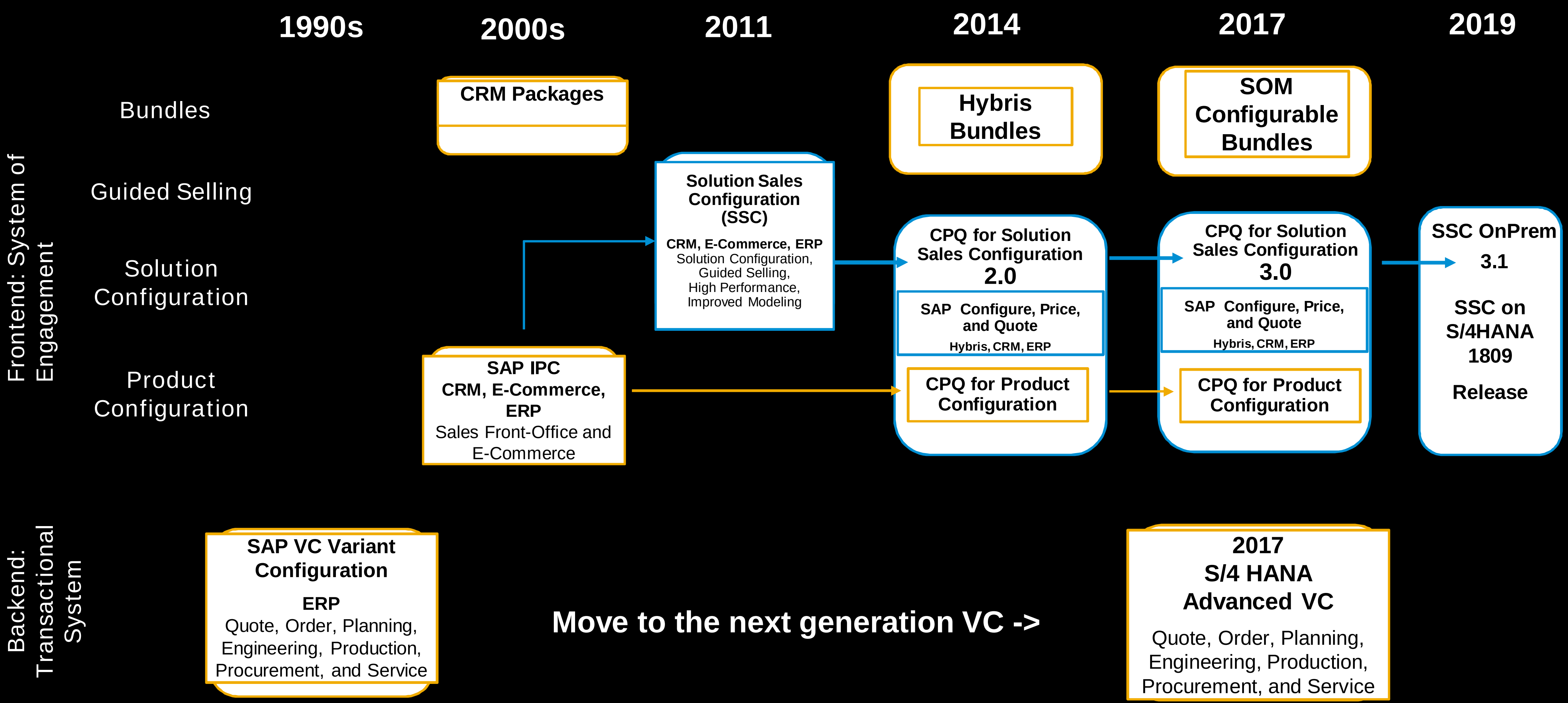
SAP Solution Sales Configuration

SCE and SME Overview

PUBLIC

THE BEST RUN SAP

Product and Solution Configuration with SAP: On-Premise



1990s

2000s

2011

2014

2017

2019

Frontend: System of Engagement

Bundles

Guided Selling

Solution Configuration

Product Configuration

CRM Packages

Solution Sales Configuration (SSC)

CRM, E-Commerce, ERP

Solution Configuration, Guided Selling, High Performance, Improved Modeling

SAP IPC

CRM, E-Commerce, ERP

Sales Front-Office and E-Commerce

Hybris Bundles

CPQ for Solution Sales Configuration 2.0

SAP Configure, Price, and Quote

Hybris, CRM, ERP

CPQ for Product Configuration

SOM Configurable Bundles

CPQ for Solution Sales Configuration 3.0

SAP Configure, Price, and Quote

Hybris, CRM, ERP

CPQ for Product Configuration

SSC OnPrem 3.1

SSC on S/4HANA 1809

Release

Backend: Transactional System

SAP VC Variant Configuration

ERP

Quote, Order, Planning, Engineering, Production, Procurement, and Service

Move to the next generation VC ->

2017

S/4 HANA

Advanced VC

Quote, Order, Planning, Engineering, Production, Procurement, and Service

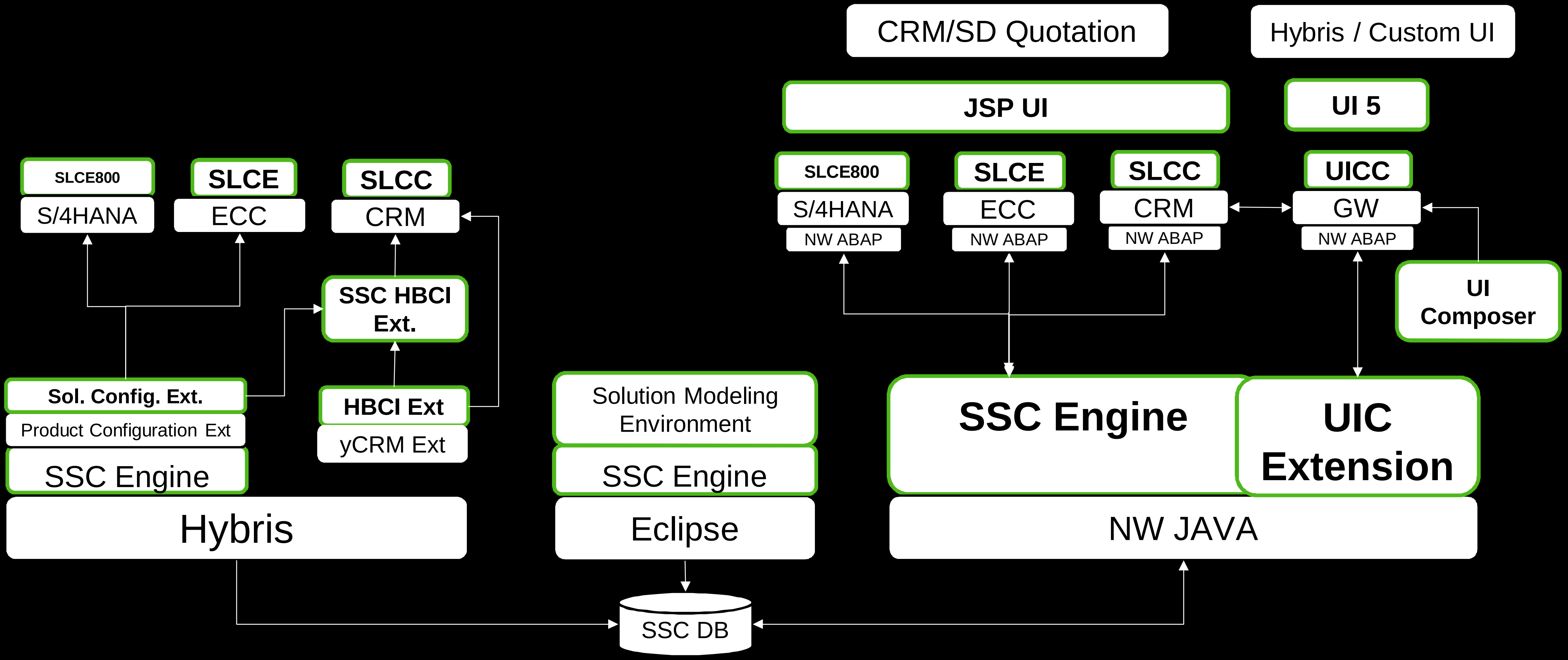
SAP Solution Sales Configuration (SSC)

SSC enables customers to build complex configurable solutions comprising of ‘any type’ of products and services across all SAP sales processes in ‘Standalone’ mode.

It comprises of :

- Solution Configuration Engine(SCE): Advanced inference engine, object oriented, and automated reasoning based upon declarative constraints.
- Solution Modeling Environment (SME):
  - Eclipse plugin for offline development and testing in distributed development setup.
  - Customizable and open development environment for specialized needs.
  - Devops support to ensure consistent performance and quality rollouts continuously.
- Sales and Service Process Integration: Seamless Integrations into SAP CRM, ERP, S/4HANA, and SAP Hybris Platforms

SSC , HBCI, and UIC Integrations Overview



Solution Configuration Engine

Advanced Mode for Bottom-Up Configuration

Classical Mode: Top-Down Configuration

- Maximum Bill of Material: Traditional super BOM

Advanced Mode: Bottom-Up Configuration

- Dynamic Bill of Material
    - Multi Instantiation: Multiple sub-components under main or root item at same BOM position.
    - Abstract Data Types: Link or group instances through reference cstics (ADT's)
- Attribute Facets in Constraints : Enhanced constraints process to consider attribute facets such as invisibility, assigned value, dynamic domain, user domain, persistent or sticky default, and not specified.
- Add Non-Parts : 'Non-Part' is an item that is not in a bill of material. They are related to other items, but not as component parts and relationships expressed using ADT's. (e.g. a service is not part of the items it serves, but does have a 'service relationship' with each of them)

Declarative Constraints and Rules

- Declarative: Define logic of computation without defining the control flow. Both constraints and rules are declarative in nature since they are triggered simultaneously when their trigger conditions are met.
  - Constraints: A relationship or condition that must hold true for configuration instances and their characteristics. Used to derive or infer contradictions or conflicts , Value Assignments , Domain Restrictions and exclusions , PART_OF instance hierarchy , Specialization of an instance etc.
  - Rules: Procedural form of dependency which are used for aggregations , access and modify external data and trigger any built in or user defined functions

Knowledge Base Decomposition , Orchestration and MCI

- KB Decomposition
  - Possible to break down big monolithic KBs into smaller KB . Easier management with relatively smaller unit of work.
  - Distributed, independent, and collaborative development
- KB Orchestration
  - Interlinking between two different KBs in a coordinated manner.
  - Unified representation of interlinked configurations.
  - Enables reuse of classical KBs within a solution model
- Multi Configuration Instances
  - Instances which can be made accessible in more than one configuration object simultaneously.

User Domain Calculations

User domain is set of allowed values in a restrictable cstic’s domain resulting from restrictions triggered by the values of all other characteristics. Two modes are supported:

- One at a Time : For the restrictable cstic selected by user, domains are calculated for that particular restrictable cstic. This is enabled by setting ‘userDomainPreview’ flag as true, and is the same as in IPC.
- All at Once: New algorithms enables SSC to:
  - Calculate user domains for every restrictable cstic at once, resulting in a simplified UI experience.
  - Handle domain conflicts, remove duplicates, and flag them adequately.

This is enabled by setting ‘userDomainCalculation’ flag as true in SSC.

Irrespective of the mode, it is possible to see the set of disallowed values in SME for easier modeling purpose .

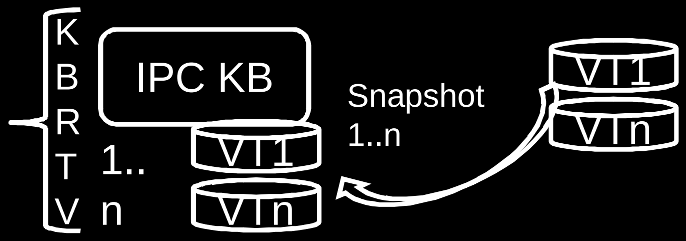
Care should be taken to limit the number of cstics to be restrictable to one’s which are used for user inputs on the UI to avoid performance degradation.

External Variant and Text Tables

External Variant Tables enables business users to maintain configuration data without the need of regenerating the configuration model.

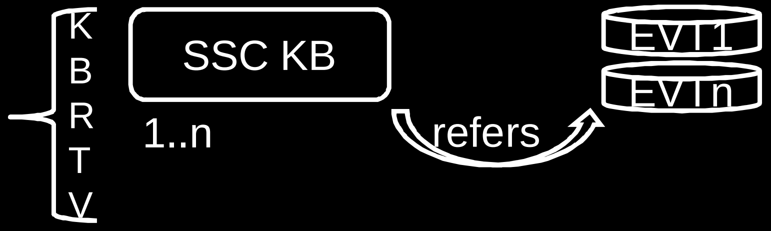
Similarly, External Texts enables management of Language text without generating the KBs again.

Classical Embedded Tables



- ❖ Requires KBRTV generation for any change in variant table data
- ❖ Multiple copies exists in the database and in SSC engine cache

SSC External Tables



- ❖ EVT’s can be mapped to a database table.
- ❖ Only one copy exists in db and in SSC engine cache
- ❖ KBRTV is not required to be generated for any change in data by business users
- ❖ Effectivity of the data is controlled by SSC_FROM_DATE and SSC_TO_DATE

Exclusions , Restrictions , Conflicts and Aggregations

- Exclusions: Exclude features which are incompatible with already selected features. Supported through ‘excluded values’ attribute facet which can also be filled dynamically.

- Restrictions: Restrict possible feature selections to only those which are compatible with already selected features. Supported through ‘dynamic domain’ attribute facet which can also be filled dynamically.

- Conflicts: Selection of a feature which is incompatible with previous selection results in a conflict. Flexible messaging capability with any number of message objects dynamically created with guidance on why this is an incompatible selection and how it can be resolved.

- Aggregation: Attributes of individual parts can contribute to summation at different levels, for example, weight, length, height, depth, bandwidth, capacity, number of ports, and number of slots etc. SSC provides reliable and declarative syntax for calculating aggregations.

Configuration Restore

SSC allows a user to quickly start a new configuration process based upon a previous saved configuration. Configuration is restored against the current valid KB and only valid options are listed. Two modes of restore are supported :

Classical Restore

- Default saving of configuration state. Final state of configuration is saved in xml.
- Does not require modeling changes as instances are created and deleted dynamically during restore process by SSC engine.
- Suitable for most configurations without many custom user exits.
- Restore performance degrades with extremely large configuration due to heavy engine activity.
- If custom user exits are used, then special handling of restore scenarios need to be accounted for .

User Input Based Restore

- Specialized saving of ‘User Action’ in configuration xml.
- Requires small modeling change to uniquely identify the instances during restore.
- Accomplishes restore by way of executing the recorded configuration steps in sequence.
- Simplifies custom user exit development process since restore specific coding is not required
- Performance consistent to initial configuration in case steps are optimized.

Modeling Frameworks and Engine Extensions

- Frameworks: To reduce redundant and repetitive model development activities, a set of core models carrying definitions and behaviors can be grouped together which can act as a base framework . This framework can then be used by other models simply by project referencing capabilities of eclipse workbench.
- Engine Extensions: SSC engine provide special hooks or user exits for customer specific and general purpose extension of SSC engine .
  - Using these hooks, it is possible to provide triggers to the customer specific code to meet any specific requirements.
  - Since custom code might change frequently, it is possible to deploy these changes without requiring a system restart.

Scaling and Performance

- Industry standard AI algorithms have been significantly modified and improved for performance and memory optimization.
- Delta Bean concept for data transfer optimization for each update.
- Dynamic variant table views to limit set of data to be processed by constraint or rules for faster processing.
- On-demand instantiation of data objects.
- Significant enhancements in basic data structures used in algorithms for performance.
- Hyper scaling through NW Java or Hybris clusters.

SAP SSC
Solution Modeling Environment

PUBLIC

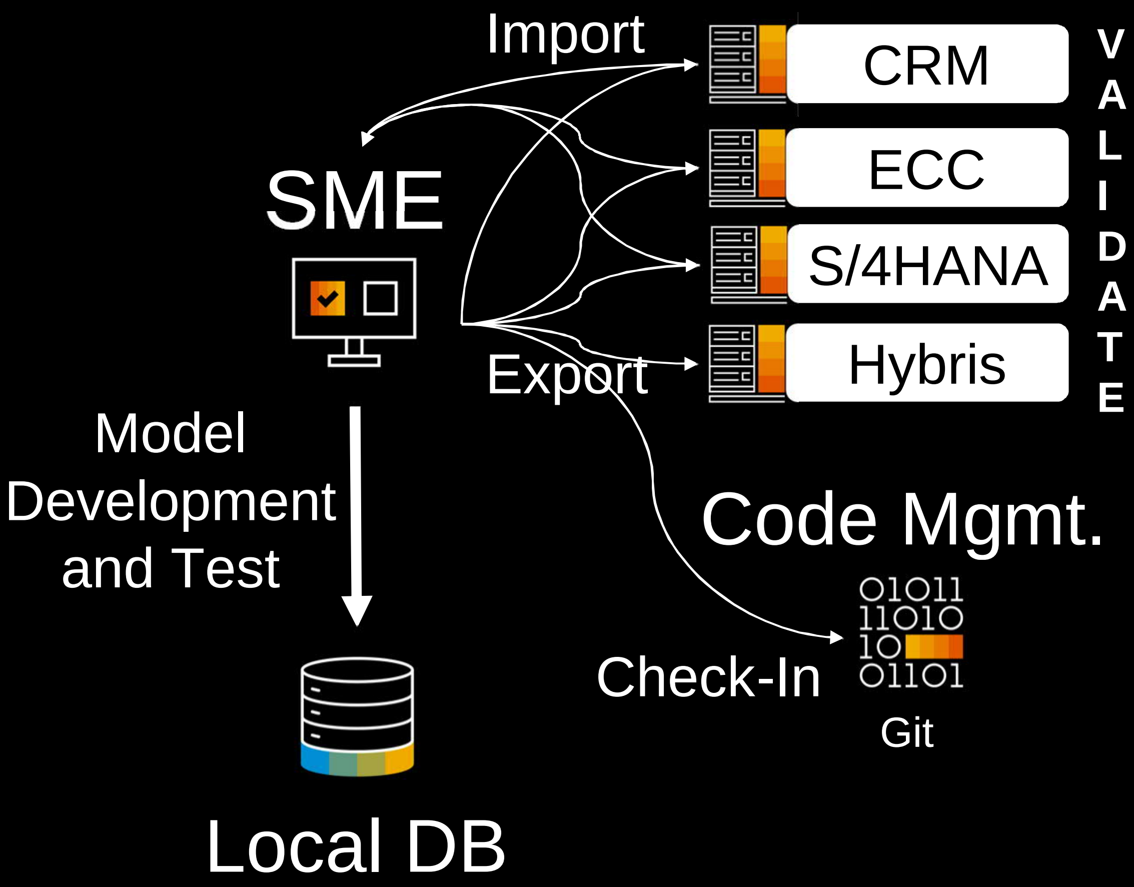
THE BEST RUN SAP

About SSC Solution Modeling Environment (SME)

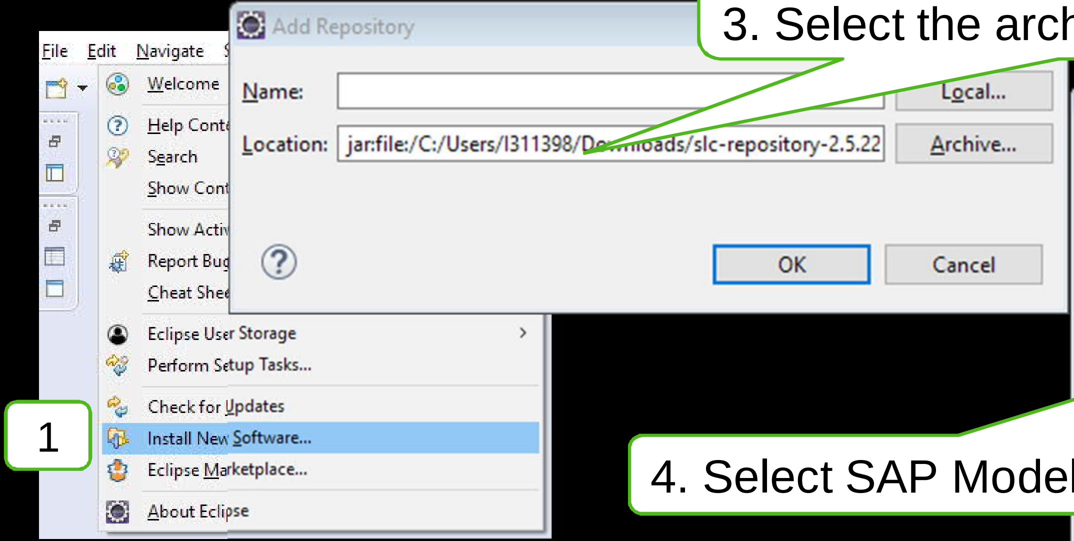
- Eclipse plug-In facilitating building of complex solution configuration models.
- Enables development in both online and offline modes .
- Modeler’s working in distributed development environment setup can collaborate in agile mode to build big and complex solutions.
- VC and IPC models can be reused and wrapped in solution models.
- Devops support for automated continuous build and regression testing of solution models.

Solution Model Development Process

- Master data can be downloaded from S/4HANA , ERP, or CRM systems.
- Solution models can be developed and tested against a local database.
- Models can be versioned in source code management repositories, like Git, for collaboration with other modeler's.
- Solution models can be exported to backend systems where they are integrated with NW Java or Hybris.
- Solution models can be exported to a database connected with NW Java or Hybris in 'Ready To Integrate' scenario mode .
- SME can operate in 'headless' mode , supporting automated KB generation/export and testing.



Installing SME in Eclipse

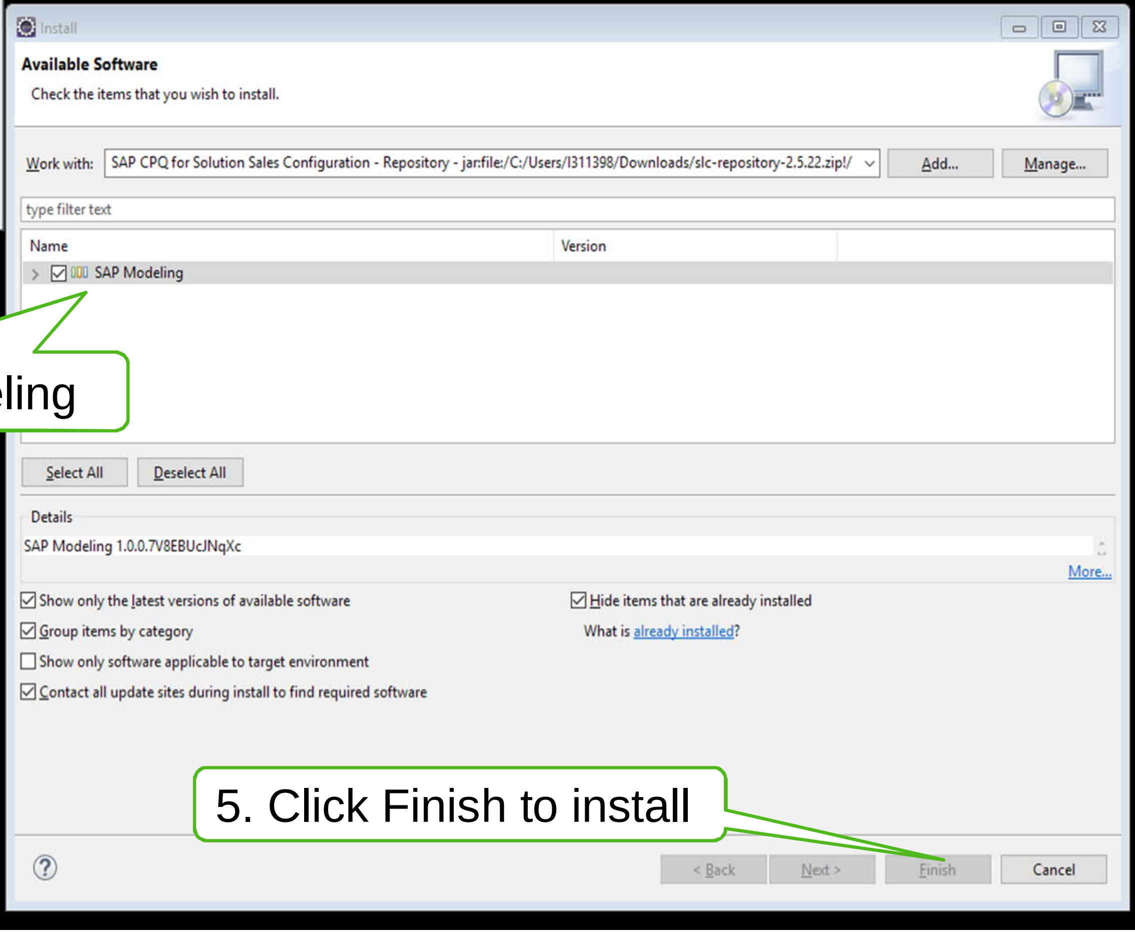


1

Name:

Location: jar:file:/C:/Users/I311398/Downloads/slc-repository-2.5.22

2. Download latest SME zip from SMP
3. Select the archive File



Available Software

Check the items that you wish to install.

Work with: SAP CPQ for Solution Sales Configuration - Repository - jar:file:/C:/Users/I311398/Downloads/slc-repository-2.5.22.zip!/

type filter text

Name	Version
☑ SAP Modeling	

4. Select SAP Modeling

Details

SAP Modeling 1.0.0.7V8EBUcJNqXc

☑ Show only the latest versions of available software

☑ Group items by category

☐ Show only software applicable to target environment

☑ Contact all update sites during install to find required software

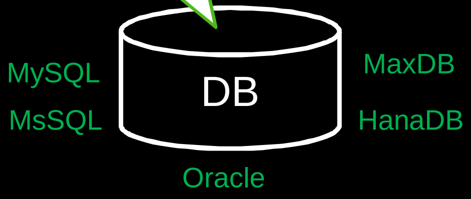
☑ Hide items that are already installed

What is already installed?

