



Teamcenter Business Modeler IDE

Bob Donovan
Development Manager BMIDE
UGS



Agenda



- ▶ Vision & Strategy
- ▶ What's New In the Unified Platform
- ▶ Development Process Support
- ▶ Features
- ▶ Future Deliverables
- ▶ Questions



Agenda



- ▶ **Vision & Strategy**
- ▶ What's New
- ▶ Development Process Support
- ▶ Features
- ▶ Future Deliverables
- ▶ Questions



Teamcenter Business Modeler IDE Unification



- ▶ Best in class product tailoring environment
- ▶ BMIDE unifies all aspects of both codeless extensions and programming extensions in a coherent manner
- ▶ Business analyst friendly
 - ▶ UML based interface
 - ▶ Focus on usability
- ▶ Supports codeless and programming extensions in a single environment
 - ▶ Emphasis on codeless extension
 - ▶ Enforce best practices when programming extensions become essential

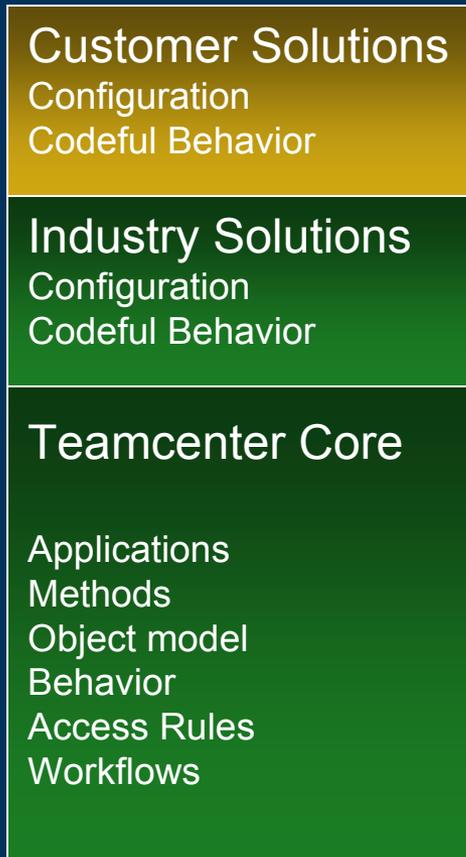
The image displays three overlapping screenshots of the Teamcenter Business Modeler IDE. The top screenshot shows a Java code editor with a class definition for 'Class1'. The middle screenshot shows the 'Naming Rule 1' configuration dialog, where users can define naming conventions for business objects, including project, name, initial value, maximum value, and patterns. The bottom screenshot shows a UML class diagram with classes like 'WorkspaceObject', 'Item', 'ItemExtension', and 'ClassExt', along with a table listing properties for these classes.



Driving Extension Configuration



Existing Development Paradigm



Driving codeful behavior into codeless BMIDE configuration

Evolving Configuration Paradigm

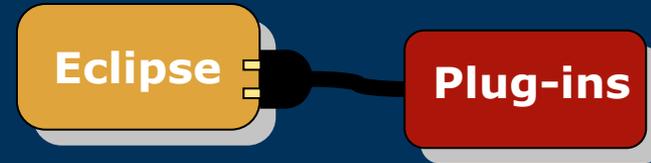




BMIDE Framework



- ▶ UGS is focusing on building products on open standards
- ▶ BMIDE is built on Eclipse Rich Client Platform
 - ▶ Plug-ins that extend framework
 - ▶ Eclipse RCP provides clear interface for building extensions
 - ▶ Plug-ins are upgradeable and interoperable
- ▶ Leveraging integrated tool set
 - ▶ Source Control Management: ClearCase, Perforce, CVS, Subversion
 - ▶ C/C++ and Java Editors, Debuggers
- ▶ Open framework
 - ▶ Tools can be extended by customers to create additional tools
- ▶ BMIDE operations in both
 - ▶ RAC Applications
 - ▶ Standalone BMIDE Client





Agenda



- ▶ Vision & Strategy
- ▶ **What's New**
- ▶ Development Process Support
- ▶ Features
- ▶ Future Deliverables
- ▶ Questions



Unified Platform - Teamcenter 2007

- ▶ BMIDE
- ▶ RAC Application
- ▶ Standalone BMIDE
- ▶ Business Objects
- ▶ UML
- ▶ Rules Engine Framework
- ▶ Templates
- ▶ Concurrent Development
- ▶ File Repository
- ▶ CMM Repository
- ▶ Test and Deploy
- ▶ Deploy Dry Run



Agenda



- ▶ Vision & Strategy
- ▶ What's New
- ▶ Development Process Support
- ▶ Features
- ▶ Future Deliverables
- ▶ Questions

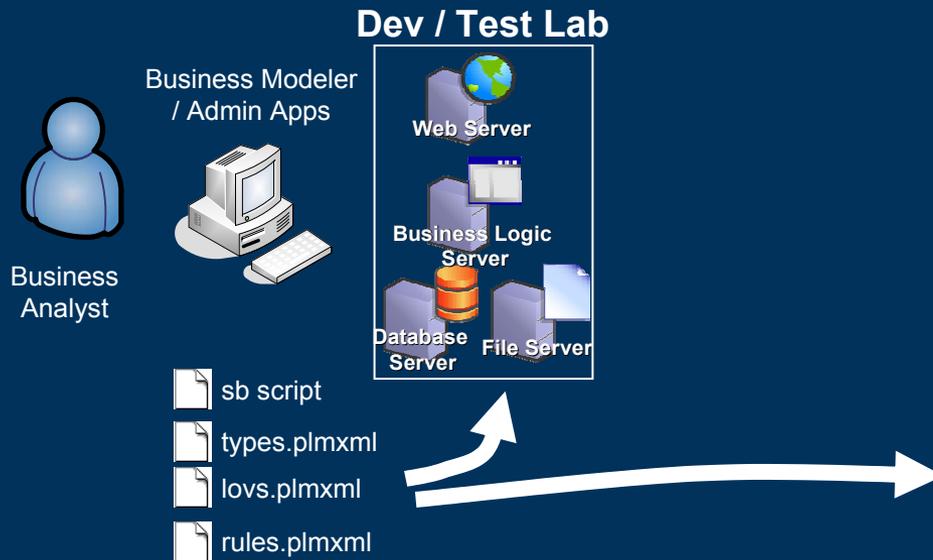


Business Extension Development Cycle





Current Process



Installation 1



Admin

Installation 2



Admin

Installation 3



Admin

- ▶ Business analyst create “sb” scripts and plmxml files to transfer definitions to each site.
- ▶ Each site admin takes the “sb” script and runs the script to update the schema
- ▶ “sb” utility and script file are crucial to synchronizing the schema for all sites



Templates



- ▶ New for Teamcenter 2007
- ▶ Container of business extensions, best practices, schema, LOVs
- ▶ All Teamcenter COTS schema, types, LOVS, etc will be contained in a template
- ▶ Templates can be extended, example: Industry Solutions
- ▶ Customers can create their own templates that extend Teamcenter templates
- ▶ Template dependencies can be declared so that a template can not be installed unless dependent templates are present in the system.



