

SmartPlant Foundation

Configuration and Administration II

Course Guide Volume 1

February 2007

Version 3.8

Process, Power & Marine



 **INTERGRAPH**

DSPF1-TP-1000010A

SmartPlant Foundation Configuration and Administration II *Course Guide*

February 2007

Version 3.8

Copyright

Copyright © 2002 - 2007 Intergraph Corporation. All Rights Reserved.

Including software, file formats, and audiovisual displays; may be used pursuant to applicable software license agreement; contains confidential and proprietary information of Intergraph and/or third parties which is protected by copyright law, trade secret law, and international treaty, and may not be provided or otherwise made available without proper authorization.

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) of the *Contractor Rights in Technical Data* clause at DFARS 252.227-7013, subparagraph (b) of the *Rights in Computer Software or Computer Software Documentation* clause at DFARS 252.227-7014, subparagraphs (b)(1) and (2) of the *License* clause at DFARS 252.227-7015, or subparagraphs (c) (1) and (2) of *Commercial Computer Software--Restricted Rights* at 48 CFR 52.227-19, as applicable.

Unpublished---rights reserved under the copyright laws of the United States.

Intergraph Corporation
Huntsville, Alabama 35894-0001

Warranties and Liabilities

All warranties given by Intergraph Corporation about equipment or software are set forth in your purchase contract, and nothing stated in, or implied by, this document or its contents shall be considered or deemed a modification or amendment of such warranties. Intergraph believes the information in this publication is accurate as of its publication date.

The information and the software discussed in this document are subject to change without notice and are subject to applicable technical product descriptions. Intergraph Corporation is not responsible for any error that may appear in this document.

The software discussed in this document is furnished under a license and may be used or copied only in accordance with the terms of this license.

No responsibility is assumed by Intergraph for the use or reliability of software on equipment that is not supplied by Intergraph or its affiliated companies. THE USER OF THE SOFTWARE IS EXPECTED TO MAKE THE FINAL EVALUATION AS TO THE USEFULNESS OF THE SOFTWARE IN HIS OWN ENVIRONMENT.

Trademarks

Intergraph, the Intergraph logo, SmartSketch, FrameWorks, SmartPlant, SmartPlant Foundation, SmartPlant P&ID, SmartPlant Instrumentation and INtools are registered trademarks of Intergraph Corporation. Microsoft and Windows are registered trademarks of Microsoft Corporation. Other brands and product names are trademarks of their respective owners.

This courseware was developed by Roni Carpenter, Bill Crego and Mitch Harbin, PPM-PIM Training, Huntsville, Alabama.

SmartPlant® SPF Configuration and *Foundation* Administration II (SPFA2)

Course Outline

Day 1

1. The SmartPlant Schema Model: An Overview

- Class Definitions
- Interface Definitions
- Relationships
- Property Definitions
- Shared Object Definitions

SmartPlant® Foundation SPF Configuration and Administration II (SPFA2)

- Smartplant Foundation Structure
 - SPF Databases
 - Schema Load Process
 - Resetting IIS
-  Activity – Loading the Modified Schema's into SPF

2. Schema Object Administration

- Starting Schema Object Administration
- Class Definitions
- Interface Definitions

SmartPlant® Foundation SPF Configuration and Administration II (SPFA2)

- **Property Definitions**
- **Enumerated List Sets**
- **Viewing and Modifying Navigation Objects**
- **Edge Definitions in Schema Object Admin**
- **Graph Definitions in Schema Object Admin**
- **View Definitions in Schema Object Admin**
- **Class View Maps**
- **RelDef Security**
- **Access Control**
- **Configuring SmartPlant Foundation E-mail**
- **Email Digest Setup**
- **Introduction to Vault Replication**

SmartPlant® Foundation SPF Configuration and Administration II (SPFA2)



Activity – Schema Object Administration

3. SmartPlant Forms

- Converting to a Form
 - Form Builder Editor
 - Configuring the Engineering Numbering System
 - Additional Form Builder Functions
- Activities – Generating and Modifying Forms**

SmartPlant® Foundation SPF Configuration and Administration II (SPFA2)

Day 2

4. Understanding the Loader Interface

- Load File Formats
- Processing Load Files
- Introduction to SmartPlant Loader Manager
 -  Activity 1 – Loading SPF Users and Groups
 -  Activity 2 – Create Data Load File

SmartPlant® **Foundation** **SPF Configuration and Administration II (SPFA2)**

5. Introduction to Methods

- **Configuring Methods**
- **Setting Method Security**
- **Attaching Methods**
- **Configuring New Menu Items**
- **Adding Menu Item Security**
 -  **Activity 1 – Configuring Methods**
 -  **Activity 2 – Configuring Menu Items**

SmartPlant® Foundation SPF Configuration and Administration II (SPFA2)

Day 3

- Additional Model Components
- Context Sensative Menus
- Creating Toolbar and Menu Methods
- Creating New Toolbar Item
- Adding Toolbar Security
- Defining Custom Icons
- Testing Toolbars and Menus
- Units of Measurement
- UOM Sets

SmartPlant® **SPF Configuration and** **Foundation** **Administration II (SPFA2)**

- Revision Schemes
 - Activity 3 - Configuring Toolbars and Menus

6. SmartPlant Change Manager (Workflow)

(with an interactive activity during the chapter)

- User Groups and Change Management
- Creating Workflows
- Viewing Workflows
- Attaching Workflows
- Other Workflow Functions
- Maintain Calendar Work Week

SmartPlant® Foundation SPF Configuration and Administration II (SPFA2)

- **Maintain Calendar Work Week**
- **Options**
- **Using Workflows**
- **Setting the Active Scope**
- **Submitting a Document to the Workflow**
- **Viewing a Workflow**
- **Workflow Signoffs**
- **Workflow History**

SmartPlant® **SPF Configuration and** ***Foundation*** **Administration II (SPFA2)**

7. Introduction to Transmittals

- **Transmittal Project Roles**
 - **Transmittal Structure**
 - **Transmittal Administrative Setup**
 - **Transmittal Users and User Groups**
 - **Modifying Transmittal Workflows**
 - **Distribution Matrix Overview**
 - **Creating a New Reason for Issue**
-  **Activity 1 – Transmittal Administrative Setup**

SmartPlant® Foundation SPF Configuration and Administration II (SPFA2)

- Creating an Internal Transmittal
 - Creating a Transmittal Section
 - Expanding the Transmittal Structure
 - Creating an External Transmittal
 - Saving a Transmittal Template
 - Create a Transmittal for a Selected Set of Documents
 - Creating a Transmittal from a Template
-  Activity 2 – Creating Transmittals

SmartPlant® **SPF Configuration and** **Foundation** **Administration II (SPFA2)**

Day 4

8. Using and Managing Transmittals

- Searching for Transmittals
- Processing a Transmittal
- Preparing a Transmittal
- Approve Transmittal
- Issuing a Transmittal and Recipient Response
- Completing a Transmittal
- Exporting a Transmittal Structure
- Transmittal Manipulations
- Activity – Using and Managing Transmittals

SmartPlant® Foundation SPF Configuration and Administration II (SPFA2)

9. Adhoc Reporting

- **Creating a Report Graph Definition**
 - **Creating a Report View Definition**
 - **Custom Line List Adhoc Report Contents**
 - **Creating a Graph Definition for the Line List Report**
 - **Creating a View Definition for the Line List Report**
 - **Adding View Definition Security**
 - **Generating a Report**
 - **Additional Adhoc Report Options**
 - **Generating a Line List Adhoc Report**
-  **Activity – Creating and Running Reports**

SmartPlant® Foundation SPF Configuration and Administration II (SPFA2)

10. Creating Custom Enum Trees and Plant Hierarchies

- Modeling the Custom Enum List Tree
 - Configuring the EnumListType Tree Structure in SPF
 - Testing the Custom Enum Tree
 - Creating a Custom Plant Hierarchy in the Schema Editor
 - Configuring the PBS Hierarchy in SPF
 - Testing the Custom PBS Hierarchy
-  Activity – Creating Custom Enum Trees and Plant Hierarchies

SmartPlant® **Foundation** **SPF Configuration and Administration II (SPFA2)**

11. Introduction to the SPF Upgrade Wizard

- Upgrade wizard control file
- Upgrade Wizard
- Appendix Log File

12. Creating and Using Datasheets

- Working with Datasheets in the Schema Object Model
- Creating Document Templates
- Editing a Document Template
- Mapping a String Property Value

SmartPlant® Foundation SPF Configuration and Administration II (SPFA2)

- Creating a Report Definition
- Adding a Report Definition to the Interface Definition
- Adding a Method to the Interface Definition
- Testing Datasheet Functionality
- Using Datasheets Offline

SmartPlant® Foundation SPF Configuration and Administration II (SPFA2)

Day 5

13. Introduction to the SmartPlant Adapter for Microsoft Excel

- Install and Configure Prerequisites
- SmartPlant Adapter for Microsoft Excel Delivered Components
- Review the Menus and Methods
- Review the delivered SmartPlant Adapter for Microsoft Excel Tool Metadata
- Configure the SmartPlant Adapter for Microsoft Excel Owning Group

SmartPlant® Foundation SPF Configuration and Administration II (SPFA2)

- Configure the SmartPlant SmartPlant Adapter for Microsoft Excel Samples
- Update the Plant, Area, and Unit in the sample Excel file
 - Lab 1
 - Create the Excel EquipmentList Document Type
 - Update the ExcelEquipmentList Class Definition
 - Creating/Updating the Component Schema
 - Update the ExcelEquipment class definition
 - Creating/Updating the Tool Schema (Map) File

SmartPlant® Foundation SPF Configuration and Administration II (SPFA2)

- Create and Map the Equipment Types Enumerated List
- Create and Map the Steam Out Req Enumerated List
- Create and Map the psi Unit of Measure
- Create and Map the F Unit of Measure
- Create the Equipment MapClassDef
- Map the Equipment MapClassDef
- Create the Plant MapClassDef
- Create the Unit MapClassDef
- Create the Plant to Equipment MapRelDef
- Create the Equipment to Unit MapRelDef

SmartPlant® Foundation SPF Configuration and Administration II (SPFA2)

- View the Completed Mappings
- Load the SmartPlant Schema additions into SmartPlant Foundation
- Creating/Validate the Document Template
- Publishing the Excel File
- Grant view access to the published relationships
- Create the edge/graph/view definitions
- Configure the view definition as the default view
- Create the menu and method to find the Excel Equipment Lists
- View the published file and data

SmartPlant® Foundation SPF Configuration and Administration II (SPFA2)

- Updating the SmartPlant P&ID Tool Metadata
- Retrieving the Excel File into SmartPlant P&ID

Wrap up and Dismissal



SPF Configuration and Administration II (SPFA2)

Table of Contents

1. The SmartPlant Schema Model: An Overview	1-3
1.1 Class Definitions	1-4
1.2 Interface Definitions	1-5
1.3 Relationships	1-7
1.3.1 Componentization Relationship	1-9
1.3.2 Realizes Relationship	1-10
1.3.3 Implies Relationship	1-11
1.4 Property Definitions	1-12
1.4.1 Exposes Relationship	1-14
1.4.2 ScopedBy Relationship	1-15
1.4.3 Contains Relationship	1-16
1.5 Shared Object Definitions	1-17
1.6 SPF Databases	1-18
1.7 SmartPlant Foundation Structure	1-19
1.8 Schema Load Process	1-20
1.9 Resetting IIS	1-27
1.10 Activity – Loading the Modified Schema's into SPF	1-31
2. Schema Object Administration	2-3
2.1 Starting Schema Object Administration	2-5
2.2 Class Definitions	2-7
2.2.1 Viewing a Class Definition	2-8
2.2.2 Update a Class Definition	2-11
2.2.3 Delete a Class Definition	2-14
2.3 Interface Definitions	2-16
2.3.1 Viewing an Interface Definition	2-17
2.3.2 Update an Interface Definition	2-19
2.3.3 Delete an Interface Definition	2-20
2.4 Property Definitions	2-22
2.4.1 Viewing a Property Definition	2-23
2.4.2 Update a Property Definition	2-27
2.4.3 Delete a Property Definition	2-29
2.5 Enumerated List Sets	2-31
2.5.1 Viewing Enumerated Sets	2-32
2.5.2 Update an Enumerated Set	2-36
2.5.3 Delete an Enumerated Set	2-38
2.5.4 Viewing Enumerated Level Sets	2-40
2.6 Viewing and Modifying Navigation Objects	2-43
2.6.1 Edge Definitions in Schema Object Administration	2-44
2.6.2 Viewing Edge Definitions	2-45
2.6.3 Creating Edge Definitions	2-48
2.6.4 Setting Edge Definition Security	2-53
2.6.5 EdgeDefs with Conditions	2-56

