



TAURUS

Integrating Tools in RDz
Serena ChangeMan Integration

May 02, 2011

Udo Partsch (SoforTe GmbH)

partsch@soforte.de

<http://www.soforte.com>

- **Focus**

- SoforTe is specialized in optimizing mainframe based application development processes and team collaboration
- RDz, RTC, RAA,
- SCLM, Endeavor, ChangeMan, ...

- **Clients**

- Mainframe customers across all industries

- **Technology**

- TAURUS



- **Lessons learned**

- Integrating existing development processes in RDz is time consuming and requires special skills
- Risk – Implementation is not done to the full extend and developers reject RDz
- Current “out of the box” solutions are not sufficient for 80% of European RDz implementations

- **Consequence**

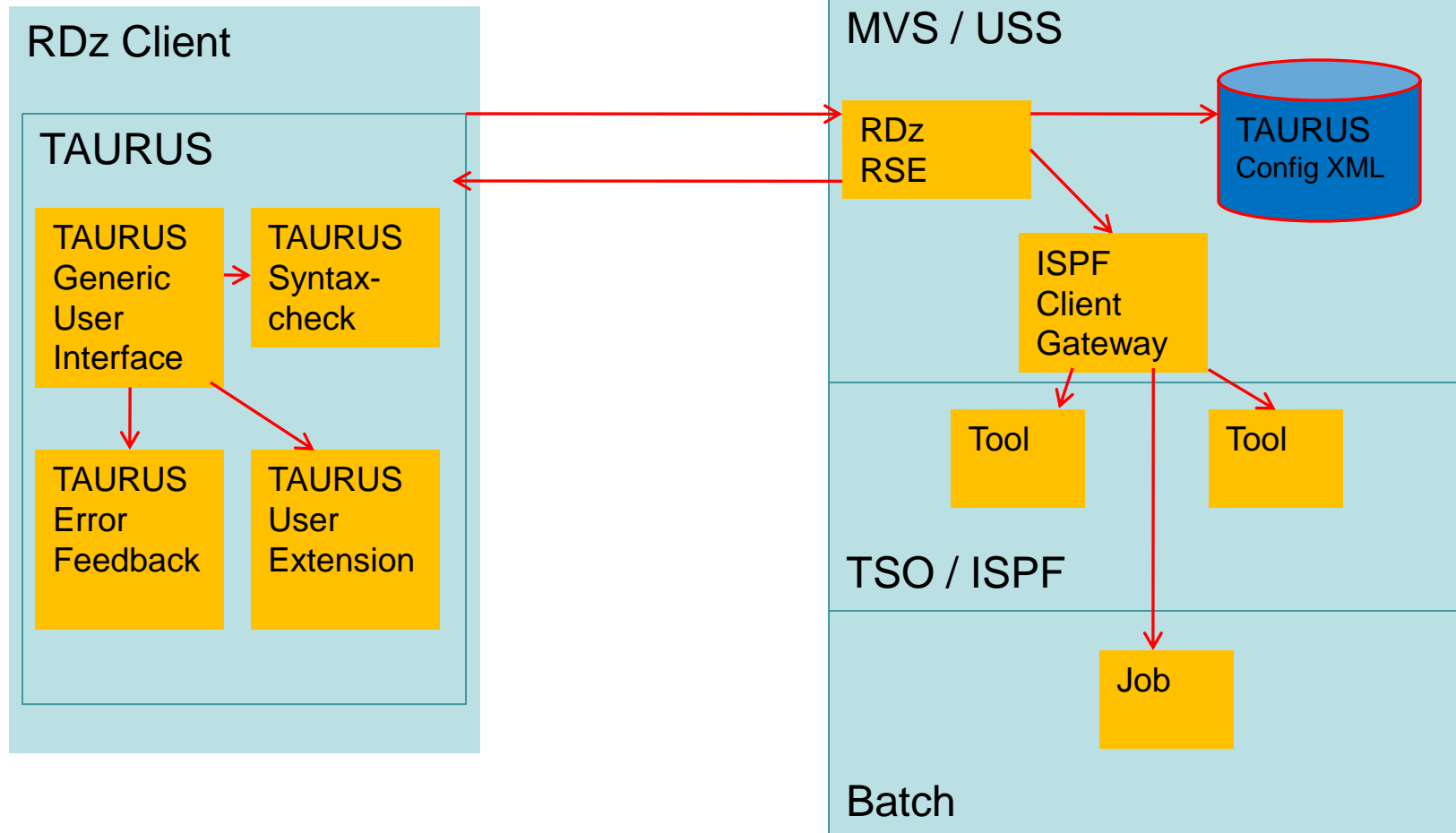
- RDz Implementations need to be simplified and easily customizable for the customers existing development process
- => development of TAURUS Integration toolkit

- **Model driven tool integration**
 - Modeling of user interfaces and tool integration
 - Supports integration of SCLM, Endeavor, ChangeMan, ...
 - Fully customizable
 - Fully integrated into Eclipse and RDz
- **Reducing effort**
 - Integration with major SCM systems out of the box
 - Predefined models for different development processes
 - Integration with all kinds of tools like RAA, RTC, ...
 - Existing skill is sufficient – no plug-in development skill required
- **10 times faster compared to plug-in development**
 - Plug-in development becomes obsolete
 - Complete and deep integration becomes affordable
 - High functionality and quality of UI increase developer acceptability and productivity
 - The financial hurdle for RDz implementations is reduced significantly

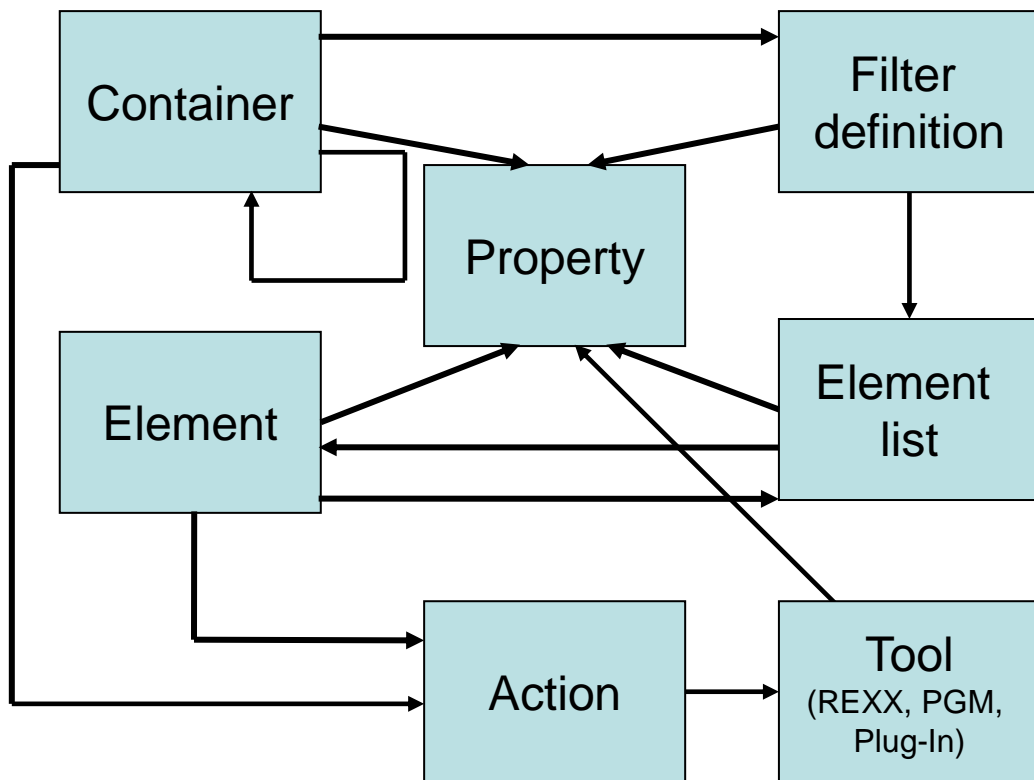
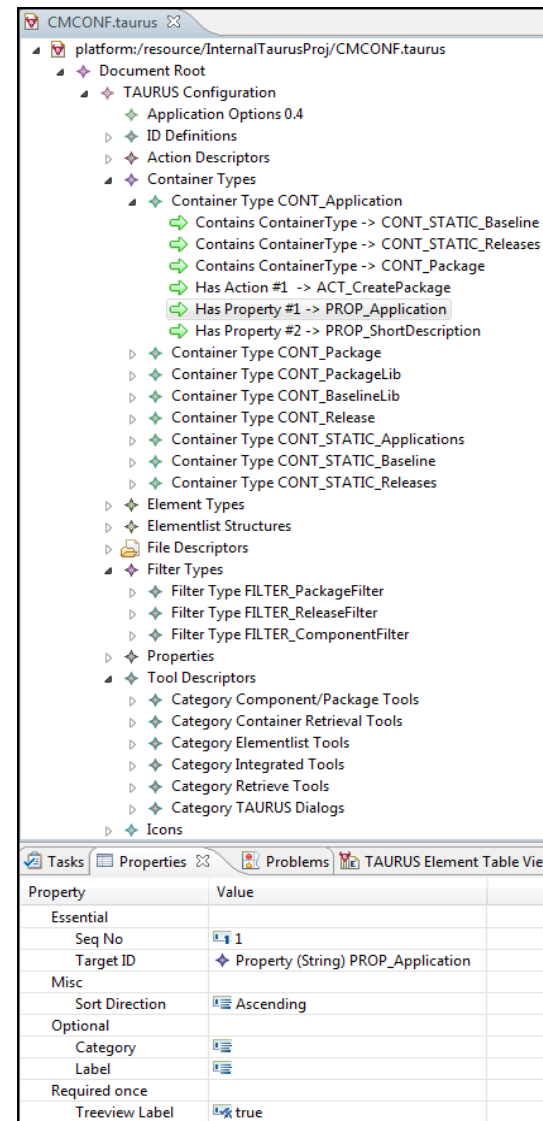
SoforTe – TAURUS – architecture