Templates



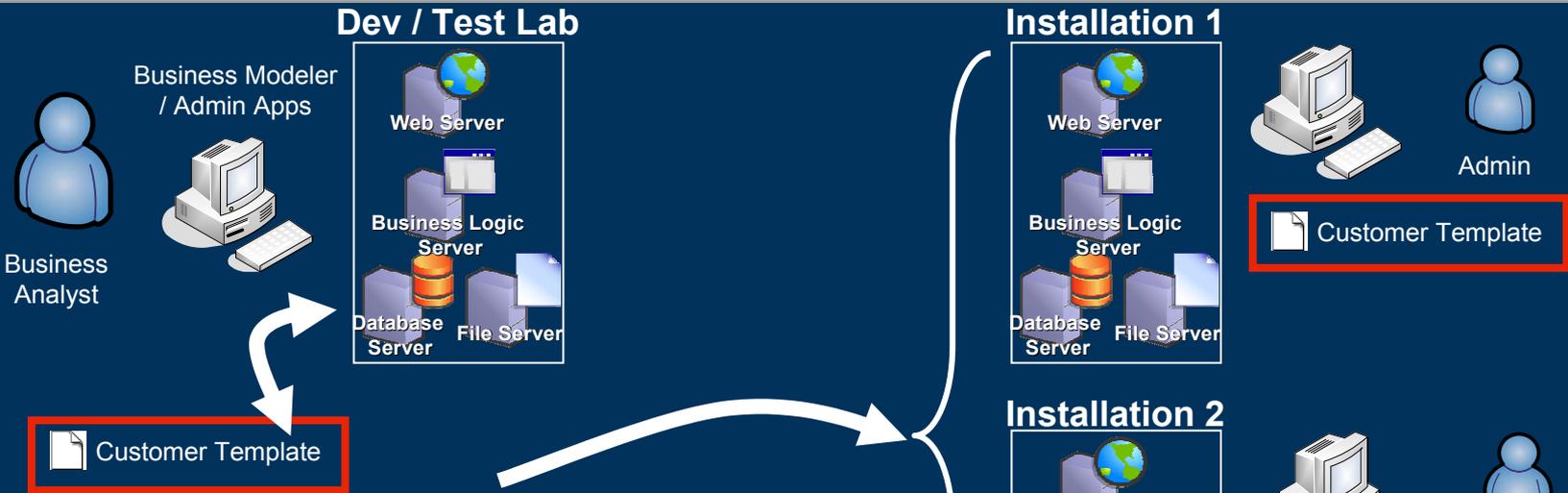
- ▶ The BMIDE manages all extensions through a template
- ▶ Business Analysts create a Extension Environment Project in the BMIDE that manages a customer template
- ▶ As Types, Classes, Attributes, LOVs, etc are created in the BMIDE that are automatically placed into the template
- ▶ Analyst can track COTS versus custom extensions
- ▶ The template becomes the mechanism for deploying custom extensions to any Teamcenter environment
- ▶ Customer extensions will be moved to template



- ▶ New Customers
 - ▶ Customers desiring to tailor Teamcenter business behavior can start with a new clean template
- ▶ Existing Customers
 - ▶ Must move custom extensions into a template
 - ▶ The BMIDE will provide tools to separate existing customer extensions from Teamcenter and move them into a template



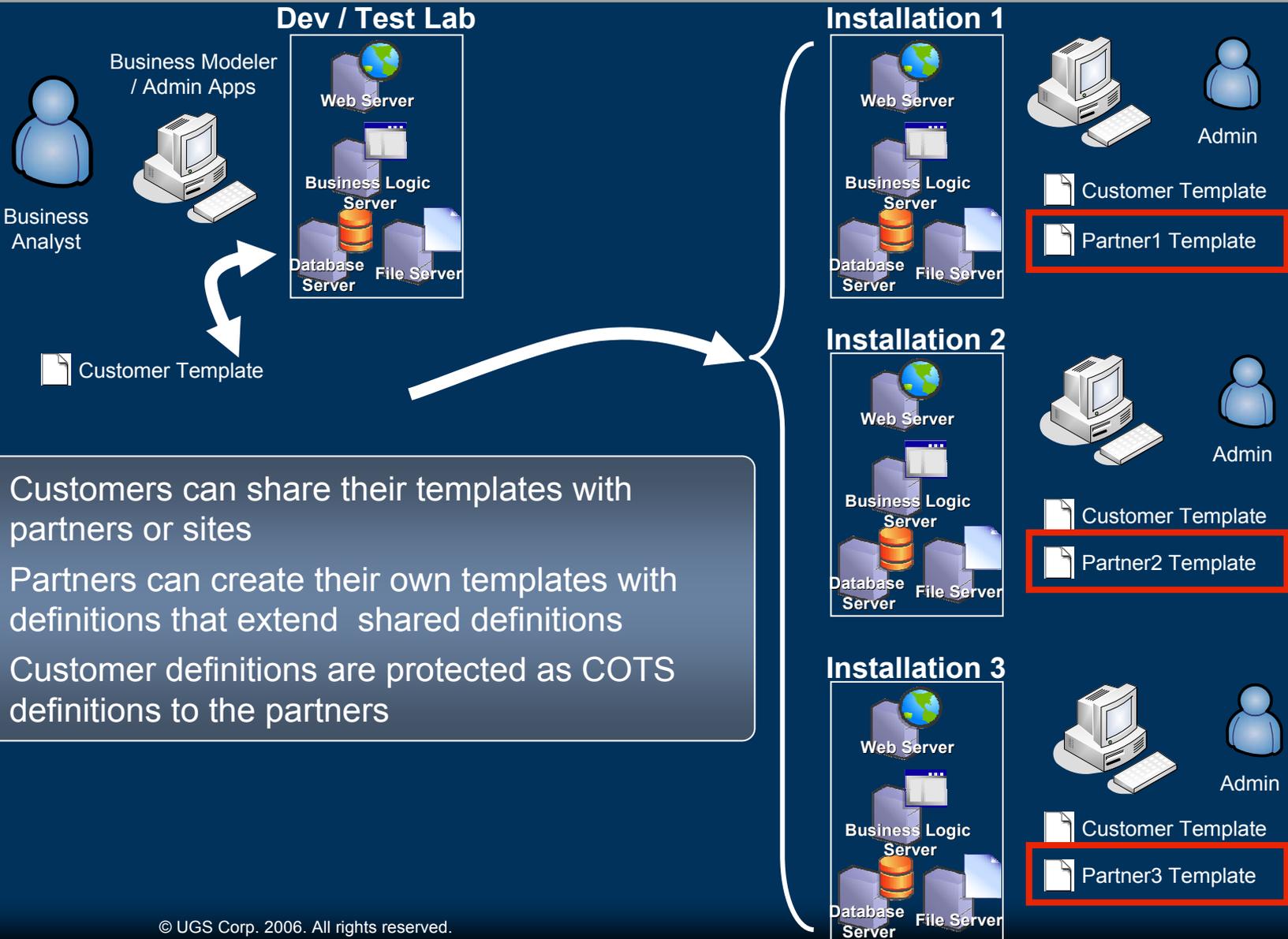
BMIDE Templates & Deploy



- ▶ In Teamcenter 2007, the customer template replaces all the individual scripts and XML files.
- ▶ The template is deployed as a whole unit to any Teamcenter server
- ▶ Each Teamcenter server will synchronize itself with the contents of the template



Template Sharing With Partners



- ▶ Customers can share their templates with partners or sites
- ▶ Partners can create their own templates with definitions that extend shared definitions
- ▶ Customer definitions are protected as COTS definitions to the partners



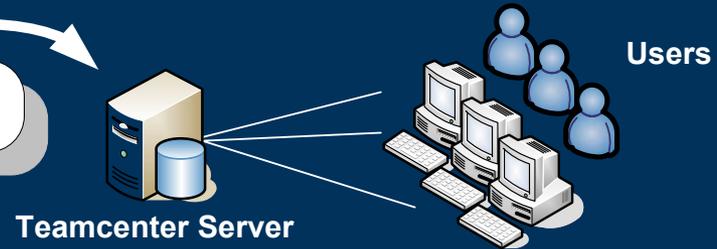
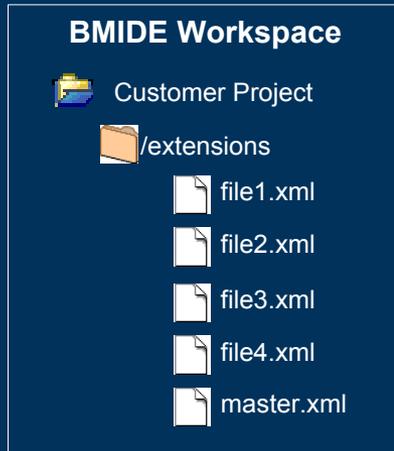
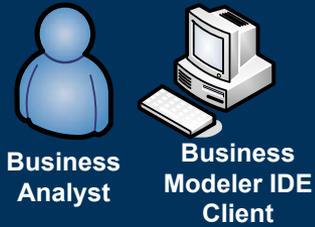
Better Extension Management



