Requests
Output File Write Requests
Control Program Configured Model Files Write Process
User Request to Store Configured Model in Repository (Button 3)
Configured Models Repository Tooling Import Process
Control Program Configured Model Queries Requests
User Request to Generate Configured Model in Wizard (Button 2)
Pattern Input and Refresh Process
Configured Model Generation Process
Configured Model Output Process
SHAPES & COLORS LEGEND
Process (Query, Program, Human Process)
Data Table
Data File
S*Pattern Configuration Wizard:
Internals on Wizard Side
V1.4.22 05.07.2023 Page 2 of 4 Implementation: V1.13.27


User Request to Refresh Pattern (Button 1)
Control Program Pattern Input Source Files Read Process
File Parameters For Read-Write Access
tableConfig
fParam
File Parameters for Queries Access
Sequenced Query Refresh Requests
Patterns Repository Tooling Export Process
1. REFRESH PATTERN
2. CONFIGURE PATTERN TO POPULATE MODEL
3. TRANSFER CONFIGURED MODEL
IO 60
Pattern Attribute Couplings
IO 64
Pattern Failure Impacts
IO 65A
Pattern Failure Impacts-Counter Requirements
IO 65
Pattern Counter Requirements
IO 66
Pattern Failure Modes
IO 66A
Pattern Failure Modes-Counter Requirements
IO 67
Pattern Failure Mode Context
ATTRIBUTE COUPLINGS
(FITNESS, DECOMP, CHARACTERIZATION, I/O)
RISK ANALYSIS MODEL
(FAILURE MODES, PROBABILITIES, SEVERITIES, COUNTER-REQS, IMPACTS, CAUSES, DETECTIONS, PREVENTIONS, PROGNOSTICS, MITIGATIONS)
Populated Features
IO30 1D Popd Ftrs, FPKAs, FPKVs
Previous Page
Populated Roles
Populated Roles and RPKs
Pop Design Info
Populated Design Components
Updated Popd ICT
Populated Input-Outputs
Populate Attribute Couplings
Popd Attribute Couplings and Relats
Previous Page
IO 7A 1-D Popd Features and PKVs
IO 25 Popd Requirements
IO 15A Popd Des Compons
IO 8 Popd Interactions and IPKs
Populate Risk Model, FMECA
Popd Risk Model, FMECA
Sequenced Query Requests
Popd Attribute Couplings and PKs
Popd Attribute Coupling Relats to Atts
Control Program Output File Writer
Popd Attribute Couplings and PKs.csv
Popd Attribute Coupling Relats to Atts.csv
Popd Failure Impacts
Popd Failure Impacts-Counter Reqs
Popd Counter Requirements
Popd Counter Reqs-Failure Modes
Popd Failure Modes
Popd Failure Modes-Des Comps
Popd Failure Mode Context
Popd FMEA View
Popd Failure Impacts.csv
Popd Failure Impacts-Counter Reqs.csv
Popd Counter Requirements
Popd Counter Reqs-Failure Modes
Popd Failure Modes
Popd Failure Modes-Des Comps
Popd Failure Mode Context
Popd FMEA View
IO 70
IO 71
IO 77
IO 78
IO 79
IO 80
IO 81
IO 82
IO 83
IO 84
IO 85
IO 86
Output File Write Requests
Control Program Configured Model Files Write Process
User Request to Store Configured Model in Repository (Button 3)
Control Program Configured Model Queries Requests
User Request to Generate Configured Model in Wizard (Button 2)
Configured Models Repository Tooling Import Process
Pattern Input and Refresh Process
Configured Model Generation Process
Configured Model Output Process
SHAPES & COLORS LEGEND
Process (Query, Program, Human Process)
Data Table
Data File
S*Pattern Configuration Wizard:
Internals on Wizard Side
V1.4.22 05.07.2023 Page 3 of 4 Implementation: V1.13.27



S*Pattern Configuration Wizard: Internals on Wizard Side

Attachment II—VBA Control Code

Attachment III—Database Query M-Code