2.6.6 Graph Definitions in Schema Object Administration _____	2-59
2.6.7 Creating Graph Definitions _____	2-63
2.6.8 View Definitions in Schema Object Administration _____	2-67
2.6.9 Creating View Definitions _____	2-68
2.7 Class View Maps _____	2-74
2.7.1 Creating Class View Maps _____	2-75
2.7.2 Setting Class View Map Security _____	2-79
2.8 RelDef Security _____	2-81
2.9 Access Control _____	2-88
2.10 Configuring SmartPlant Foundation E-mail _____	2-99
2.10.1 Testing the E-mail Configuration _____	2-101
2.11 Email Digest _____	2-105
2.11.1 Email Digest Setup _____	2-107
2.12 Introduction to Vault Replication _____	2-109
2.12.1 Setting up a standalone File Server _____	2-110
2.12.2 Setting up Vaults for Replication _____	2-112
2.12.3 Setting up Users for Replication _____	2-113
2.12.4 File Replication Mechanisms _____	2-115
2.12.5 Using Robocopy for SPF Replication _____	2-116
2.12.6 Constant Synchronization _____	2-118
2.12.7 Periodic Synchronization _____	2-119
2.12.8 Recovery options _____	2-120
2.12.9 Installation as a Service _____	2-120
2.13 Activity – Schema Object Administration _____	2-121
 3. SmartPlant Forms _____	3-3
3.1 Converting to a Form _____	3-3
3.2 Form Builder Editor _____	3-6
3.2.1 Form Editing _____	3-7
3.2.2 Modifying Data Form Sections _____	3-10
3.2.3 Modifying Display Items _____	3-15
3.2.4 Display Item to a Form Relationships _____	3-29
3.3 Configuring the Engineering Numbering System _____	3-32
3.4 Additional Form Builder Functions _____	3-43
3.5 Activity – Generating and Modifying Forms _____	3-45
 4. Understanding the Loader Interface _____	4-3
4.1 Load File Formats _____	4-6
4.1.1 Object Types _____	4-8
4.1.2 Field Types _____	4-12
4.1.3 Relationship Data _____	4-13
4.1.4 Attach a File _____	4-15

4.2 Processing Load Files	4-17
4.3 Introduction to SmartPlant Loader	4-24
4.3.1 Starting the SmartPlant Loader	4-26
4.3.2 Creating SmartPlant Database Connections	4-27
4.3.3 The SmartPlant Loader Queue	4-31
4.3.4 Testing Database Connections and Options	4-32
4.3.5 SmartPlant Loader Log	4-35
4.3.6 SmartPlant Loader Properties	4-37
4.4 Activity 1 – Loading SPF Users and Groups	4-39
4.7 Activity 2 – Create Data Load File	4-41

5. <i>Introduction to Methods</i>	5-3
5.1 Configuring Methods	5-9
5.1.1 Setting Method Security	5-20
5.2 Attaching Methods	5-23
5.3 Configuring New Menu Items	5-28
5.3.1 Adding Menu Item Security	5-33
5.4 Activity 1 – Configuring Methods	5-37
5.5 Activity 2 – Configuring Menu Items	5-41
5.6 Additional Model Components	5-43
5.7 Context Sensitive Menus	5-44
5.8 Configuring Toolbar and Menu Methods	5-46
5.8.1 Creating New Toolbar Item	5-51
5.8.2 Adding Toolbar Security	5-54
5.8.3 Defining Custom Icons	5-56
5.9 Testing Toolbars	5-60
5.10 Testing Menus	5-62
5.11 Units of Measurement	5-64
5.11.1 UOM Sets	5-66
5.12 Revision Schemes	5-68
5.13 Activity 3 – Configuring Toolbars and Menus	5-71

6. <i>Introduction to SPF Change Management Administration</i>	6-3
6.1 User Groups and Change Management	6-4
6.2 Creating Workflows	6-7
6.2.1 Step Types	6-9
6.2.2 Update a Step Type	6-11
6.2.3 Step Classes	6-13
6.2.4 Create a Step Class	6-16
6.2.5 Steps	6-18

6.2.6 Create a Step _____	6-21
6.2.7 Using Reasons for Receipt_____	6-23
6.2.8 Create a Reason For Receipt _____	6-25
6.2.9 Workflow Check Lists_____	6-28
6.2.10 Workflow Statuses_____	6-31
6.2.11 Building a Workflow _____	6-33
6.3 Viewing Workflows _____	6-49
6.4 Attaching Workflows _____	6-52
6.5 Other Workflow Functions_____	6-57
6.5.1 Maintain Calendar Work Week _____	6-58
6.5.2 Options _____	6-60
6.6 Using Workflows _____	6-65
6.6.1 Setting the Active Scope _____	6-66
6.6.2 Submitting a Document to the Workflow _____	6-68
6.6.3 Attaching a Workflow _____	6-73
6.6.4 Viewing a Workflow _____	6-76
6.6.5 Workflow Signoffs _____	6-80
6.7 Workflow History _____	6-98
7. SPF Transmittal Overview _____	7-3
7.1 Transmittal Project Roles _____	7-7
7.2 Transmittal Structure _____	7-9
7.3 Transmittal Administrative Setup _____	7-11
7.3.1 Creating a Department / Office _____	7-12
7.3.2 External Organizations _____	7-15
7.3.3 Creating a Supplier _____	7-18
7.3.4 Creating a Contract _____	7-21
7.3.5 Transmittal Users and User Groups _____	7-24
7.4 Modifying Transmittal Workflows _____	7-29
7.5 Distribution Matrix Overview _____	7-38
7.5.1 Creating an Internal Transmittal Matrix _____	7-40
7.5.2 Creating an External Transmittal Matrix _____	7-47
7.6 Creating a New Reason For Issue _____	7-53
7.7 Activity 1 – Transmittal Administrative Setup _____	7-57
7.8 Creating an Internal Transmittal _____	7-67
7.8.1 Creating a Transmittal Section _____	7-73
7.8.2 Expanding the Transmittal Structure _____	7-76
7.9 Creating an External Transmittal _____	7-78
7.10 Saving a Transmittal Template _____	7-80
7.11 Create a Transmittal for a Selected Set of Documents _____	7-82
7.12 Creating a Transmittal from a Template _____	7-85
7.13 Activity 2 – Creating Transmittals _____	7-87

8. Using and Managing Transmittals	8-3
8.1 Searching for Transmittals	8-7
8.2 Processing a Transmittal	8-13
8.2.1 Preparing a Transmittal	8-16
8.2.2 Approve Transmittal	8-25
8.2.3 Issuing a Transmittal and Recipient Response	8-33
8.3 Completing a Transmittal	8-41
8.4 Exporting a Transmittal Structure	8-50
8.5 Transmittal Manipulations	8-57
8.5.1 Updating Transmittals	8-57
8.5.2 Canceling a Transmittal	8-59
8.5.3 Terminating Transmittals	8-61
8.5.4 Terminating Transmittal Sections	8-63
8.6 Activity – Using and Managing Transmittals	8-65
 9. Adhoc Reporting	9-3
9.1 Creating a Report Graph Definition	9-5
9.2 Creating a Report View Definition	9-12
9.3 Custom Line List Adhoc Report Contents	9-18
9.3.1 Creating a Graph Definition for the Line List Report	9-22
9.3.2 Creating a View Definition for the Line List Report	9-27
9.4 Loading Report Navigation Objects	9-35
9.5 Adding View Definition Security	9-37
9.6 Generating a Report	9-41
9.6.1 Additional Adhoc Report Options	9-52
9.7 Generating a Line List Adhoc Report	9-59
9.8 Activity – Creating and Running Adhoc Reports	9-65
 10. Creating Custom Enum Trees and Plant Hierarchies	10-3
10.1 Creating a Custom Enum List Tree Structure	10-4
10.1.1 Modeling the Custom Enum List Tree	10-5
10.1.2 Configuring the EnumListType Tree Structure in SPF	10-24
10.1.3 Testing the Custom Enum Tree	10-48
10.2 Creating a Custom Plant Hierarchy	10-52
10.2.1 Creating the Custom Hierarchy in the Schema Editor	10-53
10.2.2 Configuring the PBS Hierarchy in SPF	10-68
10.2.3 Testing the Custom PBS Hierarchy	10-74
10.3 Activity – Creating a Custom Enum Tree	10-79

11. Introduction to the SPF Upgrade Wizard	11-3
11.1 Upgrade wizard control file	11-5
11.2 Upgrade Wizard	11-7
11.2.1 Welcome screen	11-8
11.2.2 Information for SPF Loader	11-15
11.2.3 Final Review	11-17
11.2.4 Completed and Upgrade Status	11-21
11.3 Appendix Log File	11-22
12. Introduction to Datasheets	12-3
12.1 Objectives	12-4
12.2 Working with Datasheets in the Schema Object Model	12-5
12.2.1 The SmartPlant Schema	12-7
12.2.2 Starting the Schema Editor	12-8
12.2.3 Editing an Open Schema File	12-14
12.2.4 Schema File Tree View	12-15
12.3 Creating Document Templates	12-18
12.4 Editing a Document Template	12-28
12.4.1 The Editor Window	12-29
12.4.2 Mapping the String Property Value	12-31
12.5 Create a Report Definition	12-35
12.5.1 Adding a Report Definition to the Interface Definition	12-41
12.5.2 Adding a Method to the Interface Definition	12-48
12.6 Testing Datasheet Functionality	12-52
12.6.1 Using Datasheets Offline	12-61
12.7 Activity – Creating and Using Datasheets	12-69
13. Introduction to the SmartPlant Adapter for Microsoft Excel	13-3
13.1 Install and Configure Prerequisites	13-3
13.2 SmartPlant Adapter for Microsoft Excel Delivered Components	13-4
13.3 Review the Menus and Methods	13-5
13.4 Review the delivered SmartPlant Adapter for Microsoft Excel Tool Metadata	13-8
13.5 To Review the delivered SmartPlant Foundation Workflow	13-12
13.5.1 What is the Work Process for Publishing an Excel file?	13-14
13.5.2 Configure the SmartPlant Adapter for Microsoft Excel Owning Group	13-17
13.6 Configure the SmartPlant SmartPlant Adapter for Microsoft Excel Samples	13-19
13.6.1 Update the Plant, Area, and Unit in the sample Excel file	13-19
13.7 Lab 1	13-22
13.8 Create the Excel EquipmentList Document Type	13-22

13.9 Update the ExcelEquipmentList Class Definition	13-37
13.9.1 Creating/Updating the Component Schema	13-39
13.10 Update the ExcelEquipment class definition	13-46
13.10.1 Creating/Updating the Tool Schema (Map) File	13-48
13.11 Create and Map the Equipment Types Enumerated List	13-49
13.12 Create and Map the Steam Out Req Enumerated List	13-58
13.13 Create and Map the psi Unit of Measure	13-64
13.14 Create and Map the F Unit of Measure	13-67
13.15 Create the Equipment MapClassDef	13-69
13.16 Map the Equipment MapClassDef	13-75
13.17 Create the Plant MapClassDef	13-79
13.18 Create the Unit MapClassDef	13-81
13.19 Create the Plant to Equipment MapRelDef	13-82
13.20 Create the Equipment to Unit MapRelDef	13-86
13.21 View the Completed Mappings	13-87
13.22 Load the SmartPlant Schema additions into SmartPlant Foundation	13-93
13.23 Configuring SmartPlant Foundation User Access	13-95
13.24 Creating the Document Template	13-96
13.25 Validate the Document Template	13-108
13.26 Publishing the Excel File	13-113
13.27 Grant view access to the published relationships	13-116
13.28 Create the edge definitions	13-117
13.29 Create the ExcelEquipment graph definition	13-121
13.30 Create the ExcelEquipment view definition	13-122
13.31 Configure the view definition as the default view	13-128
13.32 Create the menu and method to find the Excel Equipment Lists	13-130
13.33 View the published file and data	13-132
13.34 Updating the SmartPlant P&ID Tool Metadata	13-135
13.34.1 Retrieving the Excel File into SmartPlant P&ID	13-138