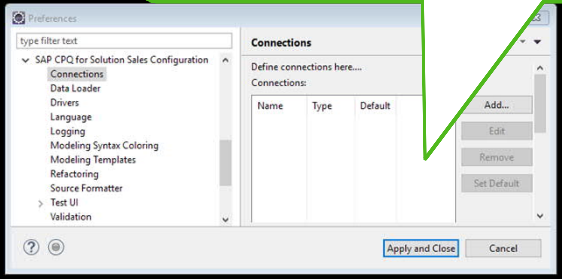
5. Click Finish to install

Setting Local Database and Connections

1. Create a database in the DB



2. Add Connection



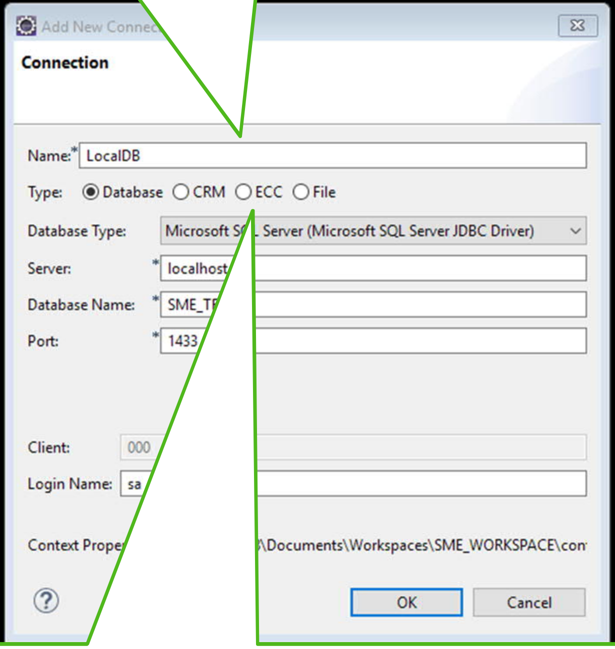
Connections

Define connections here....

Connections:

Name	Type	Default

3. Specify connection and DB details



Connection

Name:* LocalDB

Type: ● Database ○ CRM ○ ECC ○ File

Database Type: Microsoft S L Server (Microsoft SQL Server JDBC Driver)

Server:* localhost

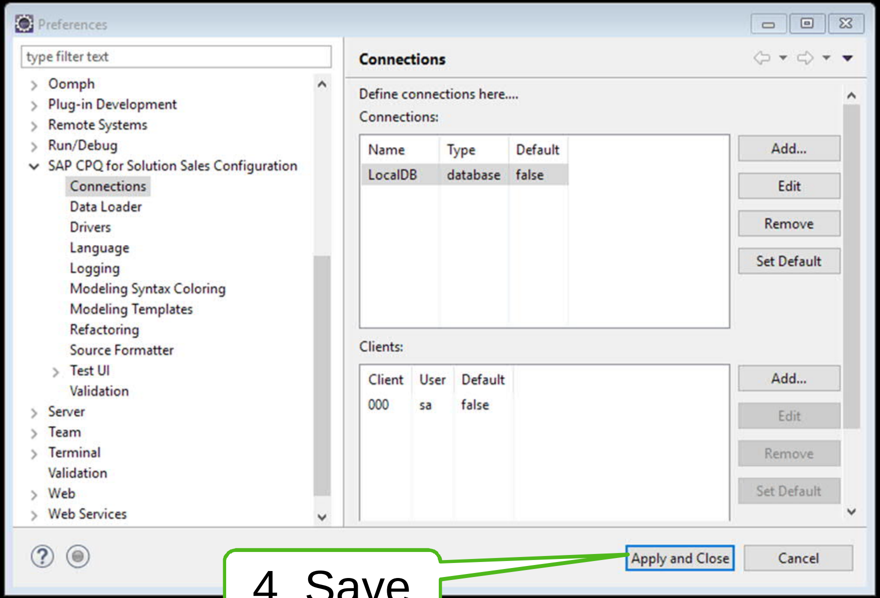
Database Name:* SME_T

Port:* 1433

Client: 000

Login Name: sa

Context Prope \Documents\Workspaces\SME_WORKSPACE\con



Connections

Define connections here....

Connections:

Name	Type	Default
LocalDB	database	false

Clients:

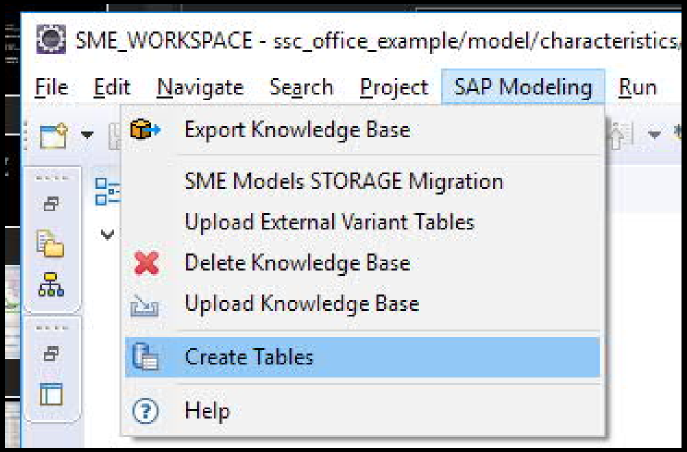
Client	User	Default
000	sa	false

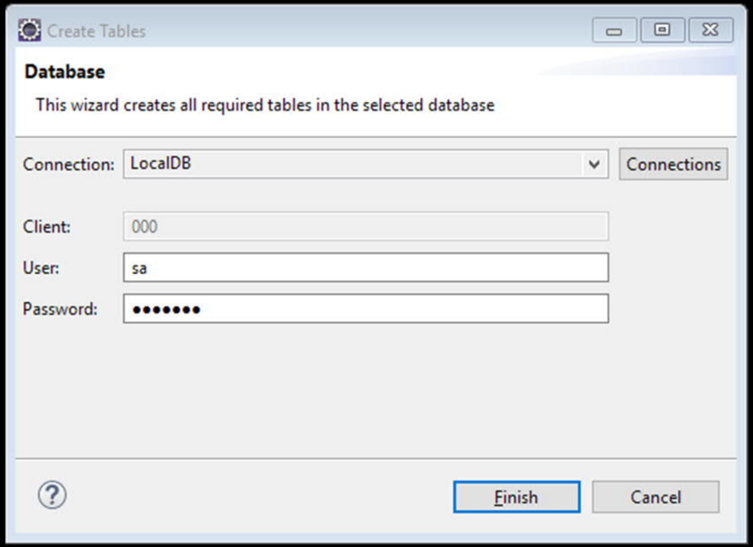
4. Save

Connections with ERP/CRM can also be created

Initializing Local Database

- Initialize the database with SSC schema and tables for a knowledge base to be exported for testing





Database

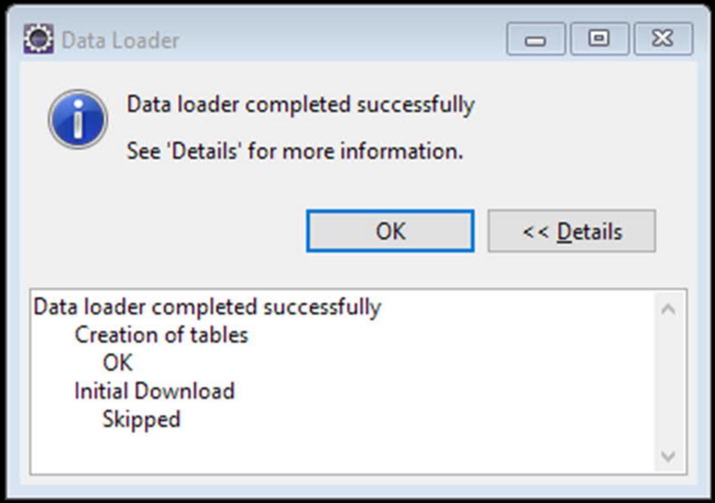
This wizard creates all required tables in the selected database

Connection: LocalDB

Client: 000

User: sa

Password: •••••••



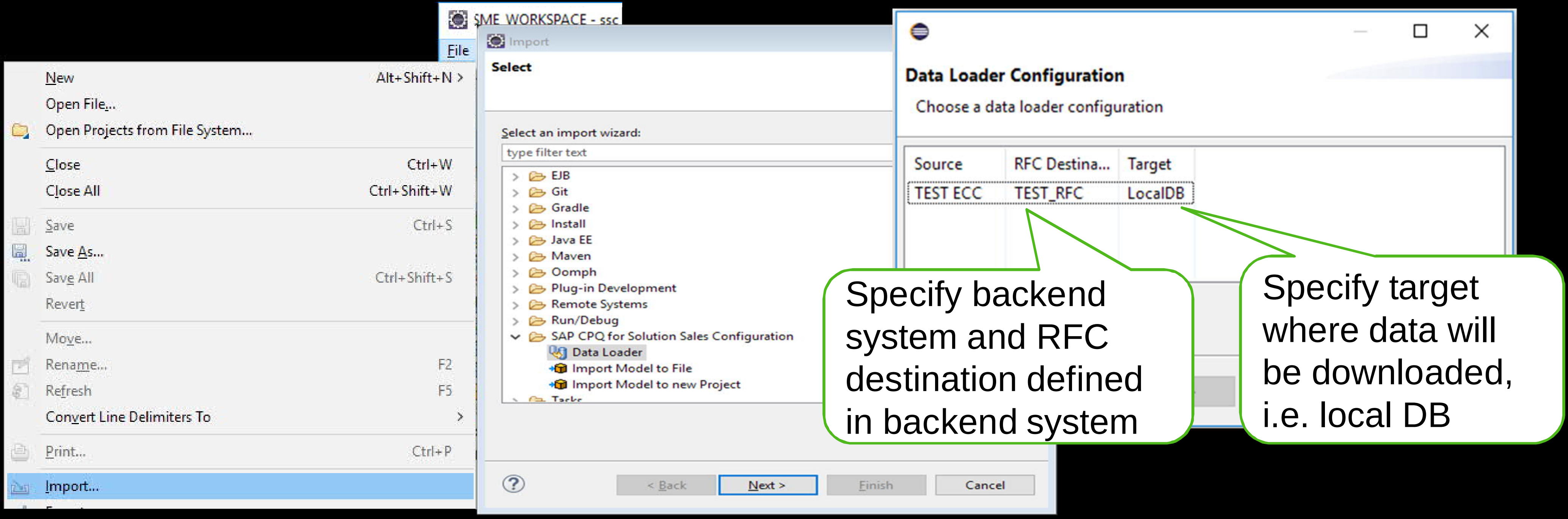
Data loader completed successfully

See 'Details' for more information.

```
Data loader completed successfully
   Creation of tables
      OK
   Initial Download
      Skipped
```

Downloading Master Data

- Dataloader is integrated in SME and can be use to download VC/IPC/SSC KBs and materials from an ECC/S4/CRM backend to the local database.
- Enables reuse of VC/IPC models (configurable products) in solution models for faster development and testing of the complete solution involving knowledge base orchestration.



Data Loader Configuration

Choose a data loader configuration

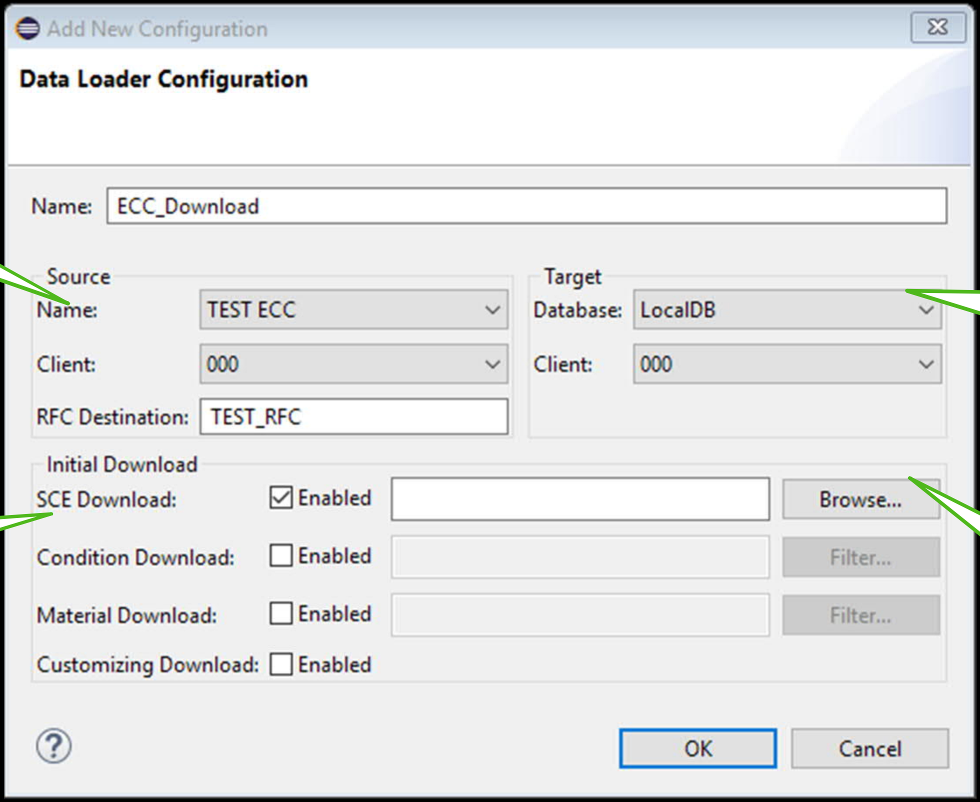
Source	RFC Destina...	Target
TEST ECC	TEST_RFC	LocalDB

Specify backend system and RFC destination defined in backend system

Specify target where data will be downloaded, i.e. local DB

Configuring Data Loader

Data loader enables fetching of master data from ECC/CRM systems into Local database for solution model development and testing.



Add New Configuration

Data Loader Configuration

Name: ECC_Download

Source

Name: TEST ECC

Client: 000

RFC Destination: TEST_RFC

Target

Database: LocalDB

Client: 000

Initial Download

SCE Download: ☑ Enabled

Condition Download: ☐ Enabled

Material Download: ☐ Enabled

Customizing Download: ☐ Enabled

Source backend system details

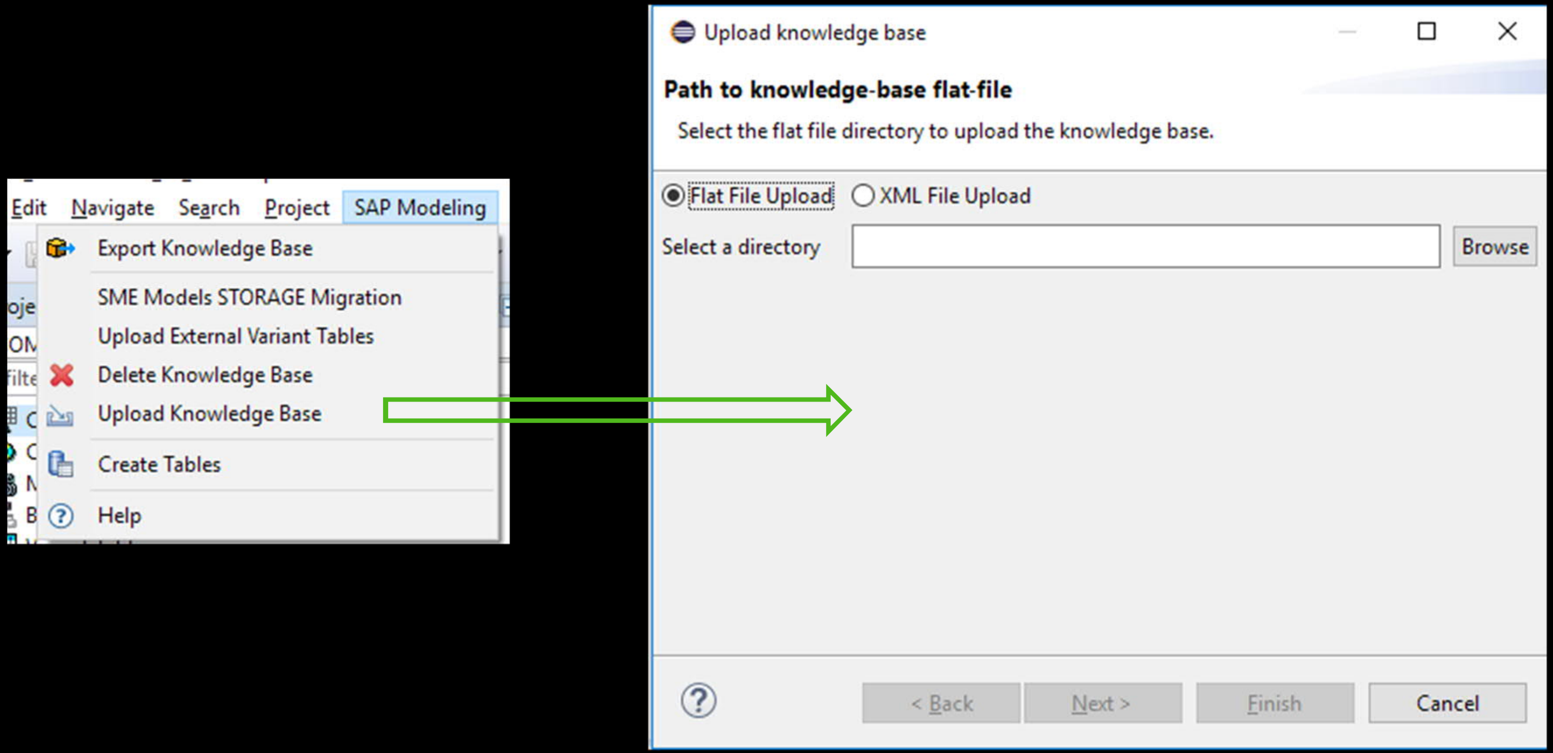
Selective data type download

Target local database connection details

Filters to reduce scope of data to be downloaded

Uploading Classical KBs Manually

In case needed, VC / IPC KBs can be downloaded from backend in xml/flat file formats and uploaded to local DB for development and testing purposes.



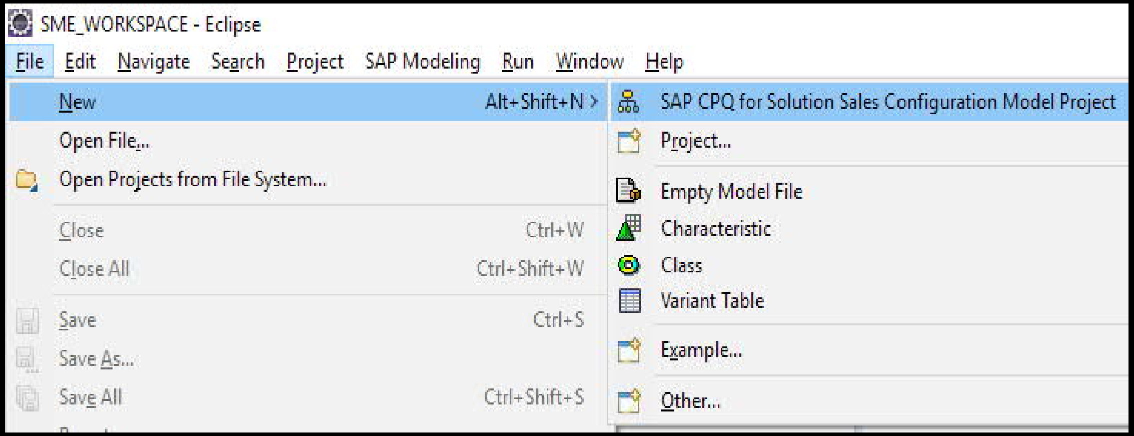
Path to knowledge-base flat-file

Select the flat file directory to upload the knowledge base.

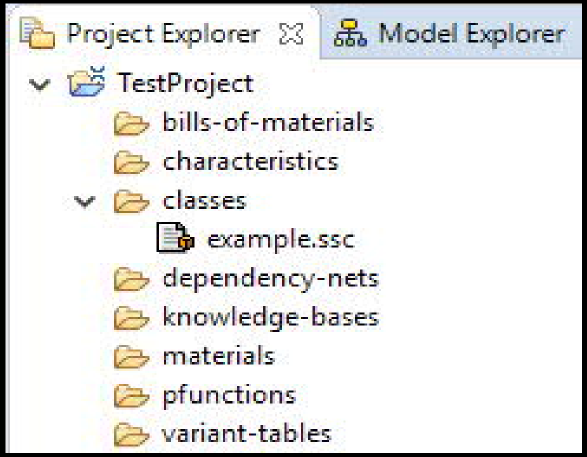
● Flat File Upload ○ XML File Upload

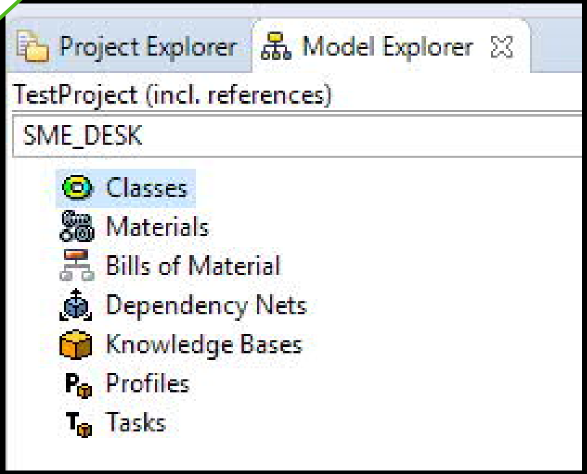
Select a directory

Creating a New Model Project



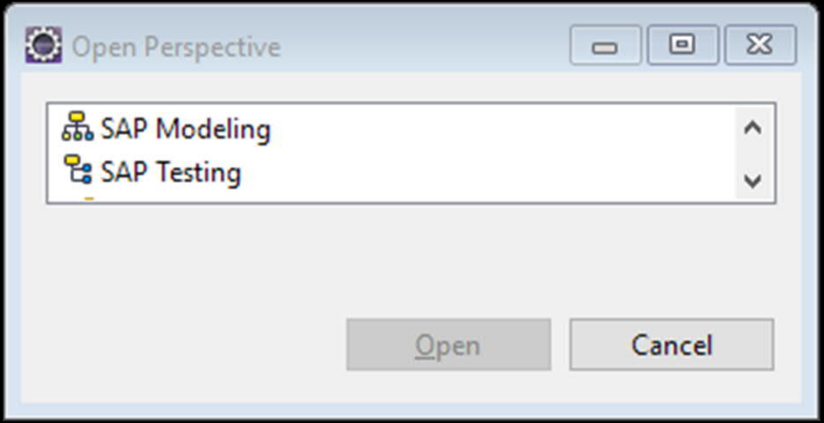
Creates basic project structure



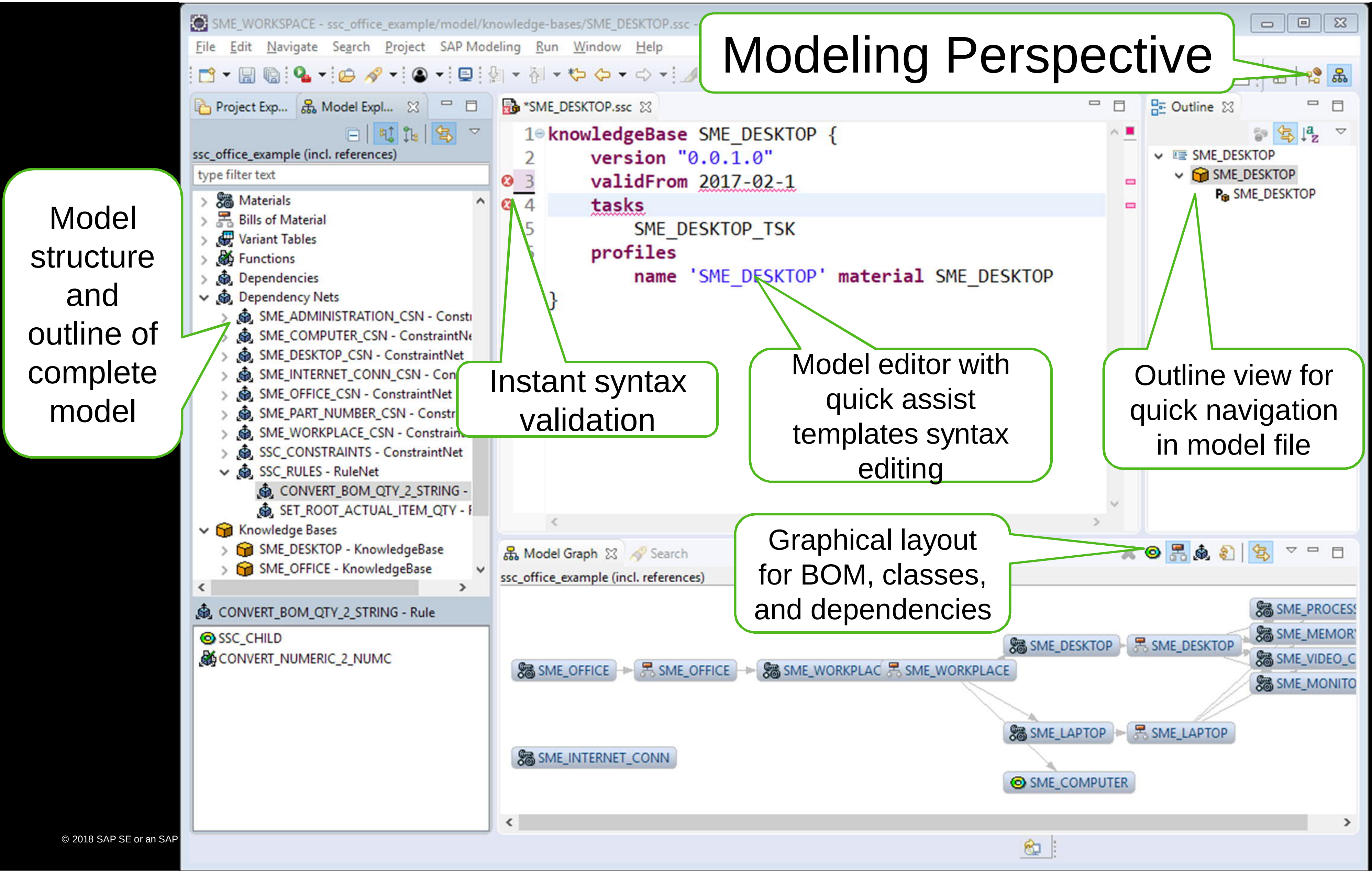


WorkBench

- Modeling Perspective: Collection of views facilitating development of a solution model
  - Model Explorer
  - Model Editor
  - Model Graph
  - Model Outline
- Testing Perspective: Collection of views facilitating testing of solution model
  - Test Runner
  - Conflicts and Justifications
  - Profiling and Tracing