Windows / Linux

z/OS



SoforTe – TAURUS – modeling

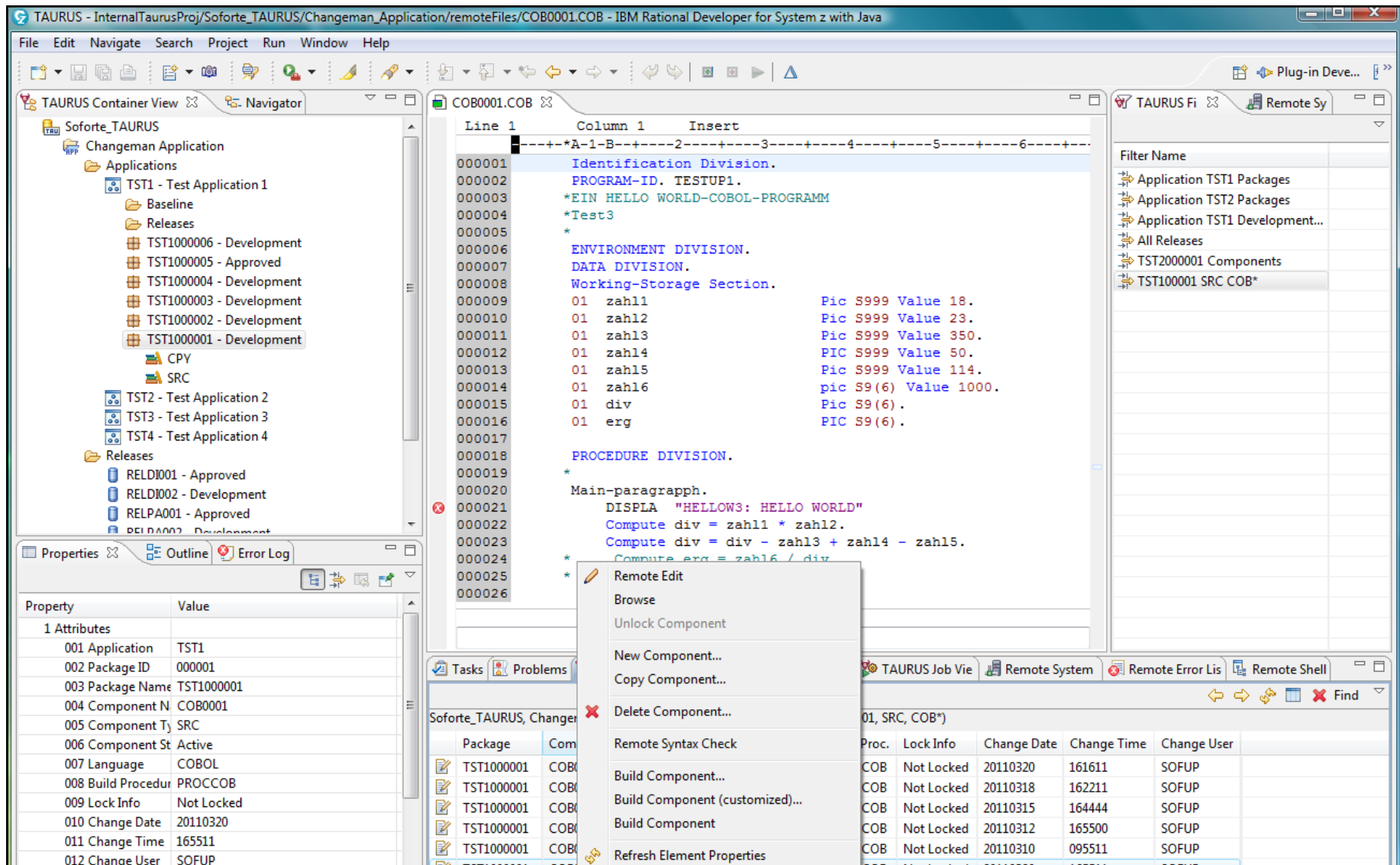



CMCONF.taurus

- platform:/resource/InternalTaurusProj/CMCONF.taurus
 - Document Root
 - TAURUS Configuration
 - Application Options 0.4
 - ID Definitions
 - Action Descriptors
 - Container Types
 - Container Type CONT_Application
 - Contains ContainerType -> CONT_STATIC_Baseline
 - Contains ContainerType -> CONT_STATIC_Releases
 - Contains ContainerType -> CONT_Package
 - Has Action #1 -> ACT_CreatePackage
 - Has Property #1 -> PROP_Application
 - Has Property #2 -> PROP_ShortDescription
 - Container Type CONT_Package
 - Container Type CONT_PackageLib
 - Container Type CONT_BaselineLib
 - Container Type CONT_Release
 - Container Type CONT_STATIC_Applications
 - Container Type CONT_STATIC_Baseline
 - Container Type CONT_STATIC_Releases
 - Element Types
 - Elementlist Structures
 - File Descriptors
 - Filter Types
 - Filter Type FILTER_PackageFilter
 - Filter Type FILTER_ReleaseFilter
 - Filter Type FILTER_ComponentFilter
 - Properties
 - Tool Descriptors
 - Category Component/Package Tools
 - Category Container Retrieval Tools
 - Category Elementlist Tools
 - Category Integrated Tools
 - Category Retrieve Tools
 - Category TAURUS Dialogs
 - Icons

Property	Value
Essential	
Seq No	1
Target ID	Property (String) PROP_Application
Misc	
Sort Direction	Ascending
Optional	
Category	
Label	
Required once	
Treeview Label	true

SoforTe – TAURUS – ChangeMan Integration



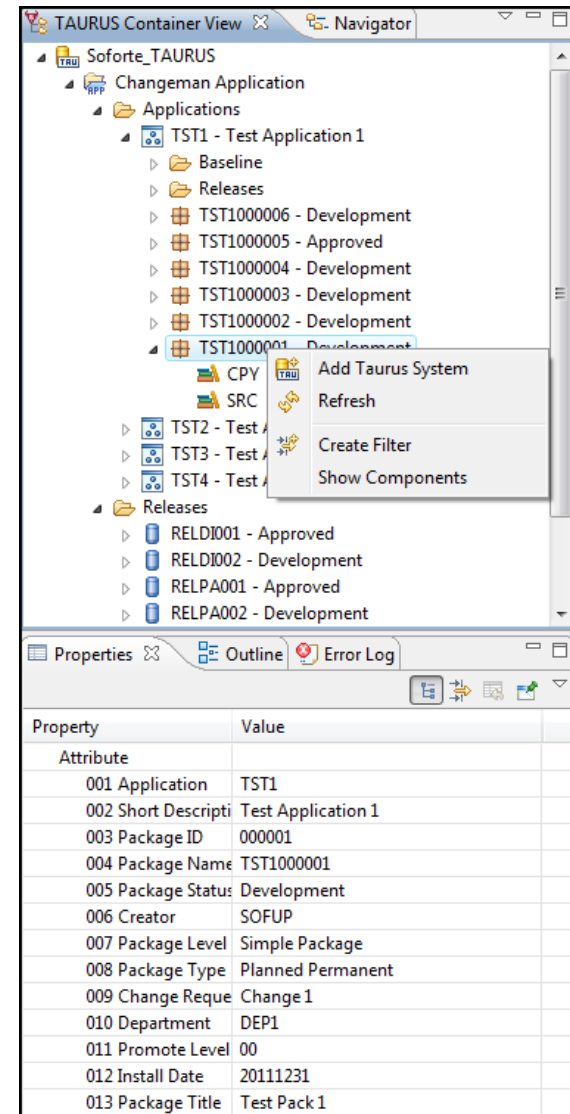
The screenshot displays the IBM Rational Developer for System z with Java interface. The main editor shows COBOL code for 'COB0001.COB'. The code includes sections for Identification, Environment, Data, Working-Storage, and Procedure Divisions. A context menu is open over the code, listing actions like 'Remote Edit', 'Browse', 'New Component...', 'Copy Component...', 'Delete Component...', 'Remote Syntax Check', 'Build Component...', 'Build Component (customized)...', 'Build Component', and 'Refresh Element Properties'. The left sidebar shows a project tree for 'Soforte_TAURUS' with sub-projects like 'ChangeMan Application' and 'Applications'. The bottom-left pane shows properties for the selected component, and the bottom-right pane shows a table of components with columns for Proc., Lock Info, Change Date, Change Time, and Change User.

Property	Value
1 Attributes	
001 Application	TST1
002 Package ID	000001
003 Package Name	TST1000001
004 Component Name	COB0001
005 Component Type	SRC
006 Component State	Active
007 Language	COBOL
008 Build Procedure	PROCCOB
009 Lock Info	Not Locked
010 Change Date	20110320
011 Change Time	165511
012 Change User	SOFUP

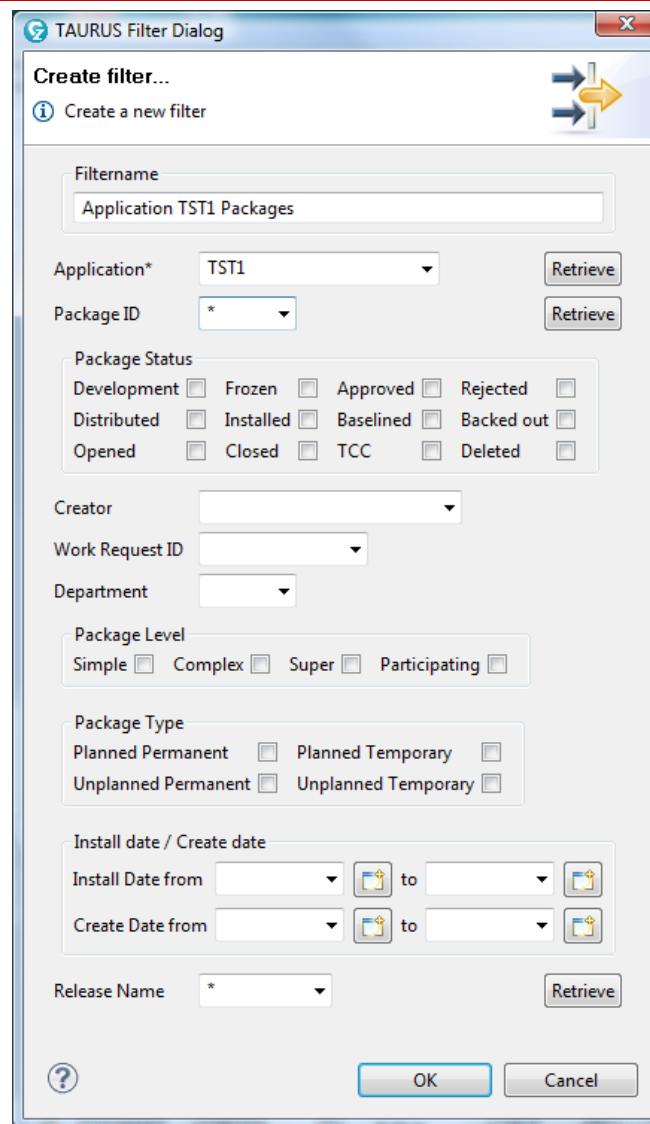
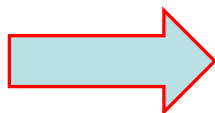
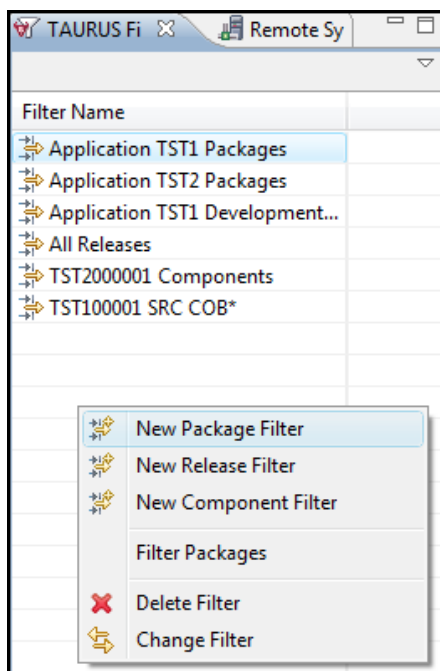
Proc.	Lock Info	Change Date	Change Time	Change User
COB	Not Locked	20110320	161611	SOFUP
COB	Not Locked	20110318	162211	SOFUP
COB	Not Locked	20110315	164444	SOFUP
COB	Not Locked	20110312	165500	SOFUP
COB	Not Locked	20110310	095511	SOFUP

Taurus - ChangeMan Tree View

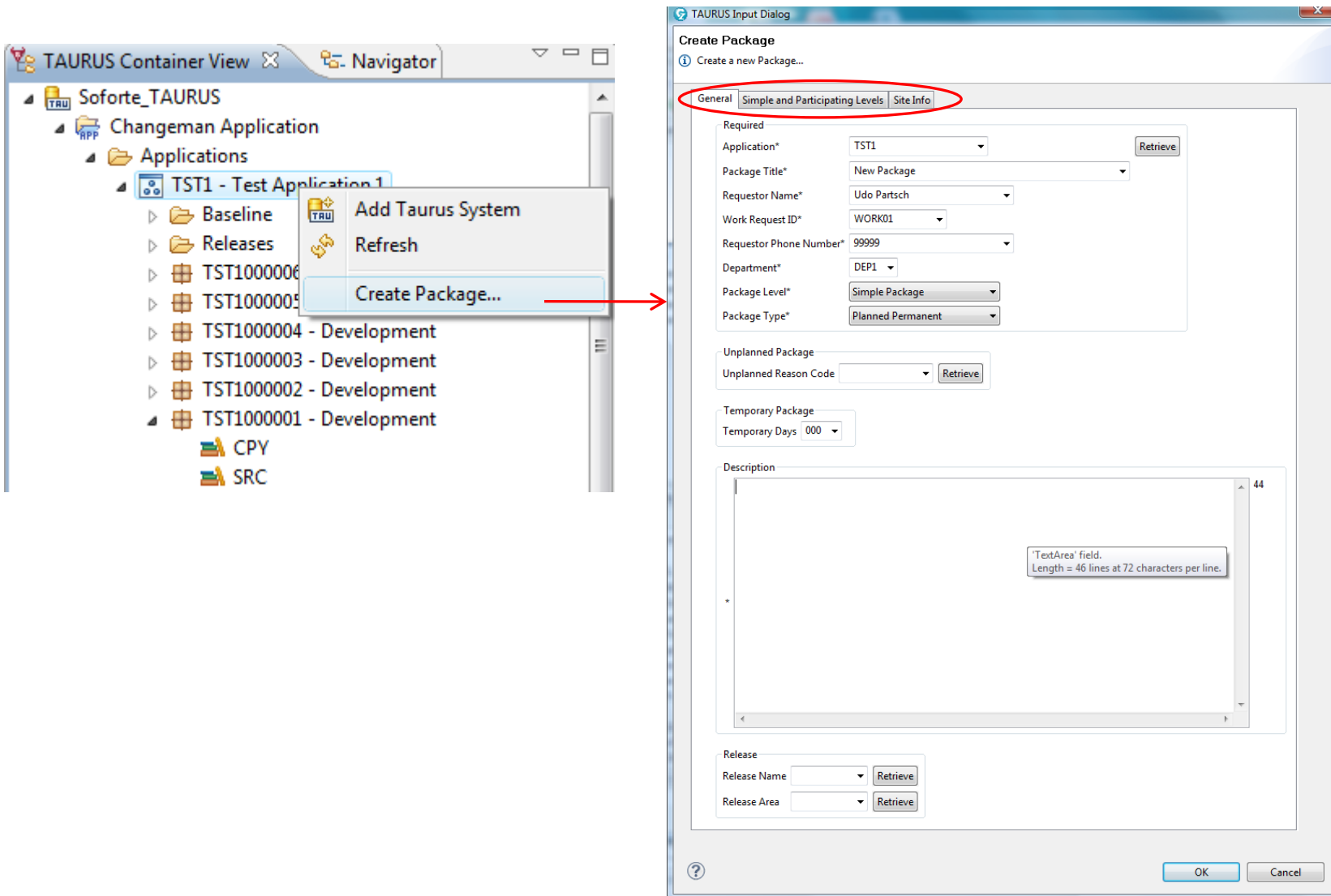
- **Supports a hierarchical view to:**
 - Applications
 - Packages
 - Baselines
 - Promotion Libs
 - Types
 - Releases (if ERO option is in use)