- ▶ Teamcenter Engineering Business Modeler
 - ▶ Supports single Business Analyst
 - ▶ Sharing extensions is difficult
 - ▶ Difficult to track which analyst has the master copy
- ▶ Teamcenter 2007 - BMIDE
 - ▶ Manages the master definition of the data model and extensions in a Repository
 - ▶ Each Business Analyst connects to the Repository to share and synchronize definitions
 - ▶ Two Options
 - ▶ File Repository
 - ▶ Teamcenter Repository

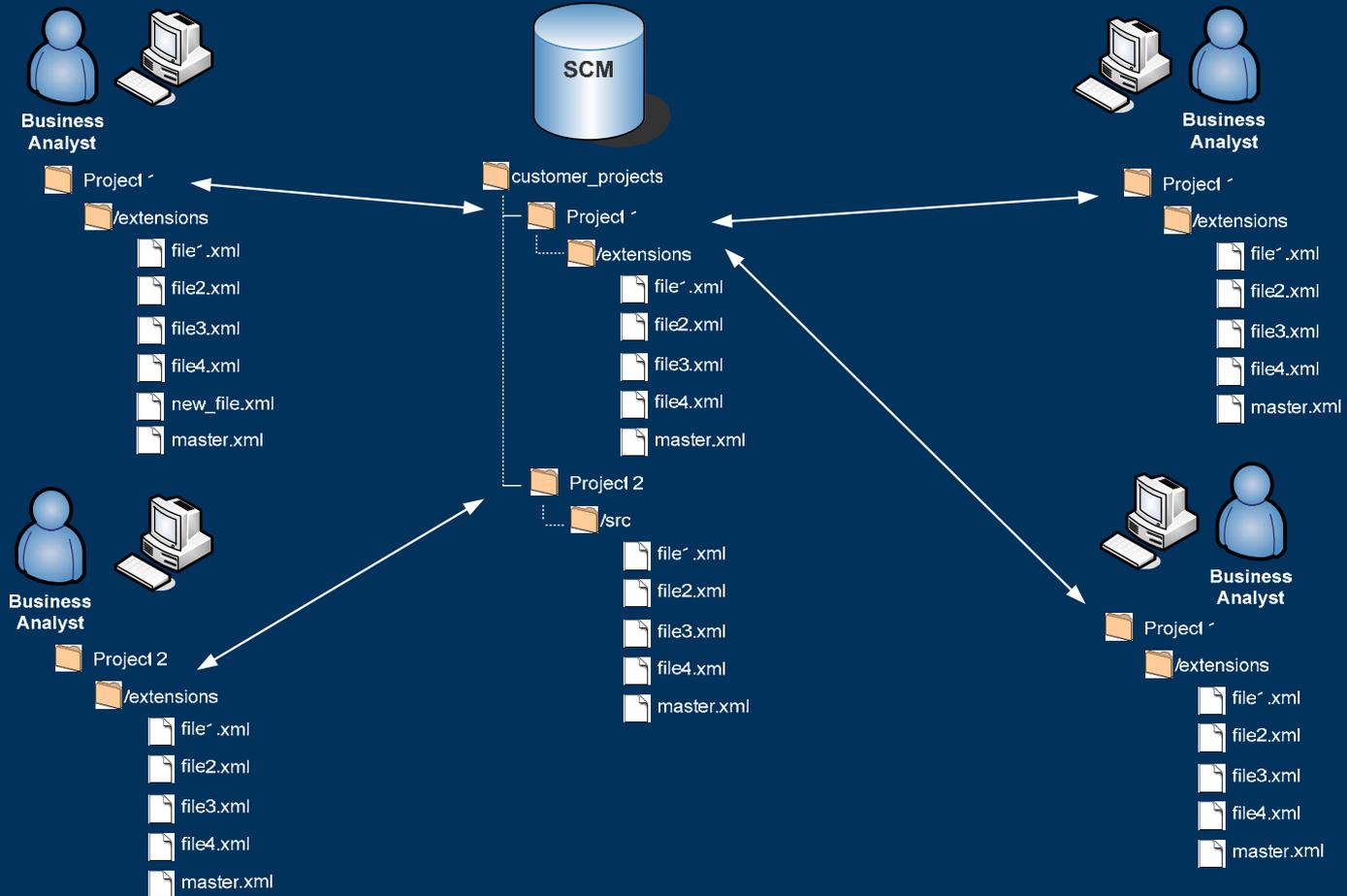


File Repository



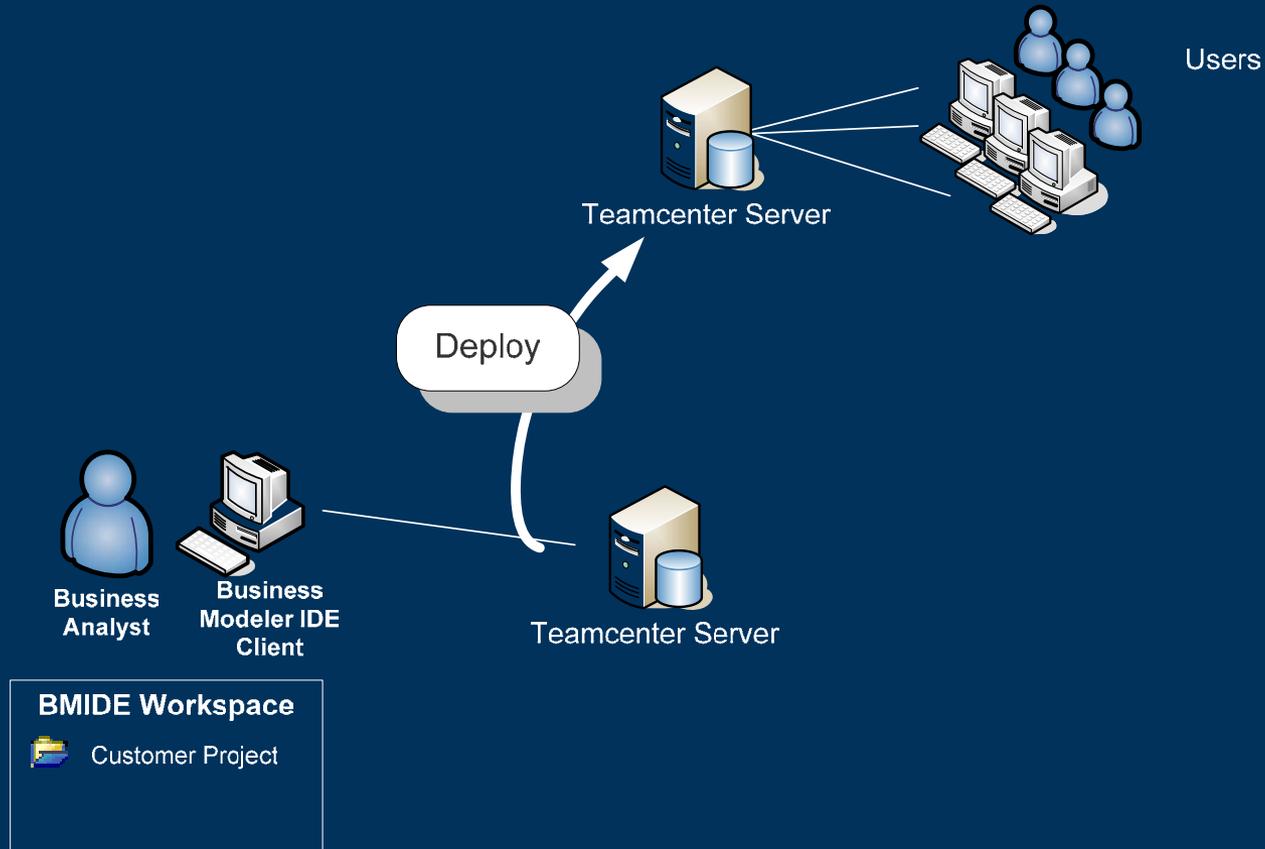


File Repository: Concurrent Development



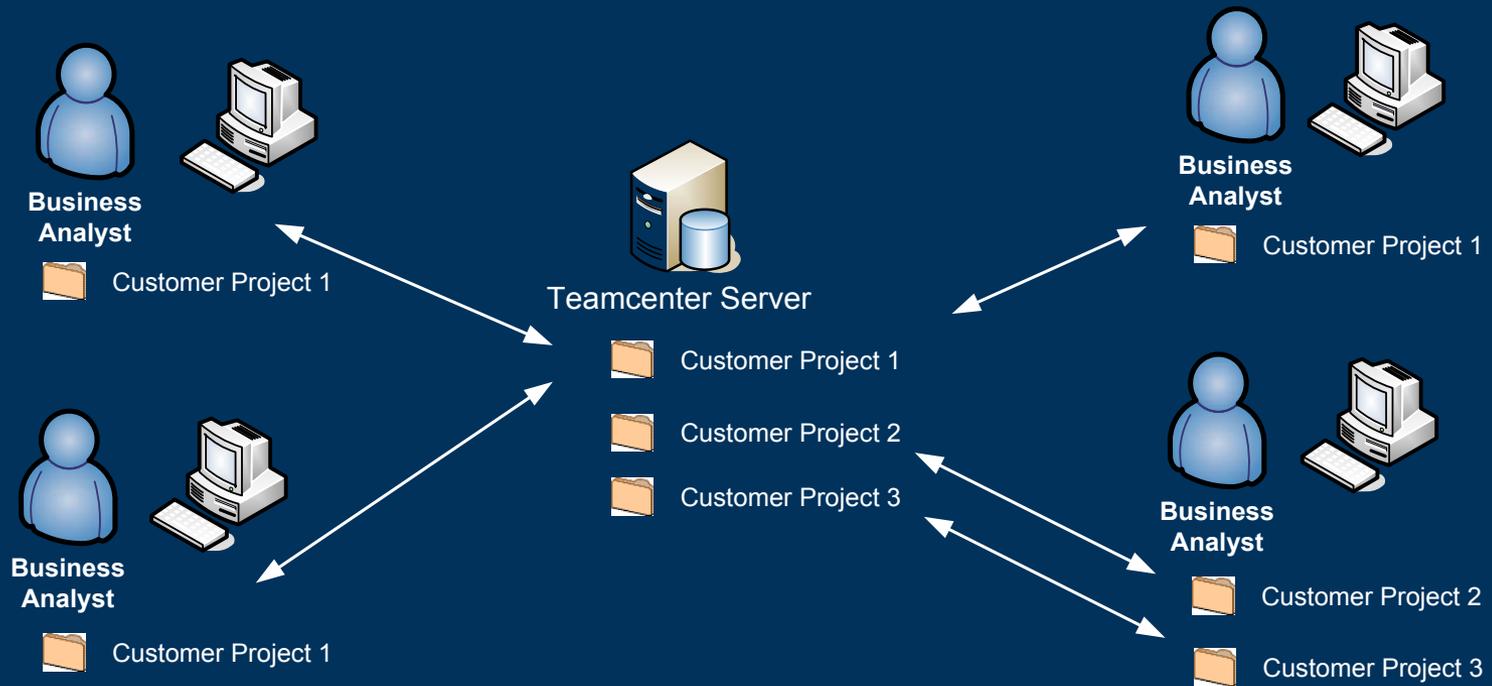


Teamcenter Repository





Teamcenter Repository: Concurrent Development





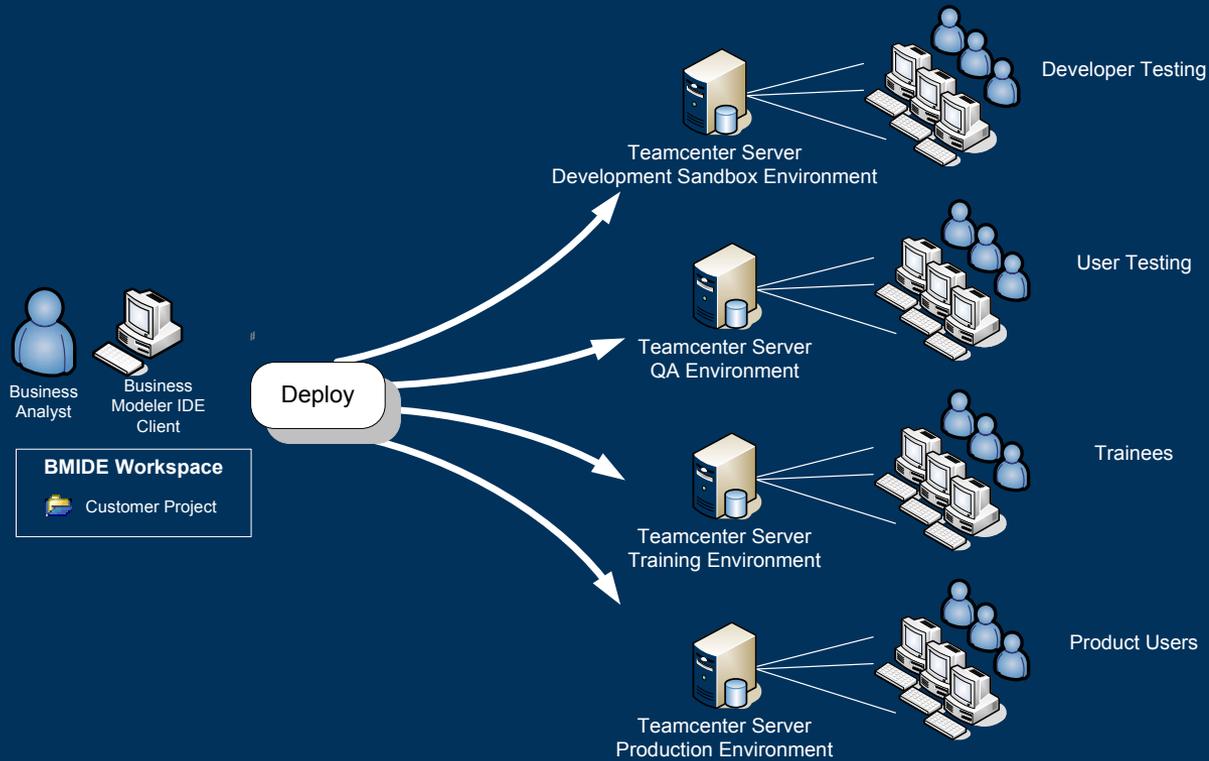
File Repository vs. Teamcenter Repository



- ▶ File Repository
- ▶ Extensions stored in XML Files