Modeling Perspective


SME_WORKSPACE - ssc_office_example/model/knowledge-bases/SME_DESKTOP.ssc -
File Edit Navigate Search Project SAP Modeling Run Window Help
Project Exp...
Model Expl...
ssc_office_example (incl. references)
type filter text
Materials
Bills of Material
Variant Tables
Functions
Dependencies
Dependency Nets
SME_ADMINISTRATION_CSN - Constr
SME_COMPUTER_CSN - ConstraintNe
SME_DESKTOP_CSN - ConstraintNet
SME_INTERNET_CONN_CSN - Con
SME_OFFICE_CSN - ConstraintNet
SME_PART_NUMBER_CSN - Constr
SME_WORKPLACE_CSN - Constrain
SSC_CONSTRAINTS - ConstraintNet
SSC_RULES - RuleNet
CONVERT_BOM_QTY_2_STRING -
SET_ROOT_ACTUAL_ITEM_QTY - F
Knowledge Bases
SME_DESKTOP - KnowledgeBase
SME_OFFICE - KnowledgeBase
CONVERT_BOM_QTY_2_STRING - Rule
SSC_CHILD
CONVERT_NUMERIC_2_NUMC
*SME_DESKTOP.ssc
Outline
SME_DESKTOP
SME_DESKTOP
SME_DESKTOP
Model Graph
Search
ssc_office_example (incl. references)
SME_OFFICE
SME_OFFICE
SME_WORKPLAC
SME_WORKPLACE
SME_DESKTOP
SME_DESKTOP
SME_PROCESS
SME_MEMOR
SME_VIDEO_C
SME_MONITO
SME_LAPTOP
SME_LAPTOP
SME_COMPUTER
SME_INTERNET_CONN

```
knowledgeBase SME_DESKTOP {
    version "0.0.1.0"
    validFrom 2017-02-1
    tasks
        SME_DESKTOP_TSK
    profiles
        name 'SME_DESKTOP' material SME_DESKTOP
}
```

Modeling Perspective

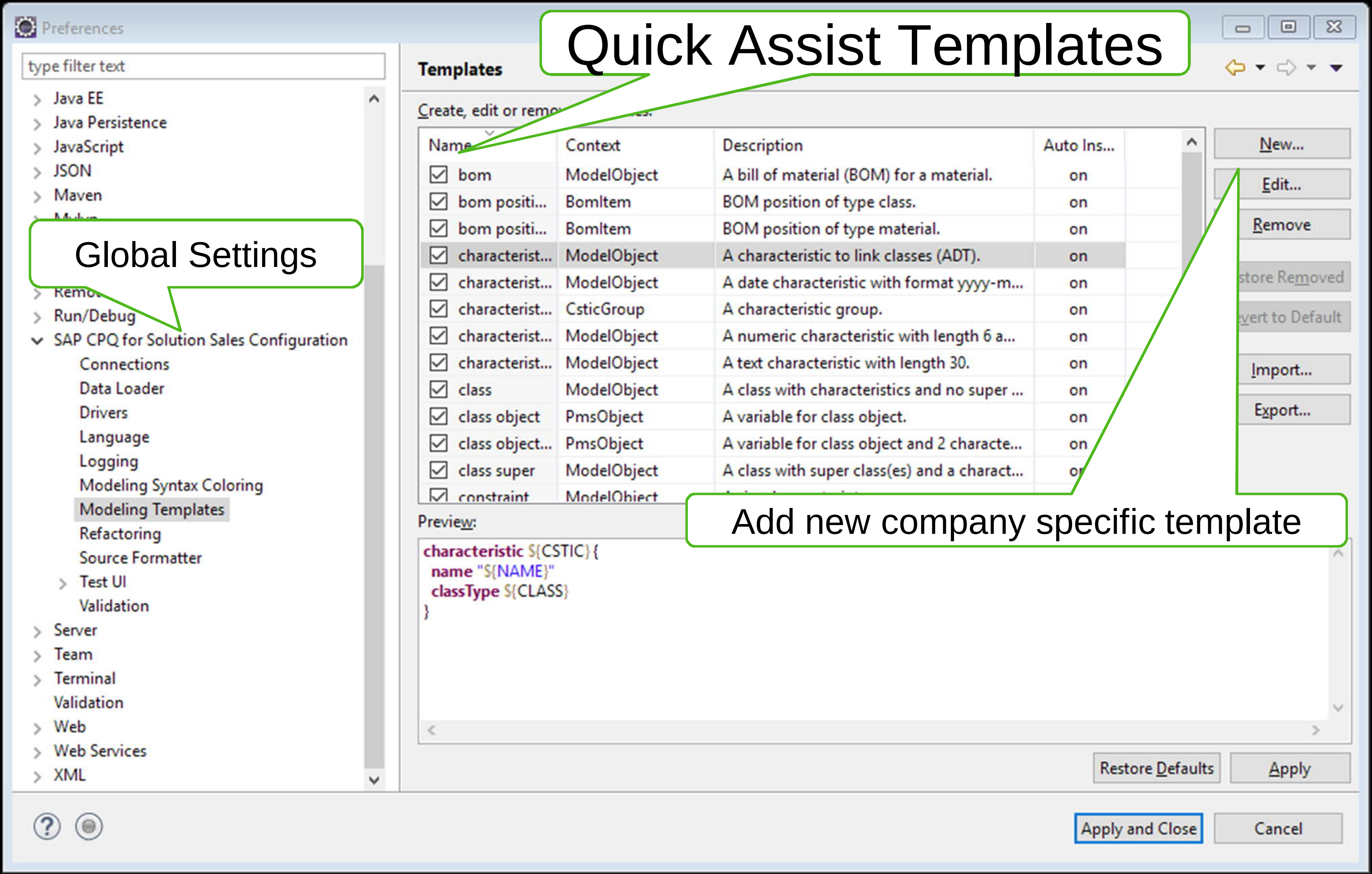
Model structure and outline of complete model

Instant syntax validation

Model editor with quick assist templates syntax editing

Outline view for quick navigation in model file

Graphical layout for BOM, classes, and dependencies



Quick Assist Templates

Global Settings

Add new company specific template

Preferences

type filter text

- Java EE
- Java Persistence
- JavaScript
- JSON
- Maven
- Mylyn
- Remo
- Run/Debug
- SAP CPQ for Solution Sales Configuration
  - Connections
  - Data Loader
  - Drivers
  - Language
  - Logging
  - Modeling Syntax Coloring
  - Modeling Templates
  - Refactoring
  - Source Formatter
  - Test UI
  - Validation
- Server
- Team
- Terminal
- Validation
- Web
- Web Services
- XML

Templates

Create, edit or remo

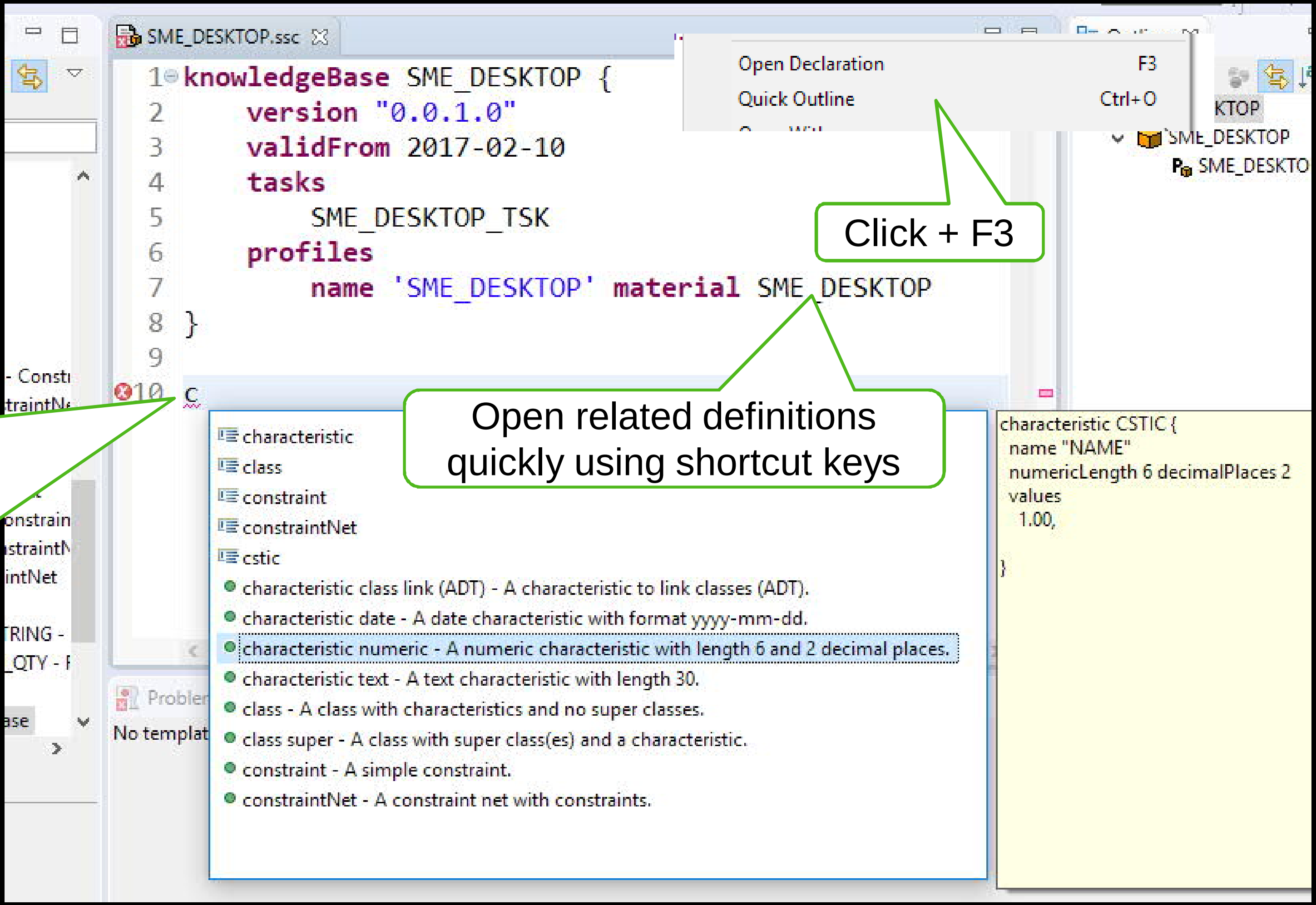
	Name	Context	Description	Auto Ins...
☑	bom	ModelObject	A bill of material (BOM) for a material.	on
☑	bom positi...	BomItem	BOM position of type class.	on
☑	bom positi...	BomItem	BOM position of type material.	on
☑	characterist...	ModelObject	A characteristic to link classes (ADT).	on
☑	characterist...	ModelObject	A date characteristic with format yyyy-m...	on
☑	characterist...	CsticGroup	A characteristic group.	on
☑	characterist...	ModelObject	A numeric characteristic with length 6 a...	on
☑	characterist...	ModelObject	A text characteristic with length 30.	on
☑	class	ModelObject	A class with characteristics and no super ...	on
☑	class object	PmsObject	A variable for class object.	on
☑	class object...	PmsObject	A variable for class object and 2 characte...	on
☑	class super	ModelObject	A class with super class(es) and a charact...	on
☑	constraint	ModelObject		

Preview:

```
characteristic ${CSTIC} {
 name "${NAME}"
 classType ${CLASS}
}
```

Quick Assist can be invoked using Ctrl + Space in the model editor.

All possible template options are shown with details and relevant selection can be made for faster creation of model constructs.



```
knowledgeBase SME_DESKTOP {
    version "0.0.1.0"
    validFrom 2017-02-10
    tasks
        SME_DESKTOP_TSK
    profiles
        name 'SME_DESKTOP' material SME_DESKTOP
}

c
```

- characteristic
- class
- constraint
- constraintNet
- cstic
- characteristic class link (ADT) - A characteristic to link classes (ADT).
- characteristic date - A date characteristic with format yyyy-mm-dd.
- characteristic numeric - A numeric characteristic with length 6 and 2 decimal places.
- characteristic text - A text characteristic with length 30.
- class - A class with characteristics and no super classes.
- class super - A class with super class(es) and a characteristic.
- constraint - A simple constraint.
- constraintNet - A constraint net with constraints.

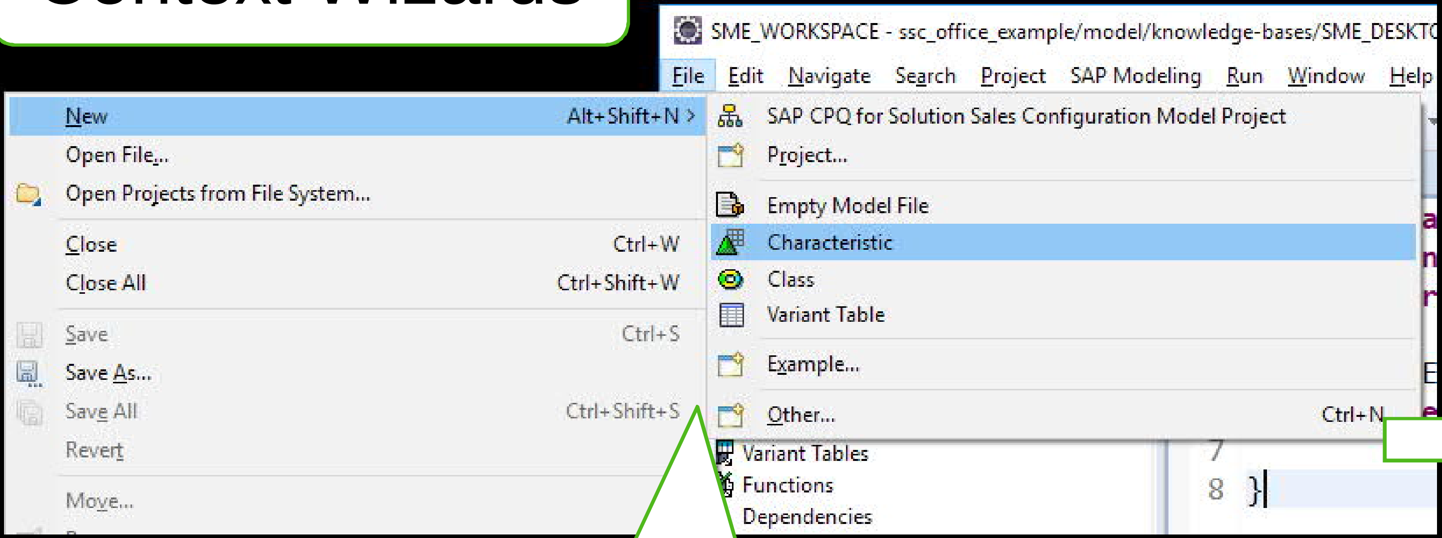
```
characteristic CSTIC {
 name "NAME"
 numericLength 6 decimalPlaces 2
 values
   1.00,

}
```

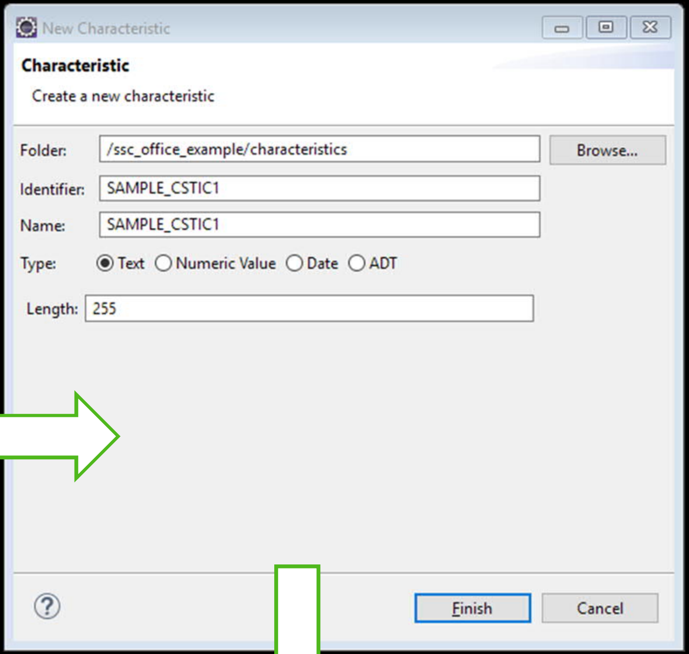
Click + F3

Open related definitions quickly using shortcut keys

Context Wizards



SME_WORKSPACE - ssc_office_example/model/knowledge-bases/SME_DESKTO



New Characteristic

Characteristic

Create a new characteristic

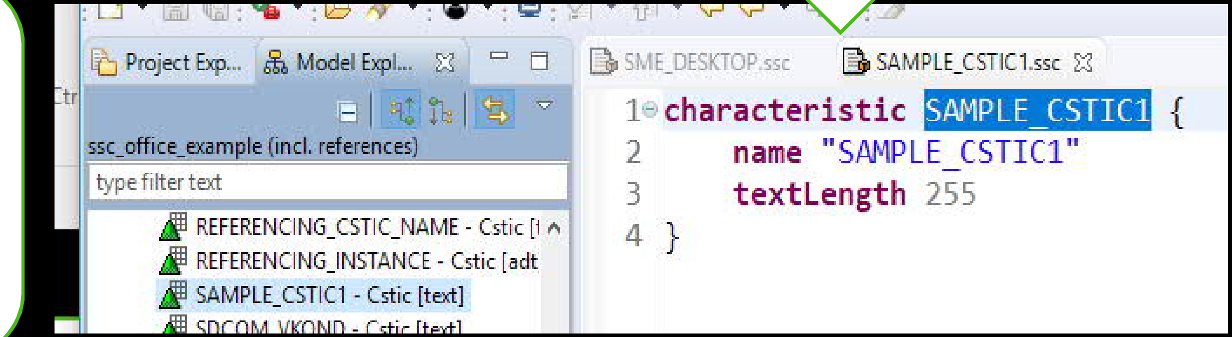
Folder: /ssc_office_example/characteristics [Browse...]

Identifier: SAMPLE_CSTIC1

Name: SAMPLE_CSTIC1

Type: ● Text ○ Numeric Value ○ Date ○ ADT

Length: 255

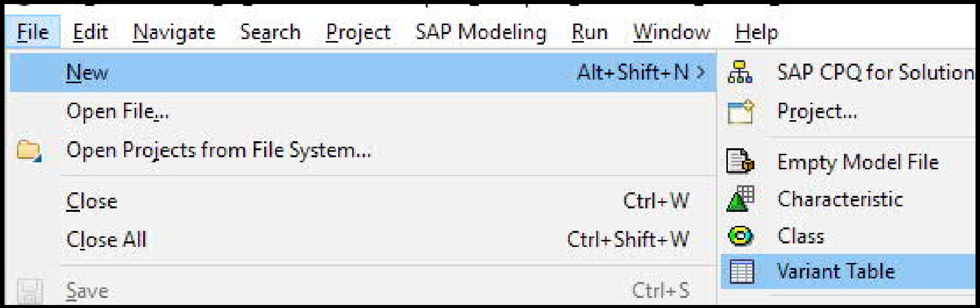


Project Exp...
Model Expl...
SME_DESKTOP.ssc
SAMPLE_CSTIC1.ssc
ssc_office_example (incl. references)
type filter text
REFERENCING_CSTIC_NAME - Cstic [t
REFERENCING_INSTANCE - Cstic [adt
SAMPLE_CSTIC1 - Cstic [text]
SDCOM_VKOND - Cstic [text]

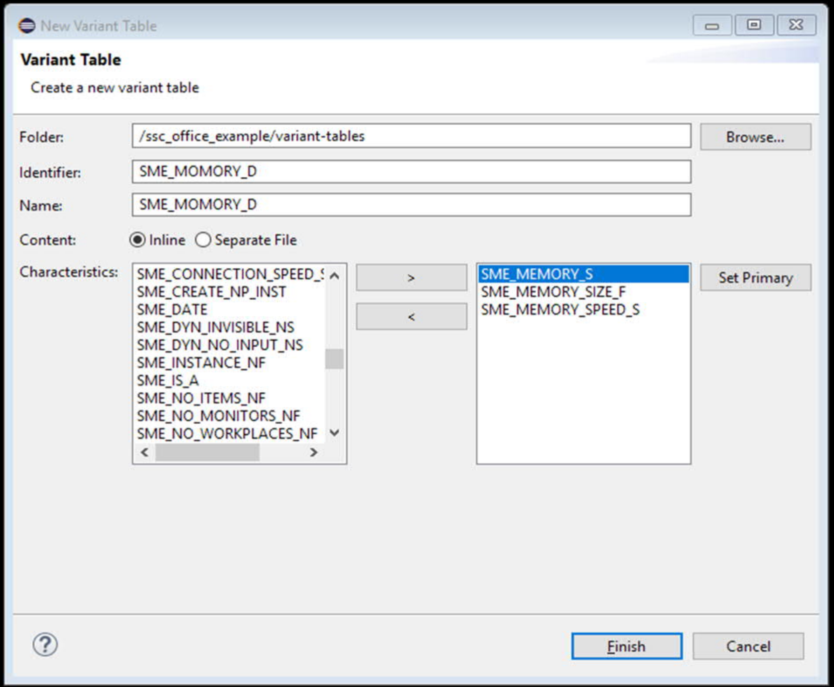
```
characteristic SAMPLE_CSTIC1 {
    name "SAMPLE_CSTIC1"
    textLength 255
}
```

- Easy-to-use wizards for rapid creation of classes, characteristics, and variant table definitions.
- Model object are automatically generated and included in the model structure

Variant Table Wizard



Easy-to-use wizard for creation and maintenance of variant tables



Variant Table

Create a new variant table

Folder: /ssc_office_example/variant-tables

Identifier: SME_MOMORY_D

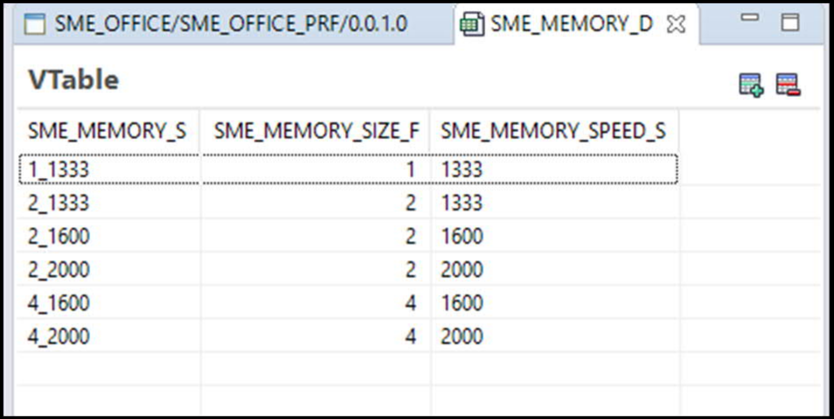
Name: SME_MOMORY_D

Content: ● Inline ○ Separate File

Characteristics:

SME_CONNECTION_SPEED_S
SME_CREATE_NP_INST
SME_DATE
SME_DYN_INVISIBLE_NS
SME_DYN_NO_INPUT_NS
SME_INSTANCE_NF
SME_IS_A
SME_NO_ITEMS_NF
SME_NO_MONITORS_NF
SME_NO_WORKPLACES_NF

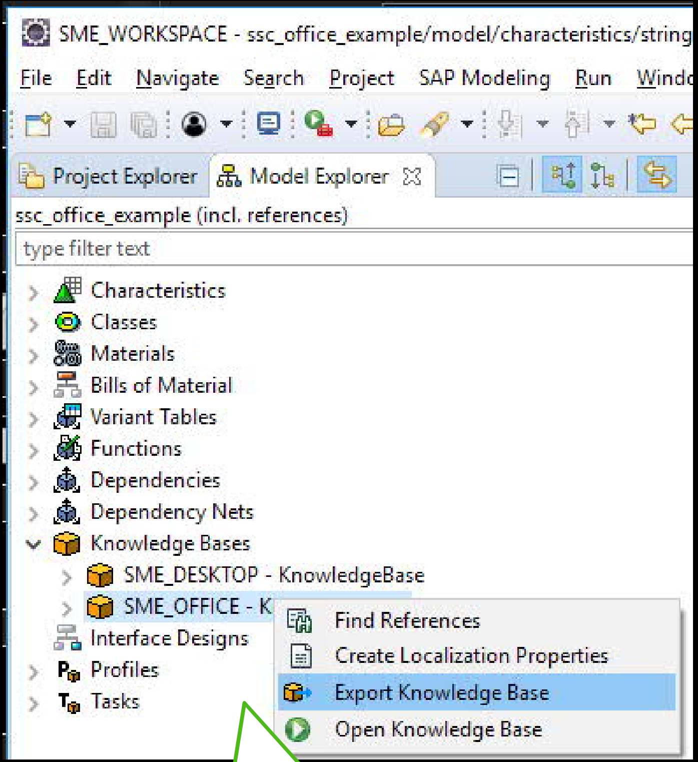
SME_MEMORY_S
SME_MEMORY_SIZE_F
SME_MEMORY_SPEED_S



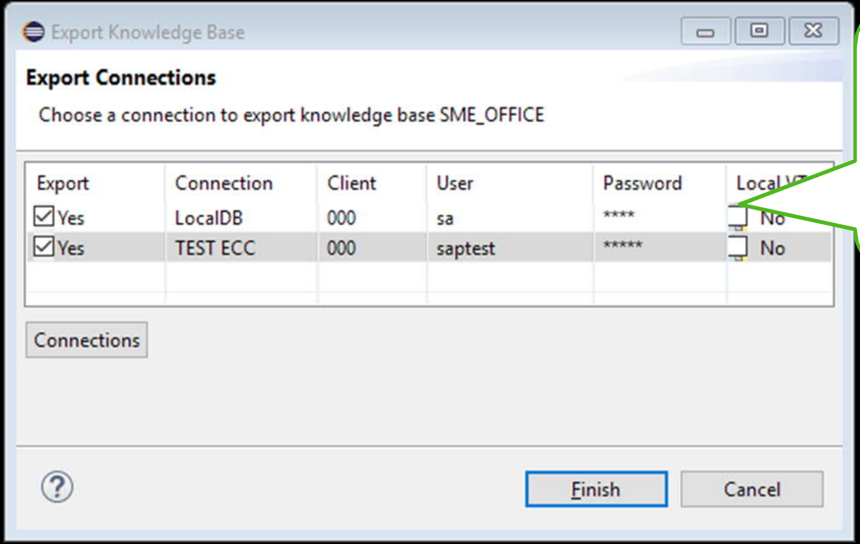
VTable

SME_MEMORY_S	SME_MEMORY_SIZE_F	SME_MEMORY_SPEED_S
1_1333	1	1333
2_1333	2	1333
2_1600	2	1600
2_2000	2	2000
4_1600	4	1600
4_2000	4	2000