C H A P T E R

1

The SmartPlant Schema Model: An Overview

1. The SmartPlant Schema Model: An Overview

To work with the SmartPlant schema in SmartPlant Foundation, you must understand the SmartPlant model. The model used in conjunction with the SmartPlant schema is a class/interface/relationship model. This model has the following characteristics:



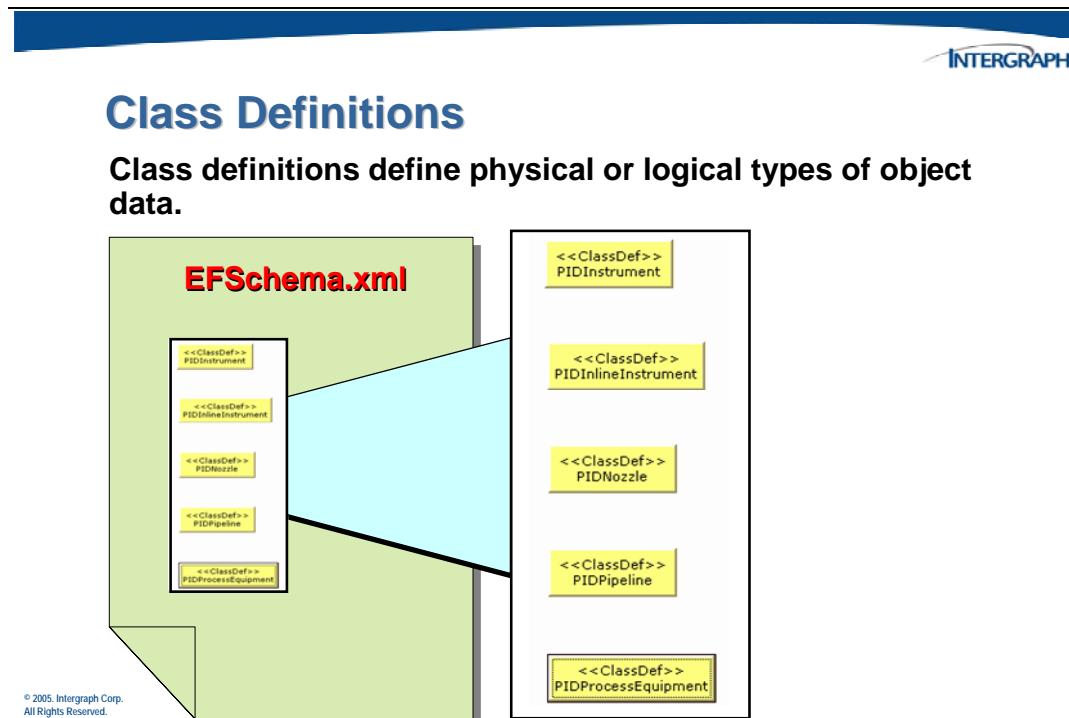
SmartPlant Overview

SmartPlant Enterprise is a class/interface/relationship model:

- Class definitions** define the schema information for concrete objects
- Class definitions realize interface definitions
- Interface definitions** can imply other interface definitions
- Interface definitions expose property definitions
- Relationships** are defined between interface definitions
- Enumerated lists** contain enumerated values
- Shared object definitions** are a group of similar class definitions

1.1 Class Definitions

A class definition is a named description of a set of objects that support or realize the same interface definitions and share the same property definitions and relationships. In the schema, class definitions can represent physical things, such as pumps, or conceptual things, such as projects.



Class definitions realize interface definitions. For example, the **PIDInstrument** class definition realizes the **IInstrument** interface definition. The **IInstrument** interface definition exposes property definitions, such as **Instrument_FluidType**. Interface definitions can be related to other interface definitions by relationships.

The property definitions for class definitions in SmartPlant Foundation contain additional properties that are not part of the schema. When you import class definitions from the schema, SmartPlant Foundation adds the following properties:

- Configuration Dependency** - Used to ensure an object of this class must be created with at least this level of configuration set (normally, **Plant** or **Project**).
- UniqueKey** - Overrides the pattern normally defined on the **EFOBJECT** business object itself.

1.2 Interface Definitions

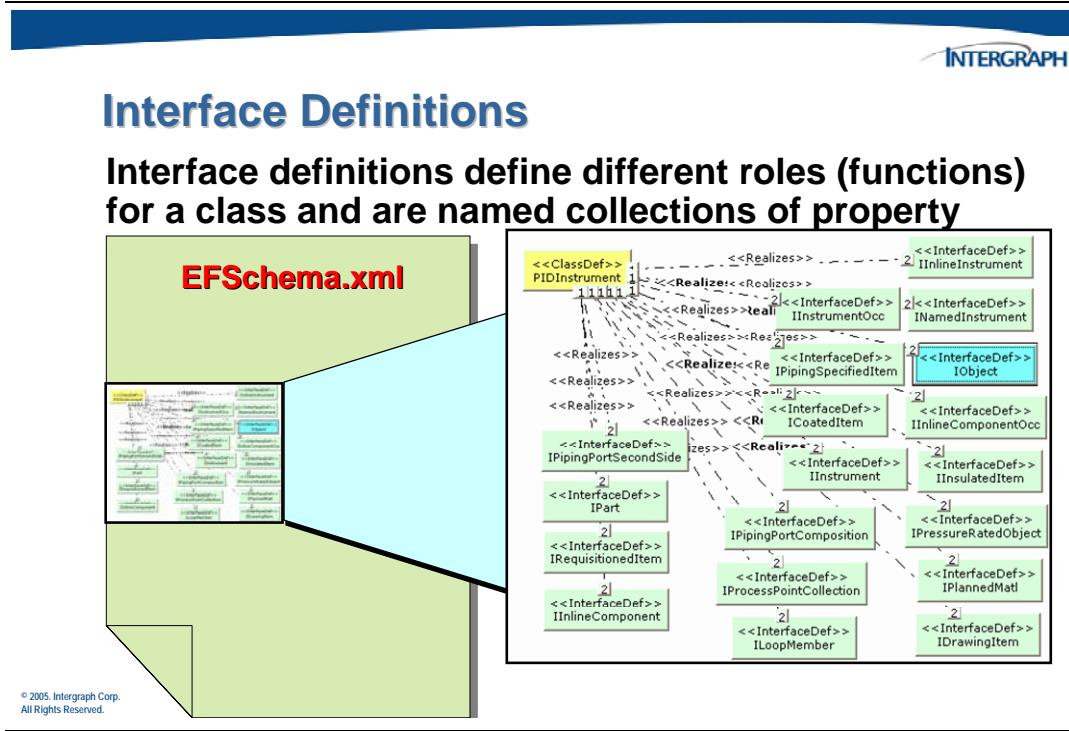
An interface definition is a named collection of property definitions. Interface definitions expose the property definitions for class definitions. By sharing specific interface definitions, class definitions can also share property definitions, but not the data associated with the properties.



Interface Definitions

- Interface definitions expose the property definitions for class definitions**
- Sharing specific interface definitions, class definitions can also share property definitions**
- Sharing interface definitions does NOT share the data associated with the properties**
- A Interface definition represents a "role"**
- Different class definitions can share the same interface definitions (or “role”)**

Each interface definition represents a "role" for a class definition. Different class definitions can share the same interface definitions, and therefore, the same role. For example, every class definition in the schema shares the **IObject** interface, which means that every class definition in the schema has the role of an object. When a class definition has this role, it has an object name, an object description, an object identifier, and any other property definitions exposed by the **IObject** interface.



However, class definitions that share the **IObject** interface also realize other interface definitions that define other roles for them. These interface definitions expose other property definitions for the objects.

In SmartPlant Foundation, interface definitions contain additional associations to methods and workflows in order to control what options appear on the shortcut menu for objects and which workflows an object can be associated with.

1.3 Relationships

Relationships are associations between interface definitions in the schema. They identify two specific objects that are related by a specific relationship type. The relationship type identifies the interface definitions on the two objects that fulfill the roles associated with the relationship type.



Relationships

- Identify two specific objects that are related by a specific relationship type**
- The relationship type also identifies the **roles** and cardinalities defined at both ends**
- Cardinalities provide the minimum and maximum numbers of participants at each end of a relationship**
- Relationships are not between class definitions**

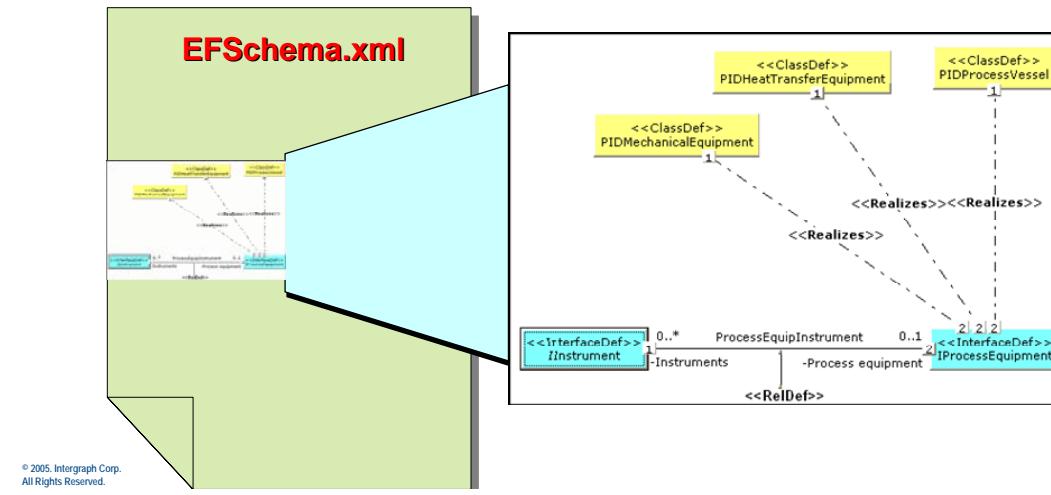
The relationship type also identifies the roles and cardinalities defined at both ends of the relationship. Cardinalities provide the constraints for relationships by defining the minimum and maximum numbers of participants at each end of a relationship.

In the schema, relationship definitions are defined between interface definitions, not between class definitions. A relationship can only exist between objects that support (realize) the interface definitions at each end of the relationship definition.



Relationships

Relationships are associations between interface definitions in the schema.



In SmartPlant Foundation, an additional relationship exists for a user group to control which users can relate objects together when working interactively.

There are several different types of relationships used in the schema:

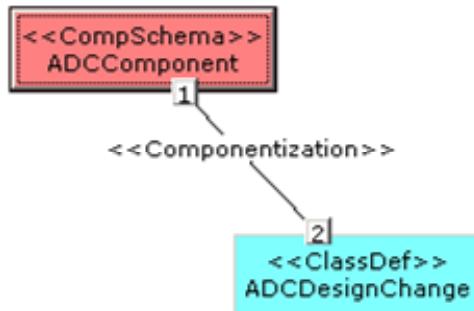
1.3.1 Componentization Relationship

Every class definition is part of one and only one component schema; therefore, a component schema has a componentization relationship with class definitions that belong to that component schema. For example, the **Vapor**, **Liquid**, **Solid**, and **BulkPhase** class definitions belong to the **PFDComponent** component schema, which means that the PFDComponent schema has componentization relationships with each of these class definitions.