- ▶ Version management at the file level
- ▶ Integrate with SCM tool
- ▶ Teamcenter Repository
- ▶ Extensions stored in Teamcenter
- ▶ Version management at the object level
- ▶ Teamcenter capabilities are leveraged for extension management
 - ▶ Version Management of objects
 - ▶ Workflow and Change Management Support
 - ▶ Baseline



Better Deployment Tools

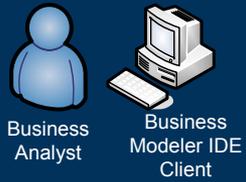




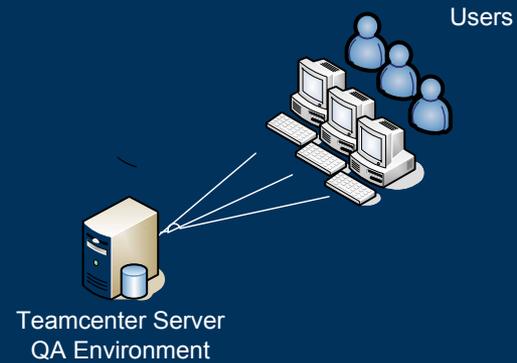
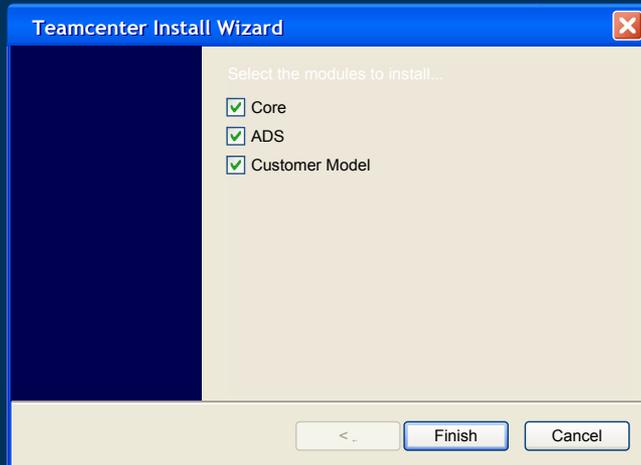
- ▶ Deploy uses web services framework
- ▶ Requires DBA access, user name and password
- ▶ Dry run support
 - ▶ Run the deploy in dry run mode to validate template against each installation
 - ▶ Returns a report of elements that will synchronize successfully and unsuccessfully
 - ▶ Can be used to validate how a template will deploy to various sites