Taurus - Create a ChangeMan Package List



TAURUS – Create a new ChangeMan Package



The image shows two screenshots from the TAURUS application. The left screenshot displays the 'TAURUS Container View' with a tree structure under 'Soforte_TAURUS' > 'Changeman Application' > 'Applications' > 'TST1 - Test Application 1'. A context menu is open over the 'TST1000001 - Development' item, showing options: 'Add Taurus System', 'Refresh', and 'Create Package...'. A red arrow points from the 'Create Package...' option to the right screenshot.

The right screenshot shows the 'TAURUS Input Dialog' window titled 'Create Package'. The 'General' tab is selected and circled in red. The dialog contains the following fields and controls:

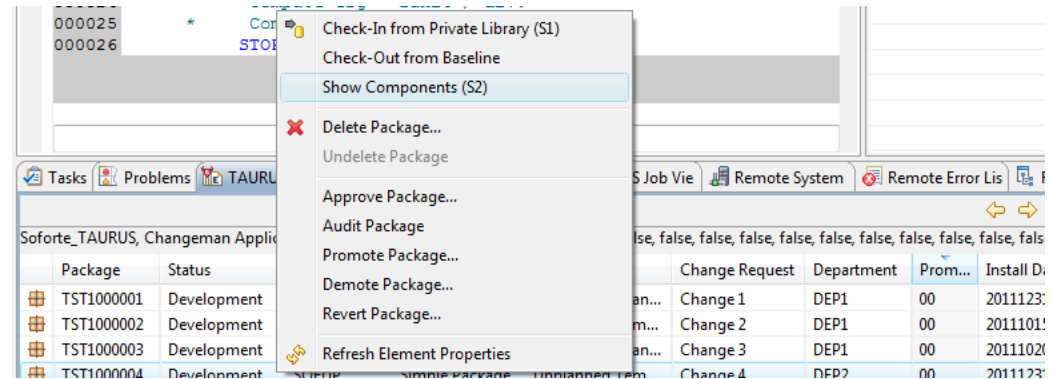
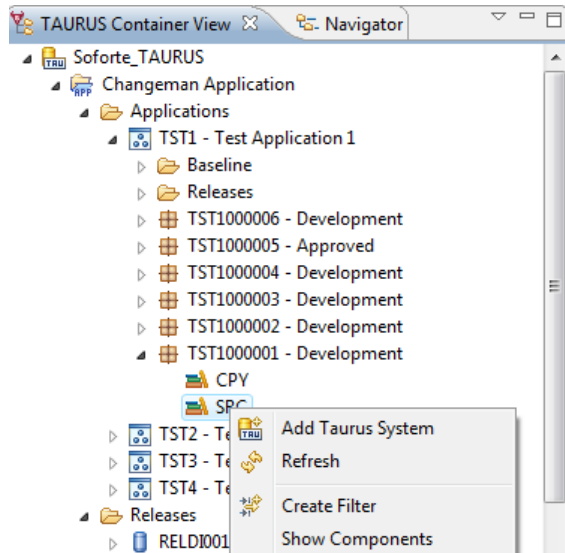
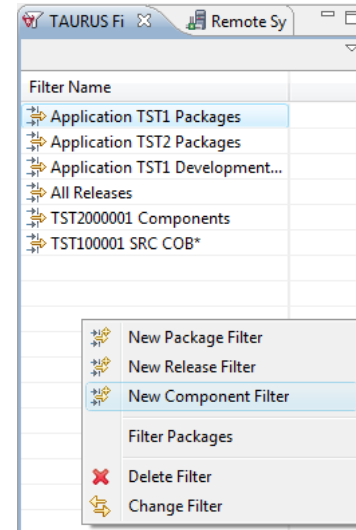
- Required:**
 - Application*: TST1 (dropdown) [Retrieve]
 - Package Title*: New Package (dropdown)
 - Requestor Name*: Udo Patsch (dropdown)
 - Work Request ID*: WORK01 (dropdown)
 - Requestor Phone Number*: 99999 (dropdown)
 - Department*: DEP1 (dropdown)
 - Package Level*: Simple Package (dropdown)
 - Package Type*: Planned Permanent (dropdown)
- Unplanned Package:**
 - Unplanned Reason Code: (dropdown) [Retrieve]
- Temporary Package:**
 - Temporary Days: 000 (dropdown)
- Description:**
 - TextArea field. Length = 46 lines at 72 characters per line. (44 lines visible)
- Release:**
 - Release Name: (dropdown) [Retrieve]
 - Release Area: (dropdown) [Retrieve]

Buttons: OK, Cancel

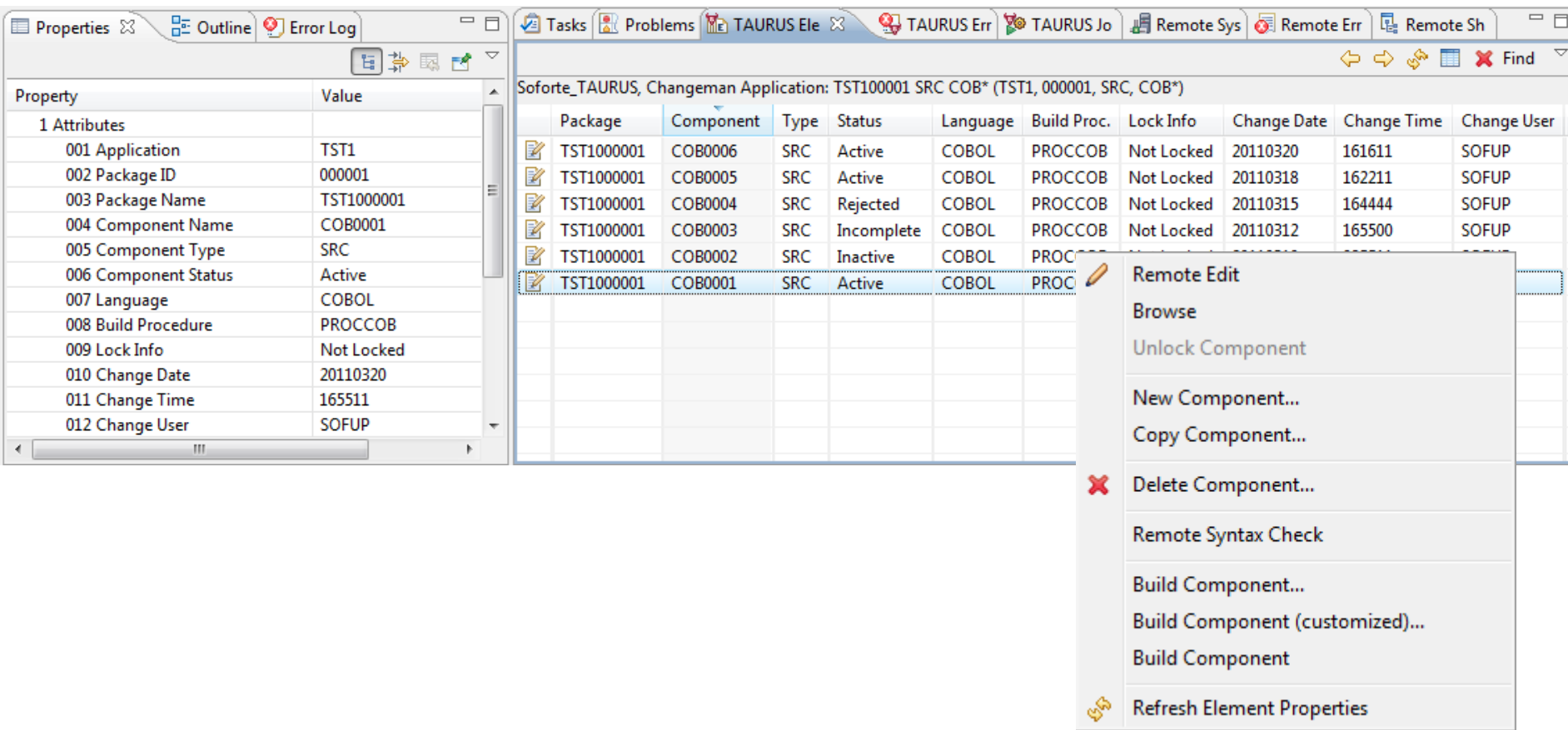
TAURUS – Create a ChangeMan Component List

- **Several ways to create a component list:**

- From the filter view
- From the package list
- From the tree view



TAURUS – ChangeMan Component Functions



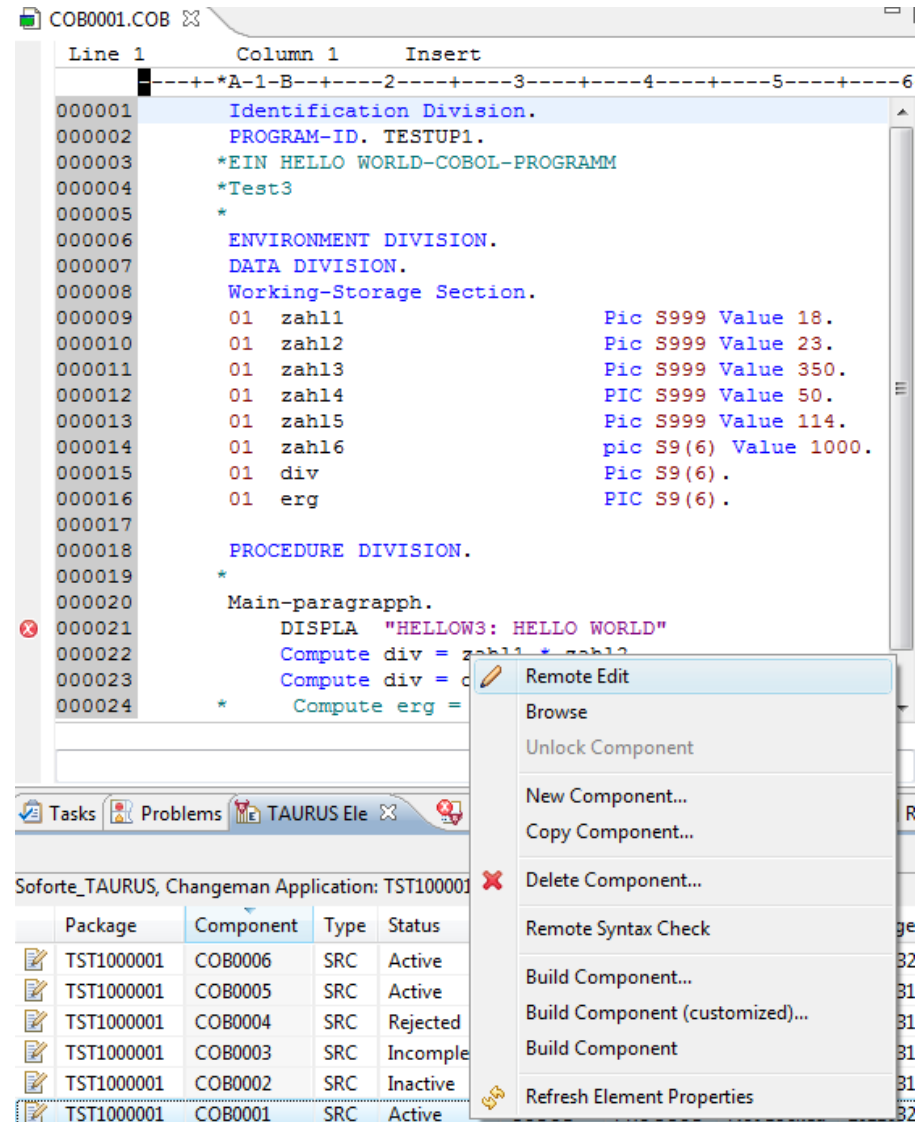
The screenshot displays the TAURUS ChangeMan interface. On the left, a 'Properties' window shows the attributes of a selected component. The main window displays a table of components for the application 'TST100001 SRC COB*'. A context menu is open over the table, listing various actions such as 'Remote Edit', 'Delete Component...', and 'Build Component...'. The table columns include Package, Component, Type, Status, Language, Build Proc., Lock Info, Change Date, Change Time, and Change User.