Relationships

Class definitions are part of one and only one component schema therefore, a component schema has a componentization relationship with class definitions



1.3.2 Realizes Relationship

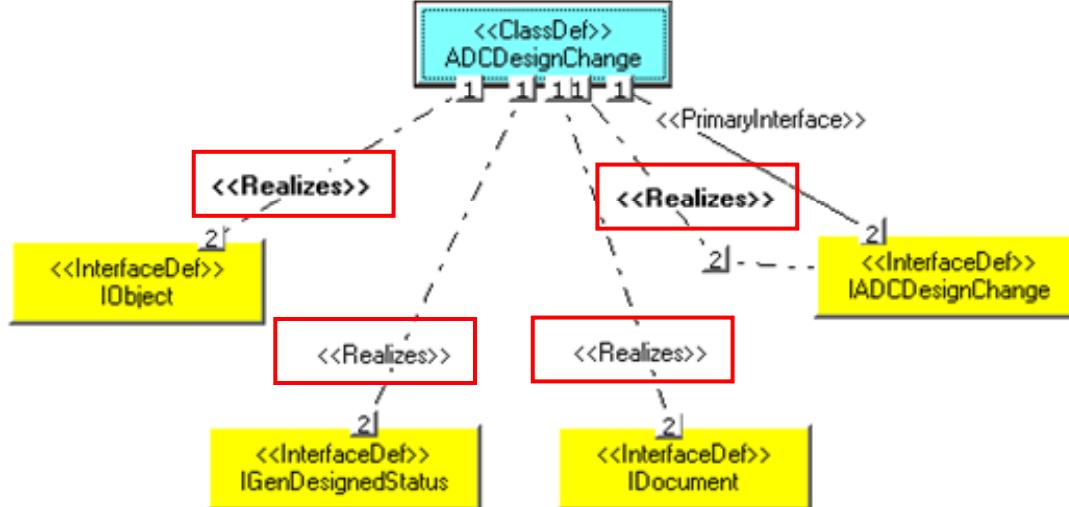
Class definitions have a **Realizes** relationship with interface definitions. This means that instances of a particular class definition support or implement the interface definitions for the realized interface definitions.

For example, the **ADCDesignChange** class definition realizes the **IObject**, **IGenDesignedStatus**, **IDocument**, and **IADCDesignChange** interface definitions, as shown in the following UML diagram.



Relationships

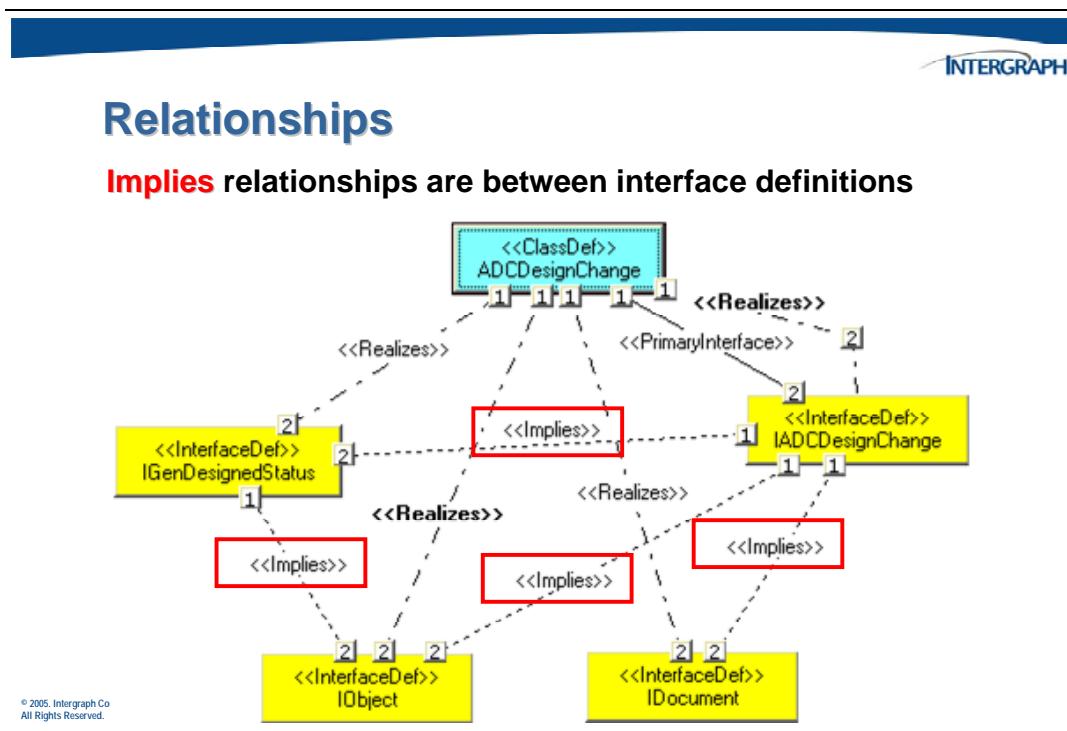
Class definitions have a **Realizes** relationship with interface definitions



1.3.3 Implies Relationship

The schema hierarchy is defined by **Implies** relationships between interface definitions, which are also referred to as implied interfaces. If an interface definition implies another interface definition, then any class definition that realizes the first interface definitions must also realize the implied interface definition.

For example, the **IGenDesignedStatus** interface definition implies the **IObject** and **IADCDesignChange** interface definitions. Therefore, any class definition that realizes the **IGenDesignedStatus** interface definition must also realize the **IObject** and **IADCDesignChange** interface definitions.



1.4 Property Definitions

All property definitions for an object are exposed through its interface definitions and never directly by the object. The property definitions that apply to a particular interface definition are defined by the **Exposes** relationship between objects of type **InterfaceDef** and objects of type **PropertyDef**.



Property Definitions

Property definitions for an object are exposed through its interface definitions and never directly by the object

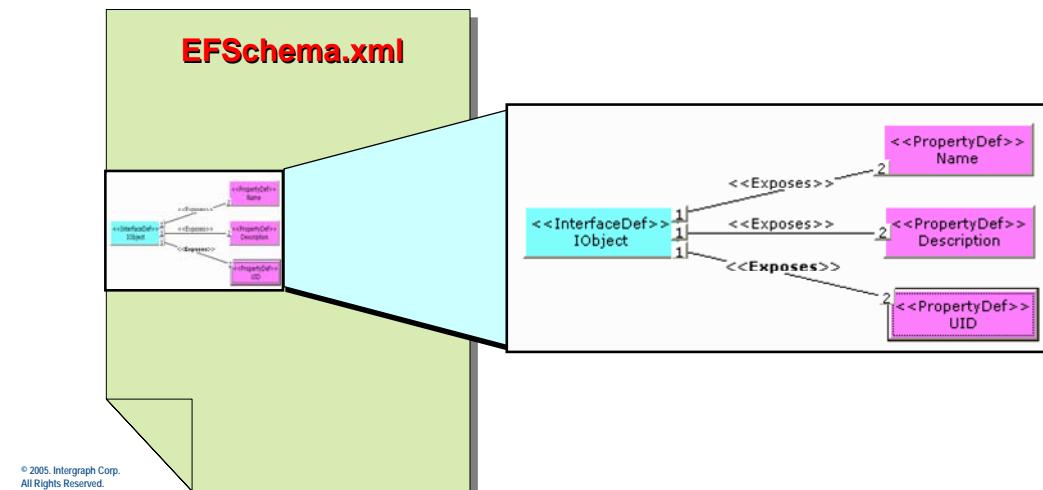
Property information can be used to determine the property type or the possible values for that property definition

A particular property definition for a given class definition is typically exposed by one, and only one, interface definition.



Property Definitions

A property is a characteristic or attribute used to describe a class.



For example, the **IObject** interface definition exposes the following property definitions:

- UID** - Unique identifier for the object. This identifier is only required to be unique within the SmartPlant.
- Name** - Name of the object
- Description** - Description of the object

Property information can be used to determine the property type or, in other words, the possible values for that property definition. Standard schema property types include Boolean, integer, double, and string.

Property definitions of the enumerated property type have a list of possible string property values defined for them. A property definition of this type must match an entry in the list of enumerated property values defined for the property type.

Unit of measure (UoM) is another property type. A unit of measure list is a special enumerated list with additional semantic information. Each enumerated entry in the unit of measure list has conversion factors to convert from the SI unit of measure for that list to a particular unit of measure selected by the user.

A property definition that is typed as a unit of measure has a double precision floating point value in the SI unit for that list. It may optionally have a preferred display unit of measure, a number of significant digits, uncertainty, and a condition, such as **@Gauge**.

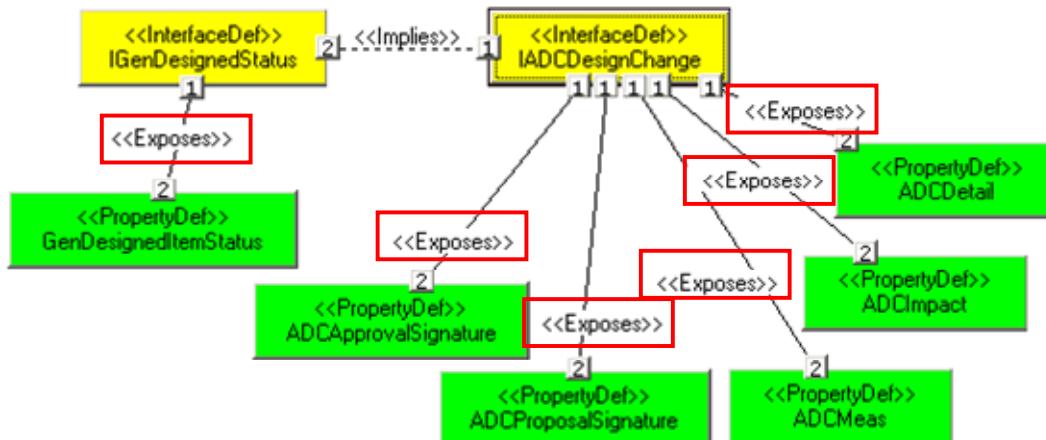
1.4.1 Exposes Relationship

Interface definitions expose property definitions. Each instance of an interface definition therefore exposes properties for each of the related property definitions. For example, the **IGenDesignedStatus** interface definition exposes the **GenDesignedItemStatus** property definition. The **IADCDesignChange** interface definition exposes the **ADCDetail**, **ADCImpact**, **ADCMeas**, **ADCSignatureProposer**, and **ADCSignatureApprover** property definitions. Each object that supports the **IGenDesignedStatus** interface allows access to the **GenDesignedItemStatus** property via that interface. Each object that supports the **IADCDesignChange** interface allows access to the **ADCDetail**, **ADCImpact**, **ADCMeas**, **ADCSignatureProposer**, and **ADCSignatureApprover** properties via that interface.



Relationships

Interface definitions **expose** property definitions



1.4.2 ScopedBy Relationship

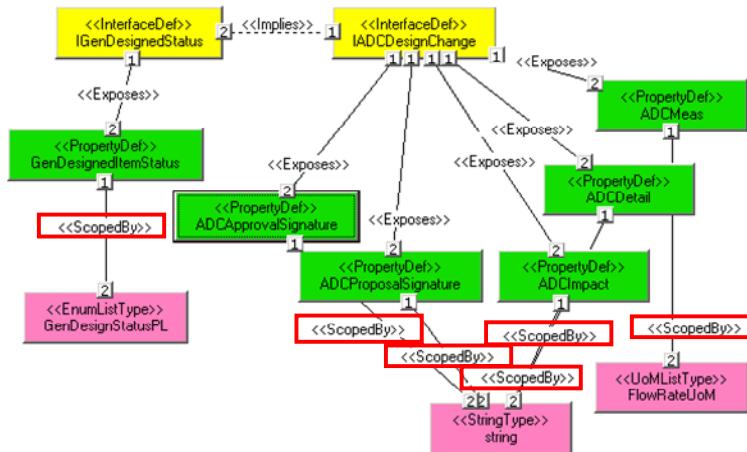
The **ScopedBy** relationship definition specifies the property type definition that defines acceptable values for, or scopes, a particular property definition. This indicates that all properties of that property definition are of that property type.