Deploy to Production Server



Generate template for Production Install





Agenda



- ▶ Vision & Strategy
- ▶ What's New
- ▶ Development Processes Support
- ▶ **Features**
- ▶ Future Deliverables
- ▶ Questions



BMIDE Features



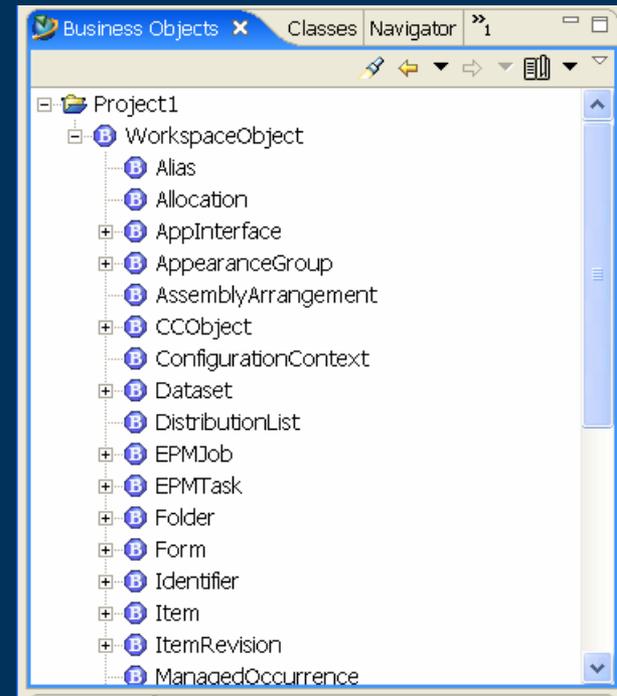
- ▶ Business Objects
 - ▶ Attribute Properties
 - ▶ Reference Properties
 - ▶ Runtime Properties
 - ▶ Compound Properties
 - ▶ Relation Properties
- ▶ Schema
 - ▶ Classes & Attributes
- ▶ UML
- ▶ Business Rules
 - ▶ Business Object Display Rules
 - ▶ Naming Rules
 - ▶ Extension Rules
 - ▶ Deep Copy Rules
 - ▶ GRM Rules
 - ▶ ID Context Rules
 - ▶ Property Rules
- ▶ Rule Engine Framework
 - ▶ Application Extension Points
 - ▶ Application Extension Rules
 - ▶ Business Context
- ▶ LOVs
 - ▶ Simple
 - ▶ Hierarchical
 - ▶ Interdependent
- ▶ Options
 - ▶ Change
 - ▶ Tool
 - ▶ Status
 - ▶ View Type
 - ▶ PS Occurrence Type
 - ▶ Validation Data
 - ▶ ID Context



Business Objects



- ▶ Primary Modeling Object of Teamcenter
- ▶ Captures business knowledge expressed by users
- ▶ Provides the ability to attach
 - ▶ Methods
 - ▶ Attributes
 - ▶ Business Rules
- ▶ Object oriented
- ▶ Supports inheritance
- ▶ Examples
 - ▶ Item, Documents, Forms,
 - ▶ Datasets, Folders





Business Objects



- ▶ New Business Objects can be created through Wizards
- ▶ Codeless configuration
- ▶ Properties can added directly to the Business Object
 - ▶ Standard Data Types
 - ▶ References to other Business objects
 - ▶ Runtime Properties
 - ▶ Relation Properties
 - ▶ Compound Properties

BMIDE Feature: Business Object editor has quick links to related items

- Deep Copy Rules
- Display Rules
- Extension Rules

The screenshot shows the 'Business Object : POM_object' editor window. A 'New Form' dialog box is open, allowing the user to create a new form. The dialog box has the following fields and controls:

- Form** section: 'Create a new form:'
- Project**: A dropdown menu.
- Name**: A text input field.
- Parent**: A text input field with a 'Browse' button.
- Advanced <<<**: A button to expand advanced options.
- Use New Class** (radio button) and **Use Existing Class** (radio button, selected).
- Name**: A text input field with a 'Browse' button.
- Attributes**: A table with columns Name, Type, Size, and RefClass.
- Add** and **Remove** buttons for the attributes table.
- < Back**, **Next >**, **Finish**, and **Cancel** buttons at the bottom.

Name	Type	Size	RefClass



Business Objects



- ▶ BMIDE features a new Item wizard for quickly creating new Business Objects and custom data
- ▶ Wizard creates:
 - ▶ New Item
 - ▶ New Item Master Form and attributes
 - ▶ New Item Revision
 - ▶ New Item Revision Master Form and attributes

BMIDE Feature: Item wizard reduces the steps necessary to create new items and associated forms

Business Object
Create a new Business Object

Project: Project1
Name: MfgItem
Parent: Item

Form
Create a new Form...

Name: MfgItem Revision
Parent: Form
Form Storage Class
Class Name: MfgItem
Parent: POM_Object

Attribute Name
mfg_state
mfg_city

Form
Create a new Form...

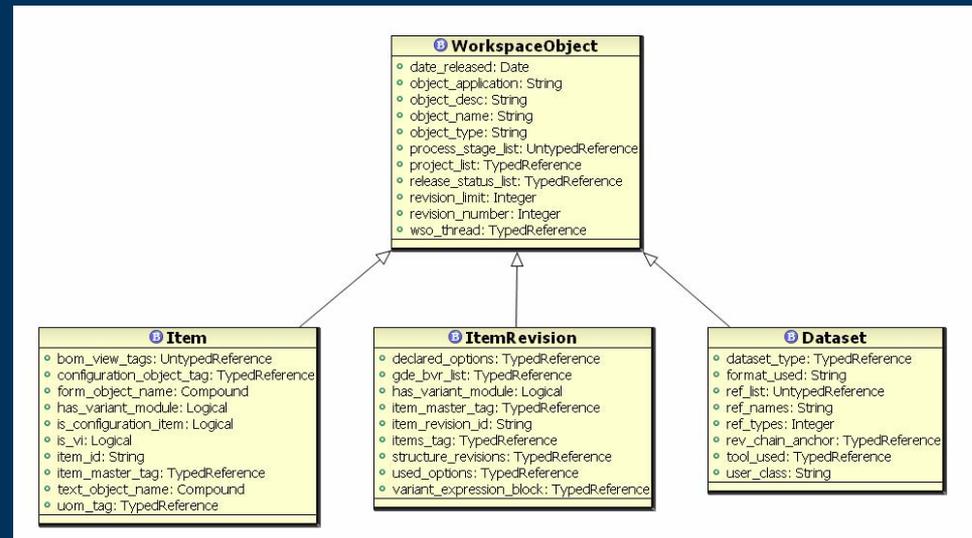
Name: MfgItem Master
Parent: Form
Form Storage Class
Class Name: MfgItemMaster
Parent: POM_object