Exporting Solution Models to Generate Runtime Version (KBRTV) for Testing



1. Select KB

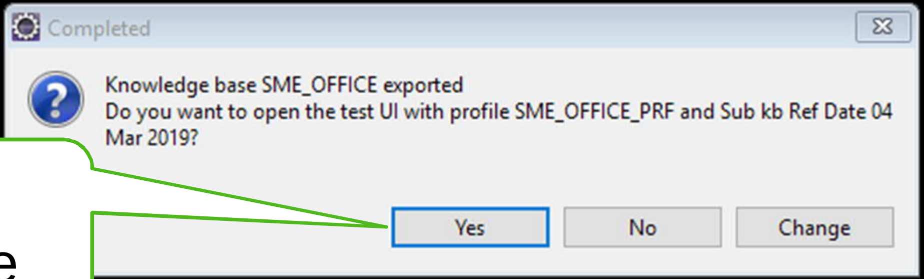


Export Connections

Choose a connection to export knowledge base SME_OFFICE

Export	Connection	Client	User	Password	Local VT
☑ Yes	LocalDB	000	sa	****	□ No
☑ Yes	TEST ECC	000	saptest	*****	□ No

2. Select connections. KBs can be exported to multiple destinations simultaneously

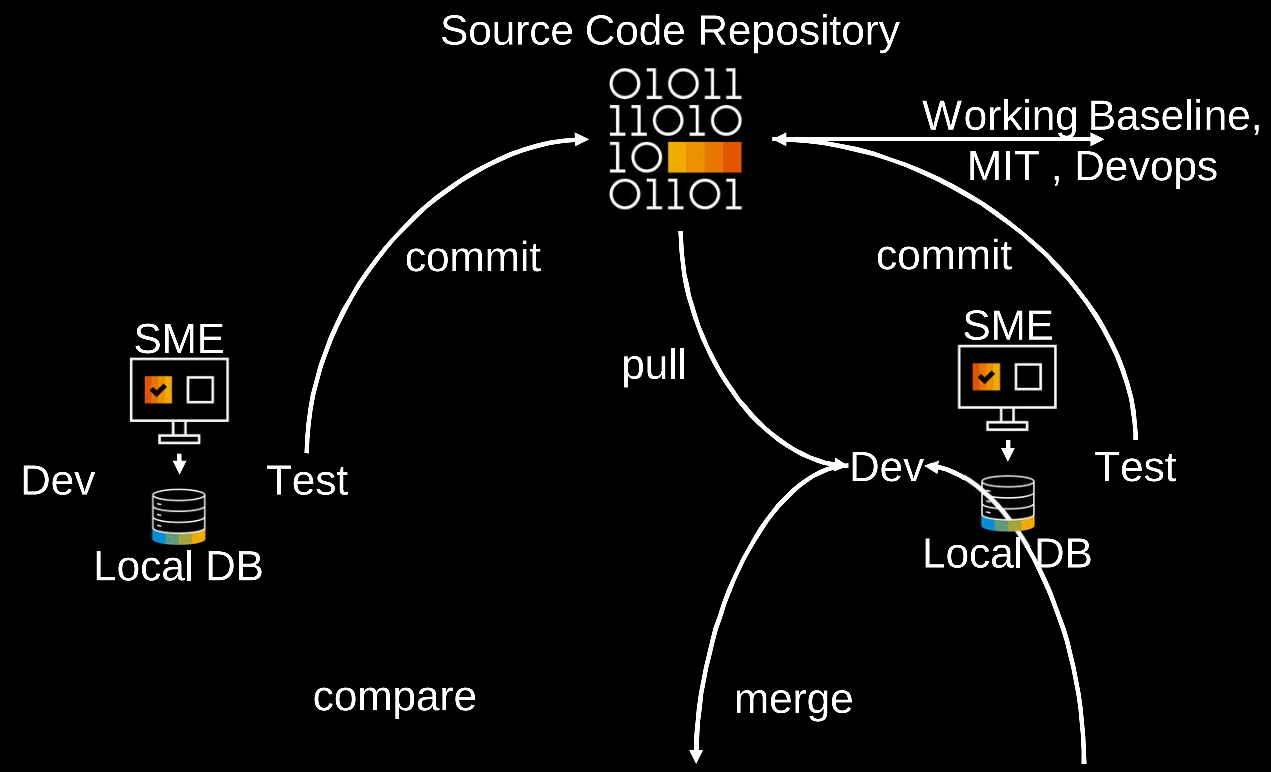


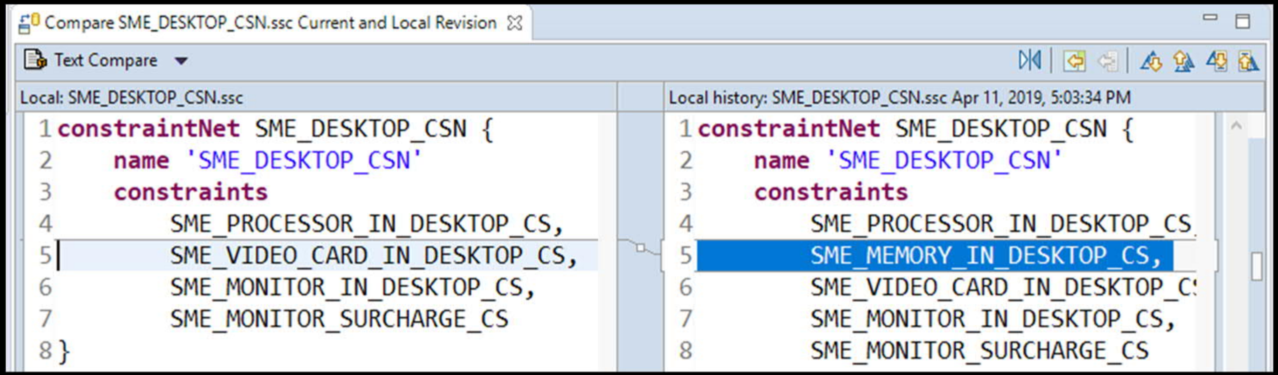
Knowledge base SME_OFFICE exported
Do you want to open the test UI with profile SME_OFFICE_PRF and Sub kb Ref Date 04 Mar 2019?

3. Opens KB in testing perspective

Collaborative Development

- Developers can collaborate through any source code repository like Git :
  - Split large solution model development amongst modelers
  - Develop and test Individually . Commit working model and tests into repository
  - Merge your changes with changes from other modelers. Test and commit
  - Create baselines of working software in repository
  - Setup devops process on the central repository





Local: SME_DESKTOP_CSN.ssc

```
constraintNet SME_DESKTOP_CSN {
    name 'SME_DESKTOP_CSN'
    constraints
        SME_PROCESSOR_IN_DESKTOP_CS,
        SME_VIDEO_CARD_IN_DESKTOP_CS,
        SME_MONITOR_IN_DESKTOP_CS,
        SME_MONITOR_SURCHARGE_CS
}
```

Local history: SME_DESKTOP_CSN.ssc Apr 11, 2019, 5:03:34 PM

```
constraintNet SME_DESKTOP_CSN {
    name 'SME_DESKTOP_CSN'
    constraints
        SME_PROCESSOR_IN_DESKTOP_CS
        SME_MEMORY_IN_DESKTOP_CS,
        SME_VIDEO_CARD_IN_DESKTOP_C
        SME_MONITOR_IN_DESKTOP_CS,
        SME_MONITOR_SURCHARGE_CS
```

Testing Perspective

Solution Model Testing Process

SME facilitates the iterative mode of model development process which is usually the norm. Development through SME involves :

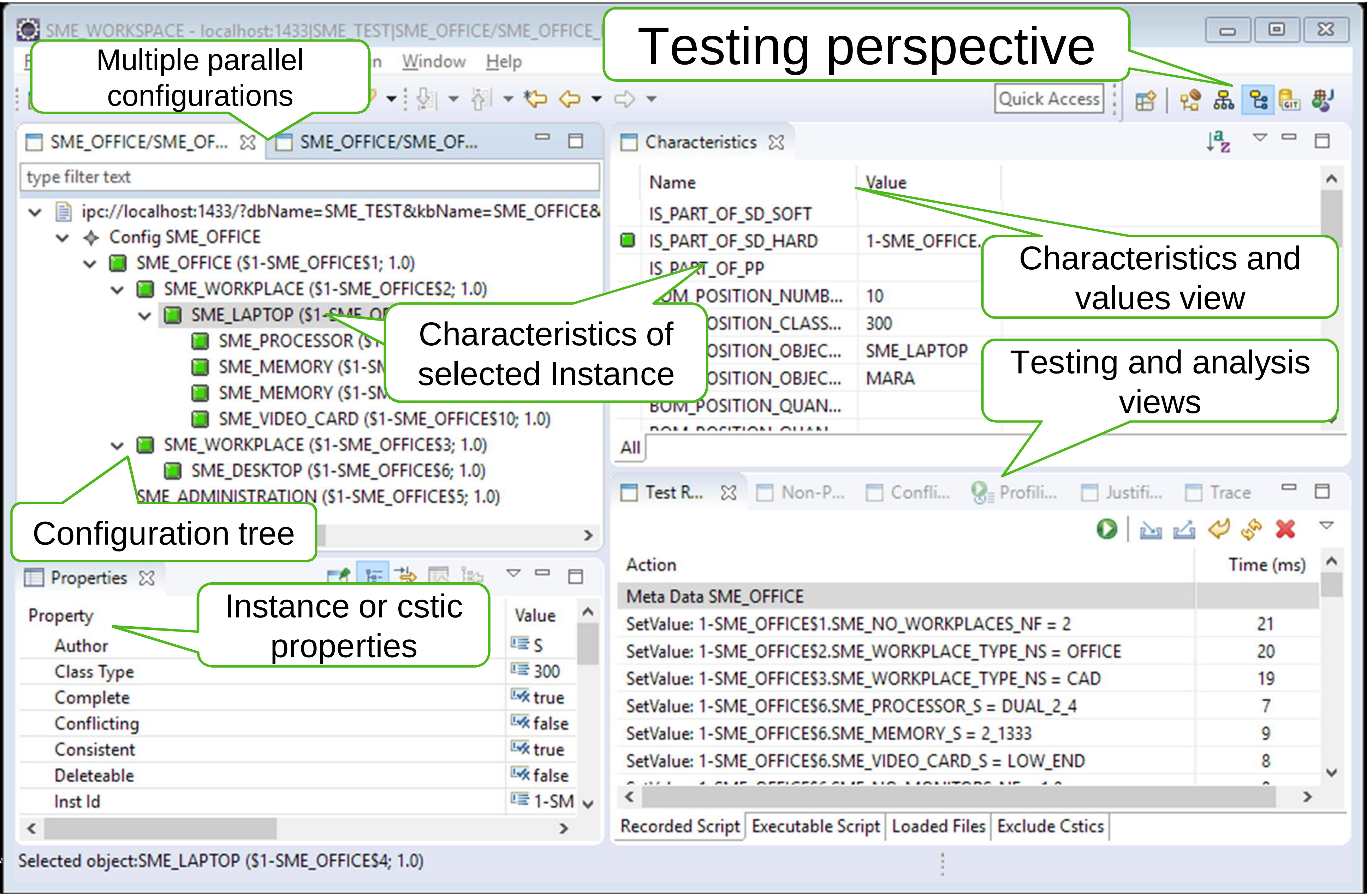
- Development of a feature, its runtime version generation by exporting to a local database.
- Testing of feature for functional and performance aspects.
- Regression testing of existing solution model using recorded test scripts.
- Test scripts generation/enhancement of existing scripts for subsequent regression.

Above steps can be repeated till the desired functionality is achieved and solution models are rolled in production.

Subsequently, the setup of continuous delivery pipeline using devops support can ensure faster delivery of changes into production with high quality.

SME Testing Perspective

- Standalone testing workbench for testing of solution models. Use same engine as used in NW Java server thus ensuring same behavior.
- Record and replay of configuration steps and values for automated regression testing
- Debug and analysis views provide deep insights into runtime behavior of developed solution model. These views are :
  - Configuration Editor: Details about configuration hierarchy and instance relationships
  - Properties View: Details about instance/class /cstic properties
  - Test Runner: Recording and replaying of configuration steps. Performance measure of each step.
  - Conflicts: Detail about conflicting ‘facts’ during configuration steps.
  - Justifications: Details about ‘facts’ which justify a configuration value.
  - Profiling: Performance indicators for dependencies which can help in performance optimization.
  - Tracing: Detailed engine logs to trace root cause of any undesired behavior.



ipc://localhost:1433/?dbName=SME_TEST&kbName=SME_OFFICE&
- Config SME_OFFICE
  - 🟩 SME_OFFICE ($1-SME_OFFICE$1; 1.0)
    - 🟩 SME_WORKPLACE ($1-SME_OFFICE$2; 1.0)
      - 🟩 SME_LAPTOP ($1-SME_OF
        - 🟩 SME_PROCESSOR ($1
        - 🟩 SME_MEMORY ($1-SM
        - 🟩 SME_MEMORY ($1-SM
        - 🟩 SME_VIDEO_CARD ($1-SME_OFFICE$10; 1.0)
    - 🟩 SME_WORKPLACE ($1-SME_OFFICE$3; 1.0)
      - 🟩 SME_DESKTOP ($1-SME_OFFICE$6; 1.0)
    - SME_ADMINISTRATION ($1-SME_OFFICE$5; 1.0)

	Name	Value
	IS_PART_OF_SD_SOFT	
🟩	IS_PART_OF_SD_HARD	1-SME_OFFICE.
	IS_PART_OF_PP	
	OM_POSITION_NUMB...	10
	OSITION_CLASS...	300
	OSITION_OBJEC...	SME_LAPTOP
	OSITION_OBJEC...	MARA
	BOM_POSITION_QUAN...	

Property	Value
Author	S
Class Type	300
Complete	true
Conflicting	false
Consistent	true
Deleteable	false
Inst Id	1-SM

Action	Time (ms)
Meta Data SME_OFFICE	
SetValue: 1-SME_OFFICE$1.SME_NO_WORKPLACES_NF = 2	21
SetValue: 1-SME_OFFICE$2.SME_WORKPLACE_TYPE_NS = OFFICE	20
SetValue: 1-SME_OFFICE$3.SME_WORKPLACE_TYPE_NS = CAD	19
SetValue: 1-SME_OFFICE$6.SME_PROCESSOR_S = DUAL_2_4	7
SetValue: 1-SME_OFFICE$6.SME_MEMORY_S = 2_1333	9
SetValue: 1-SME_OFFICE$6.SME_VIDEO_CARD_S = LOW_END	8

Selected object:SME_LAPTOP ($1-SME_OFFICE$4; 1.0)

Testing perspective

Multiple parallel configurations

Characteristics and values view

Characteristics of selected Instance

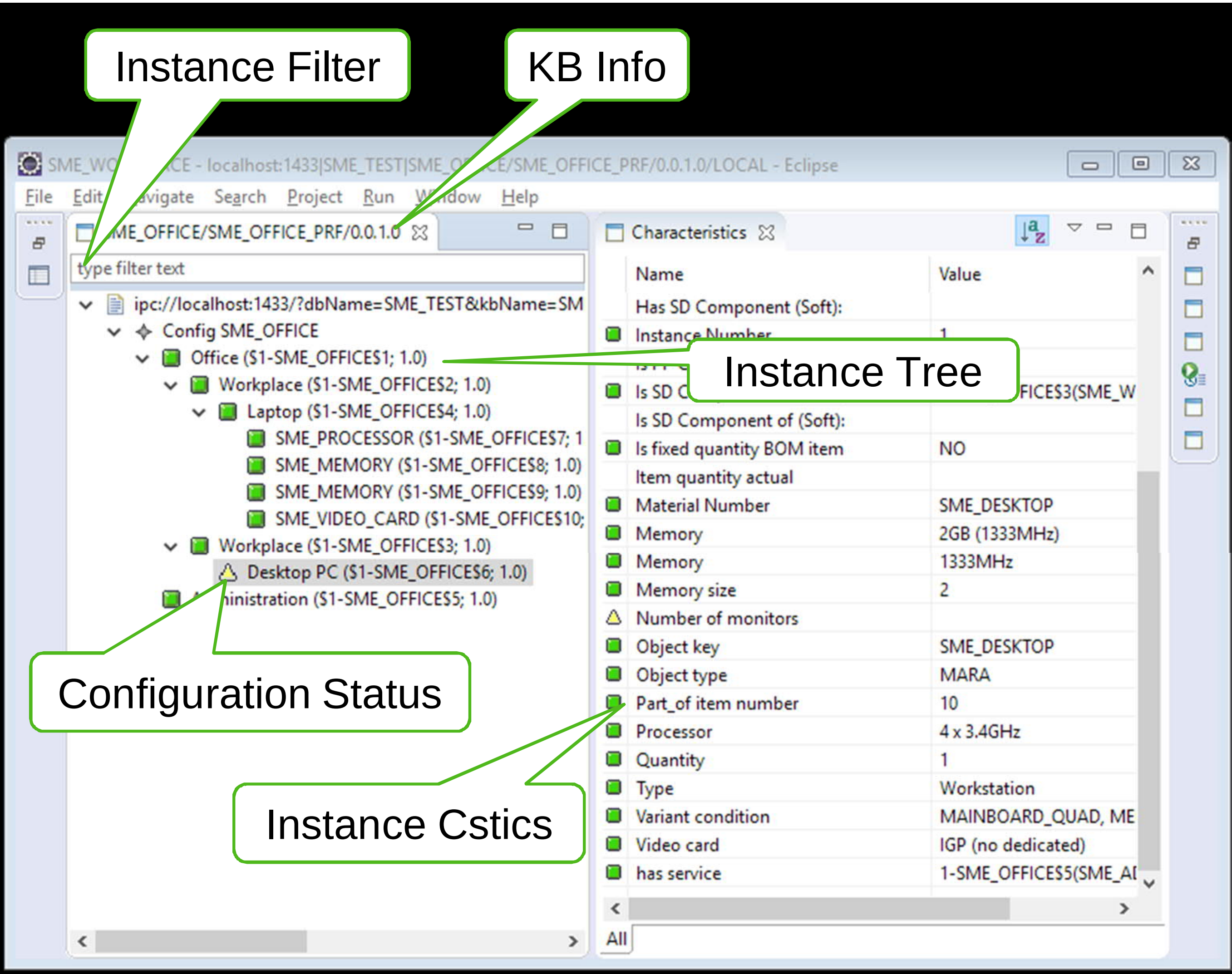
Testing and analysis views

Configuration tree

Instance or cstic properties

Configuration View

- Tree type layout for easier navigation through multi-level configurations.
- Multiple configurations can be instantiated in parallel.
- Filters for instance types for quick search in case of very large trees.



Instance Filter

KB Info

Instance Tree

Configuration Status

Instance Cstics

SME_WO…CE - localhost:1433|SME_TEST|SME_O…CE/SME_OFFICE_PRF/0.0.1.0/LOCAL - Eclipse

…ME_OFFICE/SME_OFFICE_PRF/0.0.1.0

type filter text

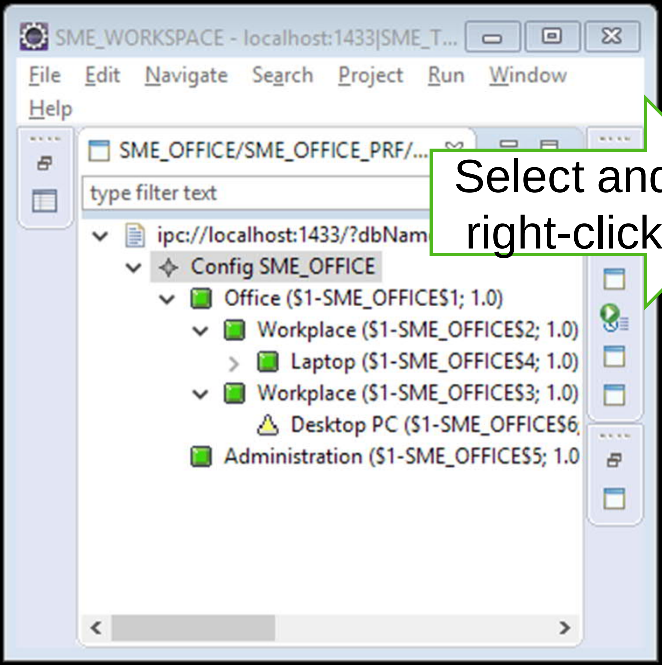
```
ipc://localhost:1433/?dbName=SME_TEST&kbName=SM
  Config SME_OFFICE
    Office ($1-SME_OFFICE$1; 1.0)
      Workplace ($1-SME_OFFICE$2; 1.0)
        Laptop ($1-SME_OFFICE$4; 1.0)
          SME_PROCESSOR ($1-SME_OFFICE$7; 1
          SME_MEMORY ($1-SME_OFFICE$8; 1.0)
          SME_MEMORY ($1-SME_OFFICE$9; 1.0)
          SME_VIDEO_CARD ($1-SME_OFFICE$10;
      Workplace ($1-SME_OFFICE$3; 1.0)
        ⚠ Desktop PC ($1-SME_OFFICE$6; 1.0)
    …inistration ($1-SME_OFFICE$5; 1.0)
```

Characteristics

	Name	Value
	Has SD Component (Soft):	
🟩	Instance Number	1
🟩	Is SD C…	…FICE$3(SME_W
	Is SD Component of (Soft):	
🟩	Is fixed quantity BOM item	NO
	Item quantity actual	
🟩	Material Number	SME_DESKTOP
🟩	Memory	2GB (1333MHz)
🟩	Memory	1333MHz
🟩	Memory size	2
⚠	Number of monitors	
🟩	Object key	SME_DESKTOP
🟩	Object type	MARA
🟩	Part_of item number	10
🟩	Processor	4 x 3.4GHz
🟩	Quantity	1
🟩	Type	Workstation
🟩	Variant condition	MAINBOARD_QUAD, ME
🟩	Video card	IGP (no dedicated)
🟩	has service	1-SME_OFFICE$5(SME_AI

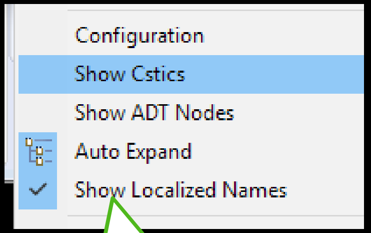
Configuration View: Cstic and Values in Tree Mode

Cstic's and its values can be displayed in the tree for quick analysis.