For example, the **GenDesignedItemStatus** property definition is scoped by the **EnumListType** property type definition. Therefore, all instances of the **GenDesignedItemStatus** property definition (all **GenDesignedItemStatus** properties) are enumerated lists (picklists). The **ADCDetail**, **ADCImpact**, **ADCSignatureProposer**, and **ADCSignatureApprover** property definitions are scoped by the **string** property type definition. Therefore, all instances of the **ADCDetail**, **ADCImpact**, **ADCSignatureProposer**, and **ADCSignatureApprover** property definitions are strings.



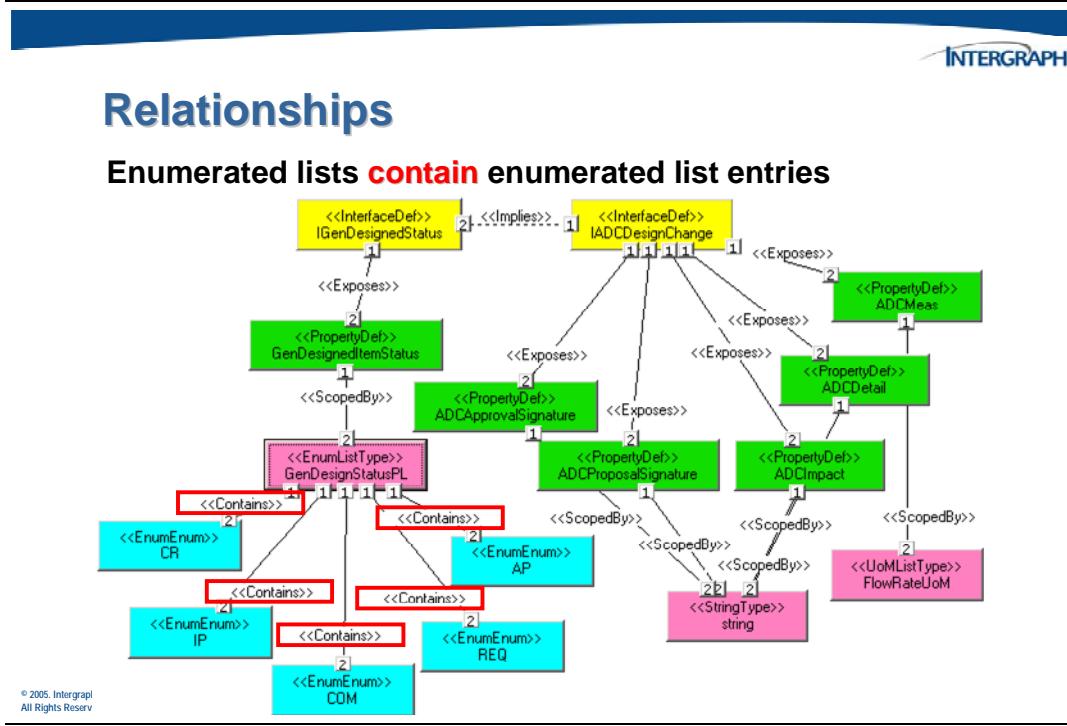
Relationships

A **ScopedBy** relationship definition specifies the property type definition that defines acceptable values for, or *scopes*, a particular property definition



1.4.3 Contains Relationship

Enumerated lists contain enumerated list entries. These entries identify the possible set of values for a property definition.

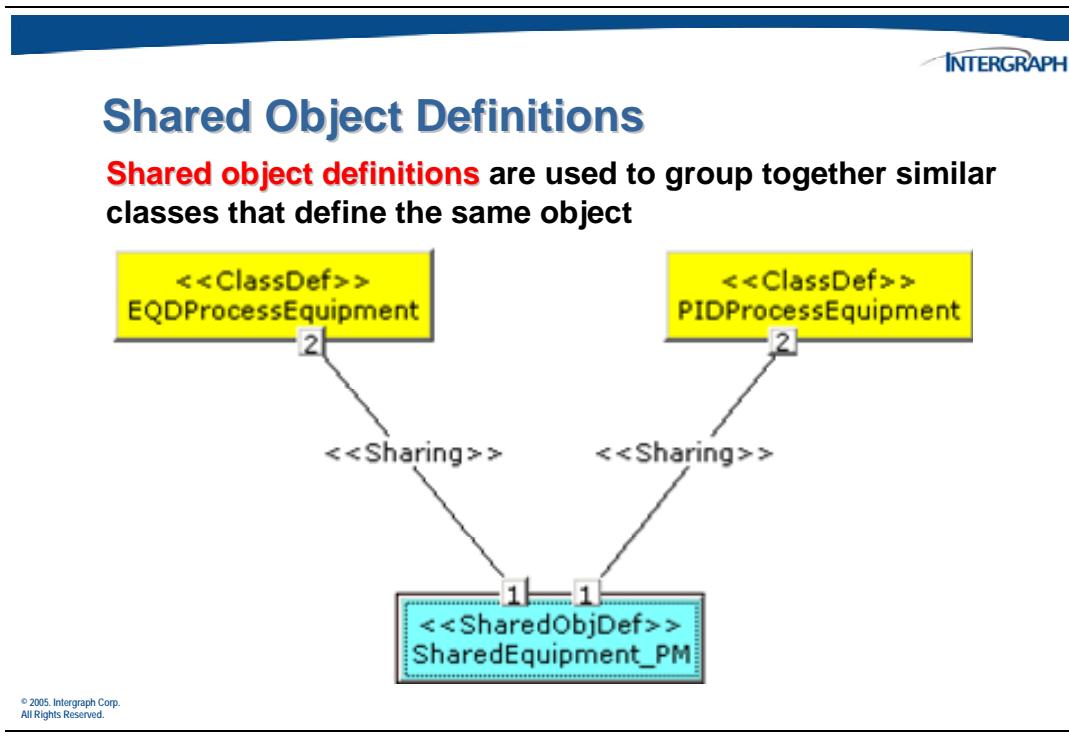


For example, the **GenDesignedItemStatus** enumerated list contains the following entries:

CR
IP
COM
REQ
AP

1.5 Shared Object Definitions

Shared object definitions are used to group together similar classes that define the same object in different domains. For example, equipment defined in a PFD has a class definition called *EQDProcessEquipment*, while equipment defined in P&ID has a class definition called *PIDProcessEquipment* therefore the object definitions are shared.



1.6 SPF Databases

SmartPlant Foundation uses two databases to store necessary information.



SPF Databases

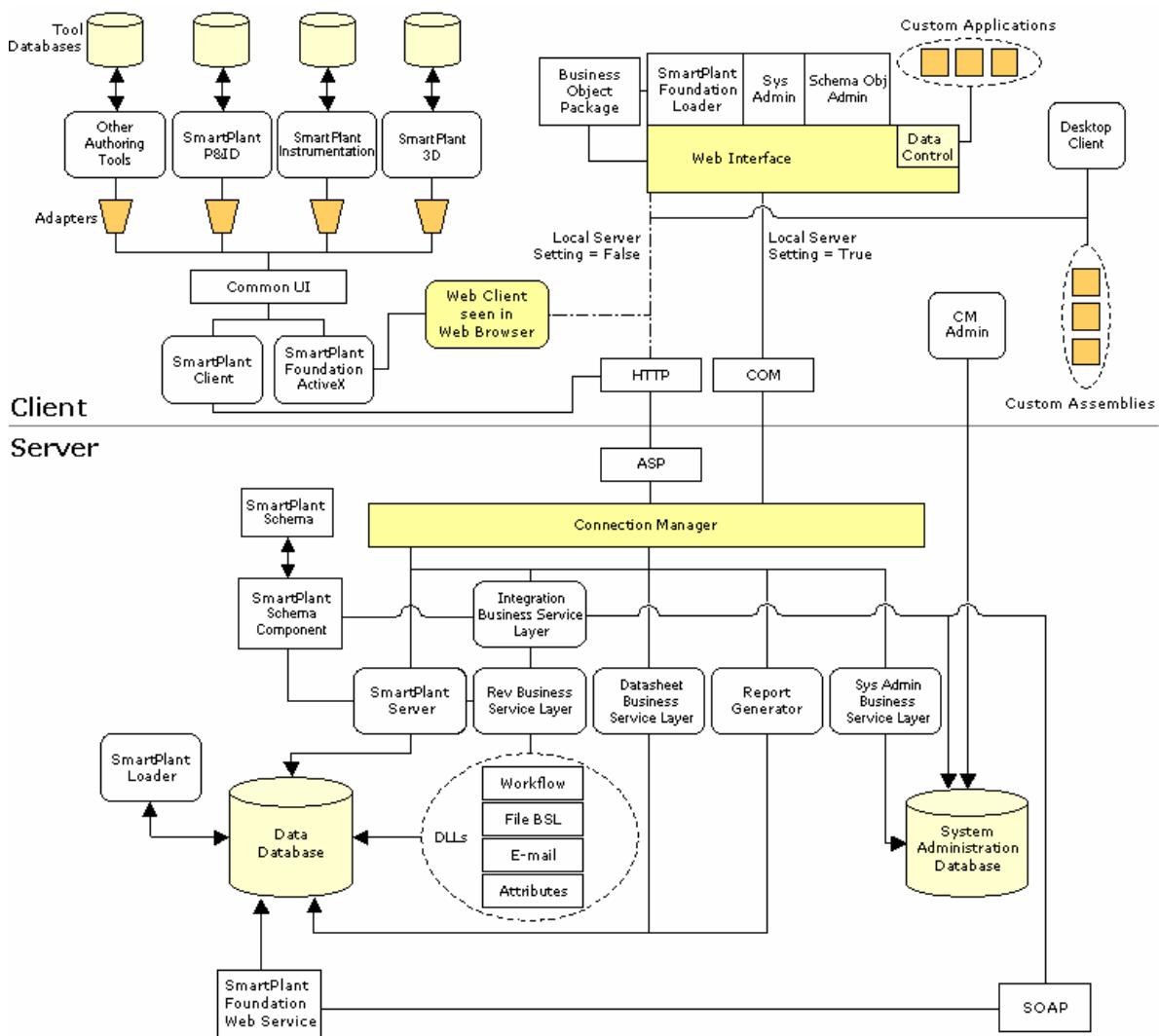
SPF has two databases

- Database for the actual data**
 - Can be populated interactively, by using SPF Loader or through application interfaces
- Administration database for system configuration**
 - Is maintained by using System Administration, Schema Object Administration, Change Management Administration, Form Builder, SPF Loader or Schema Loader

1.7 SmartPlant Foundation Structure

SmartPlant Foundation is a single data source ensuring data accuracy, integrity, and security; improving accessibility and control of work processes; and making plant data readily available to everyone in the enterprise and beyond. All plant/project information is consolidated into one consistent, managed data repository.