Attribute Name	Type	Size	Ref Class
mfg_state	String	30	
mfg_city	String	30	

Add
Modify
Remove

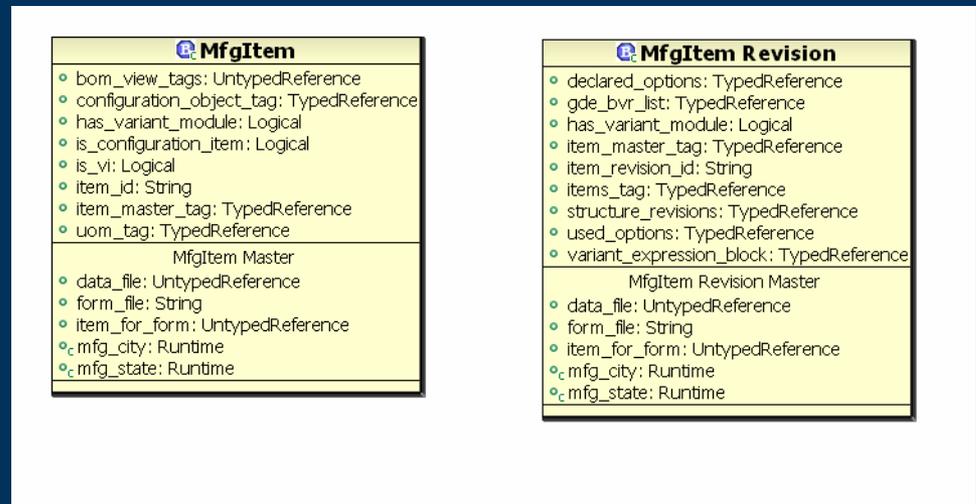
< Back Next > Finish Cancel

- ▶ Industry standard language
- ▶ Expresses Teamcenter
 - ▶ Objects
 - ▶ Attributes
 - ▶ Operations
 - ▶ Inheritance
- ▶ Supports Business Object Model and Logical Model



- ▶ UML figures for Items and Item Revision include attribute groups
- ▶ Attribute groups show associated Form properties

BMIDE Feature: Attribute groups makes it easier for analyst to see all relevant information related to the Business Object in one view.





Schema



- ▶ Define new storage classes and attributes for persisting data
- ▶ Control
 - ▶ Flag for exporting class data to other sites
 - ▶ Data types and sizes
 - ▶ Initial values
 - ▶ Upper and lower bounds for attribute values
 - ▶ Read/Write protection

The screenshot displays the Teamcenter Business Objects interface. On the left, a tree view shows a project named 'Project1' containing a 'POM_object' class with several sub-classes like 'AIEBatd', 'AIECom', 'AIEData', etc. Two dialog boxes are overlaid on the interface:

- New Class Dialog:** This dialog is titled 'New Class' and is used to create a new class. It shows the 'Project' as 'Project1', the 'Name' as 'MfgFormStorage', and the 'Parent' as 'POM_object'. There is a checkbox for 'NewClassPage' which is currently unchecked.
- Create New Attribute Dialog:** This dialog is titled 'Create New Attribute' and is used to define a new attribute. It shows the 'Project' as 'Project1', the 'Name' as 'mfg_city', the 'Attribute Type' as 'String', and the 'String Size' as '30'. The 'Attribute Access' is set to 'WRITE'. There are also checkboxes for 'Array', 'Nulls Allowed', 'Is Unique', 'Ignore Errors', and 'Is Candidate Key', all of which are currently unchecked. The 'Initial Value', 'Lower Bound', and 'Upper Bound' fields are empty.

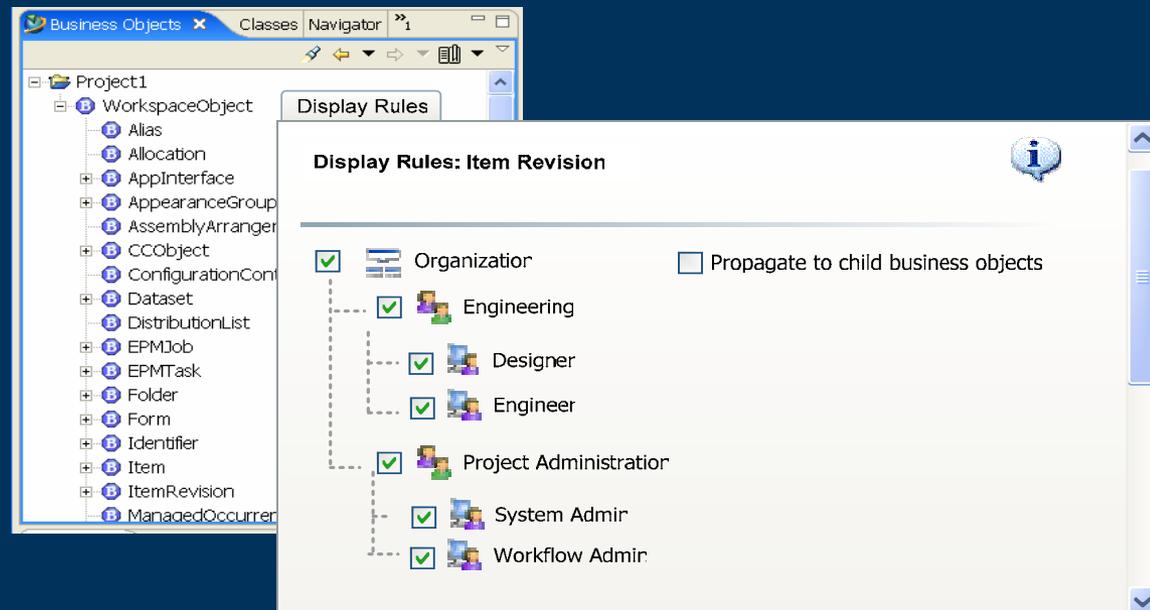


Display Rules



- ▶ Business Behavior Configuration
- ▶ Restrict the visibility of Business Object in the creation dialog windows
- ▶ Configurable based on
 - ▶ Organization
 - ▶ Group
 - ▶ Role

BMIDE Feature: Used to hide Business Objects from specific types of users





List of Values



- ▶ Business Behavior Configuration
- ▶ Create your own lists of valid choices for properties
- ▶ Can be configured so that users are required to pick from the list, suggestive list, or range of values
- ▶ Lists can be populated with Teamcenter data
- ▶ Large lists can be organized into cascading groups

BMIDE Feature: LOV constrain properties so that users pick from a configurable list of choices

The image shows three overlapping screenshots of the 'LOV Edit the LOV' configuration dialog. The dialog is titled 'LOV Edit the LOV' and contains the following fields and options:

- Name: [Text Field]
- Description: [Text Field]
- Type: [Text Field]
- Usage: [Text Field]
- Class: [Text Field]
- Attribute: [Text Field]
- Usage: Exhaustive Suggestive Range
- Lower: [Text Field] ...
- Upper: [Text Field] ...

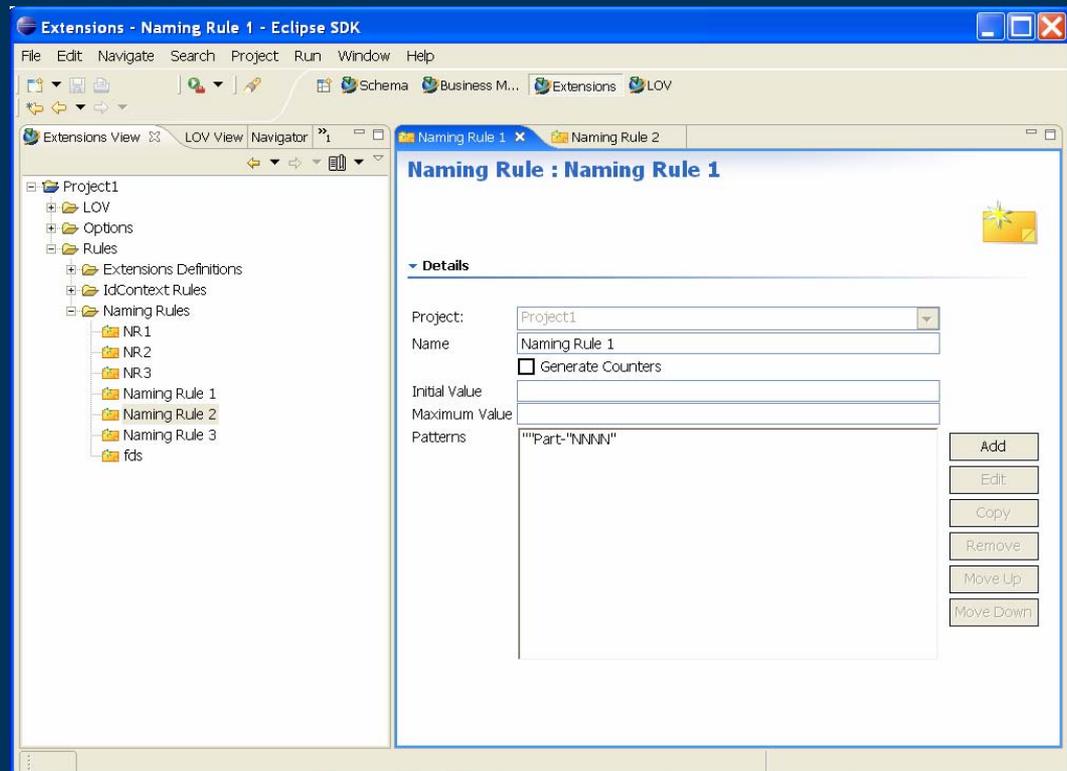


Naming Rules



- ▶ Specify valid naming convention and patterns for Items, Item Revisions, Datasets, and Forms
- ▶ Automatic name generation follows the pattern in sequence
- ▶ Examples:
 - ▶ Part-0001
 - ▶ Doc99-Kvr99