SME_WORKSPACE - localhost:1433|SME_T...
File Edit Navigate Search Project Run Window
Help
SME_OFFICE/SME_OFFICE_PRF/...
type filter text

- ipc://localhost:1433/?dbNam
  - Config SME_OFFICE
    - Office ($1-SME_OFFICE$1; 1.0)
      - Workplace ($1-SME_OFFICE$2; 1.0)
        - Laptop ($1-SME_OFFICE$4; 1.0)
      - Workplace ($1-SME_OFFICE$3; 1.0)
        - Desktop PC ($1-SME_OFFICE$6,
      - Administration ($1-SME_OFFICE$5; 1.0

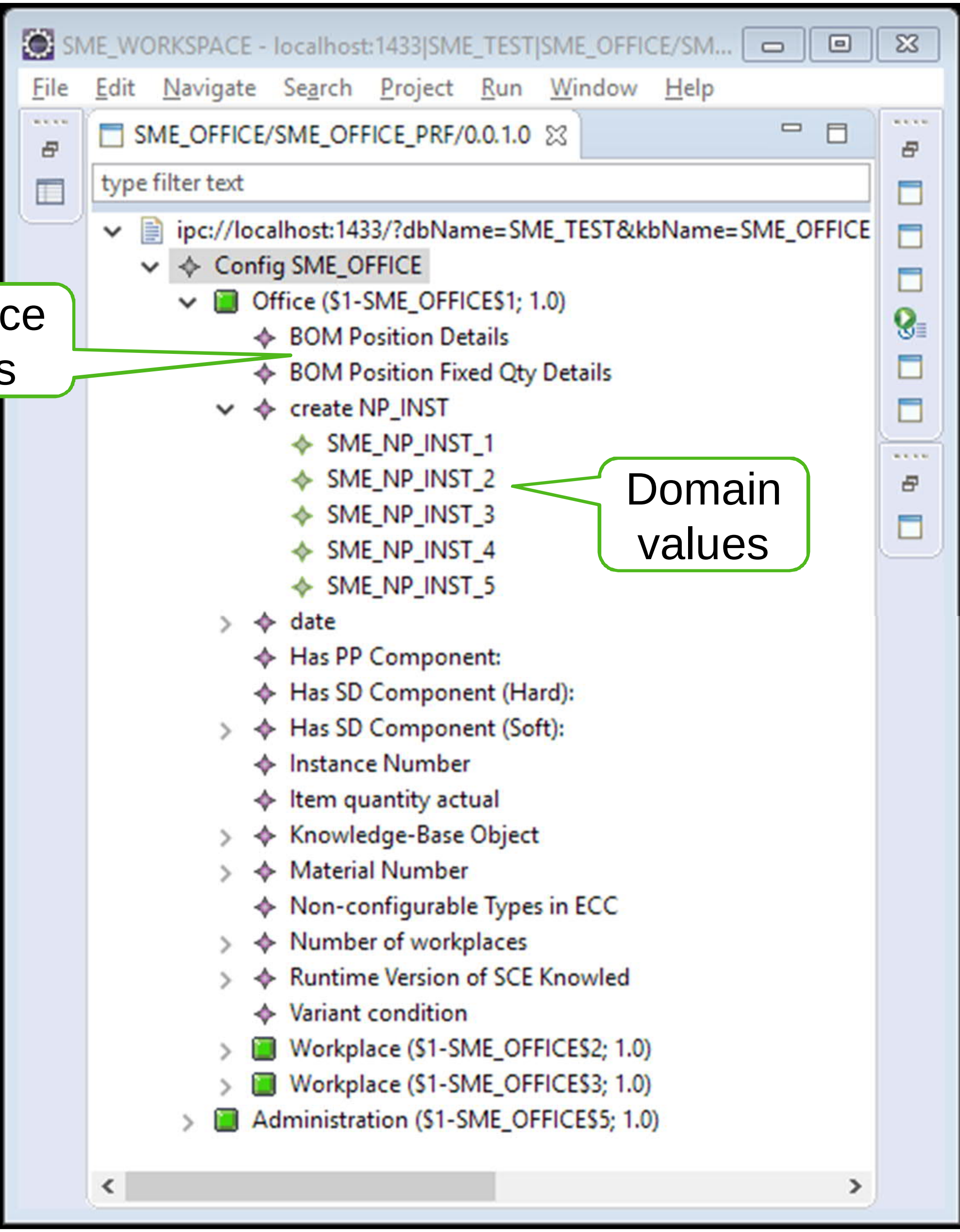
Select and right-click



More tree options

Instance cstics

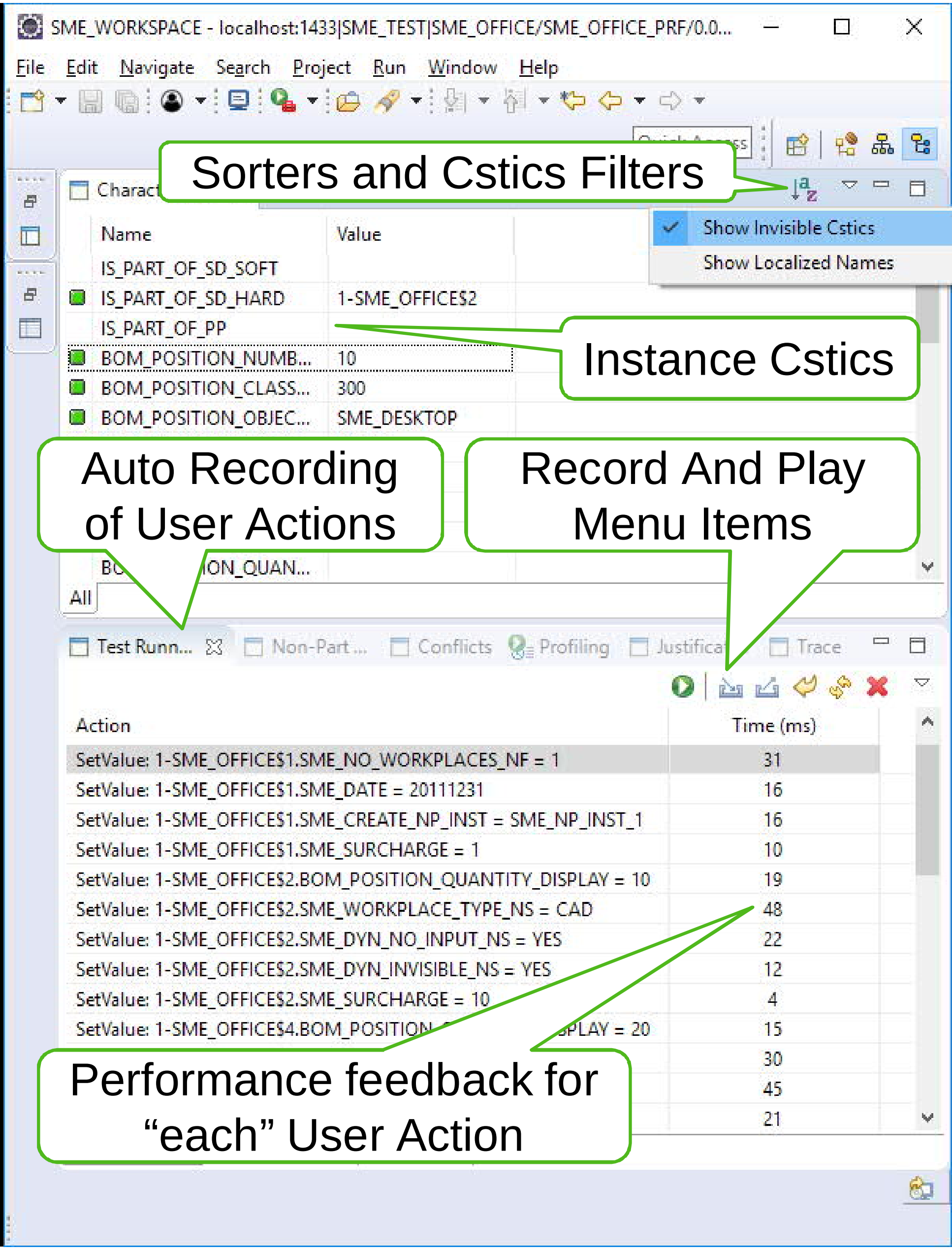
Domain values


SME_WORKSPACE - localhost:1433|SME_TEST|SME_OFFICE/SM...
File Edit Navigate Search Project Run Window Help
SME_OFFICE/SME_OFFICE_PRF/0.0.1.0
type filter text

- ipc://localhost:1433/?dbName=SME_TEST&kbName=SME_OFFICE
  - Config SME_OFFICE
    - Office ($1-SME_OFFICE$1; 1.0)
      - BOM Position Details
      - BOM Position Fixed Qty Details
      - create NP_INST
        - SME_NP_INST_1
        - SME_NP_INST_2
        - SME_NP_INST_3
        - SME_NP_INST_4
        - SME_NP_INST_5
      - date
      - Has PP Component:
      - Has SD Component (Hard):
      - Has SD Component (Soft):
      - Instance Number
      - Item quantity actual
      - Knowledge-Base Object
      - Material Number
      - Non-configurable Types in ECC
      - Number of workplaces
      - Runtime Version of SCE Knowled
      - Variant condition
      - Workplace ($1-SME_OFFICE$2; 1.0)
      - Workplace ($1-SME_OFFICE$3; 1.0)
    - Administration ($1-SME_OFFICE$5; 1.0)

Characteristics and Test Runner View

- Characteristic view shows visible cstics of the instance selected in the config view. Invisible cstic and localized names can be shown if needed.
- Values which are set on the cstics are automatically recorded in the Test Runner view which can be exported if needed.
- Time taken by engine for each action is shown along with the action. This gives immediate feedback to the modeler for any performance improvements.


SME_WORKSPACE - localhost:1433|SME_TEST|SME_OFFICE/SME_OFFICE_PRF/0.0...
File Edit Navigate Search Project Run Window Help
Charact
Show Invisible Cstics
Show Localized Names
All
Test Runn...
Non-Part ...
Conflicts
Profiling
Justificat
Trace

Sorters and Cstics Filters

Instance Cstics

	Name	Value
	IS_PART_OF_SD_SOFT	
🟩	IS_PART_OF_SD_HARD	1-SME_OFFICE$2
	IS_PART_OF_PP	
🟩	BOM_POSITION_NUMB...	10
🟩	BOM_POSITION_CLASS...	300
🟩	BOM_POSITION_OBJEC...	SME_DESKTOP
	BO ION_QUAN...	

Auto Recording of User Actions

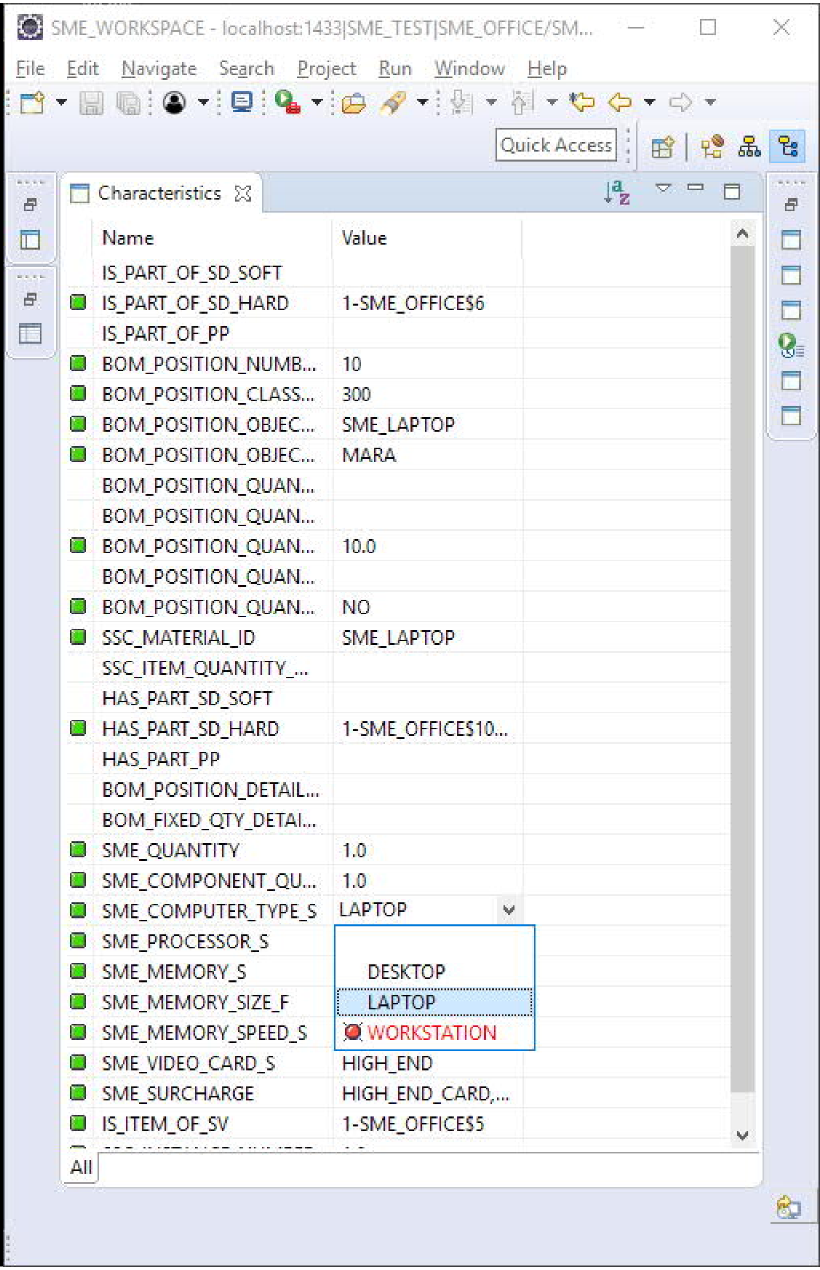
Record And Play Menu Items

Action	Time (ms)
SetValue: 1-SME_OFFICE$1.SME_NO_WORKPLACES_NF = 1	31
SetValue: 1-SME_OFFICE$1.SME_DATE = 20111231	16
SetValue: 1-SME_OFFICE$1.SME_CREATE_NP_INST = SME_NP_INST_1	16
SetValue: 1-SME_OFFICE$1.SME_SURCHARGE = 1	10
SetValue: 1-SME_OFFICE$2.BOM_POSITION_QUANTITY_DISPLAY = 10	19
SetValue: 1-SME_OFFICE$2.SME_WORKPLACE_TYPE_NS = CAD	48
SetValue: 1-SME_OFFICE$2.SME_DYN_NO_INPUT_NS = YES	22
SetValue: 1-SME_OFFICE$2.SME_DYN_INVISIBLE_NS = YES	12
SetValue: 1-SME_OFFICE$2.SME_SURCHARGE = 10	4
SetValue: 1-SME_OFFICE$4.BOM_POSITION PLAY = 20	15
	30
	45
	21

Performance feedback for "each" User Action

Cstics View: Selections and Restrictions

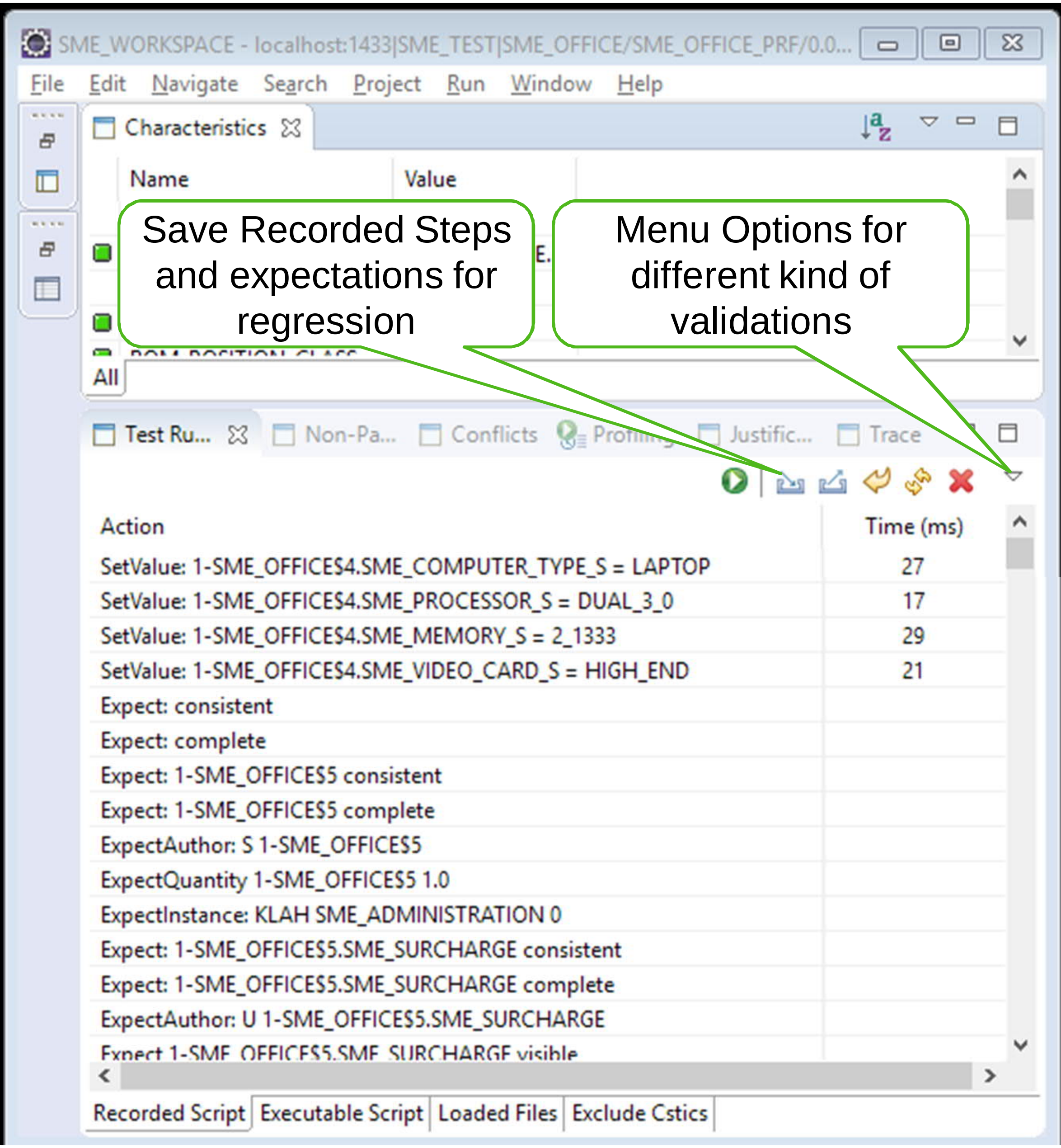
- In the characteristic view, values can be selected/unselected using easy to use dropdowns.
- Restrictions are shown in ‘red’ color. They can be selected but it will lead into a conflict situation, that is, more than one contradictory fact has been detected in the engine.
- Conflicts and justifications views provide more insights into this “Why Not ?” situation.



	Name	Value
	IS_PART_OF_SD_SOFT	
🟩	IS_PART_OF_SD_HARD	1-SME_OFFICE$6
	IS_PART_OF_PP	
🟩	BOM_POSITION_NUMB...	10
🟩	BOM_POSITION_CLASS...	300
🟩	BOM_POSITION_OBJEC...	SME_LAPTOP
🟩	BOM_POSITION_OBJEC...	MARA
	BOM_POSITION_QUAN...	
	BOM_POSITION_QUAN...	
🟩	BOM_POSITION_QUAN...	10.0
	BOM_POSITION_QUAN...	
🟩	BOM_POSITION_QUAN...	NO
🟩	SSC_MATERIAL_ID	SME_LAPTOP
	SSC_ITEM_QUANTITY_...	
	HAS_PART_SD_SOFT	
🟩	HAS_PART_SD_HARD	1-SME_OFFICE$10...
	HAS_PART_PP	
	BOM_POSITION_DETAIL...	
	BOM_FIXED_QTY_DETAI...	
🟩	SME_QUANTITY	1.0
🟩	SME_COMPONENT_QU...	1.0
🟩	SME_COMPUTER_TYPE_S	LAPTOP
🟩	SME_PROCESSOR_S	
🟩	SME_MEMORY_S	
🟩	SME_MEMORY_SIZE_F	
🟩	SME_MEMORY_SPEED_S	
🟩	SME_VIDEO_CARD_S	HIGH_END
🟩	SME_SURCHARGE	HIGH_END_CARD,...
🟩	IS_ITEM_OF_SV	1-SME_OFFICE$5

Test Runner View

- Test Runner enables faster testing of solution models :
  - User action on cstics are automatically recorded which can be saved (.performer) for regression testing later.
  - Config state, that is, expectations are also captured automatically. Expectations are asserted/validated upon rerun of the test scripts against the new configuration created in engine to highlight any discrepancies which might arise due to changes in solution model.
  - Different kind of validations can be enabled/disabled through menu options.



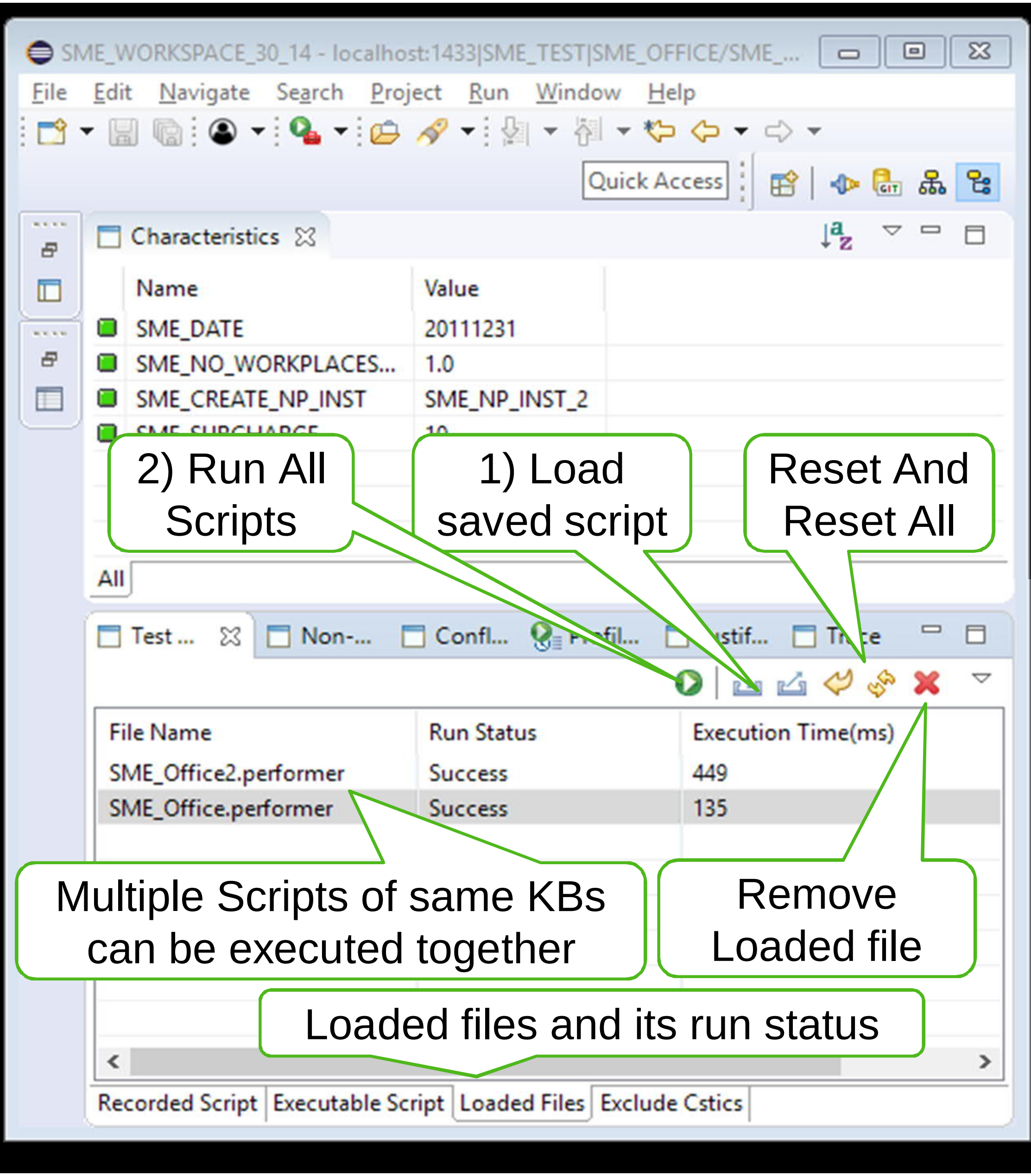
Save Recorded Steps and expectations for regression

Menu Options for different kind of validations

Action	Time (ms)
SetValue: 1-SME_OFFICE$4.SME_COMPUTER_TYPE_S = LAPTOP	27
SetValue: 1-SME_OFFICE$4.SME_PROCESSOR_S = DUAL_3_0	17
SetValue: 1-SME_OFFICE$4.SME_MEMORY_S = 2_1333	29
SetValue: 1-SME_OFFICE$4.SME_VIDEO_CARD_S = HIGH_END	21
Expect: consistent	
Expect: complete	
Expect: 1-SME_OFFICE$5 consistent	
Expect: 1-SME_OFFICE$5 complete	
ExpectAuthor: S 1-SME_OFFICE$5	
ExpectQuantity 1-SME_OFFICE$5 1.0	
ExpectInstance: KLAH SME_ADMINISTRATION 0	
Expect: 1-SME_OFFICE$5.SME_SURCHARGE consistent	
Expect: 1-SME_OFFICE$5.SME_SURCHARGE complete	
ExpectAuthor: U 1-SME_OFFICE$5.SME_SURCHARGE	
Expect 1-SME_OFFICE$5.SME_SURCHARGE visible	

Test Runner: Regression Testing

- Two Step Process: Load and Execute previously saved scripts.
- More than one test script can be loaded and executed at same time, that is, bulk regression.
- Test scripts of different KBs can be loaded and executed to validate if related models are impacted with recent changes.
- Configuration tree is created and displayed for each execution.
- Execution can be repeated after reset/resetAll of loaded configurations
- Total execution time can be monitored for each script to verify any performance impact.