Package	Component	Type	Status	Language	Build Proc.	Lock Info	Change Date	Change Time	Change User
TST1000001	COB0006	SRC	Active	COBOL	PROCCOB	Not Locked	20110320	161611	SOFUP
TST1000001	COB0005	SRC	Active	COBOL	PROCCOB	Not Locked	20110318	162211	SOFUP
TST1000001	COB0004	SRC	Rejected	COBOL	PROCCOB	Not Locked	20110315	164444	SOFUP
TST1000001	COB0003	SRC	Incomplete	COBOL	PROCCOB	Not Locked	20110312	165500	SOFUP
TST1000001	COB0002	SRC	Inactive	COBOL	PROCCOB	Not Locked			
TST1000001	COB0001	SRC	Active	COBOL	PROCCOB	Not Locked			

- Remote Edit
- Browse
- Unlock Component
- New Component...
- Copy Component...
- Delete Component...
- Remote Syntax Check
- Build Component...
- Build Component (customized)...
- Build Component
- Refresh Element Properties

Taurus – Editing a ChangeMan Component

- **Modeling your own Edit Scenario:**
 - Scenario A:
 - Remote Edit of a package component
 - Scenario B:
 - Check-Out from a package to the RDz workspace
 - Local Edit
 - Check-In
 - Scenario C:
 - Check-out from the baseline to a private PDS
 - Remote Edit
 - Check-In package
 -
- **Support of the local and remote syntax check**



The screenshot displays the Taurus IDE interface. The main window shows a COBOL program editor for 'COB0001.COB'. The code is structured as follows:

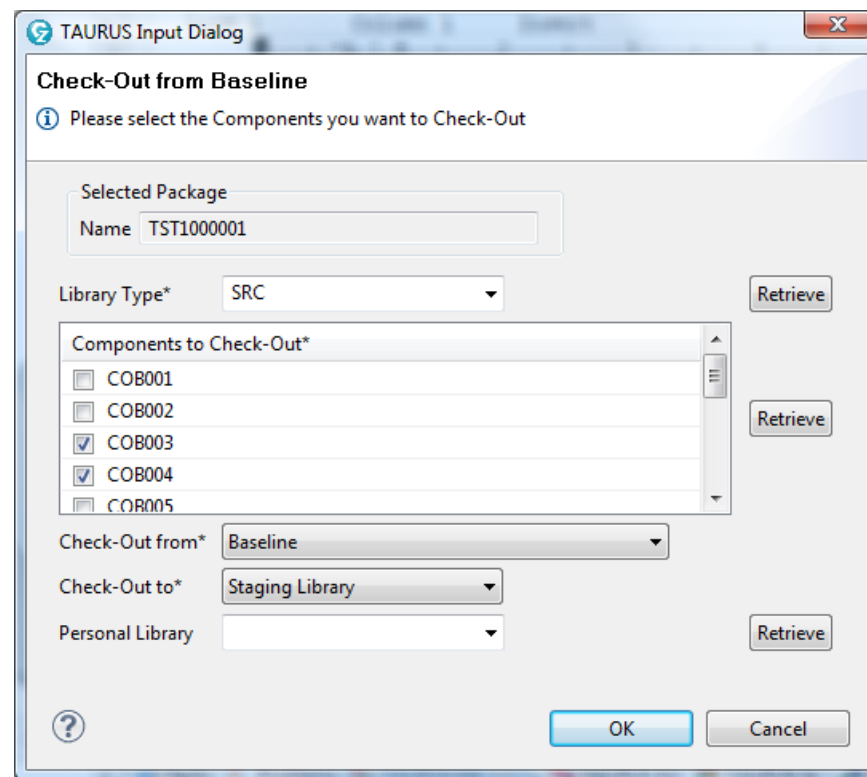
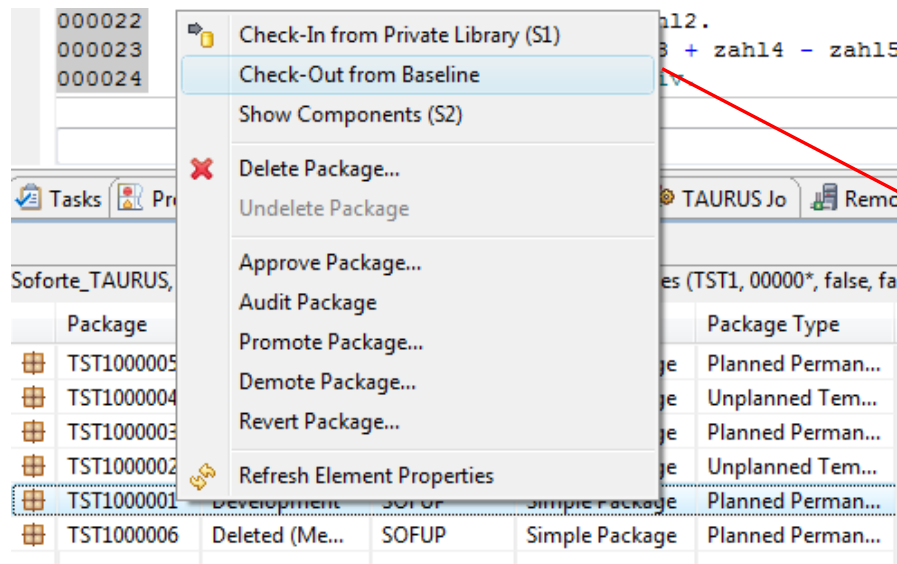
```
Line 1      Column 1      Insert
-----*A-1-B-----2-----3-----4-----5-----6
000001      Identification Division.
000002      PROGRAM-ID. TESTUP1.
000003      *EIN HELLO WORLD-COBOL-PROGRAMM
000004      *Test3
000005      *
000006      ENVIRONMENT DIVISION.
000007      DATA DIVISION.
000008      Working-Storage Section.
000009      01  zahl1                      Pic S999 Value 18.
000010      01  zahl2                      Pic S999 Value 23.
000011      01  zahl3                      Pic S999 Value 350.
000012      01  zahl4                      PIC S999 Value 50.
000013      01  zahl5                      Pic S999 Value 114.
000014      01  zahl6                      pic S9(6) Value 1000.
000015      01  div                        Pic S9(6).
000016      01  erg                        PIC S9(6).
000017
000018      PROCEDURE DIVISION.
000019      *
000020      Main-paragraph.
000021      DISPLA  "HELLOWS: HELLO WORLD"
000022      Compute div = zahl1 * zahl2
000023      Compute div = c
000024      *      Compute erg =
```

A context menu is open over the code, with 'Remote Edit' selected. Other options include 'Browse', 'Unlock Component', 'New Component...', 'Copy Component...', 'Delete Component...', 'Remote Syntax Check', 'Build Component...', 'Build Component (customized)...', 'Build Component', and 'Refresh Element Properties'.

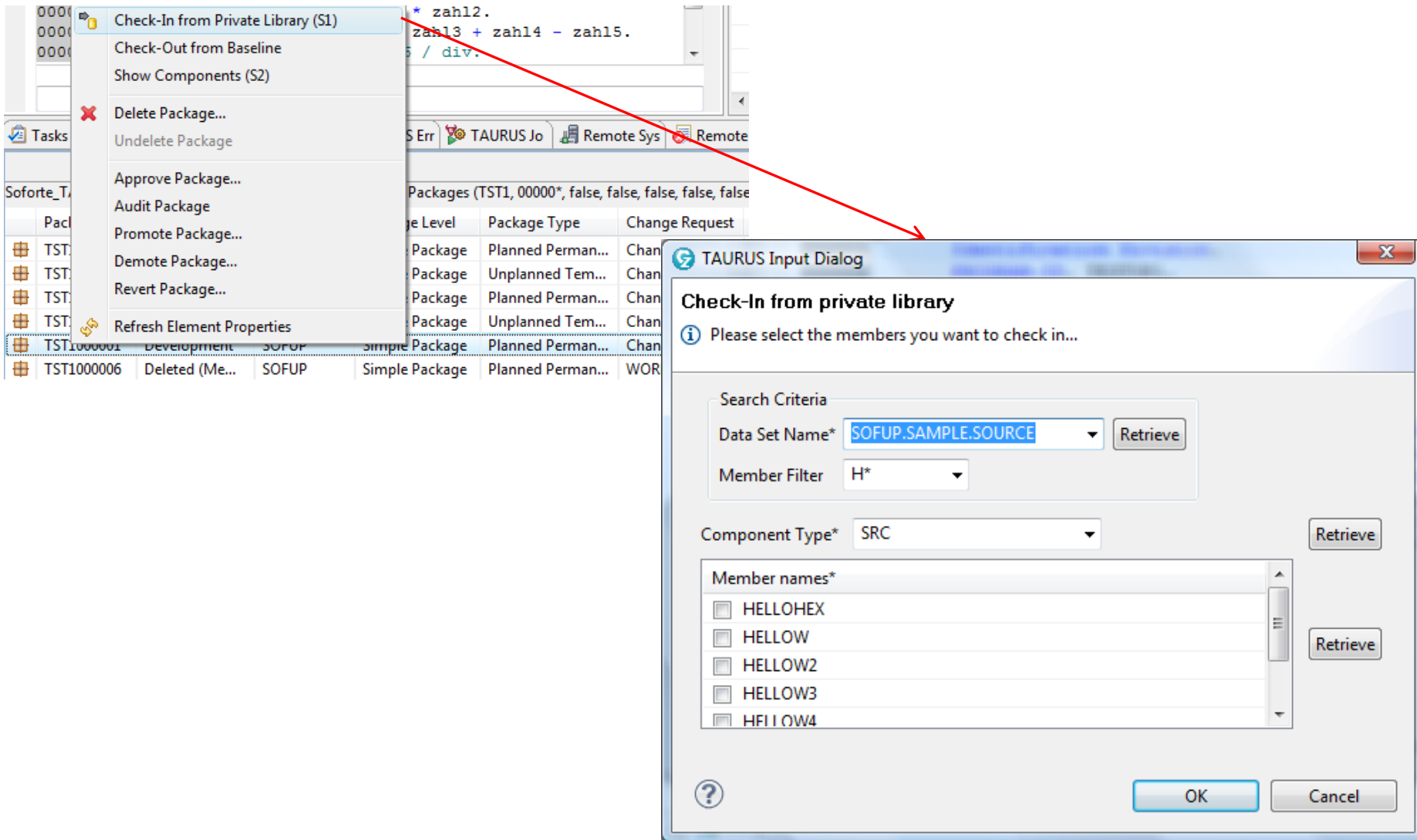
At the bottom, a table shows the component structure:

Package	Component	Type	Status
TST1000001	COB0006	SRC	Active
TST1000001	COB0005	SRC	Active
TST1000001	COB0004	SRC	Rejected
TST1000001	COB0003	SRC	Incomplete
TST1000001	COB0002	SRC	Inactive
TST1000001	COB0001	SRC	Active