BMIDE Feature: Naming Rules constrain properties so that users enter information that conforms to a format





Property Rules



- ▶ Control access to the properties on an object
- ▶ Constrain properties
 - ▶ Modifiable
 - ▶ Visible
 - ▶ Enabled
 - ▶ Required
 - ▶ Initial Value
 - ▶ Complex property
 - ▶ Combination of properties and strings that assign a value to the property



Compound Properties



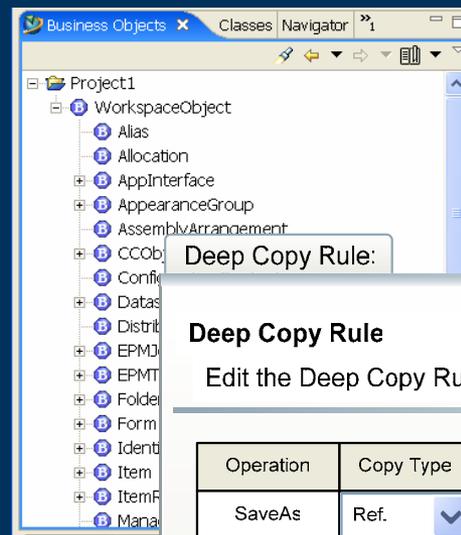
- ▶ Show properties from one object on another object
- ▶ Properties can be gathered from various related objects and shown on a central object
- ▶ Can be configured to get a property by traversing from one object to another by using Reference or Relation properties
- ▶ Any level of traversal



Deep Copy Rules



- ▶ Used in combination with creation of Item Revisions using the Save As and Revise menu commands
- ▶ During respective operations, deep copy rules can control which attached business objects are copied forward, referenced, or not copied



Deep Copy Rule
Edit the Deep Copy Rule

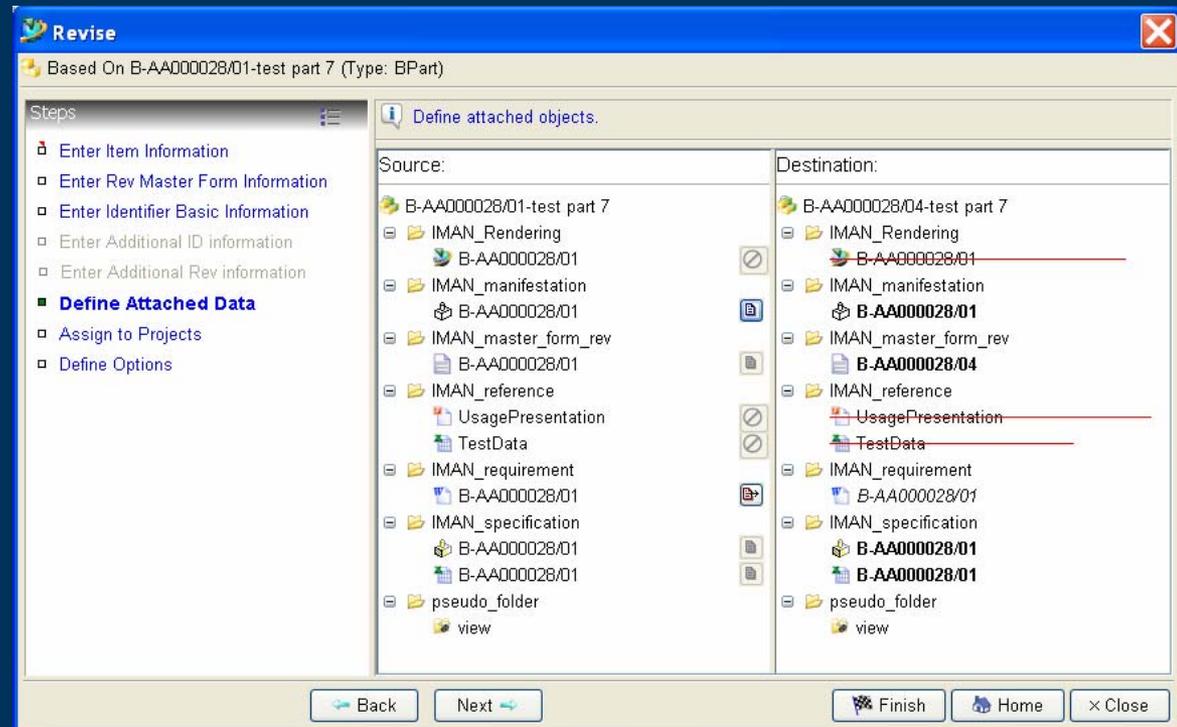
Operation	Copy Type	Relation Type	Object Type	Rec.
SaveAs	Ref.	Specification	Form	<input checked="" type="checkbox"/>
SaveAs	Object	Reference	Dataset	<input checked="" type="checkbox"/>
Revise	Never	Alias	UGMaster	<input checked="" type="checkbox"/>



Deep Copy Rules



- ▶ During a Save As or Revise operation, the Deep Copy Rules can be visualized in the Client in a comparison view.
- ▶ The left view shows your current object and its associated objects.
- ▶ The right view shows which associated objects will be copied and which will not (strike through)





Relationship Management



- ▶ Define new relations between two objects
- ▶ Constrain the usage of the relation between two objects
- ▶ Relationships can be constrained in terms of:
 - ▶ Cardinality (0, 1, 2...)
 - ▶ Changeability
 - ▶ Attachability
 - ▶ Detachability

GRM Rules

GRM Rules

Mode Query Selection

Primary

Secondary

Relation

Relation	Primary	Secondary	1 st Cardinality	2 nd Cardinality	Attachability	Changeability	Detachability	Secured
ComponentRelation	Item	ItemRevision	0..1	0..3	Write	Add	Write	<input checked="" type="checkbox"/>
3DMarkup	WorkspaceObject	DirectModel	0..2	0..2	Write	Add	Write	<input type="checkbox"/>
TC_Derived	Item	Document	0..1	0..*	Write	Add	Write	<input checked="" type="checkbox"/>



Extension Rules



- ▶ Extend OOTB behavior with additional business behaviors
- ▶ Pre-canned behaviors can be configured to execute at different hook points...
 - ▶ Pre-Condition, Pre-Action, Post-Action
 - ▶ User Exits
- ▶ Behavior can be added to operations on the Business Object
 - ▶ Create, Save, Save As, Import, Export, etc
- ▶ Behavior can be added to properties of the Business Object
- ▶ New functions can be developed by the site using classic coding techniques using programming extension and connected to hook points

ItemRevision Extension Rule

Extension Rule

Edit the Extension Rule

Extension Rule for ItemRevision : CreateRevision

▼ Pre-Condition

Name	Active	COTS	Inherited
ValidateName	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Extension1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Extension2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UserExtension1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

▼ Pre Action

Name	Active	COTS	Inherited
ReviseName	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Extension3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Extension4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UserExtension2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

▼ Post Action

Name	Active	COTS	Inherited
CheckName	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Extension5	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UserExtension3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UserExtension4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Properties and Operations