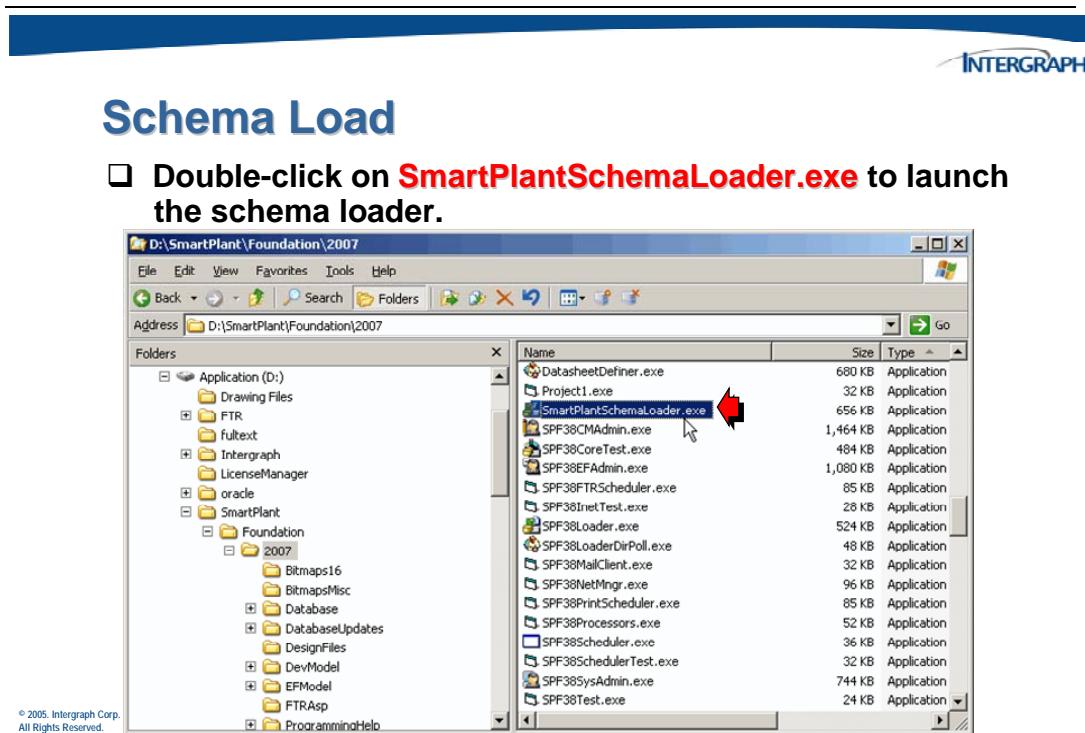
In SPF data objects are not deleted when updated in the database. They are terminated so the data stays in the database so a history for the data can be viewed.



1.8 Schema Load Process

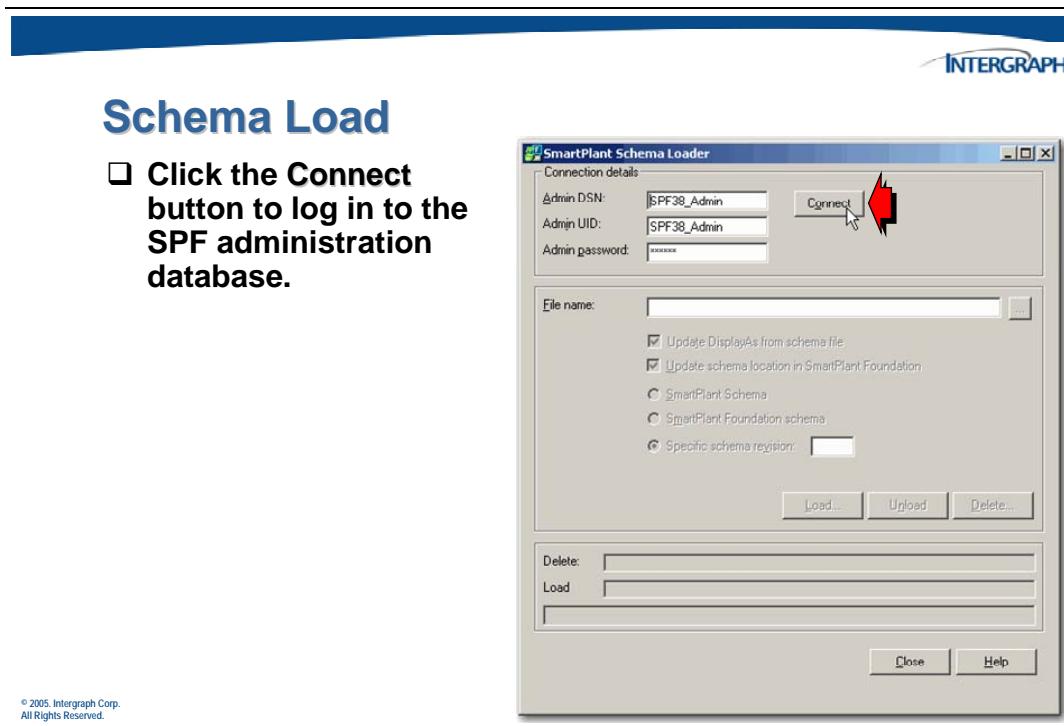
Use the SmartPlant Schema Loader to load the custom data model that was defined by using the SmartPlant Schema Editor into the SmartPlant Foundation administration database.

To launch the SmartPlant Schema Loader you can double-click **SmartPlantSchemaLoader.exe** or you can open a *Command Prompt* window and **cd** to the location of the executable.

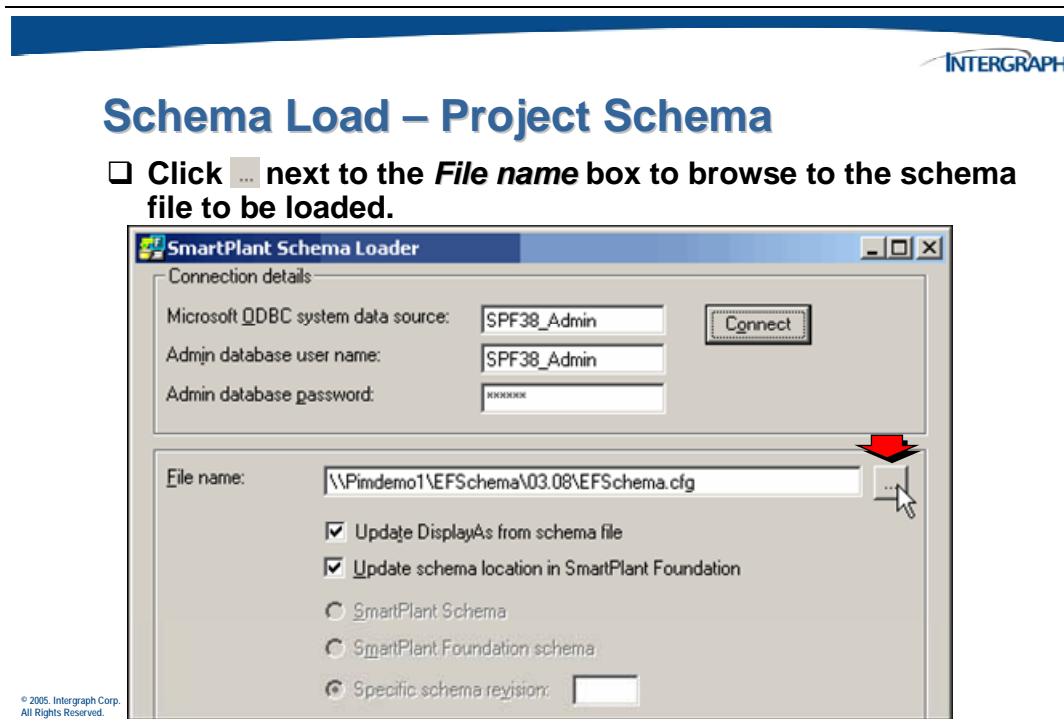


Note: Never load a component schema as the SmartPlant schema. Doing so will cause everything else to be deleted.

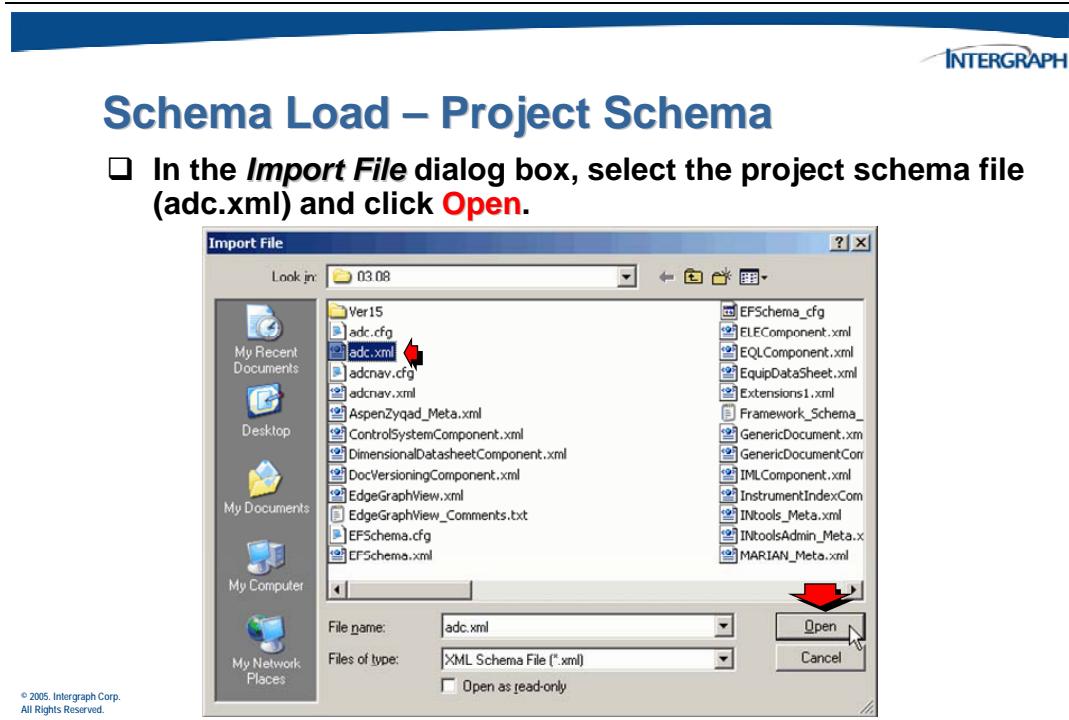
The *SmartPlant Schema Loader* window appears.



The software automatically uses the data source name (DSN), database user ID, and database password that you entered for the system administration database during SmartPlant Foundation installation.

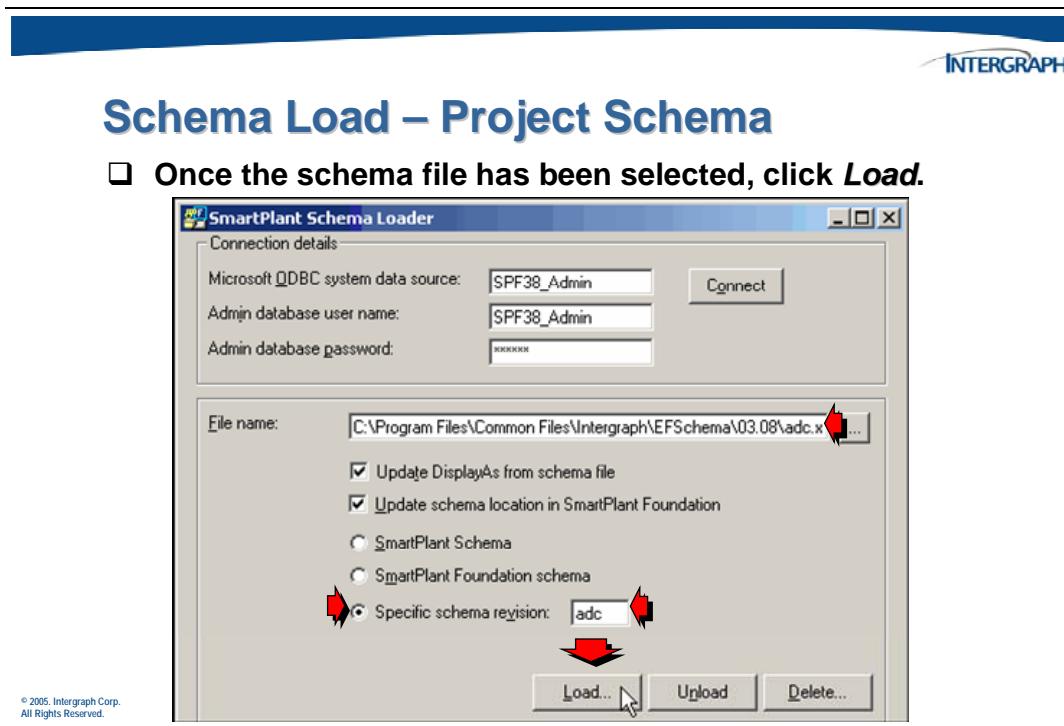


The default installation location for the SmartPlant schema files is *C:\Program Files\Common Files\Intergraph\EFSchema\03.08*.