Name	Value
SME_DATE	20111231
SME_NO_WORKPLACES...	1.0
SME_CREATE_NP_INST	SME_NP_INST_2
SME_SURCHARGE	10

File Name	Run Status	Execution Time(ms)
SME_Office2.performer	Success	449
SME_Office.performer	Success	135

2) Run All Scripts

1) Load saved script

Reset And Reset All

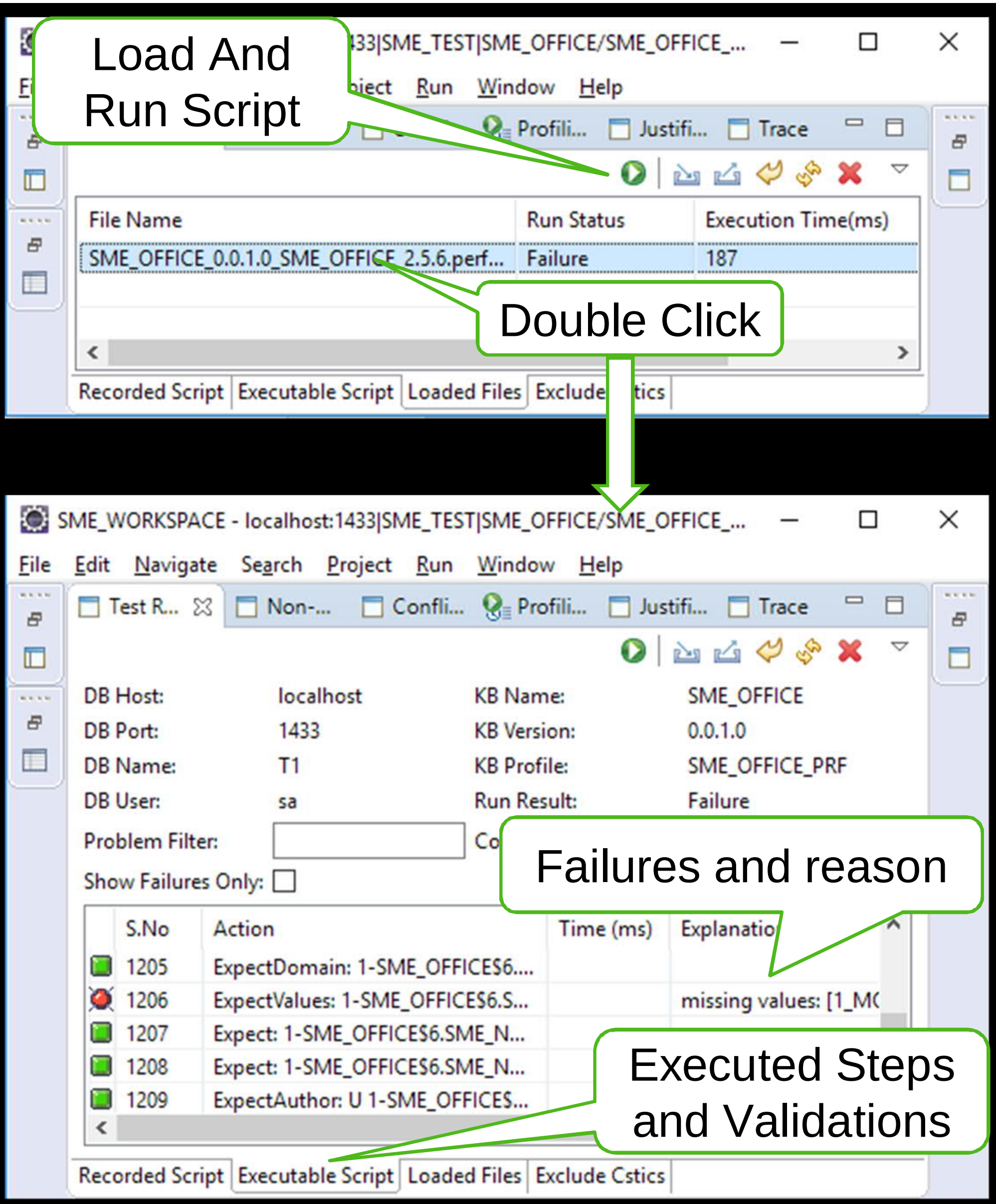
Multiple Scripts of same KBs can be executed together

Remove Loaded file

Loaded files and its run status

Test Runner: Verifying Execution Result

- Upon script execution, for each configuration step, it’s execution result is validated against previously saved values.
- Execution results are shown in Executable Script Tab, which shows step-wise execution time and its validation success or failure.
- Explanations are also provided for the reason of failure.


Load And Run Script
Double Click
Failures and reason
Executed Steps and Validations
...433|SME_TEST|SME_OFFICE/SME_OFFICE_...
...oject Run Window Help
Profili... Justifi... Trace
Recorded Script | Executable Script | Loaded Files | Exclude...tics
SME_WORKSPACE - localhost:1433|SME_TEST|SME_OFFICE/SME_OFFICE_...
File Edit Navigate Search Project Run Window Help
Test R... Non-... Confli... Profili... Justifi... Trace
Recorded Script | Executable Script | Loaded Files | Exclude Cstics

File Name	Run Status	Execution Time(ms)
SME_OFFICE_0.0.1.0_SME_OFFICE_2.5.6.perf...	Failure	187

DB Host: localhost

DB Port: 1433

DB Name: T1

DB User: sa

KB Name: SME_OFFICE

KB Version: 0.0.1.0

KB Profile: SME_OFFICE_PRF

Run Result: Failure

Problem Filter: ______ Co

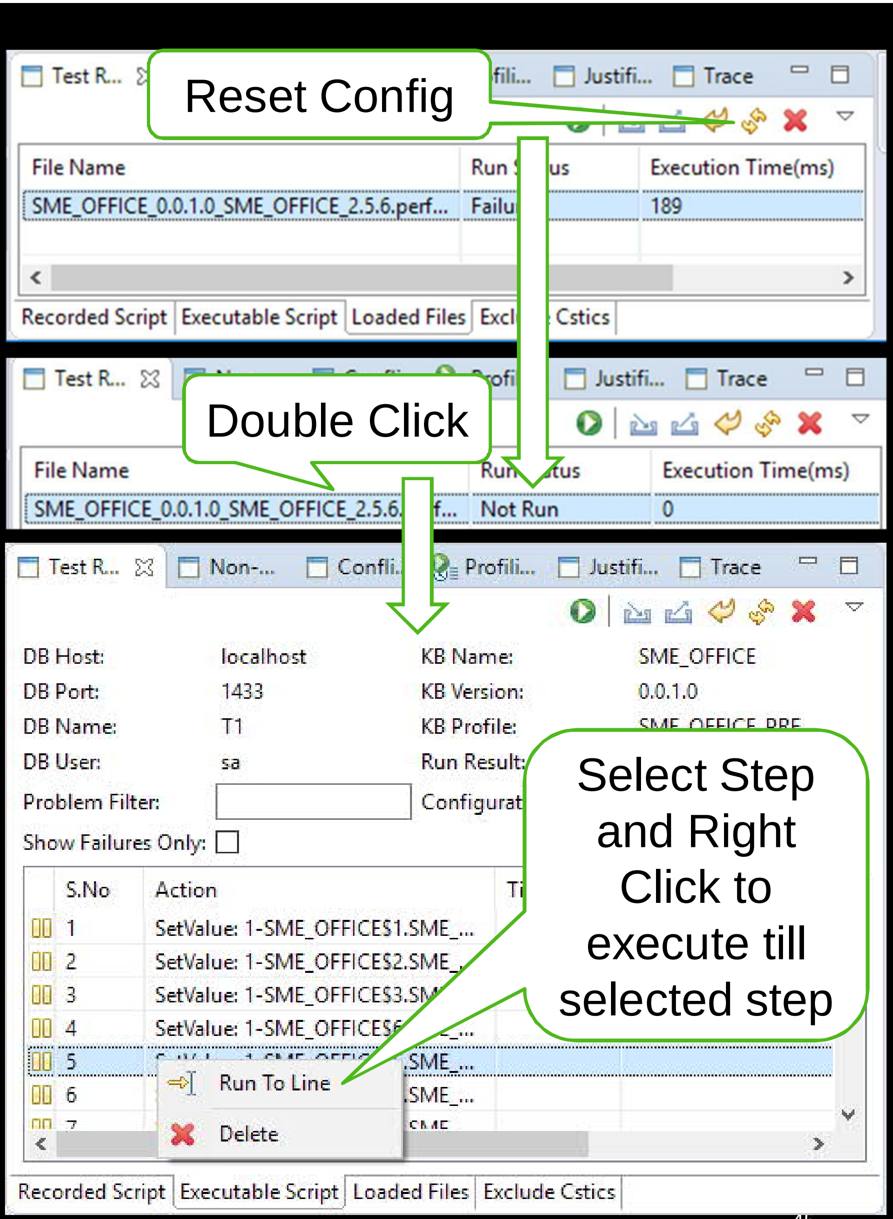
Show Failures Only: □

	S.No	Action	Time (ms)	Explanatio
🟩	1205	ExpectDomain: 1-SME_OFFICE$6....		
🟥	1206	ExpectValues: 1-SME_OFFICE$6.S...		missing values: [1_MC
🟩	1207	Expect: 1-SME_OFFICE$6.SME_N...		
🟩	1208	Expect: 1-SME_OFFICE$6.SME_N...		
🟩	1209	ExpectAuthor: U 1-SME_OFFICE$...		

Test Runner :
Adding New Configuration Steps

To reduce tedious manual effort of executing preceding configuration steps in order to add new steps or a new test case, configuration step execution features can be used :

- Reset the configuration for which a new test steps is to be added.
- Double-click on the script to load all executable steps.
- Identify configuration step till which the configuration should be done.
- Select the step -> Right-click -> Run To Line. This will execute all preceding steps and prepare the configuration for next steps.
- Make changes in the configuration as needed. Steps will be automatically recorded.
- Save the new test script.



File Name	Run Status	Execution Time(ms)
SME_OFFICE_0.0.1.0_SME_OFFICE_2.5.6.perf...	Failu...	189

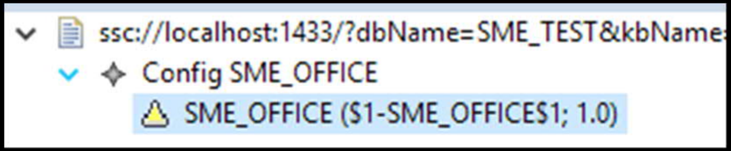
File Name	Run Status	Execution Time(ms)
SME_OFFICE_0.0.1.0_SME_OFFICE_2.5.6...f...	Not Run	0

DB Host: localhost
KB Name: SME_OFFICE
DB Port: 1433
KB Version: 0.0.1.0
DB Name: T1
KB Profile: SME_OFFICE_PRF
DB User: sa
Run Result:
Problem Filter:
Configurat...
Show Failures Only: □

S.No	Action	Ti...
1	SetValue: 1-SME_OFFICE$1.SME_...	
2	SetValue: 1-SME_OFFICE$2.SME_...	
3	SetValue: 1-SME_OFFICE$3.SM...	
4	SetValue: 1-SME_OFFICE$...	
5	...SME_...	
6	...SME_...	
7	...SME...	

Test Runner: Validations

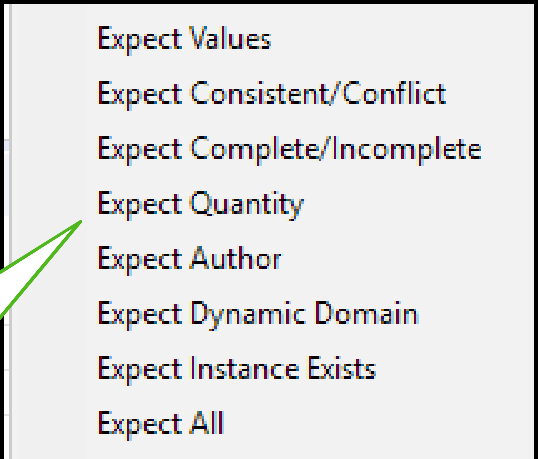
Configuration state validations are supported through concept of ‘expectations’ which are data records containing what is to be expected after execution of a configuration step. These validations can be applied at instance level selectively and can be automatically recorded after each step or at end of entire configuration.

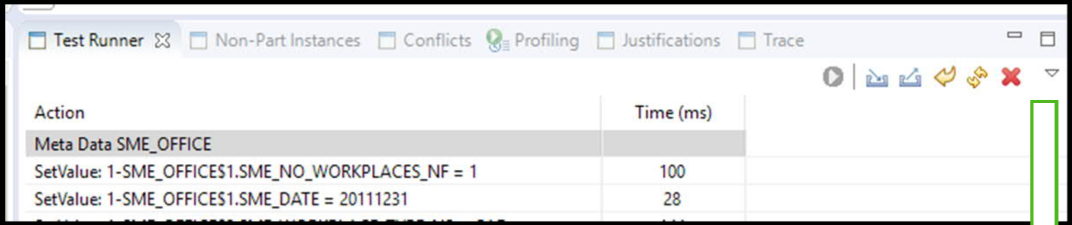

ssc://localhost:1433/?dbName=SME_TEST&kbName
Config SME_OFFICE
SME_OFFICE ($1-SME_OFFICE$1; 1.0)

Select Configuration

Right Click

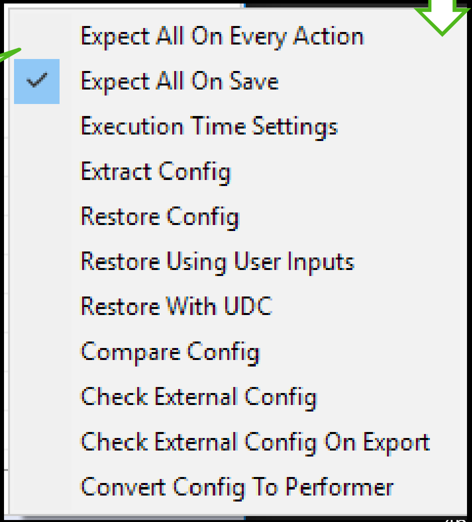
Choose Validation Granularity





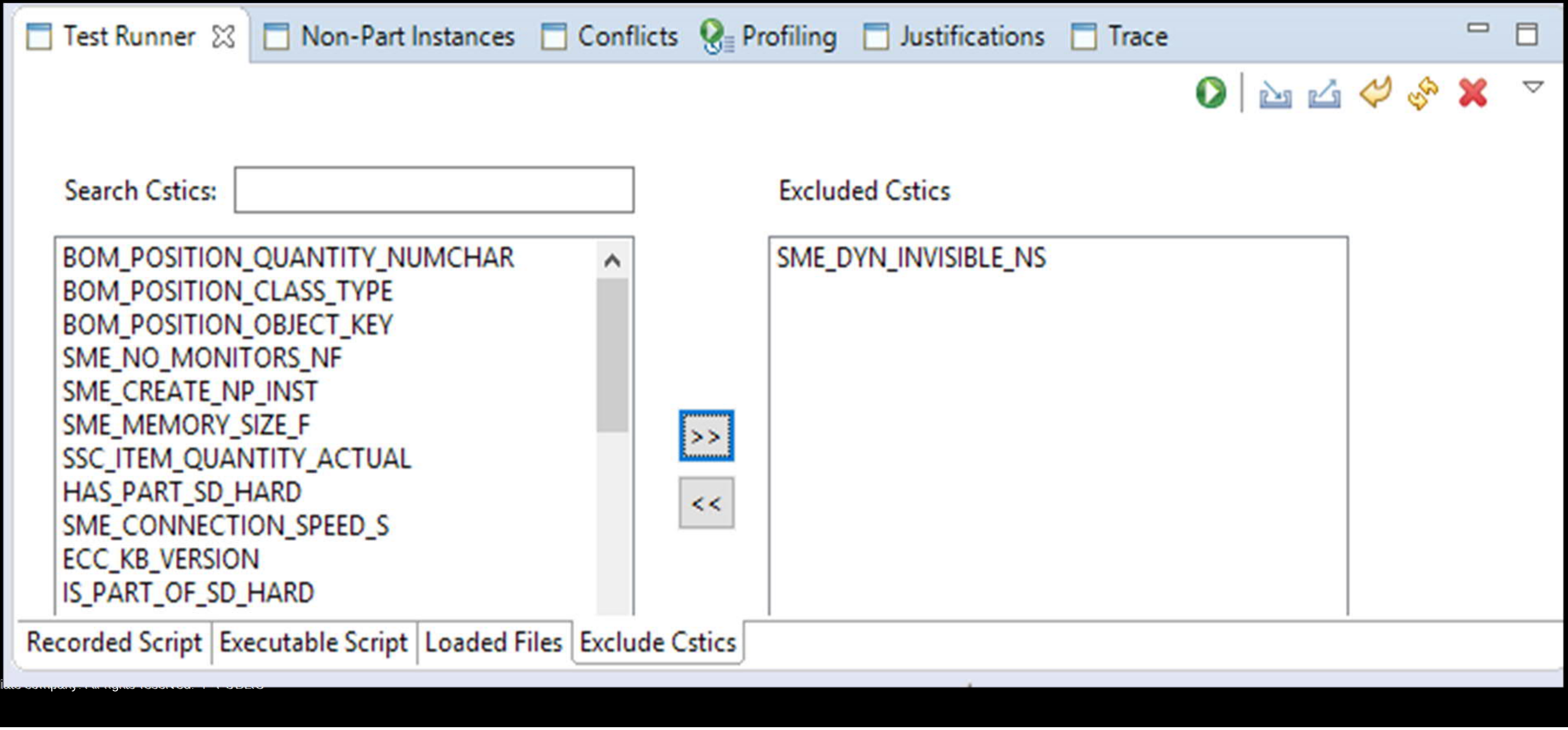
Action	Time (ms)
Meta Data SME_OFFICE	
SetValue: 1-SME_OFFICE$1.SME_NO_WORKPLACES_NF = 1	100
SetValue: 1-SME_OFFICE$1.SME_DATE = 20111231	28

Validation Options



Test Runner: Excluding Cstics

In certain situations, cstic values can be dynamically calculated randomly in each configuration. Thus, the same cannot be validated and test scripts would fail. Thu, these cstic’s do not contribute actively in configuration and can be excluded. Test runner provides mechanism to exclude such cstics.



Search Cstics:

BOM_POSITION_QUANTITY_NUMCHAR
BOM_POSITION_CLASS_TYPE
BOM_POSITION_OBJECT_KEY
SME_NO_MONITORS_NF
SME_CREATE_NP_INST
SME_MEMORY_SIZE_F
SSC_ITEM_QUANTITY_ACTUAL
HAS_PART_SD_HARD
SME_CONNECTION_SPEED_S
ECC_KB_VERSION
IS_PART_OF_SD_HARD

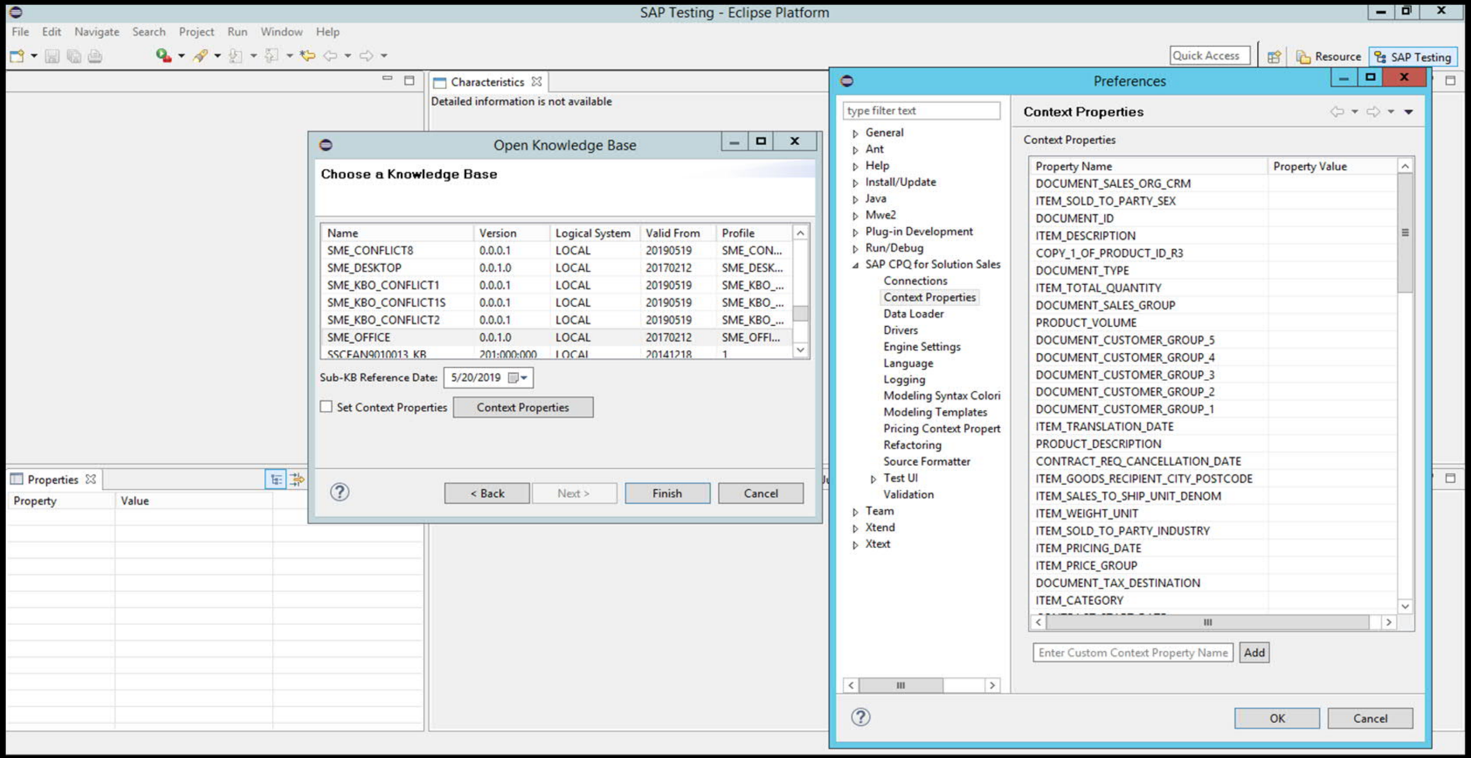
Excluded Cstics

SME_DYN_INVISIBLE_NS

Test Runner: Configuration Context Properties

To enable testing of configurations which depend upon context properties injected by system at runtime, context properties can be maintained in SME during testing process.

Recorded test scripts store the context properties and its values, enabling simulation and testing of different scenarios within SME itself, thus saving significant effort of cycles of test and fixing with the backend system.



SAP Testing - Eclipse Platform
File Edit Navigate Search Project Run Window Help
Quick Access
Resource
SAP Testing
Characteristics
Properties
Property
Value

Detailed information is not available

Open Knowledge Base

Choose a Knowledge Base

Name	Version	Logical System	Valid From	Profile
SME_CONFLICT8	0.0.0.1	LOCAL	20190519	SME_CON...
SME_DESKTOP	0.0.1.0	LOCAL	20170212	SME_DESK...
SME_KBO_CONFLICT1	0.0.0.1	LOCAL	20190519	SME_KBO_...
SME_KBO_CONFLICT1S	0.0.0.1	LOCAL	20190519	SME_KBO_...
SME_KBO_CONFLICT2	0.0.0.1	LOCAL	20190519	SME_KBO_...
SME_OFFICE	0.0.1.0	LOCAL	20170212	SME_OFFI...
SSCEAN9010013_KB	201:000:000	LOCAL	20141218	1

Sub-KB Reference Date: 5/20/2019

□ Set Context Properties

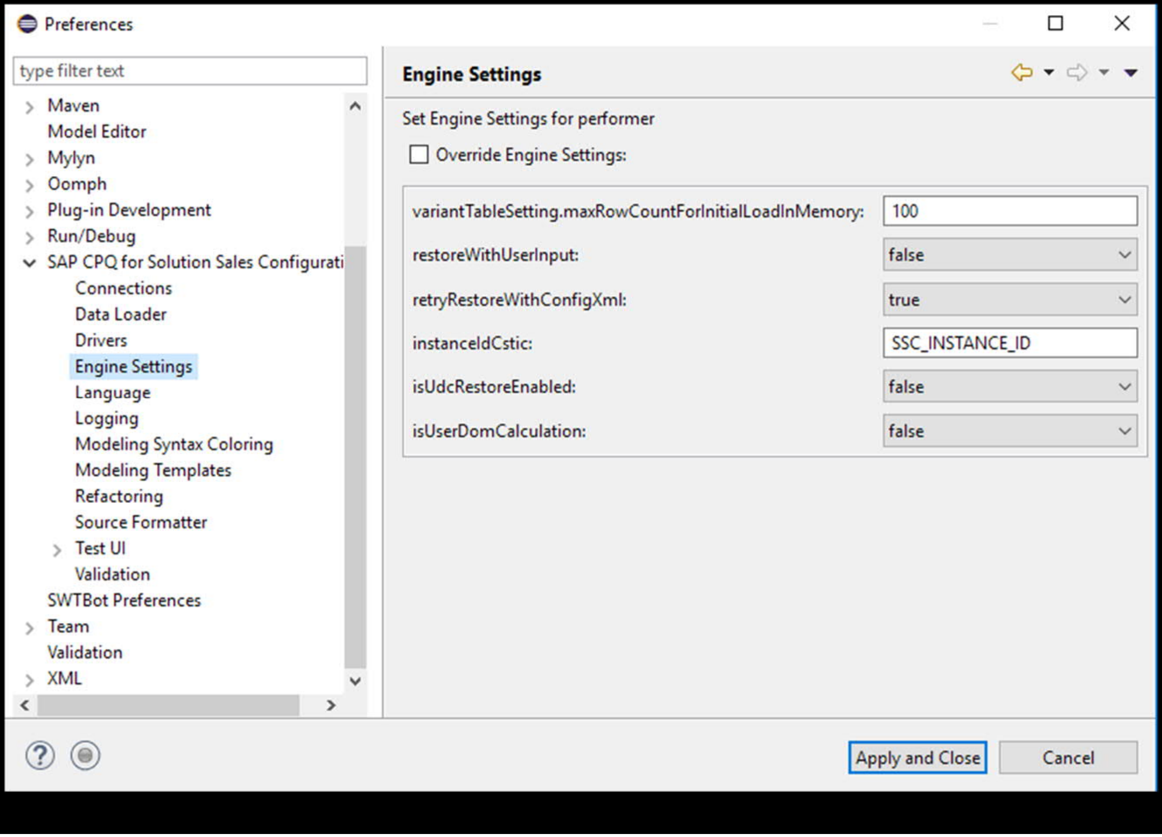
Context Properties

Context Properties

Property Name	Property Value
DOCUMENT_SALES_ORG_CRM	
ITEM_SOLD_TO_PARTY_SEX	
DOCUMENT_ID	
ITEM_DESCRIPTION	
COPY_1_OF_PRODUCT_ID_R3	
DOCUMENT_TYPE	
ITEM_TOTAL_QUANTITY	
DOCUMENT_SALES_GROUP	
PRODUCT_VOLUME	
DOCUMENT_CUSTOMER_GROUP_5	
DOCUMENT_CUSTOMER_GROUP_4	
DOCUMENT_CUSTOMER_GROUP_3	
DOCUMENT_CUSTOMER_GROUP_2	
DOCUMENT_CUSTOMER_GROUP_1	
ITEM_TRANSLATION_DATE	
PRODUCT_DESCRIPTION	
CONTRACT_REQ_CANCELLATION_DATE	
ITEM_GOODS_RECIPIENT_CITY_POSTCODE	
ITEM_SALES_TO_SHIP_UNIT_DENOM	
ITEM_WEIGHT_UNIT	
ITEM_SOLD_TO_PARTY_INDUSTRY	
ITEM_PRICING_DATE	
ITEM_PRICE_GROUP	
DOCUMENT_TAX_DESTINATION	
ITEM_CATEGORY	

Test Runner: Engine Flags

SME is shipped with default engine settings by SAP. However, deployments in productive environment can be different or might need to be tuned in accordance with the solution model being developed. For quick testing, engine flags can be changed .