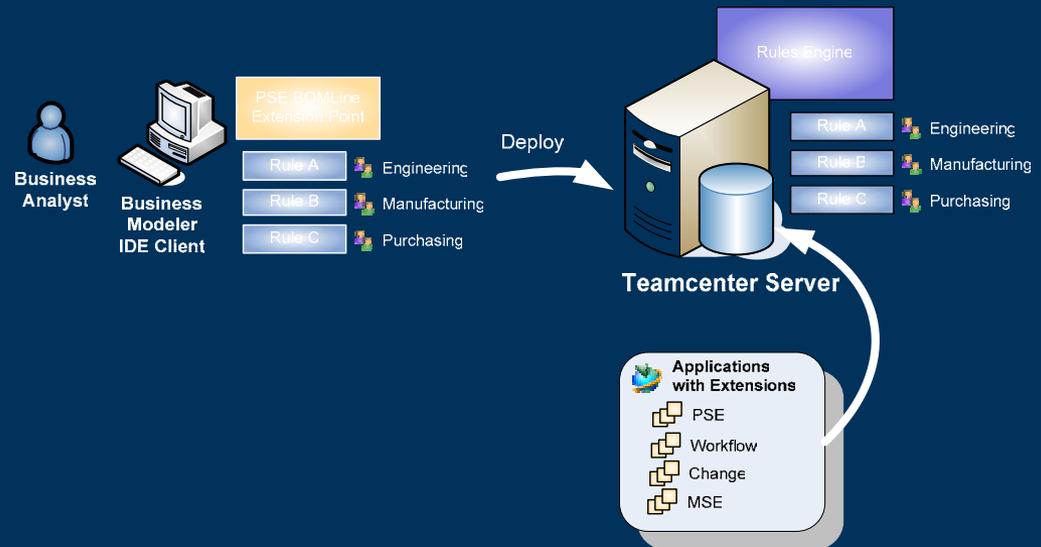
- Properties
 - Description
 - ID
 - Last Modified Date
 - Last Modified User
 - Name
 - Owning Group
 - Owning Site
 - Owner User
 - Released Date
 - Unit Of Measure
- Property Operations
 - ASK
 - SET
- Type Operations
 - CreateRevision**
 - Delete
 - Export
 - Import
 - Save
 - SaveAs
 - Refresh
 - Revise



Rules Engine Framework



- ▶ Rules Engine used to evaluate complex business logic
- ▶ Extension Points are defined in the application code and exposed in the BMIDE for configuration
- ▶ Business Analyst can review the extension points in the BMIDE and configure each to suit the business objectives
- ▶ Behaviors can be varied based on Group and Role
- ▶ Rules Engine can evaluate



- ▶ Primitives: String, Integer, Date, Char, Logical
- ▶ Business objects
- ▶ Expressions
 - ▶ Less than, greater than
 - ▶ Equal to, Not Equal to
 - ▶ Between x and y

- ▶ Example Rules...
- ▶ If the status is released, then send to partners
- ▶ If the material is copper and the pressure is between 10 and 20, then display in red
- ▶ If the document type is Quality Assessment and the priority is critical, then submit the document for secondary review



Rules Engine Framework



- ▶ Application Developer Tasks
 - ▶ Extensions Points are defined in the application code by application developer
 - ▶ Developer determines the expected input and output
 - ▶ Exposes the extension point by creating an Application Extension Point in the BMIDE and constrains the input and output to the analyst
 - ▶ Adds Rule Engine Execute API to application Class API
 - ▶ Passes expected input as arguments
 - ▶ SOA enabled
 - ▶ Rules can be stored in Teamcenter and executed from 3rd party applications

New Application Extension Point

Create a new Application Extension Point

Project: Project1

ID: tc.core.pse.backgroundcolor

Name: Backgroundcolor

Type: Decision Table

Folder: PSE BOM-Line Rendering

Description: Input are material and properties and outputs are background colors(RGB)

Inputs

Input Type	Type Name	Display Name	Is Expression	List of Choices
Primitive	String	Material	false	
Primitive	Integer	Pressure	true	

Outputs

Output Type	Type Name	Display Name	List of Choices
Primitive	Integer	Red	color_list
Primitive	Integer	Green	color_list
Primitive	Integer	Blue	color_list

< Back Next > Finish Cancel



Rules Engine Framework



- ▶ Business Analyst Tasks
 - ▶ Analyzes list of application extension points in the BMIDE
 - ▶ Creates an Application Extension Rule for the target extension point
 - ▶ Based on the constrained input and output, configures the rule to achieve the business objects
 - ▶ Can configure multiple Application Extension Rules for an extension point by
 - ▶ Can vary the behavior of the rule
 - ▶ Deploys the extension rules along with other elements in BMIDE template to a Teamcenter server

New Application Extension Rule

Application Extension Rule
Create a new application extension rule

Project:

ID:

Name:

Description:

Secure

Decision Table

Material	Pressure	Red	Green	Blue
Steel	Equal to 10	255	0	0
Plastic	Less than 25	0	255	0
Bronze	Between 50 and 75	255	0	255
Copper	Greater than 100	255	255	0

Config Inputs
Add
Remove
Edit

Business Context

<input type="text" value="Business Context1"/>
<input type="text"/>
<input type="text"/>
<input type="text"/>
<input type="text"/>

Add
Remove

< Back Next > Finish Cancel



Rules Engine Framework



- ▶ Advanced Rules
- ▶ Business Object can be passed as input arguments
- ▶ Application Extension Point
 - ▶ Specifies Business Object input and output
- ▶ Inputs columns are configured later by the Business Analyst on the Application Extension Rule
- ▶ Business analyst can pick any valid property on the Business Object for configuration

New Application Extension Point
Create a new Application Extension Point

Project: Project1
ID: tc.core.compressor.itemmaterial
Name: Item Material
Type: Decision Table
Folder: Compressor Design
Descriptor: Item is input and material is output.

Input Type	Type Name	Column Name	Is Expression	List of Choices
Type	Item			

Output Type	Type Name	Column Name	List of Choices
Primitive	String	Material	material

< Back Next > Finish Cancel

New Application Extension Rule
Create a new application extension rule

Project: Project1
ID: tc.core.compressor.itemmaterial
Name: Material Selector
Descriptor: Materials for the values of type and size
 Secure

Type	Size	Material
Valve	1"	CUNI
Flange	3"	Copper
Pipe	4"	Steel
Pipe	2"	Plastic

Business Context

< Back Next > Finish Cancel



Agenda



- ▶ Vision & Strategy
- ▶ What's New
- ▶ Development Processes Support
- ▶ Features
- ▶ **Future Deliverables**
- ▶ Questions



Unified Platform – beyond Teamcenter 2007

- ▶ Business Context
 - ▶ Extending to support Project and Program
- ▶ Support for programming Behavior Extensions
- ▶ Document Management Tailoring
- ▶ BMIDE Extension Reports
- ▶ Managing Display Names
- ▶ REF Enhancements
 - ▶ More COTS Extension Points
 - ▶ Decision Tree, Forward Chaining
- ▶ CMM – Packaged Workflows
- ▶ Role Based UI Tailoring
 - ▶ Property Style Sheets
 - ▶ Panel arrangement



Teamcenter Business Modeler IDE Realized Benefits



- ▶ Reduces Project Risks
- ▶ Reduces Implementation Time and Costs
- ▶ Improves Productivity and Efficiencies
- ▶ Exercises and Guides Best Practices and Standards Compliance
- ▶ Consistent application customization
 - ▶ Published controls, verifications, and validations
- ▶ Promotes Application and Functional Synergy through Open IDE Architecture
- ▶ Provide Easy-to-Use Business Analyst Interface for Extending the System
- ▶ Wizards guide the Tailoring and System Configuration
- ▶ Provides Basis for Streamlined Release Upgrades
- ▶ Distribute Pre-defined and Tested System Configurations to Remote Sites
- ▶ Reduced Operation Disruption with System Runtime Extension Recognition



Summary



- ▶ Business Modeler IDE
 - ▶ Best in class and unified product tailoring environment
 - ▶ Reducing cost of ownership
 - ▶ Built on open standards and tools



www.ugs.com

Bob Donovan