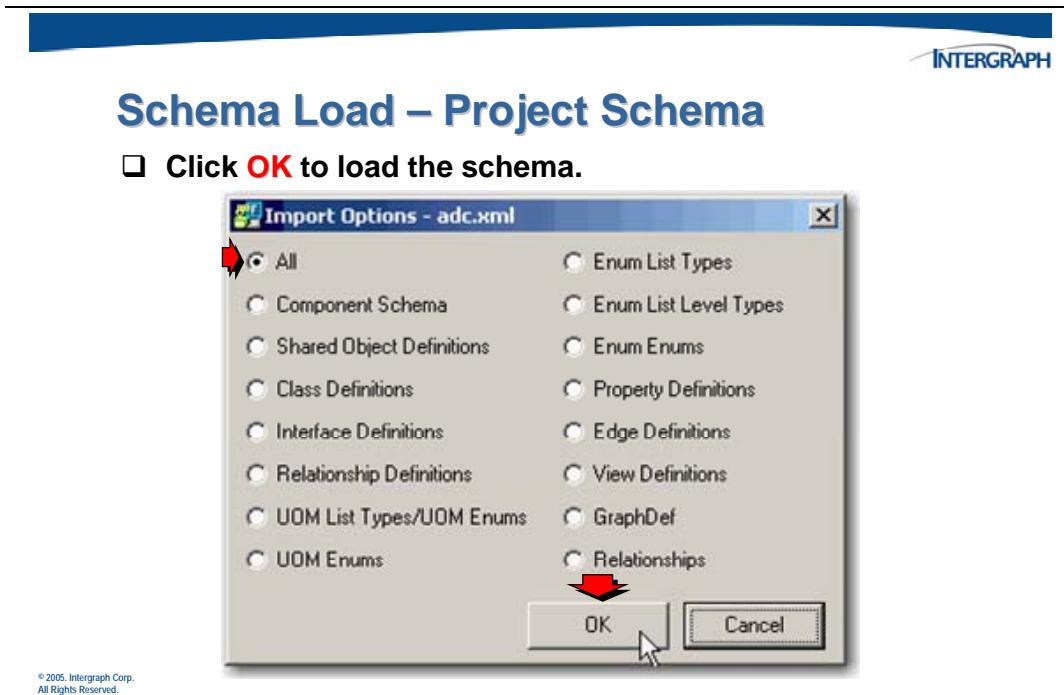


The file name to choose for the first load is **adc.xml**. This is the project schema that was created in the SmartPlant Mapping and Modeling training course.

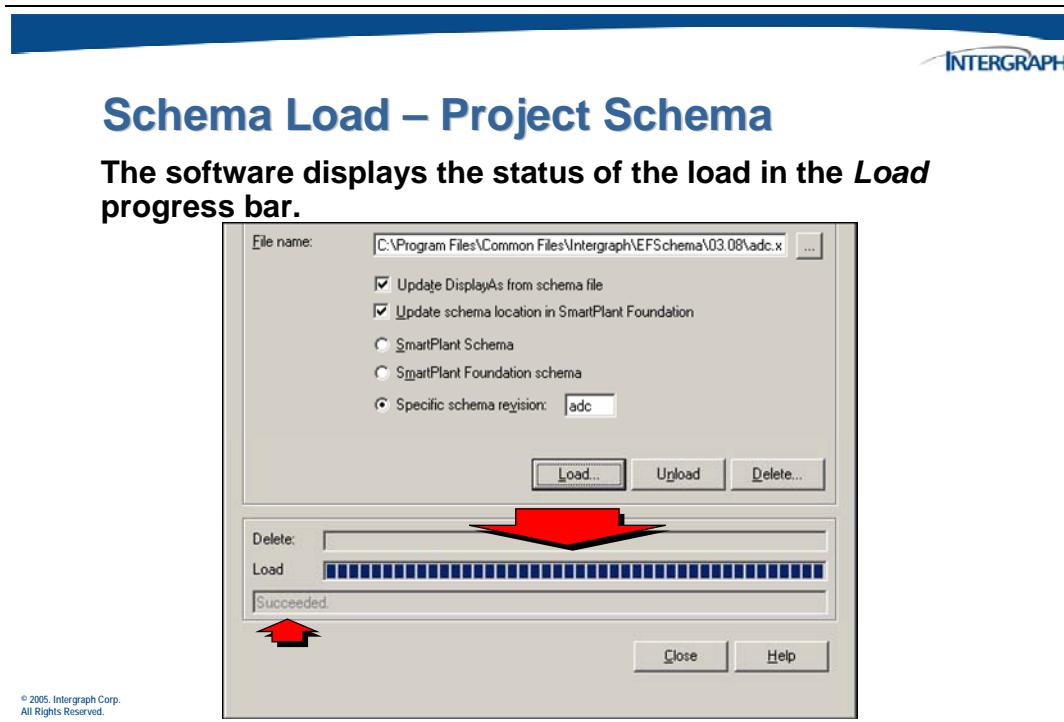
Enable the **Specific schema revision** radio button and enter a revision designator.



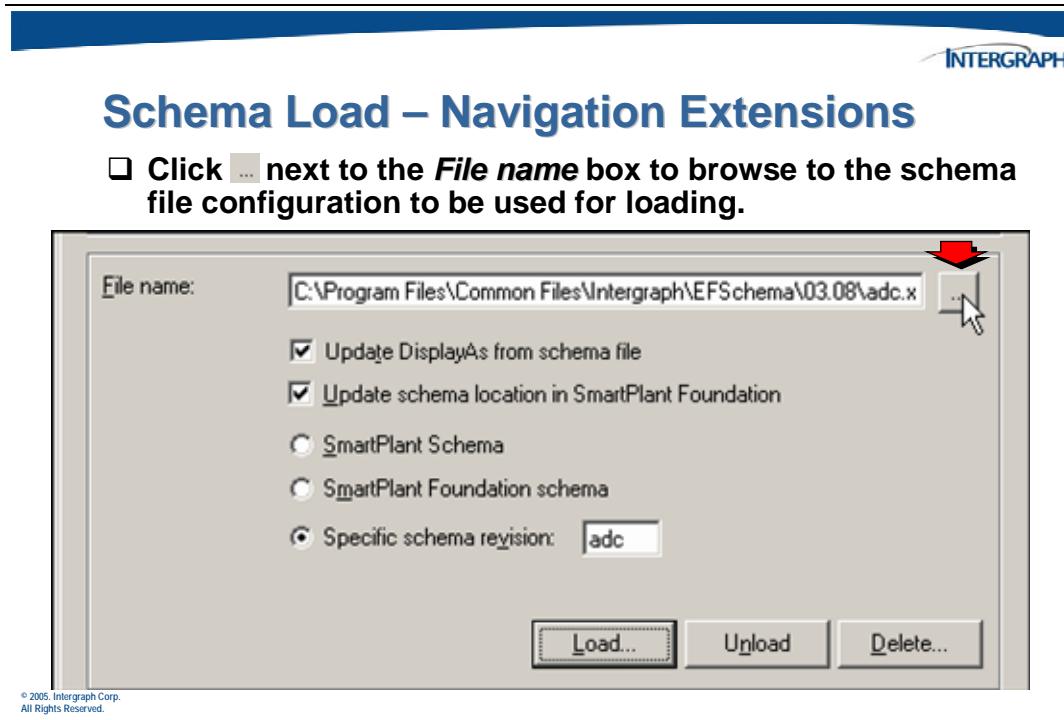
In the **Import Options** dialog box, select the type of data that you want to load from the schema into the system administration database. By default, the software loads the entire SmartPlant schema. However, you can limit the data that you load by selecting another option in the **Import Options** dialog box.



The load software will load the custom project schema into the SPF Admin database.



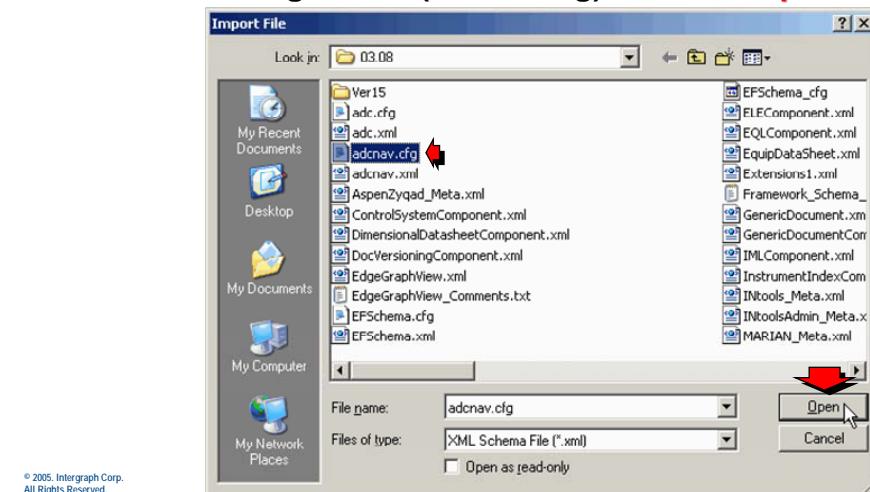
The next step is to load the navigation schema . Repeat the schema load procedure to accomplish this.





Schema Load – Navigation Extensions

- In the *Import File* dialog box, select the navigation schema file configuration (adcnav.cfg) and click **Open**.

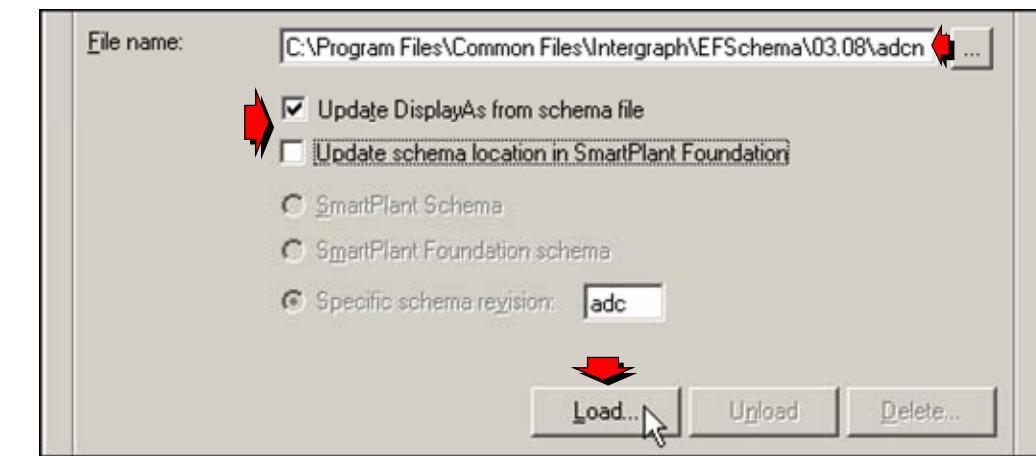


© 2005, Intergraph Corp.
All Rights Reserved.