Taurus – Check-out Components from Baseline



Taurus – Check-in Components from a PDS

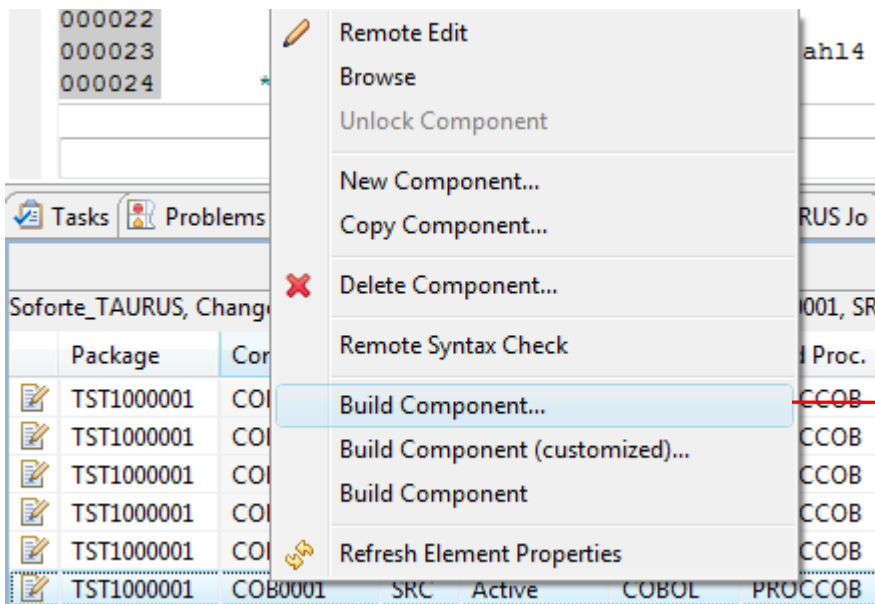


The screenshot shows the Taurus software interface. A context menu is open over a package, with the 'Check-In from Private Library (S1)' option selected. A red arrow points from this menu item to the 'TAURUS Input Dialog' window. The dialog box is titled 'Check-In from private library' and contains the following fields and controls:

- Search Criteria:**
 - Data Set Name*: SOFUP.SAMPLE.SOURCE (with a 'Retrieve' button)
 - Member Filter: H* (with a dropdown arrow)
- Component Type*:** SRC (with a dropdown arrow and a 'Retrieve' button)
- Member names*:** A list box containing:
 - HELLOHEX
 - HELLOW
 - HELLOW2
 - HELLOW3
 - HELLOW4(with a 'Retrieve' button)

At the bottom of the dialog are 'OK' and 'Cancel' buttons, and a help icon (?) on the left.

Taurus – Build a Component (standard Dialog)



000022
000023
000024

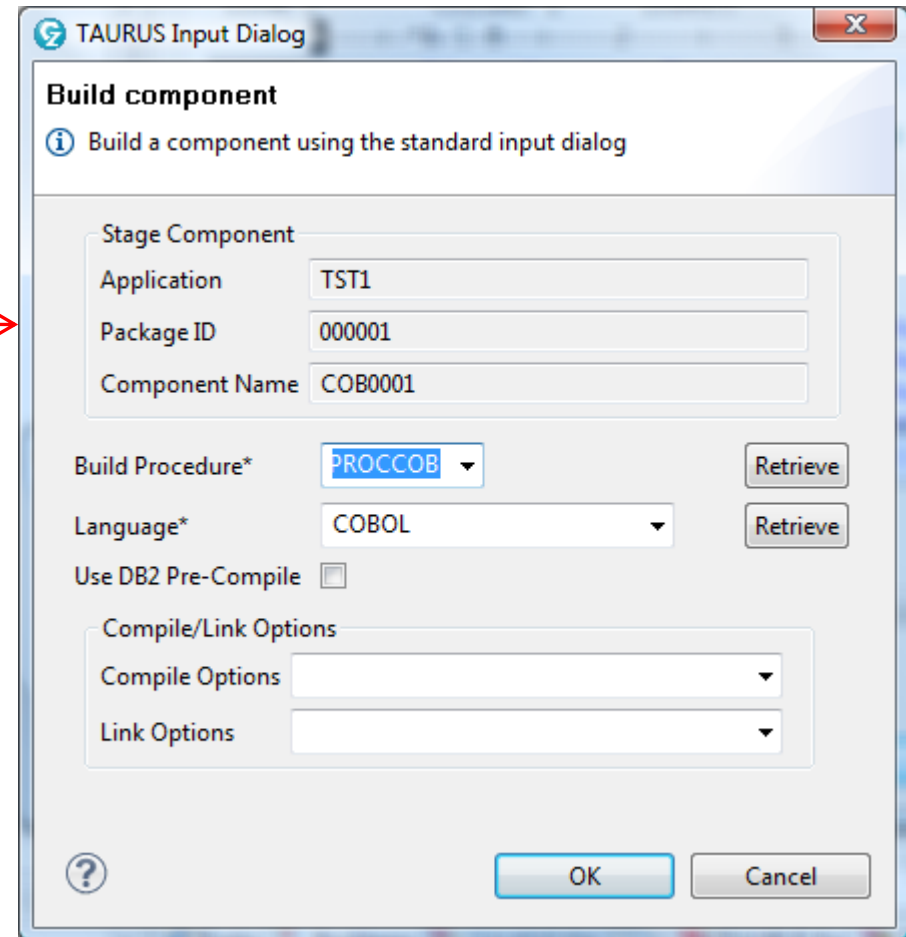
ah14

Tasks Problems

Soforte_TAURUS, Change

Package	Cor
TST1000001	CO
TST1000001	CO
TST1000001	CO
TST1000001	CO
TST1000001	CO
TST1000001	CO
TST1000001	COB0001 SRC Active COBOL PROCCOB

- Remote Edit
- Browse
- Unlock Component
- New Component...
- Copy Component...
- Delete Component...
- Remote Syntax Check
- Build Component...**
- Build Component (customized)...
- Build Component
- Refresh Element Properties



TAURUS Input Dialog

Build component

Build a component using the standard input dialog

Stage Component

Application TST1

Package ID 000001

Component Name COB0001

Build Procedure* PROCCOB Retrieve

Language* COBOL Retrieve

Use DB2 Pre-Compile

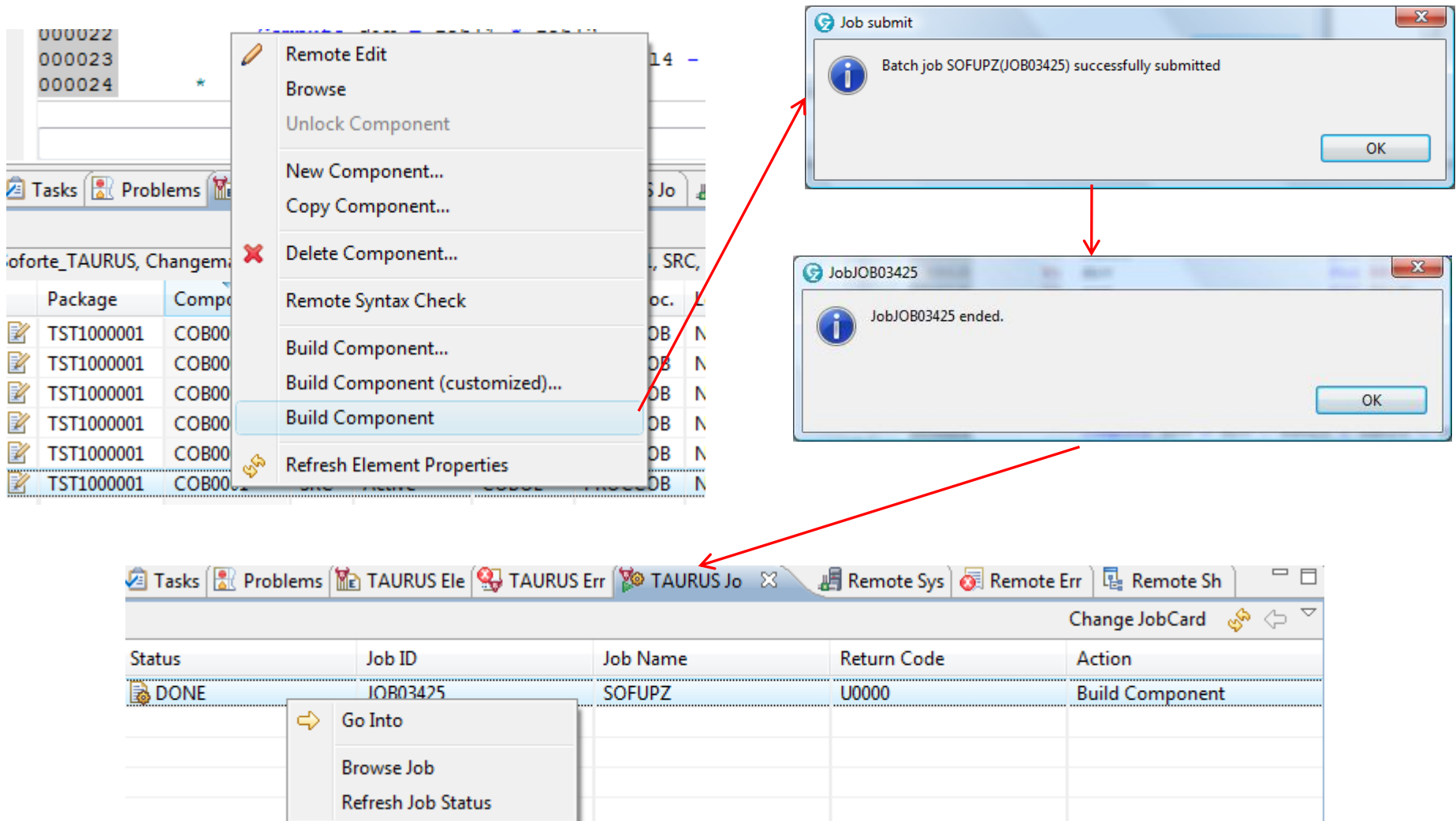
Compile/Link Options

Compile Options

Link Options

OK Cancel

Taurus – Build a Component (reuse options)

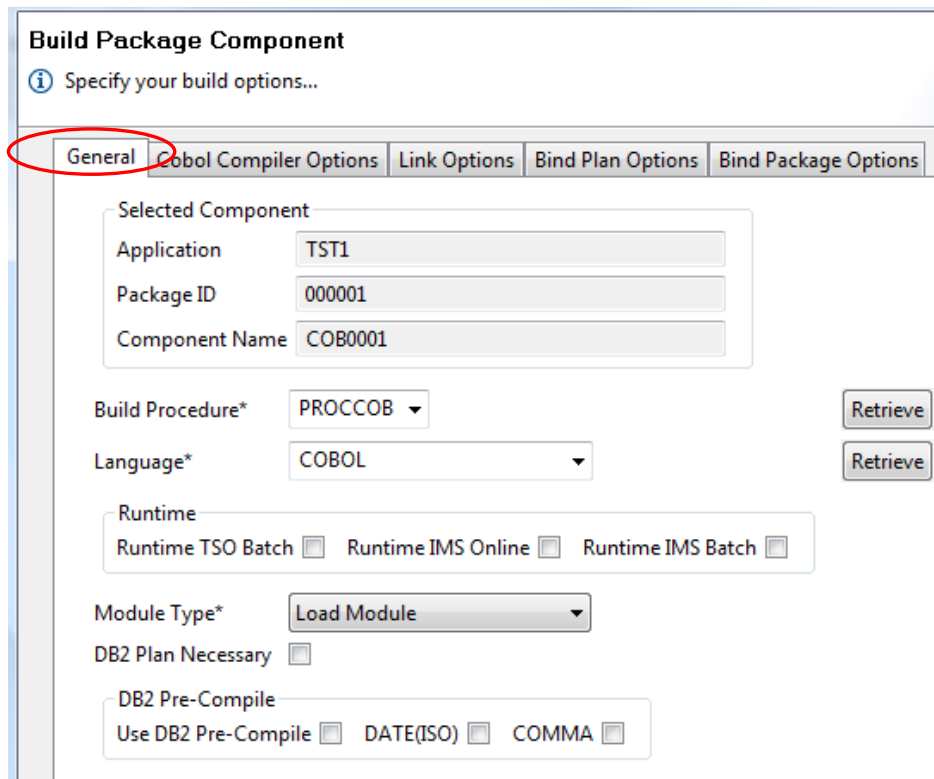


The screenshot illustrates the workflow for building a component in the SoforTe IDE. A context menu is open over a component list, with 'Build Component' selected. This triggers a 'Job submit' dialog box with the message 'Batch job SOFUPZ(JOB03425) successfully submitted'. A second dialog box, 'JobJOB03425', shows 'JobJOB03425 ended.' Below, the IDE's job status table shows the job as 'DONE' with a return code of 'U0000' and an action of 'Build Component'. A context menu is also visible over the 'Build Component' action in the table.