Engine Settings

Set Engine Settings for performer

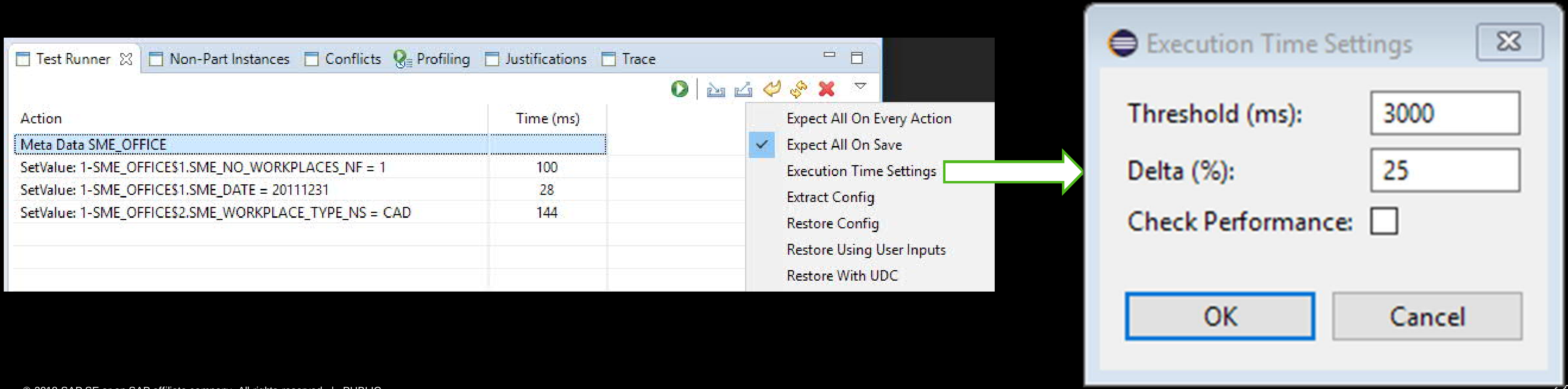
□ Override Engine Settings:

variantTableSetting.maxRowCountForInitialLoadInMemory:	100
restoreWithUserInput:	false
retryRestoreWithConfigXml:	true
instanceIdCstic:	SSC_INSTANCE_ID
isUdcRestoreEnabled:	false
isUserDomCalculation:	false

Test Runner: Auto-Performance Validation

While configuration state, which is binary can be easily asserted through expectations, the same cannot be done in straightforward manner for asserting performance. Test runner provides an approximate way to control performance issues :

- Maximum acceptable performance degradation can be defined in SME as threshold both in absolute or % terms.
- During test recording, time taken for each action is saved in the performer file.
- During rerun, actions taking more than the 'Threshold' time plus delta would be highlighted in red but will not fail test.
- If “Check Performance“ is selected, then test script would fail .



Action	Time (ms)
Meta Data SME_OFFICE	
SetValue: 1-SME_OFFICE$1.SME_NO_WORKPLACES_NF = 1	100
SetValue: 1-SME_OFFICE$1.SME_DATE = 20111231	28
SetValue: 1-SME_OFFICE$2.SME_WORKPLACE_TYPE_NS = CAD	144

Threshold (ms): 3000

Delta (%): 25

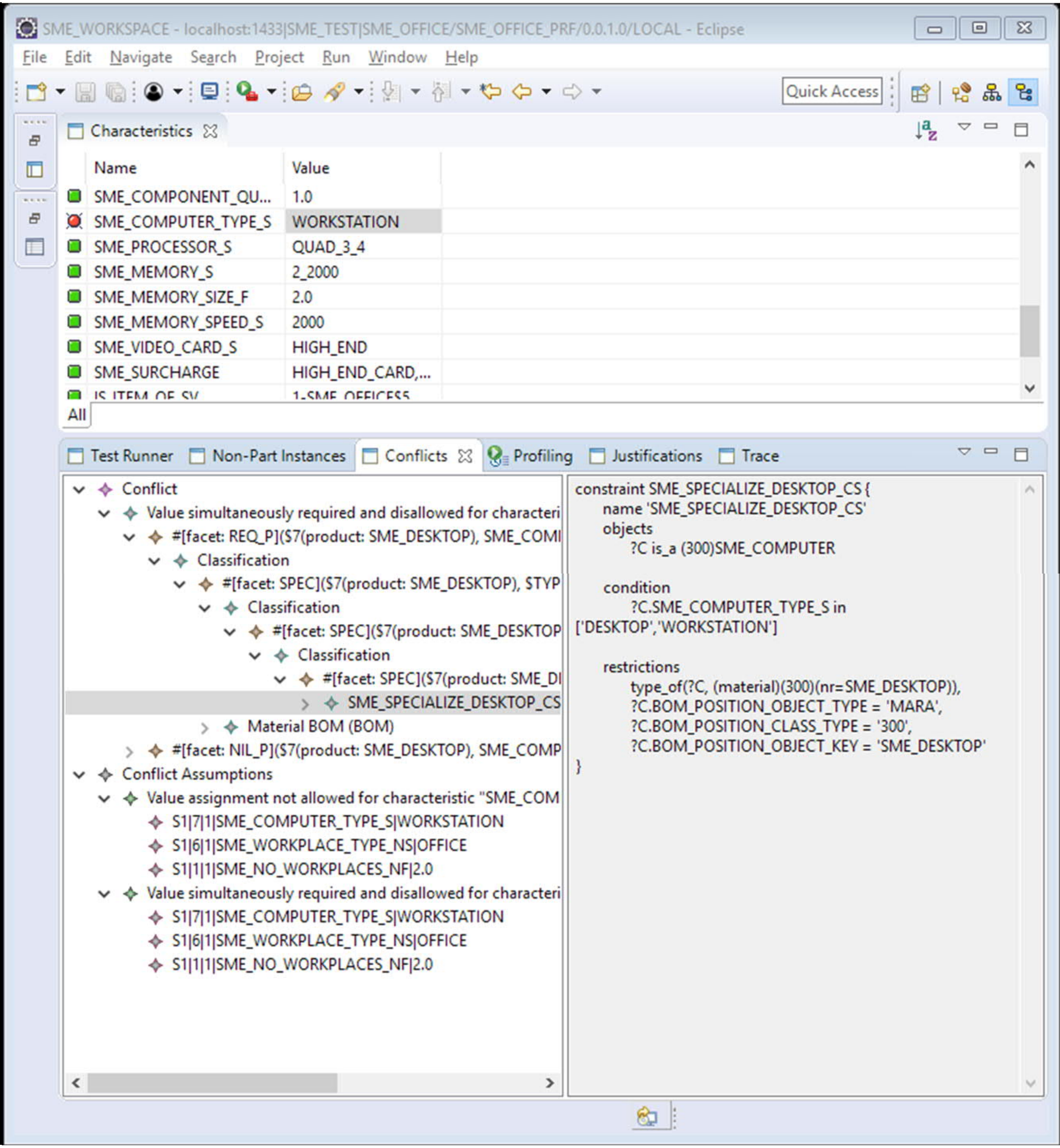
Check Performance: □

Test Runner: Performer Script Structure

```
kbName "SME_OFFICE" ;
kbVersion "0.0.1.0" ;
kbProfile "SME_OFFICE_PRF" ;
databaseName "SME_TEST" ;
databaseHostname "localhost" ;
databasePort "1433" ;
databaseUser "sa" ;
setValues "1" onCharacteristic "SME_NO_WORKPLACES_NF" inInstance "1-SME_OFFICE$1" obyKey "SME_OFFICE" obyType "MARA"
occurrence 0 time 28 ;
setValues "SME_NP_INST_1" onCharacteristic "SME_CREATE_NP_INST" inInstance "1-SME_OFFICE$1" obyKey "SME_OFFICE"
obyType "MARA" occurrence 0 time 38 ;
expect consistent ;
expect complete ;
expect consistent inInstance "1-SME_OFFICE$5" obyKey "SME_ADMINISTRATION" obyType "KLAH" occurrence 0 ;
expect complete inInstance "1-SME_OFFICE$5" obyKey "SME_ADMINISTRATION" obyType "KLAH" occurrence 0 ;
expectAuthor "S" inInstance "1-SME_OFFICE$5" obyKey "SME_ADMINISTRATION" obyType "KLAH" occurrence 0 ;
expectQuantity inInstance "1-SME_OFFICE$5" obyKey "SME_ADMINISTRATION" obyType "KLAH" occurrence 0 quantity 1.0 ;
expectInstance obyKey "SME_ADMINISTRATION" obyType "KLAH" occurrence 0 ;
expect consistent onCharacteristic "SME_SURCHARGE" inInstance "1-SME_OFFICE$5" obyKey "SME_ADMINISTRATION" obyType
"KLAH" occurrence 0 ;
expect complete onCharacteristic "SME_SURCHARGE" inInstance "1-SME_OFFICE$5" obyKey "SME_ADMINISTRATION" obyType
"KLAH" occurrence 0 ;
```

Conflicts View

- When a conflict gets triggered during the configuration execution, detailed explanations are displayed in the conflicts view.
- For the conflicting cstic, conflicts view shows :
  - Conflict: Description and chain of events leading to conflict. It also shows the source code of dependency leading to conflict for quick analysis.
  - Conflict Assumptions: Suggestions on how a given conflict can be solved. Depending upon type of conflict and involved instances, cstics and values, the engine might suggest to change or delete a cstic, or delete unspecialized instance


SME_WORKSPACE - localhost:1433|SME_TEST|SME_OFFICE/SME_OFFICE_PRF/0.0.1.0/LOCAL - Eclipse
File Edit Navigate Search Project Run Window Help
Quick Access
Characteristics
All
Test Runner
Non-Part Instances
Conflicts
Profiling
Justifications
Trace

	Name	Value
🟩	SME_COMPONENT_QU...	1.0
●	SME_COMPUTER_TYPE_S	WORKSTATION
🟩	SME_PROCESSOR_S	QUAD_3_4
🟩	SME_MEMORY_S	2_2000
🟩	SME_MEMORY_SIZE_F	2.0
🟩	SME_MEMORY_SPEED_S	2000
🟩	SME_VIDEO_CARD_S	HIGH_END
🟩	SME_SURCHARGE	HIGH_END_CARD,...
🟩	IS_ITEM_OF_SV	1-SME_OFFICESS

```
Conflict
    Value simultaneously required and disallowed for characteri
        #[facet: REQ_P]($7(product: SME_DESKTOP), SME_COMI
            Classification
                #[facet: SPEC]($7(product: SME_DESKTOP), $TYP
                    Classification
                        #[facet: SPEC]($7(product: SME_DESKTOP
                            Classification
                                #[facet: SPEC]($7(product: SME_DI
                                    SME_SPECIALIZE_DESKTOP_CS
                    Material BOM (BOM)
        #[facet: NIL_P]($7(product: SME_DESKTOP), SME_COMP
Conflict Assumptions
    Value assignment not allowed for characteristic "SME_COM
        S1|7|1|SME_COMPUTER_TYPE_S|WORKSTATION
        S1|6|1|SME_WORKPLACE_TYPE_NS|OFFICE
        S1|1|1|SME_NO_WORKPLACES_NF|2.0
    Value simultaneously required and disallowed for characteri
        S1|7|1|SME_COMPUTER_TYPE_S|WORKSTATION
        S1|6|1|SME_WORKPLACE_TYPE_NS|OFFICE
        S1|1|1|SME_NO_WORKPLACES_NF|2.0
```

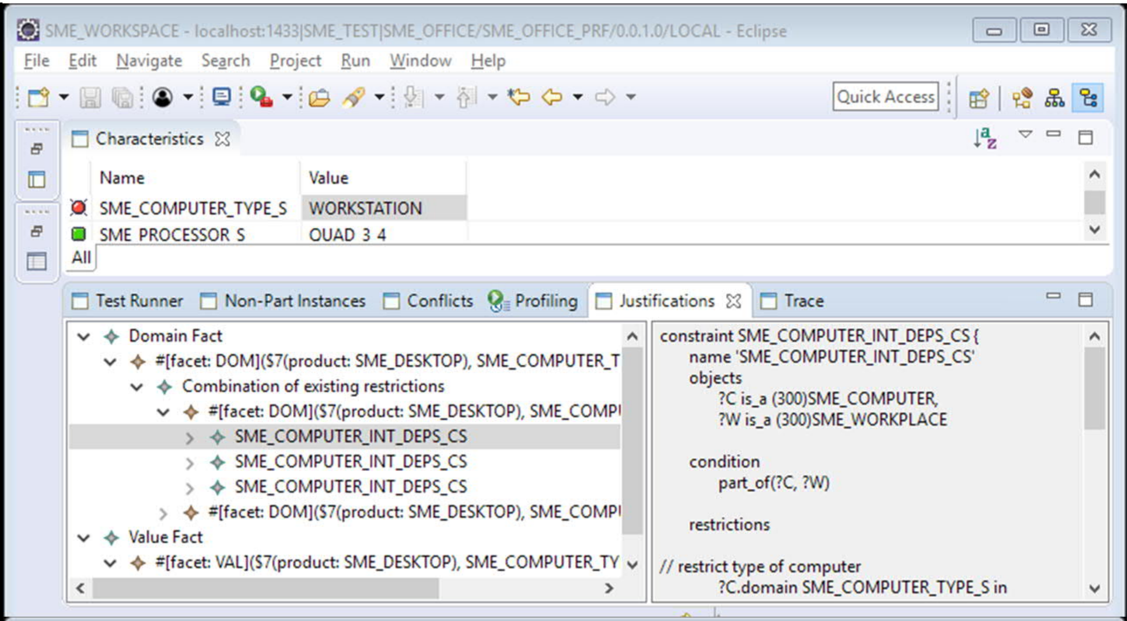
```
constraint SME_SPECIALIZE_DESKTOP_CS {
    name 'SME_SPECIALIZE_DESKTOP_CS'
    objects
        ?C is_a (300)SME_COMPUTER

    condition
        ?C.SME_COMPUTER_TYPE_S in
['DESKTOP','WORKSTATION']

    restrictions
        type_of(?C, (material)(300)(nr=SME_DESKTOP)),
        ?C.BOM_POSITION_OBJECT_TYPE = 'MARA',
        ?C.BOM_POSITION_CLASS_TYPE = '300',
        ?C.BOM_POSITION_OBJECT_KEY = 'SME_DESKTOP'
}
```

Justifications View

- Each Instance or characteristic value is justified by a set of facts asserted by the user or by dependency.
- Justifications for the setting of any characteristic values, for example, either by user, or by system, or component instantiation is displayed in the justification view.



SME_WORKSPACE - localhost:1433|SME_TEST|SME_OFFICE/SME_OFFICE_PRF/0.0.1.0/LOCAL - Eclipse
File Edit Navigate Search Project Run Window Help
Quick Access
Characteristics
Test Runner
Non-Part Instances
Conflicts
Profiling
Justifications
Trace

Name	Value
SME_COMPUTER_TYPE_S	WORKSTATION
SME_PROCESSOR_S	QUAD 3 4

All

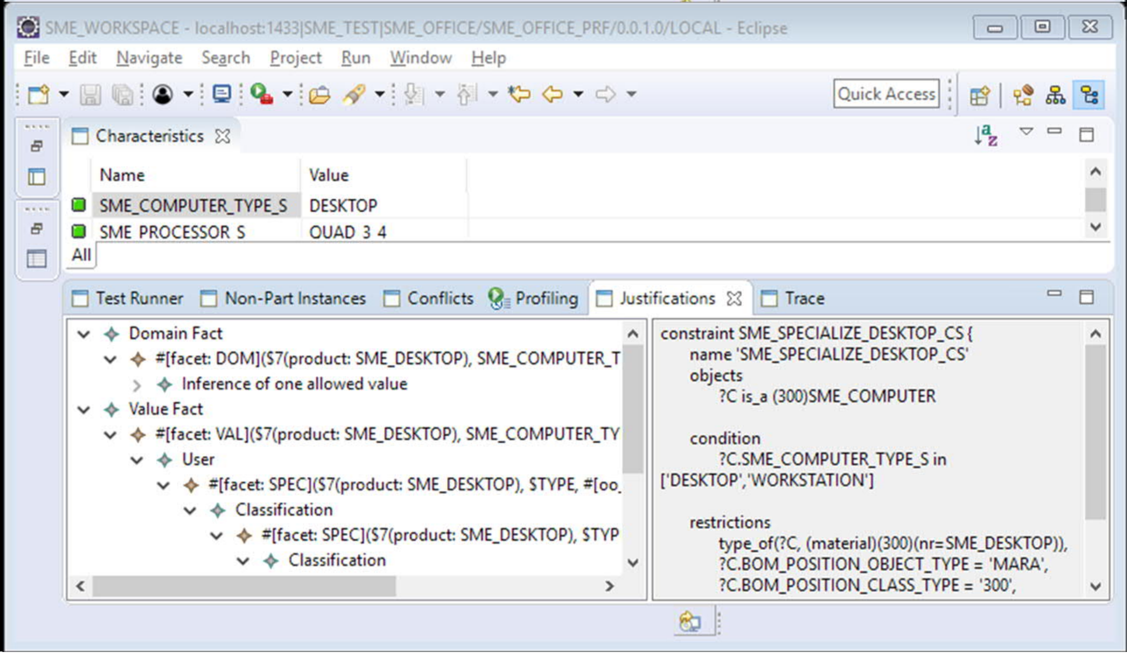
```
Domain Fact
  #[facet: DOM]($7(product: SME_DESKTOP), SME_COMPUTER_T
    Combination of existing restrictions
      #[facet: DOM]($7(product: SME_DESKTOP), SME_COMPI
        SME_COMPUTER_INT_DEPS_CS
        SME_COMPUTER_INT_DEPS_CS
        SME_COMPUTER_INT_DEPS_CS
      #[facet: DOM]($7(product: SME_DESKTOP), SME_COMPI
Value Fact
  #[facet: VAL]($7(product: SME_DESKTOP), SME_COMPUTER_TY
```

```
constraint SME_COMPUTER_INT_DEPS_CS {
    name 'SME_COMPUTER_INT_DEPS_CS'
    objects
        ?C is_a (300)SME_COMPUTER,
        ?W is_a (300)SME_WORKPLACE

    condition
        part_of(?C, ?W)

    restrictions

// restrict type of computer
        ?C.domain SME_COMPUTER_TYPE_S in
```



SME_WORKSPACE - localhost:1433|SME_TEST|SME_OFFICE/SME_OFFICE_PRF/0.0.1.0/LOCAL - Eclipse
File Edit Navigate Search Project Run Window Help
Quick Access
Characteristics
Test Runner
Non-Part Instances
Conflicts
Profiling
Justifications
Trace

Name	Value
SME_COMPUTER_TYPE_S	DESKTOP
SME_PROCESSOR_S	QUAD 3 4

All

```
Domain Fact
  #[facet: DOM]($7(product: SME_DESKTOP), SME_COMPUTER_T
    Inference of one allowed value
Value Fact
  #[facet: VAL]($7(product: SME_DESKTOP), SME_COMPUTER_TY
    User
      #[facet: SPEC]($7(product: SME_DESKTOP), $TYPE, #[oo,
        Classification
          #[facet: SPEC]($7(product: SME_DESKTOP), $TYP
            Classification
```

```
constraint SME_SPECIALIZE_DESKTOP_CS {
    name 'SME_SPECIALIZE_DESKTOP_CS'
    objects
        ?C is_a (300)SME_COMPUTER

    condition
        ?C.SME_COMPUTER_TYPE_S in
['DESKTOP','WORKSTATION']

    restrictions
        type_of(?C, (material)(300)(nr=SME_DESKTOP)),
        ?C.BOM_POSITION_OBJECT_TYPE = 'MARA',
        ?C.BOM_POSITION_CLASS_TYPE = '300',
```

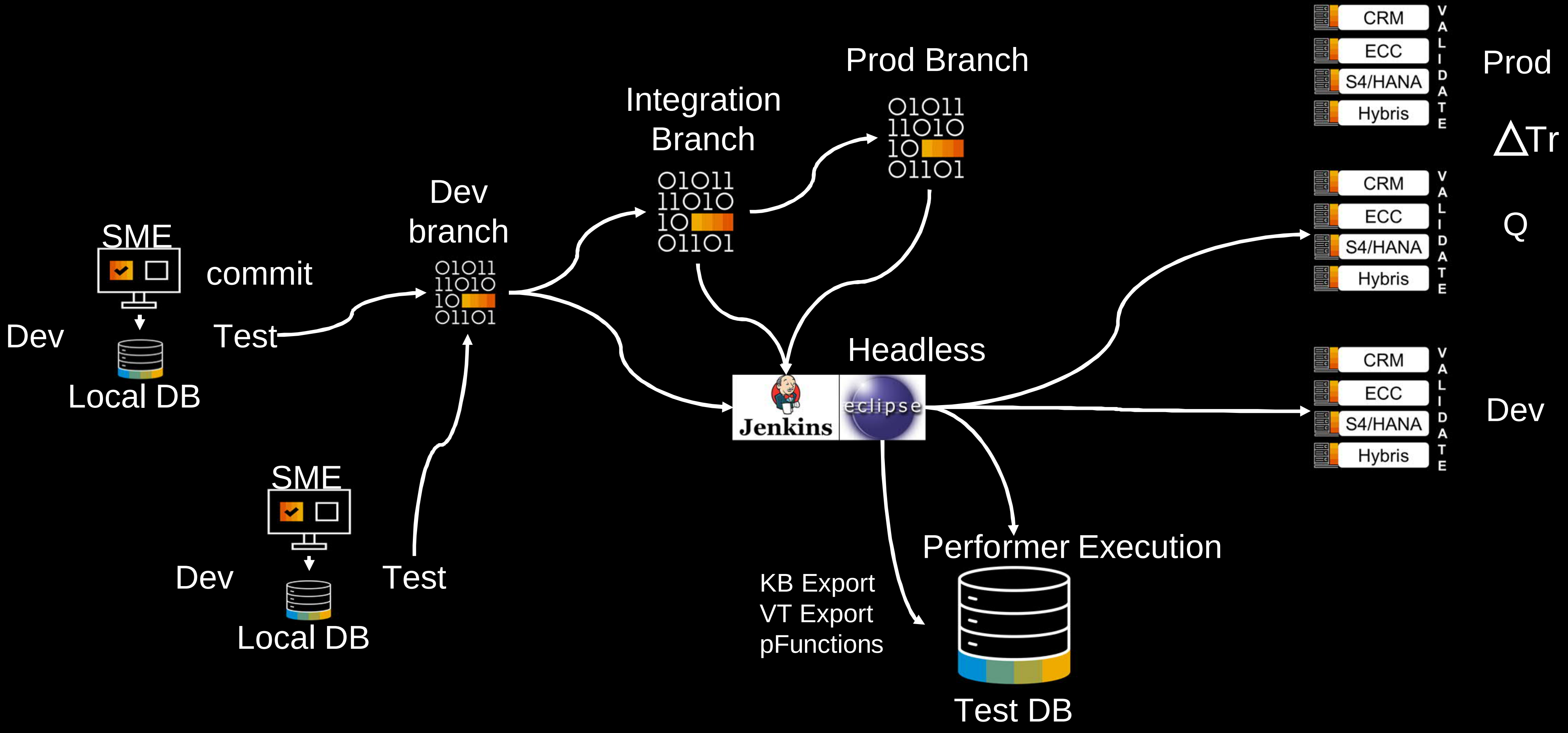
Solution Modeling Devops

Devops Support In SSC

Implementing devops practices helps reduce time between making a change in a solution model and moving this change into production while ensuring high quality. SSC provides full support for devops practices :

- Solution models can be maintained in any source code repository which enables baseline creation and change management.
- Headless mode, which is, the execution of functions without GUI in automated manner is provided for setup of continuous build, deploy, and testing pipelines.
  - Latest changes in solution models can be exported to a test DB or to a backend system.
  - Test scripts can be executed to ensure behavior consistency.
  - Any open source build server like Jenkins or simple scripting can be use for daily build and testing.
  - Any issues can be highlighted earlier and fixed adequately to ensure good quality roll out in production system

Devops Setup



Devops Process in SME

Build Phase

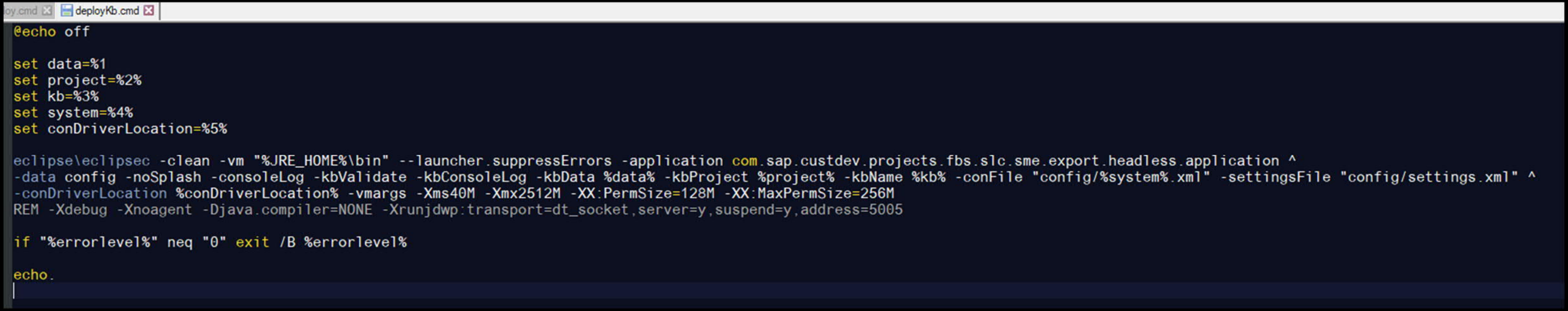
- Headless Export: Solution models are compiled and validated before export.
- Headless Export of Variant/External Variant Tables

Regression Test Phase

- Headless Execution of Performer Scripts: Execution status codes for success/failures printed in console logs enabling easy dashboard reporting.
- Headless XML Restore: Multiple combinations are supported to test different scenarios at the same time.