Status	Job ID	Job Name	Return Code	Action
DONE	JOB03425	SOFUPZ	U0000	Build Component

Taurus – Build a Component (customized)

- Defining your own Build Options Dialog without Plug-in programming effort (sample modeling result):



Build Package Component
Specify your build options...

General | Cobol Compiler Options | Link Options | Bind Plan Options | Bind Package Options

Selected Component

Application: TST1
Package ID: 000001
Component Name: COB0001

Build Procedure*: PROCCOB [Retrieve]
Language*: COBOL [Retrieve]

Runtime
Runtime TSO Batch Runtime IMS Online Runtime IMS Batch

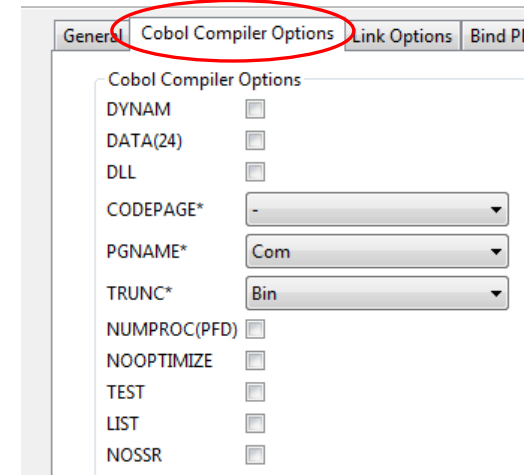
Module Type*: Load Module

DB2 Plan Necessary

DB2 Pre-Compile
Use DB2 Pre-Compile DATE(ISO) COMMA

Build Package Component

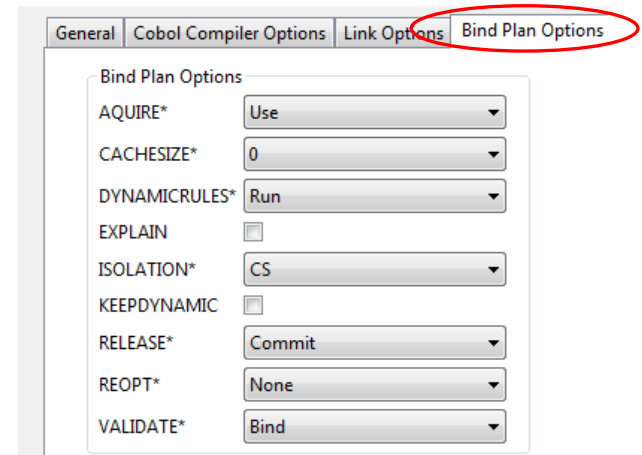
Specify your build options...



General | Cobol Compiler Options | Link Options | Bind Plan Options

Cobol Compiler Options

DYNAM
DATA(24)
DLL
CODEPAGE* -
PGMNAME* Com
TRUNC* Bin
NUMPROC(PFD)
NOOPTIMIZE
TEST
LIST
NOSSR

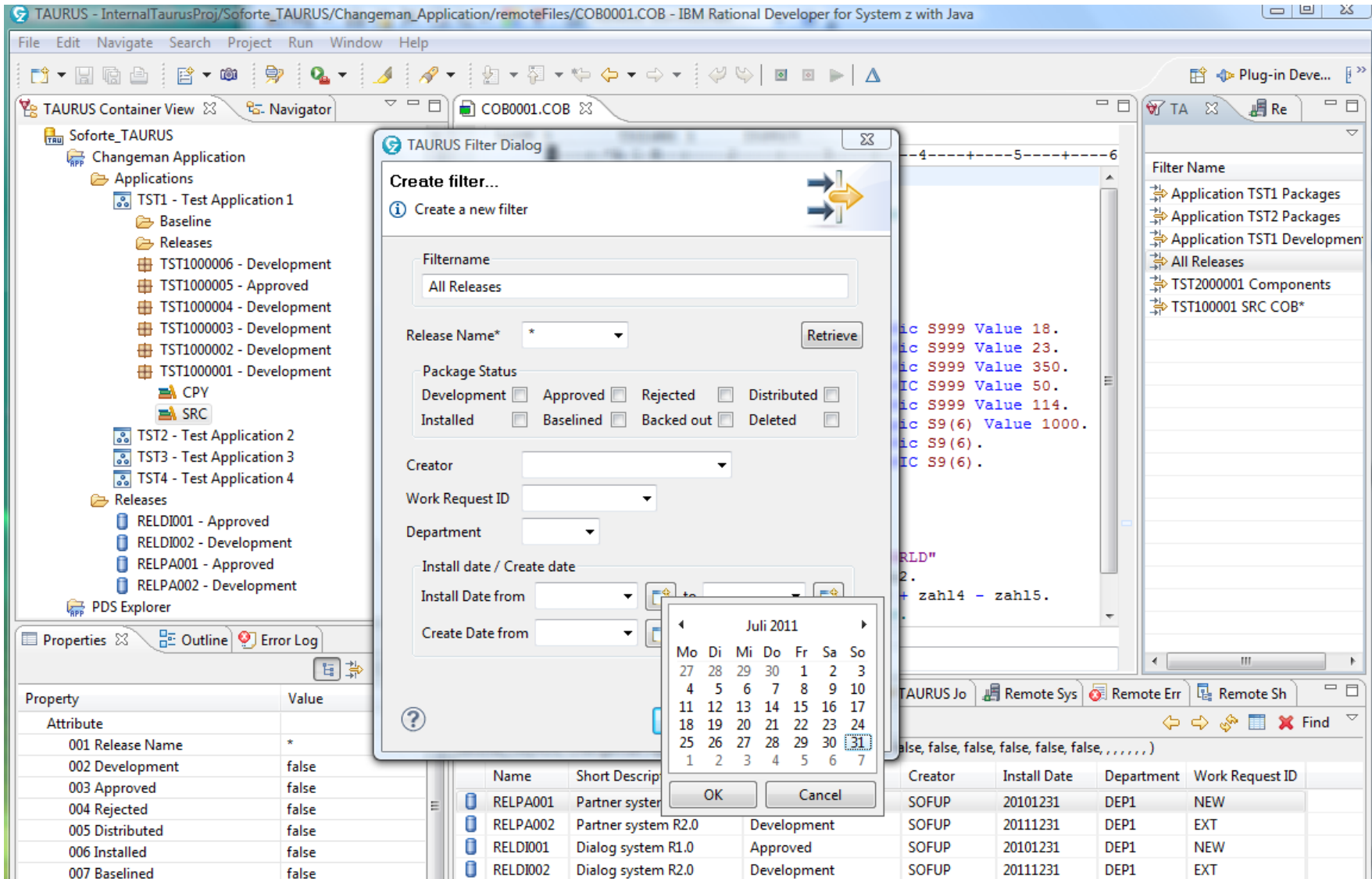


General | Cobol Compiler Options | Link Options | Bind Plan Options

Bind Plan Options

ACQUIRE* Use
CACHESIZE* 0
DYNAMICRULES* Run
EXPLAIN
ISOLATION* CS
KEEP DYNAMIC
RELEASE* Commit
REOPT* None
VALIDATE* Bind

Taurus – Release support (ERO Option only)



The screenshot displays the IBM Rational Developer for System z with Java interface. The main window shows the 'TAURUS Filter Dialog' with the following fields and options:

- Create filter...:** Create a new filter
- Filtername:** All Releases
- Release Name*:** * (dropdown)
- Package Status:** Development Approved Rejected Distributed Installed Baselined Backed out Deleted
- Creator:** (dropdown)
- Work Request ID:** (dropdown)
- Department:** (dropdown)
- Install date / Create date:** Install Date from (dropdown) Create Date from (dropdown)

A calendar for July 2011 is open, showing the date 31 selected.

The background shows the 'Soforte_Taurus' project structure in the Navigator, including 'Applications' (TST1-TST4) and 'Releases' (RELDI001-002, RELPA001-002). The Properties window shows the following values:

Property	Value
Attribute	
001 Release Name	*
002 Development	false
003 Approved	false
004 Rejected	false
005 Distributed	false
006 Installed	false
007 Baselined	false

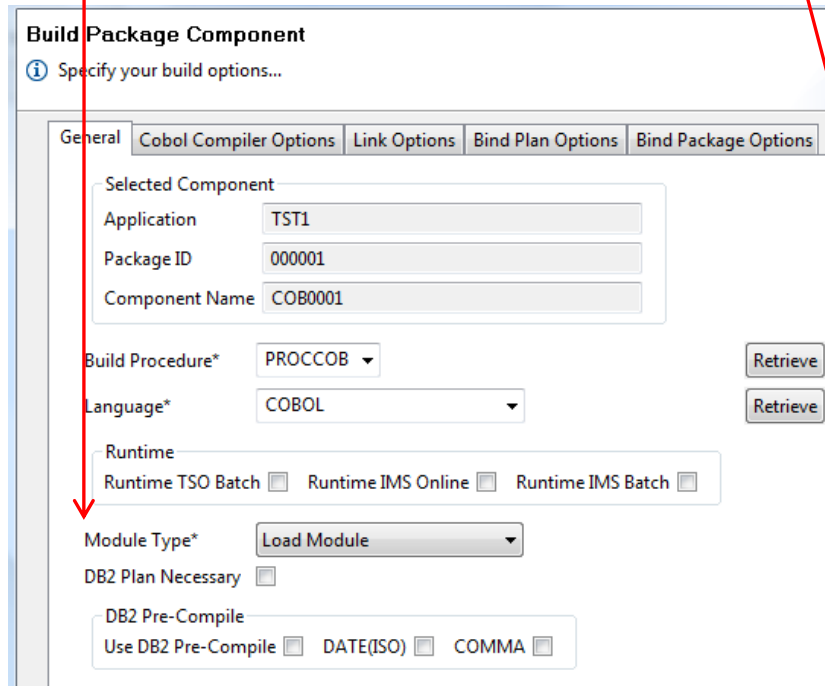
The bottom right shows a table of releases:

Name	Short Description	Creator	Install Date	Department	Work Request ID
RELPA001	Partner system	SOFUP	20101231	DEP1	NEW
RELPA002	Partner system R2.0	SOFUP	20111231	DEP1	EXT
RELDI001	Dialog system R1.0	SOFUP	20101231	DEP1	NEW
RELDI002	Dialog system R2.0	SOFUP	20111231	DEP1	EXT

Taurus – How to model the customized build

- **(1) Define a property for each dialog field**

Model
Result



Build Package Component

Specify your build options...

General | Cobol Compiler Options | Link Options | Bind Plan Options | Bind Package Options

Selected Component

Application: TST1

Package ID: 000001

Component Name: COB0001

Build Procedure*: PROCCOB

Language*: COBOL

Runtime

Runtime TSO Batch Runtime IMS Online Runtime IMS Batch

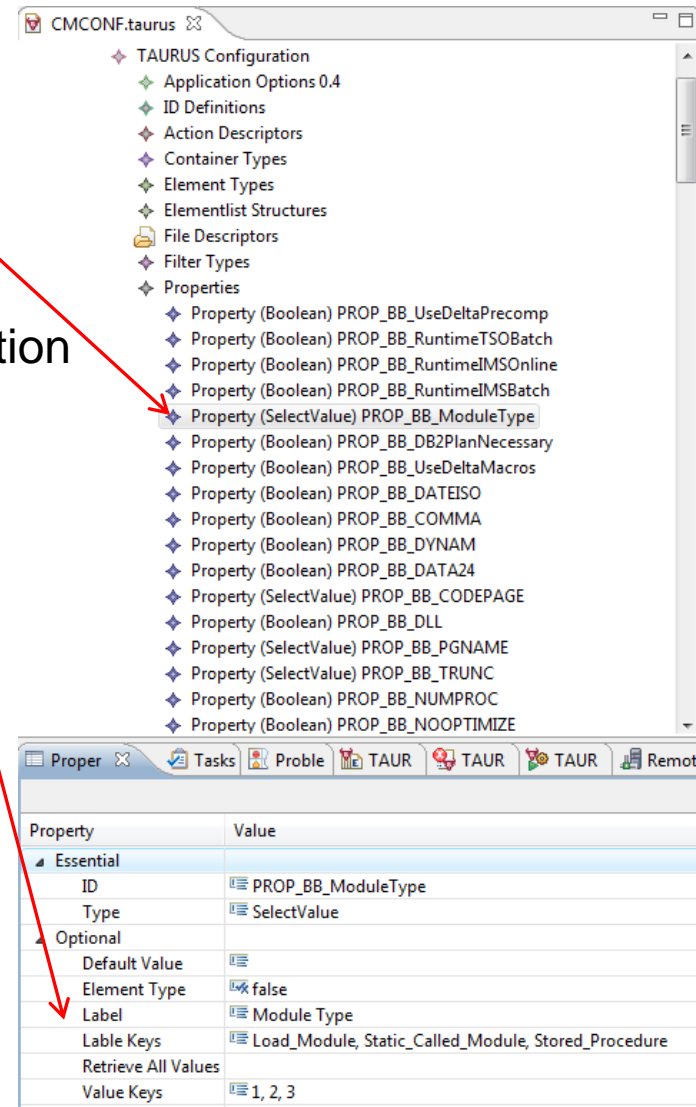
Module Type*: Load Module

DB2 Plan Necessary

DB2 Pre-Compile

Use DB2 Pre-Compile DATE(ISO) COMMA

Model
Definition



CMCONF.taurus

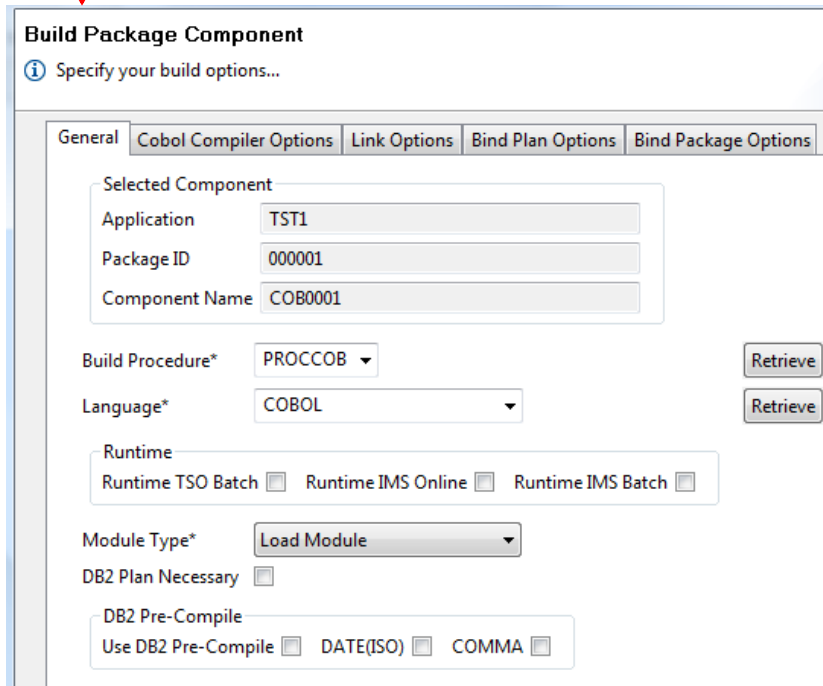
- TAURUS Configuration
 - Application Options 0.4
 - ID Definitions
 - Action Descriptors
 - Container Types
 - Element Types
 - Elementlist Structures
 - File Descriptors
 - Filter Types
 - Properties
 - Property (Boolean) PROP_BB_UseDeltaPrecomp
 - Property (Boolean) PROP_BB_RuntimeTSOBatch
 - Property (Boolean) PROP_BB_RuntimeIMSOnline
 - Property (Boolean) PROP_BB_RuntimeIMSBatch
 - Property (SelectValue) PROP_BB_ModuleType
 - Property (Boolean) PROP_BB_DB2PlanNecessary
 - Property (Boolean) PROP_BB_UseDeltaMacros
 - Property (Boolean) PROP_BB_DATEISO
 - Property (Boolean) PROP_BB_COMMA
 - Property (Boolean) PROP_BB_DYNAM
 - Property (Boolean) PROP_BB_DATA24
 - Property (SelectValue) PROP_BB_CODEPAGE
 - Property (Boolean) PROP_BB_DLL
 - Property (SelectValue) PROP_BB_PGNAME
 - Property (SelectValue) PROP_BB_TRUNC
 - Property (Boolean) PROP_BB_NUMPROC
 - Property (Boolean) PROP_BB_NOOPTIMIZE

Property	Value
Essential	
ID	PROP_BB_ModuleType
Type	SelectValue
Optional	
Default Value	
Element Type	false
Label	Module Type
Label Keys	Load_Module, Static_Called_Module, Stored_Procedure
Retrieve All Values	
Value Keys	1, 2, 3

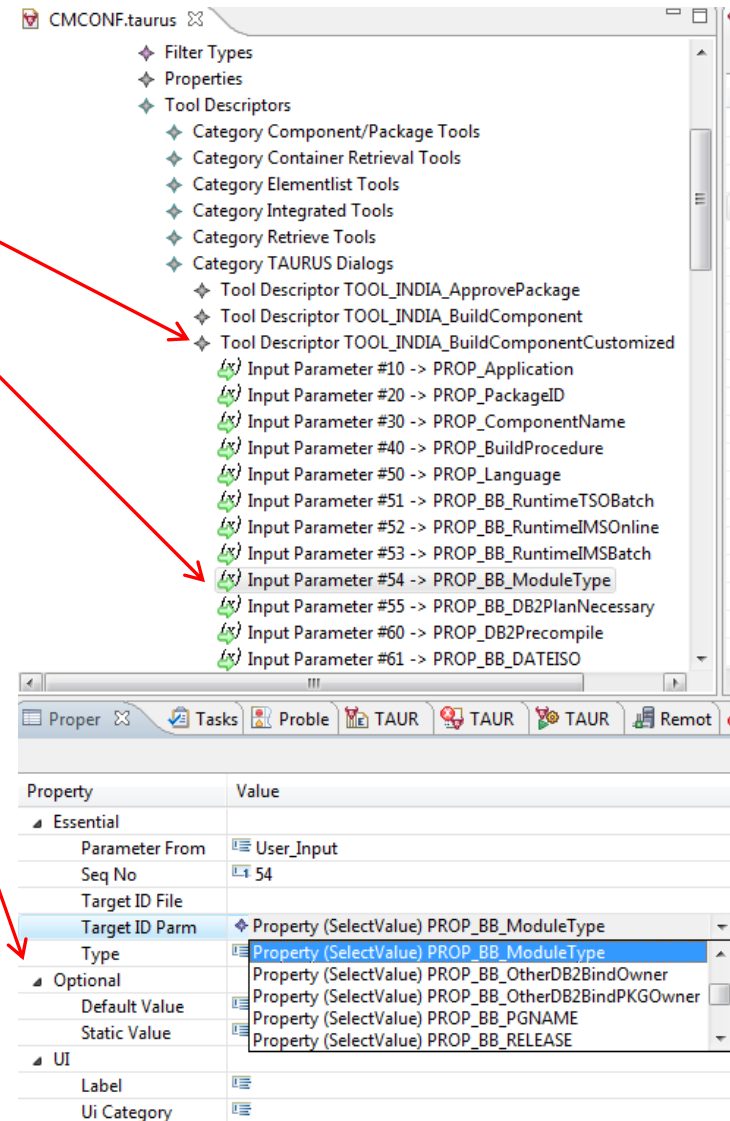
Taurus – How to model the customized build

- (2) Define the dialog and associate the dialog fields

Model
Result



Model
Definition

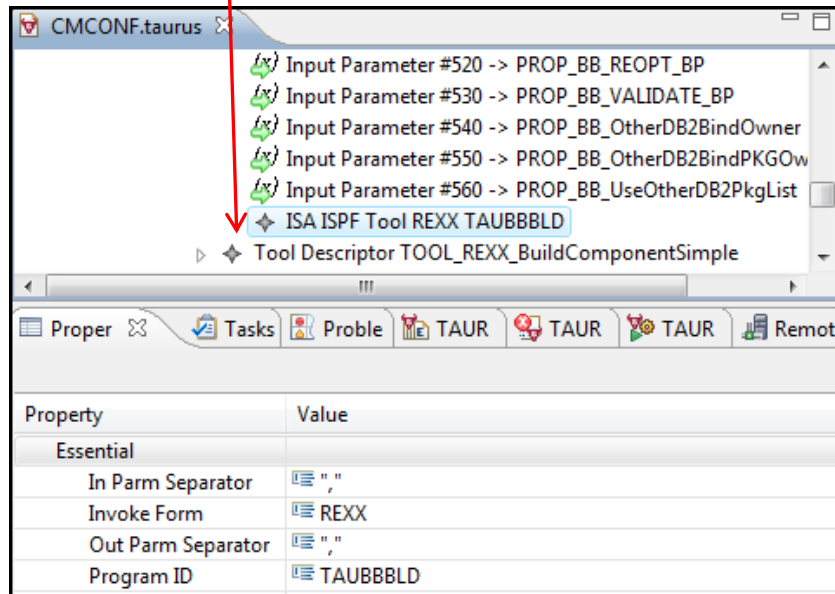


Property	Value
Essential	
Parameter From	User_Input
Seq No	54
Target ID File	
Target ID Parm	Property (SelectValue) PROP_BB_ModuleType
Type	Property (SelectValue) PROP_BB_ModuleType
Optional	
Default Value	Property (SelectValue) PROP_BB_OtherDB2BindOwner
Static Value	Property (SelectValue) PROP_BB_PGNAME
UI	
Label	
Ui Category	

Taurus – How to model the customized build

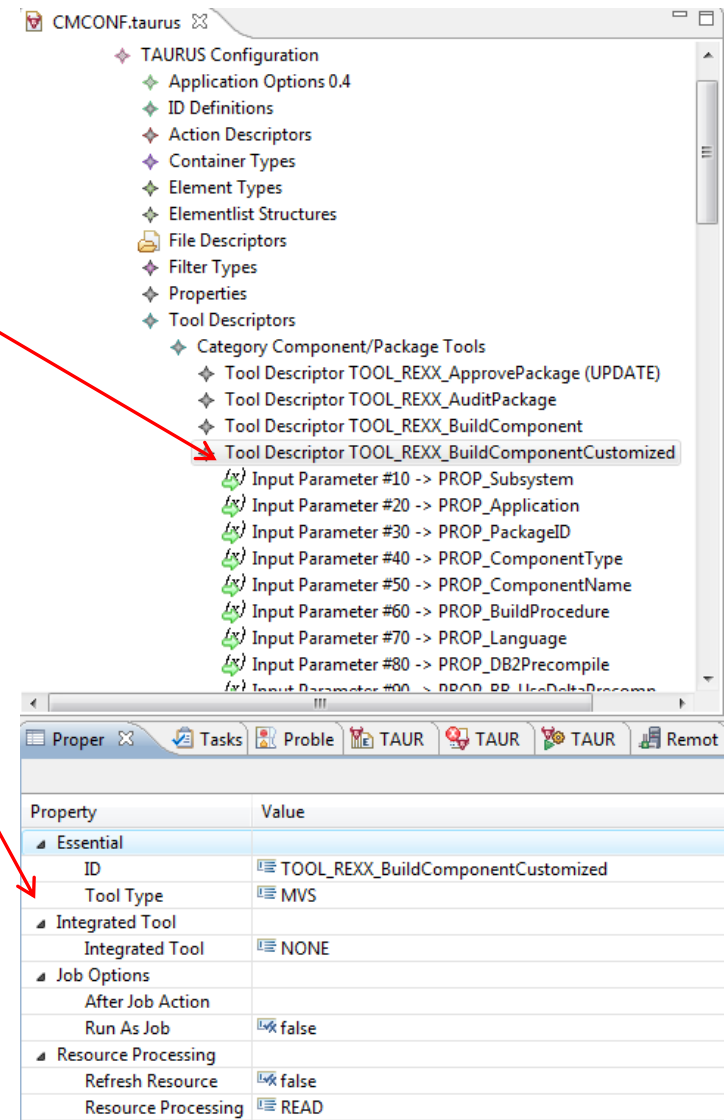
- (3) Define the REXX tool which calls the ChangeMan Build and associate the dialog fields as input parameters.