Headless Commands Sample

Simple commands which can be executed through cmd / shell scripts, and can be included as part of Jenkins build server for automated deployment, testing, and notifications.



```
@echo off

set data=%1
set project=%2%
set kb=%3%
set system=%4%
set conDriverLocation=%5%

eclipse\eclipsec -clean -vm "%JRE_HOME%\bin" --launcher.suppressErrors -application com.sap.custdev.projects.fbs.slc.sme.export.headless.application ^
-data config -noSplash -consoleLog -kbValidate -kbConsoleLog -kbData %data% -kbProject %project% -kbName %kb% -conFile "config/%system%.xml" -settingsFile "config/settings.xml" ^
-conDriverLocation %conDriverLocation% -vmargs -Xms40M -Xmx2512M -XX:PermSize=128M -XX:MaxPermSize=256M
REM -Xdebug -Xnoagent -Djava.compiler=NONE -Xrunjdwp:transport=dt_socket,server=y,suspend=y,address=5005

if "%errorlevel%" neq "0" exit /B %errorlevel%

echo.
```

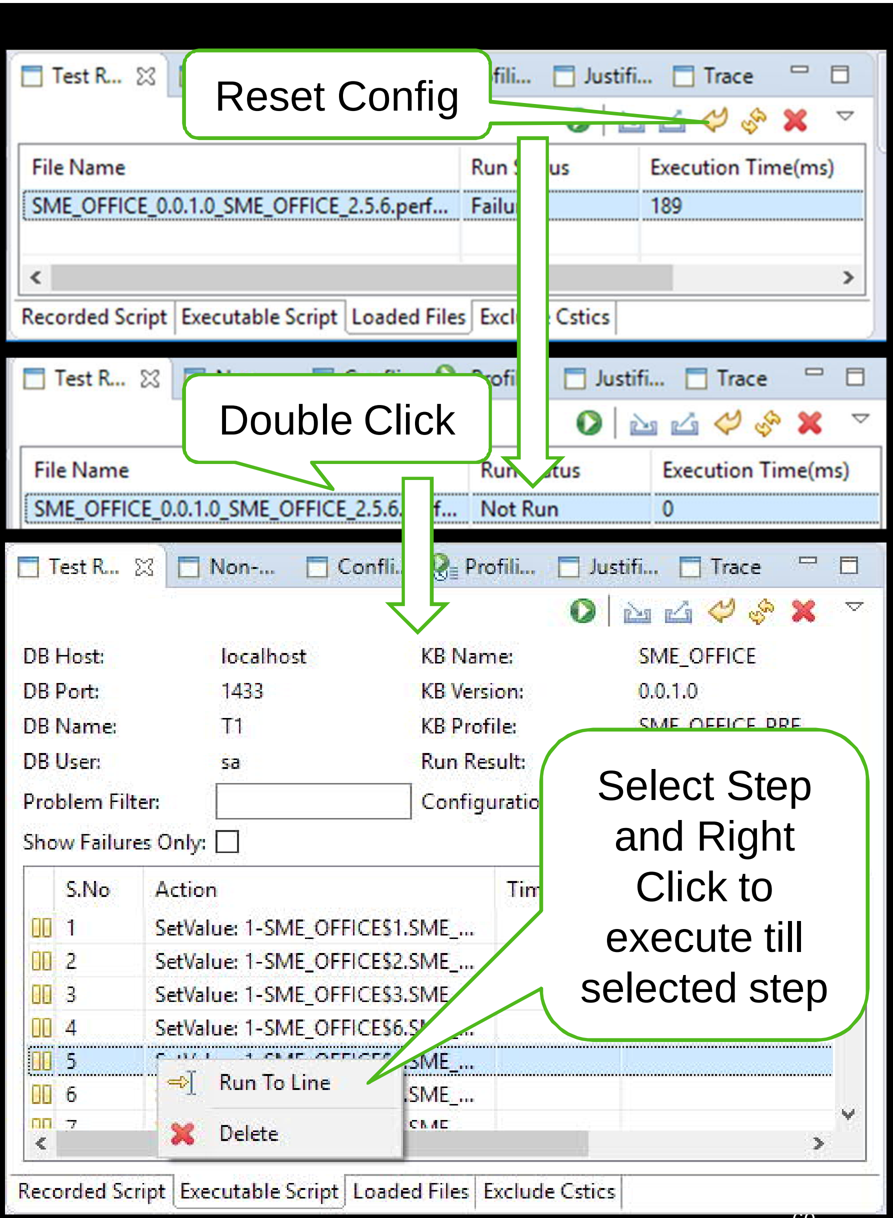
Debugging Solution Models

Debugging Solution Models

Configuration Preparation

To reduce tedious manual effort of executing preceding configuration steps before a problematic area can be profiled, configuration step execution features can be used :

- Reset the configuration which needs to be profiled.
- Double-click on the script to load all the executable steps
- Identify configuration step till which the configuration should be done.
- Select the step -> Right-click -> Run To Line. This will execute all preceding steps and prepare the configuration for next steps.
- Now, profiling and tracing can be activated to debug the problematic area.
- Subsequent steps can be selected and executed. This can help updating the configuration in steps/batches.



Reset Config

File Name	Run Status	Execution Time(ms)
SME_OFFICE_0.0.1.0_SME_OFFICE_2.5.6.perf...	Failu	189

Double Click

File Name	Run Status	Execution Time(ms)
SME_OFFICE_0.0.1.0_SME_OFFICE_2.5.6.perf...	Not Run	0

DB Host:	localhost	KB Name:	SME_OFFICE
DB Port:	1433	KB Version:	0.0.1.0
DB Name:	T1	KB Profile:	SME_OFFICE_PRF
DB User:	sa	Run Result:	
Problem Filter:		Configuratio	

Show Failures Only: ☐

S.No	Action	Tim
1	SetValue: 1-SME_OFFICE$1.SME_...	
2	SetValue: 1-SME_OFFICE$2.SME_...	
3	SetValue: 1-SME_OFFICE$3.SME	
4	SetValue: 1-SME_OFFICE$6.S...	
5	...SME_...	
6	...SME_...	
7		

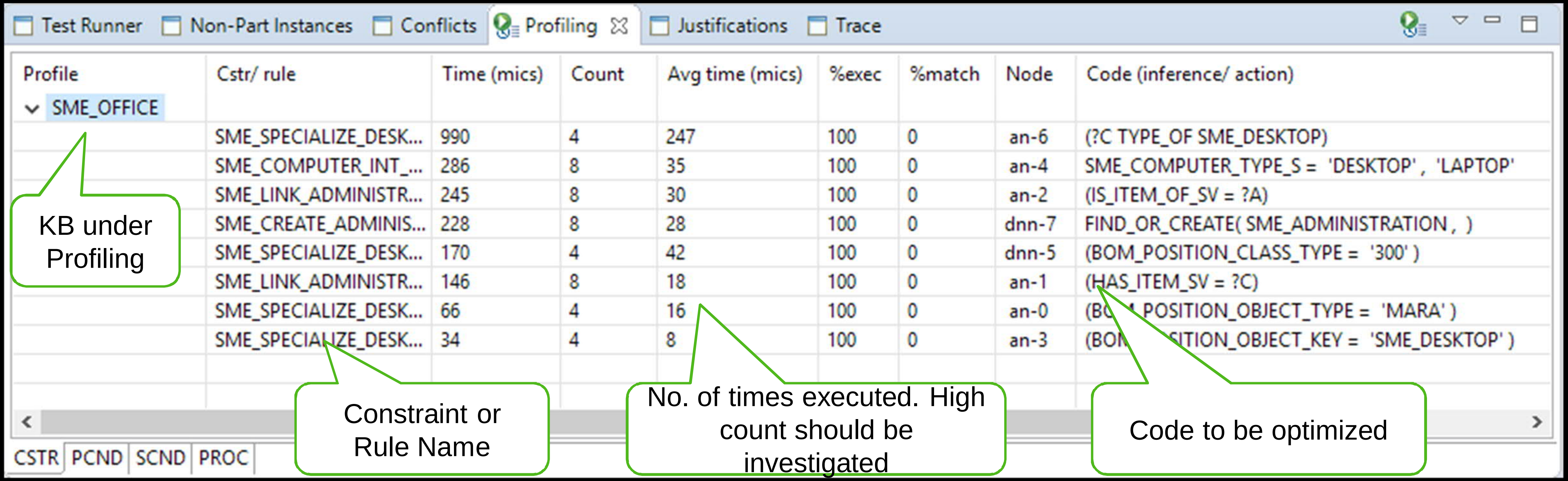
Run To Line

Delete

Select Step and Right Click to execute till selected step

Debugging Solution Models
Profiling

Profiling provides detailed information about how a solution configuration model behaves from performance perspective.



Profile	Cstr/ rule	Time (mics)	Count	Avg time (mics)	%exec	%match	Node	Code (inference/ action)
⌄ SME_OFFICE								
	SME_SPECIALIZE_DESK...	990	4	247	100	0	an-6	(?C TYPE_OF SME_DESKTOP)
	SME_COMPUTER_INT_...	286	8	35	100	0	an-4	SME_COMPUTER_TYPE_S = 'DESKTOP' , 'LAPTOP'
	SME_LINK_ADMINISTR...	245	8	30	100	0	an-2	(IS_ITEM_OF_SV = ?A)
	SME_CREATE_ADMINIS...	228	8	28	100	0	dnn-7	FIND_OR_CREATE(SME_ADMINISTRATION ,)
	SME_SPECIALIZE_DESK...	170	4	42	100	0	dnn-5	(BOM_POSITION_CLASS_TYPE = '300')
	SME_LINK_ADMINISTR...	146	8	18	100	0	an-1	(HAS_ITEM_SV = ?C)
	SME_SPECIALIZE_DESK...	66	4	16	100	0	an-0	(BOM_POSITION_OBJECT_TYPE = 'MARA')
	SME_SPECIALIZE_DESK...	34	4	8	100	0	an-3	(BOM_POSITION_OBJECT_KEY = 'SME_DESKTOP')

KB under Profiling

Constraint or Rule Name

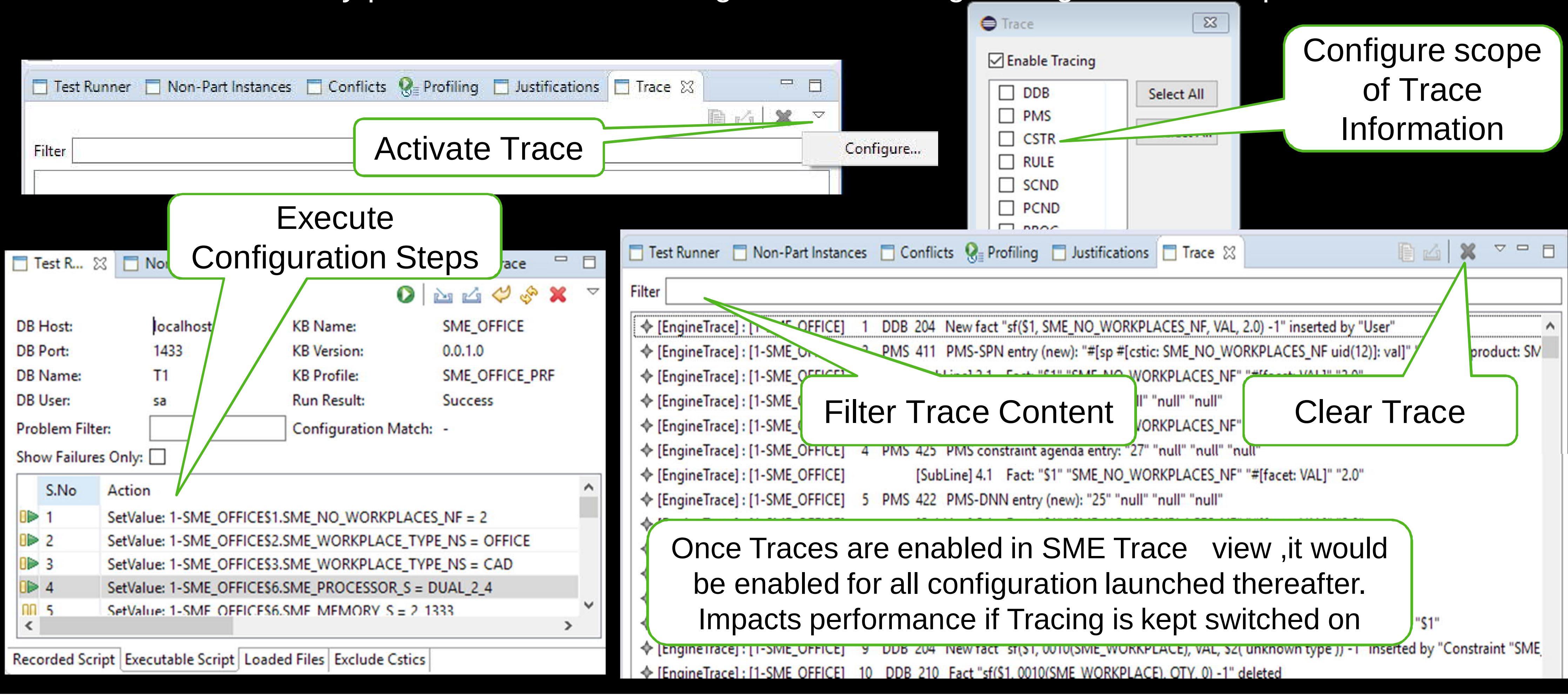
No. of times executed. High count should be investigated

Code to be optimized

CSTR : Constraints | PCND : Preconditions | SCND : Selection Conditions | PROC : Procedures

Debugging Solution Models: Tracing

Tracing feature helps in understanding processing of dependencies for a Solution Configuration. Trace can be activated at any point in time of the configuration enabling management of scope of the trace.



Activate Trace

Configure scope of Trace Information

☑ Enable Tracing

- □ DDB
- □ PMS
- □ CSTR
- □ RULE
- □ SCND
- □ PCND

Execute Configuration Steps

DB Host:	localhost
DB Port:	1433
DB Name:	T1
DB User:	sa
Problem Filter:	
Show Failures Only:	□
KB Name:	SME_OFFICE
KB Version:	0.0.1.0
KB Profile:	SME_OFFICE_PRF
Run Result:	Success
Configuration Match:	-

S.No	Action
1	SetValue: 1-SME_OFFICE$1.SME_NO_WORKPLACES_NF = 2
2	SetValue: 1-SME_OFFICE$2.SME_WORKPLACE_TYPE_NS = OFFICE
3	SetValue: 1-SME_OFFICE$3.SME_WORKPLACE_TYPE_NS = CAD
4	SetValue: 1-SME_OFFICE$6.SME_PROCESSOR_S = DUAL_2_4
5	SetValue: 1-SME_OFFICE$6.SME_MEMORY_S = 2_1333

Filter Trace Content

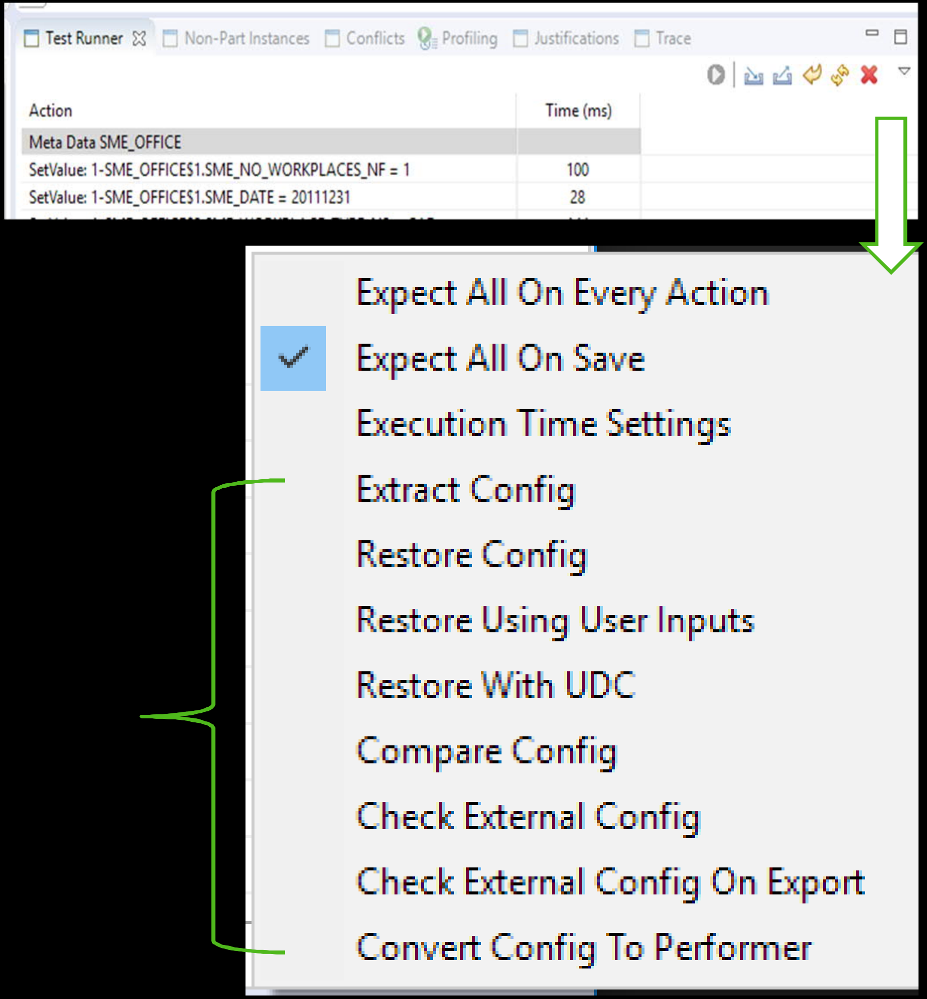
Clear Trace

```
[EngineTrace] : [1-SME_OFFICE]   1   DDB  204   New fact "sf($1, SME_NO_WORKPLACES_NF, VAL, 2.0) -1" inserted by "User"
[EngineTrace] : [1-SME_O         2   PMS  411   PMS-SPN entry (new): "#[sp #[cstic: SME_NO_WORKPLACES_NF uid(12)]: val]"          product: SM
[EngineTrace] : [1-SME_OFFICE]                          Fact: "$1" "SME_NO_WORKPLACES_NF" "#[facet: VAL]" "2.0"
[EngineTrace] : [1-SME_                                 l" "null" "null"
[EngineTrace] : [1-SME_                                 VORKPLACES_NF"
[EngineTrace] : [1-SME_OFFICE]   4   PMS  425   PMS constraint agenda entry: "27" "null" "null" "null"
[EngineTrace] : [1-SME_OFFICE]               [SubLine] 4.1    Fact: "$1" "SME_NO_WORKPLACES_NF" "#[facet: VAL]" "2.0"
[EngineTrace] : [1-SME_OFFICE]   5   PMS  422   PMS-DNN entry (new): "25" "null" "null" "null"
                                                                                                                   "$1"
[EngineTrace] : [1-SME_OFFICE]   9   DDB  204   New fact "sf($1, 0010(SME_WORKPLACE), VAL, $2( unknown type )) -1" inserted by "Constraint "SME
[EngineTrace] : [1-SME_OFFICE]   10  DDB  210   Fact "sf($1, 0010(SME_WORKPLACE), QTY, 0) -1" deleted
```

Once Traces are enabled in SME Trace view ,it would be enabled for all configuration launched thereafter. Impacts performance if Tracing is kept switched on

Test Runner View: Additional Test Options

- Extract Config: Test configuration can be extracted in xml form for restoring in test systems
- Restore Config: Configuration extracted from productive systems can be restored in SME for test and debug purpose
- Restore Using User Inputs: Recreate configuration using alternate user input restore mechanism
- Compare Config: Two configuration xml files can be compared for any differences
- Check External Config: The configuration result which is submitted to backend system can be verified.
- Check External Config On Export: Verify config result during export only.
- Convert Config To Performer: Converts the configuration result file extracted from backend systems into performer file format for easier debugging and testing



Action	Time (ms)
Meta Data SME_OFFICE	
SetValue: 1-SME_OFFICES1.SME_NO_WORKPLACES_NF = 1	100
SetValue: 1-SME_OFFICES1.SME_DATE = 20111231	28

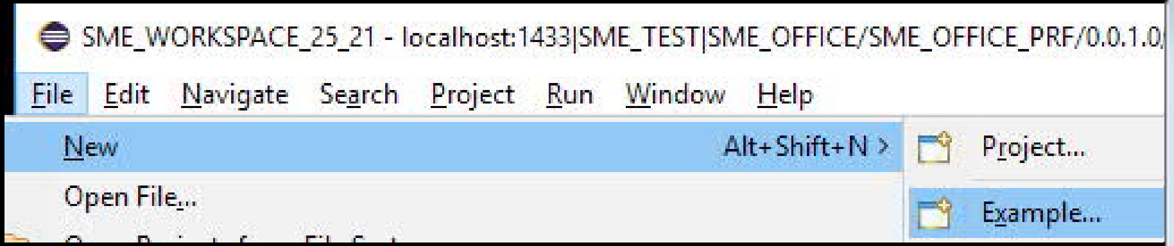
Solution Model Examples

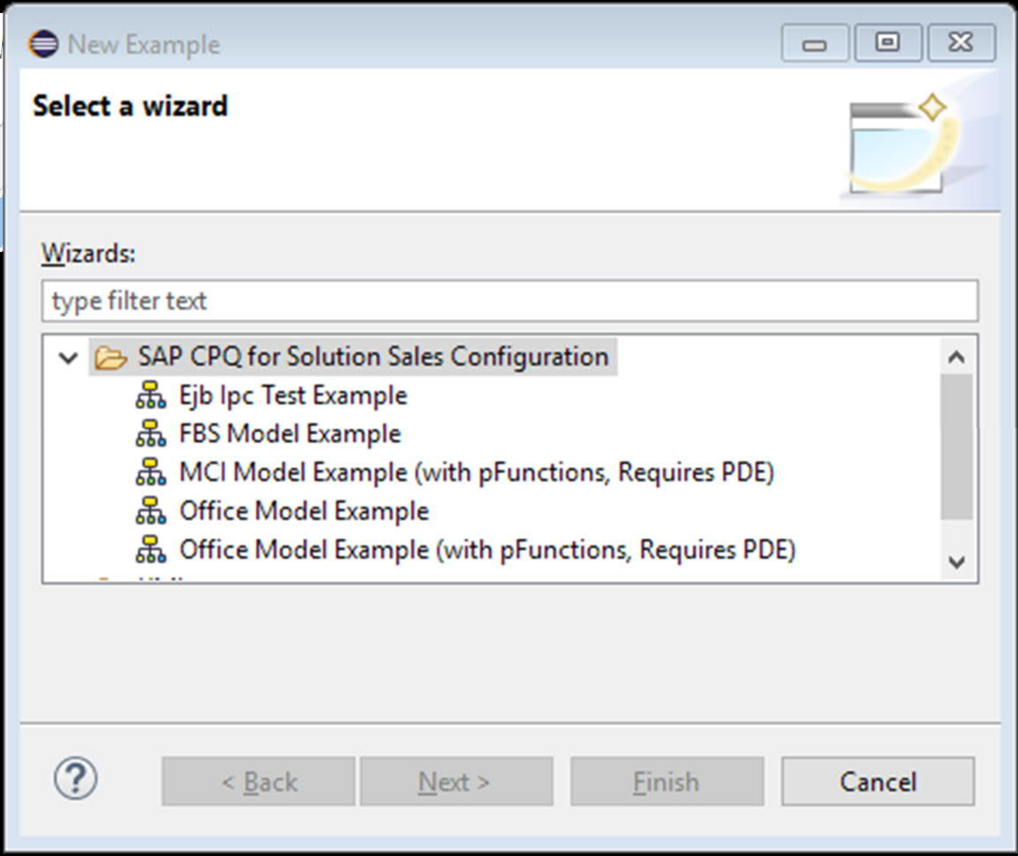
Solution Model Examples

To help modelers quickly get started and understand concepts of solution modeling, some example projects are shipped along with SME. These are :

- SME Office: Simulates configuration of an office. Also, demonstrates how custom user exists can be written. This is also accompanied along with a Junit project which lists all the engine API and it usage via testing of the SME office solution.
- KBO and MCI: Elaborates concepts of knowledge base orchestration and multi-configuration instances. A document is also provided along with as reference guide to this example.

Installing Sample Projects


SME_WORKSPACE_25_21 - localhost:1433|SME_TEST|SME_OFFICE/SME_OFFICE_PRF/0.0.1.0/
File Edit Navigate Search Project Run Window Help
New Alt+Shift+N >
Open File...
Project...
Example...


New Example
Select a wizard
Wizards:
type filter text
SAP CPQ for Solution Sales Configuration
Ejb Ipc Test Example
FBS Model Example
MCI Model Example (with pFunctions, Requires PDE)
Office Model Example
Office Model Example (with pFunctions, Requires PDE)
< Back
Next >
Finish
Cancel

- EJB IPC Test : Provides Junits for FBS Office example, thus giving insights on how SSC Engine API’s can be used
- FBS Model Example: Simple model showcasing the KBO concepts
- MCI Model Example: Sample model showcasing the KBO and multi-configuration instances concepts

Thank you.

Contact information:

Vivek Kumar Gupta
Program Lead : Solution Sales Configuration
vivek.kumar.Gupta@sap.com

THE BEST RUN SAP