Model
Definition



Input Parameter #520 -> PROP_BB_REOPT_BP
Input Parameter #530 -> PROP_BB_VALIDATE_BP
Input Parameter #540 -> PROP_BB_OtherDB2BindOwner
Input Parameter #550 -> PROP_BB_OtherDB2BindPKGOW
Input Parameter #560 -> PROP_BB_UseOtherDB2PkgList
ISA ISPF Tool REXX TAUBBLD
Tool Descriptor TOOL_REXX_BuildComponentSimple

Property	Value
Essential	
In Parm Separator	" "
Invoke Form	REXX
Out Parm Separator	" "
Program ID	TAUBBLD



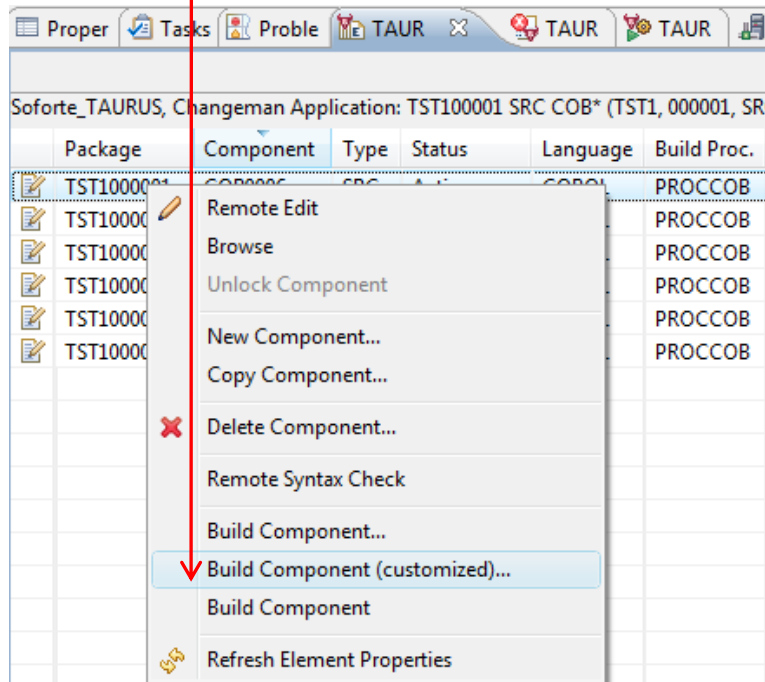
TAURUS Configuration
Application Options 0.4
ID Definitions
Action Descriptors
Container Types
Element Types
Elementlist Structures
File Descriptors
Filter Types
Properties
Tool Descriptors
Category Component/Package Tools
Tool Descriptor TOOL_REXX_ApprovePackage (UPDATE)
Tool Descriptor TOOL_REXX_AuditPackage
Tool Descriptor TOOL_REXX_BuildComponent
Tool Descriptor TOOL_REXX_BuildComponentCustomized
Input Parameter #10 -> PROP_Subsystem
Input Parameter #20 -> PROP_Application
Input Parameter #30 -> PROP_PackageID
Input Parameter #40 -> PROP_ComponentType
Input Parameter #50 -> PROP_ComponentName
Input Parameter #60 -> PROP_BuildProcedure
Input Parameter #70 -> PROP_Language
Input Parameter #80 -> PROP_DB2Precompile
Input Parameter #90 -> PROP_BB_UseDeltaPrecomp

Property	Value
Essential	
ID	TOOL_REXX_BuildComponentCustomized
Tool Type	MVS
Integrated Tool	
Integrated Tool	NONE
Job Options	
After Job Action	
Run As Job	false
Resource Processing	
Refresh Resource	false
Resource Processing	READ

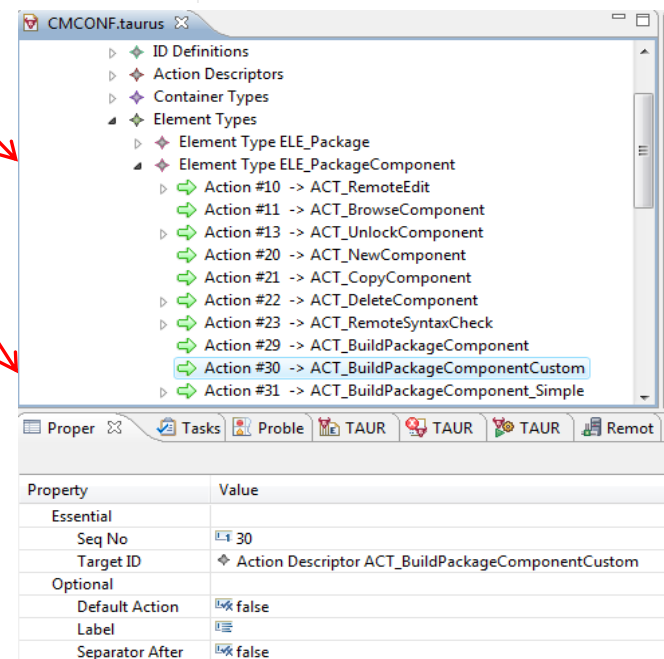
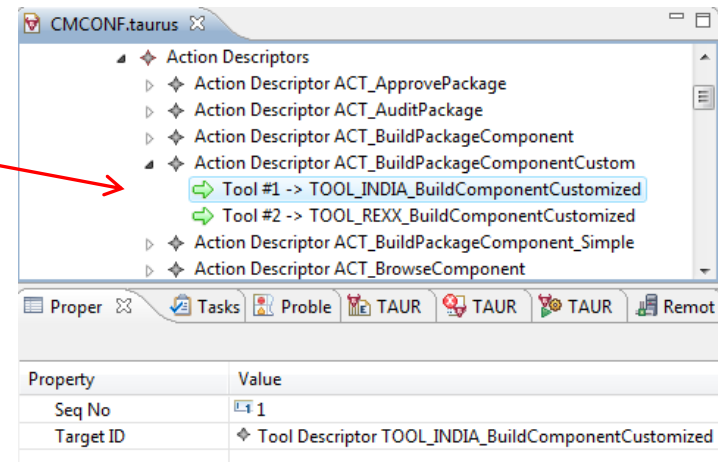
Taurus – How to model the customized build

- (4) Define an action and associate the input dialog and the REXX tool
- (5) Associate the action to the ChangeMan package component type

Model Result

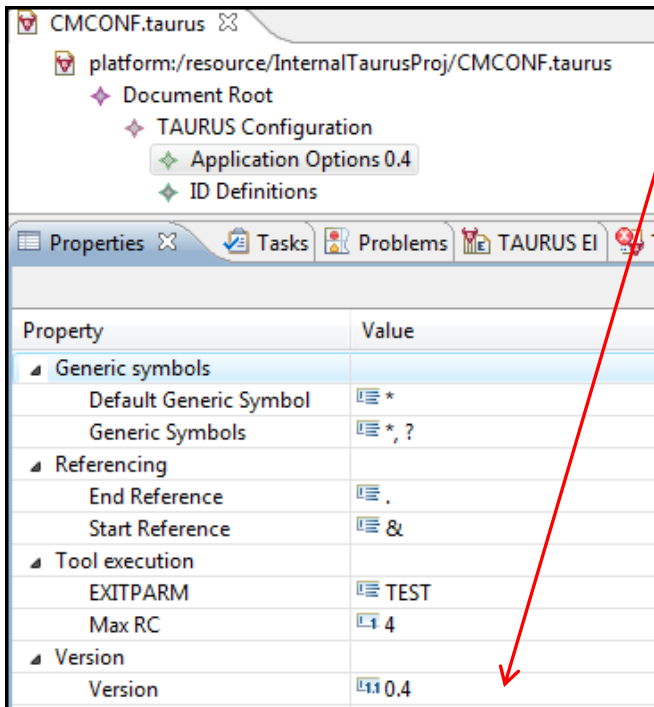


Model Definition



Taurus – How to model the customized build

- (6) validate your model (consistency check)
- (7) Store the model in a PDS or USS and test your customization
- (8) Promote the new model to all client users only by changing the version number (no client installation process necessary)

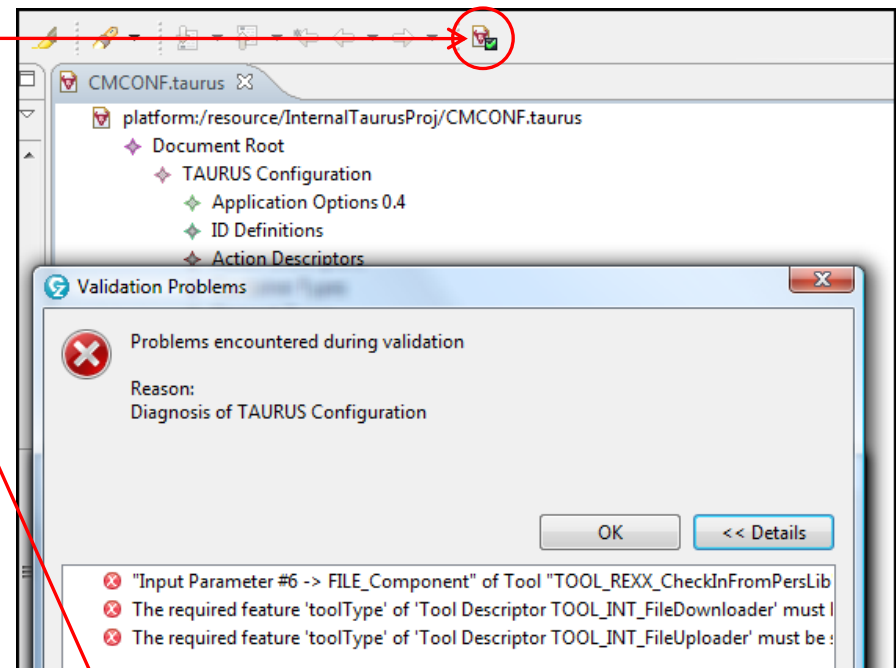


CMCONF.taurus

- platform:/resource/InternalTaurusProj/CMCONF.taurus
 - Document Root
 - TAURUS Configuration
 - Application Options 0.4
 - ID Definitions

Properties

Property	Value
Generic symbols	
Default Generic Symbol	*
Generic Symbols	*, ?
Referencing	
End Reference	.
Start Reference	&
Tool execution	
EXITPARM	TEST
Max RC	4
Version	
Version	0.4



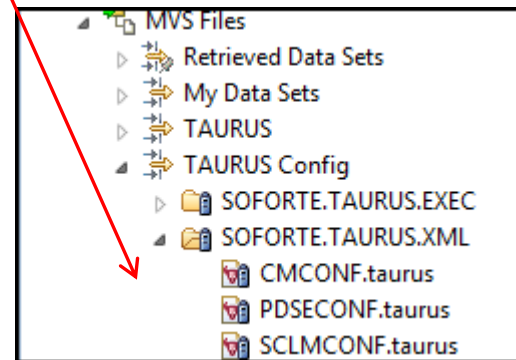
Validation Problems

Problems encountered during validation

Reason:
Diagnosis of TAURUS Configuration

OK << Details

- ✗ "Input Parameter #6 -> FILE_Component" of Tool "TOOL_REXX_CheckInFromPersLib"
- ✗ The required feature 'toolType' of 'Tool Descriptor TOOL_INT_FileDownloader' must be
- ✗ The required feature 'toolType' of 'Tool Descriptor TOOL_INT_FileUploader' must be



MVS Files

- Retrieved Data Sets
- My Data Sets
- TAURUS
 - TAURUS Config
 - SOFORTE.TAURUS.EXEC
 - SOFORTE.TAURUS.XML
 - CMCONF.taurus
 - PDSECONF.taurus
 - SCLMCONF.taurus

- **Reduces hurdle for RDz implementation**
 - Plug-in development becomes obsolete
 - Existing skill is sufficient – no plug-in development skill required
 - Supports SCLM, Endeavor, ChangeMan
 - The financial hurdle for RDz implementations is reduced significantly
- **Reduces risk for RDz Implementation**
 - Complete and deep integration becomes affordable
 - Predefined models for different development processes
 - Integration with all kinds of tools like RAA, RTC, ...
 - High functionality and quality of UI increase developer acceptability and productivity
- **It's a product**
 - It's a product, not a service
 - Existing shortage in implementation skills is not a bottleneck anymore