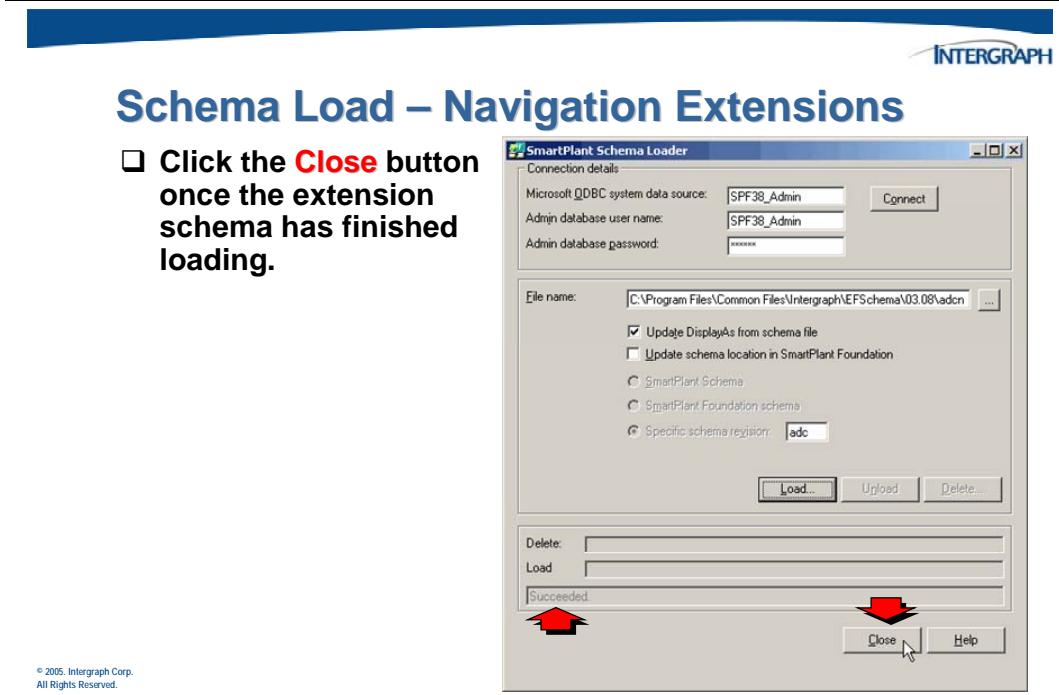
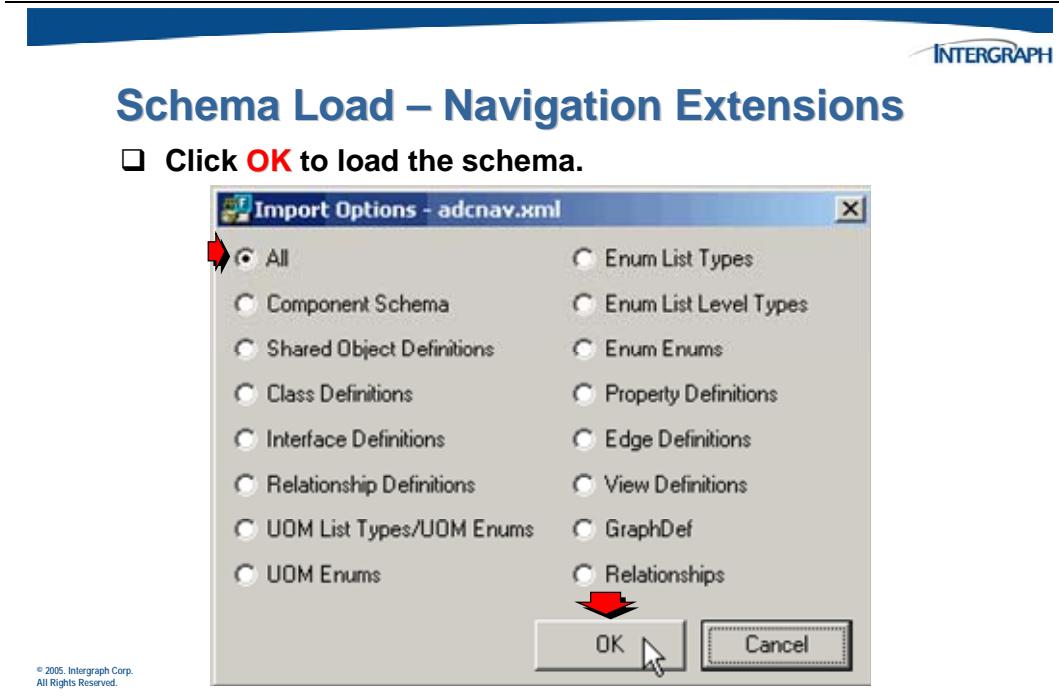
Schema Load – Navigation Extensions

- Once the schema configuration has been selected, click **Load**.



© 2005, Intergraph Corp.
All Rights Reserved.

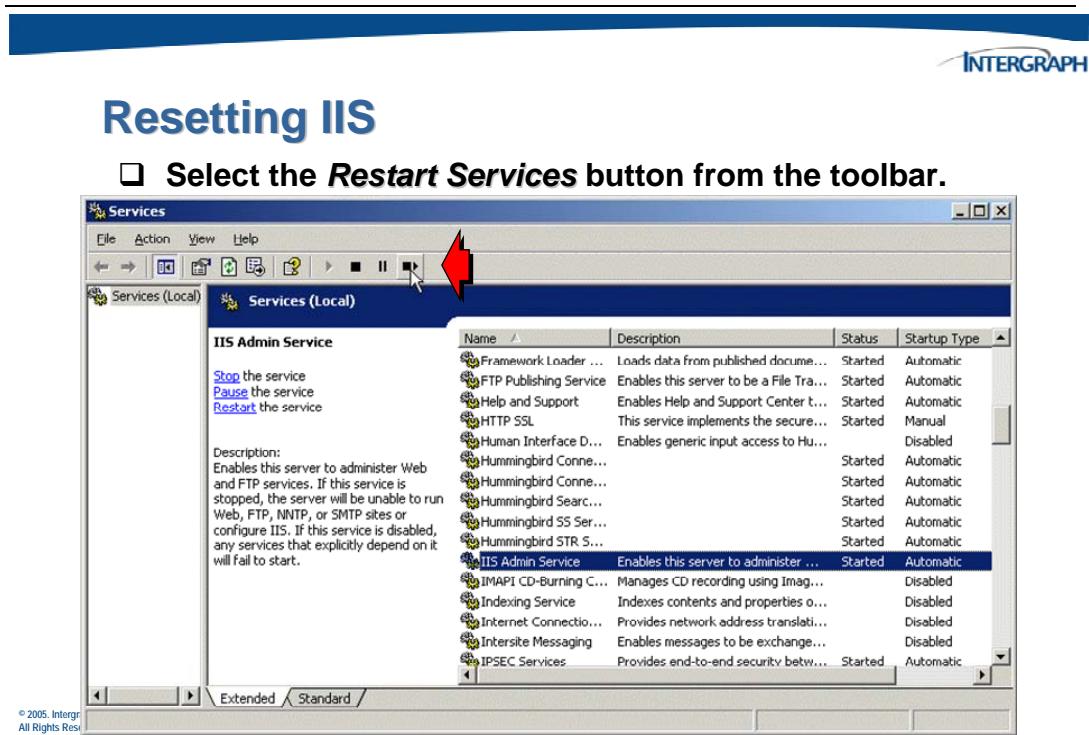
In the **Import Options** dialog box, select the type of data that you want to load from the schema into the system administration database.



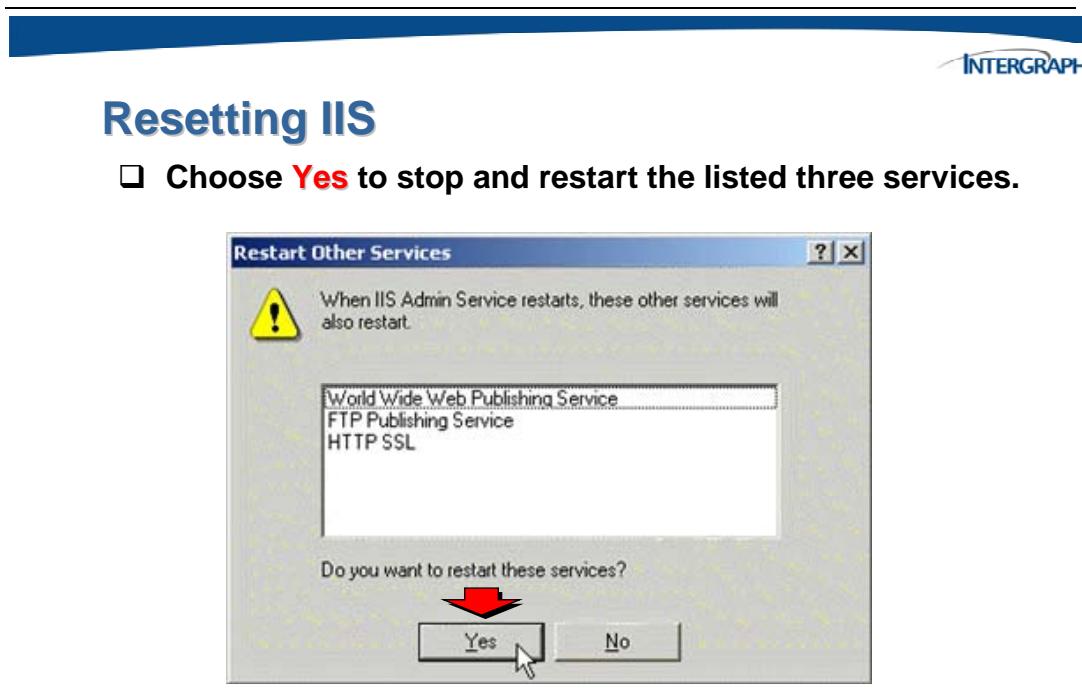
This will load the custom navigation objects into the SPF Admin data base.

1.9 Resetting IIS

Any changes to the SPF model will require that the IIS services be stopped and restarted. This is in order for the IIS information to be re-cached. There are two ways to accomplish this. The first method is to open a *Services* window.

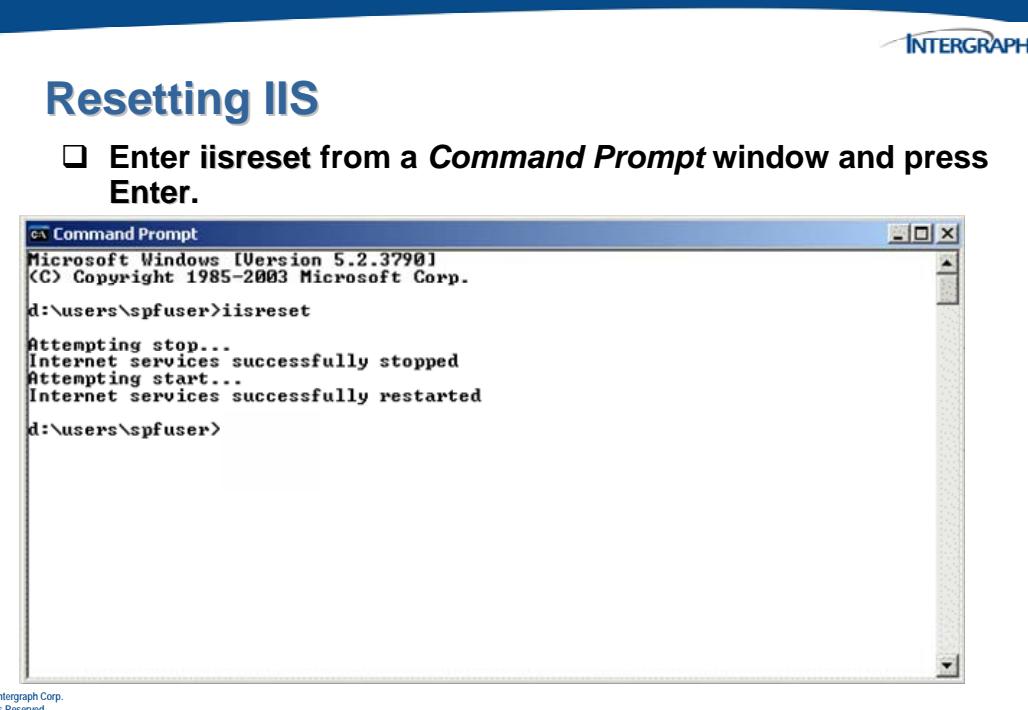


The Restart Other Services dialog will display containing the three services that need to be reset, *World Wide Web Publishing Service*, *FTP Publishing Service*, and *HTTP SSL*.



This allows you to stop and restart all three services with one operation instead of having to stop and start them individually.

The other method for stopping and starting the IIS services is by using a Command Prompt window and the *iisreset* command line entry.



Any time a model change is made (with the exception of form changes), the IIS services will need to be stopped and restarted (reset) and the changed model information re-cached.

1.10 Activity – Loading the Modified Schema’s into SPF

The objective of this activity is to use the Schema Loader to load the schemas created in the SmartPlant Modeling and Mapping course into the SPF Admin database.

Before starting this activity, copy the **adc.xml**, **adc.cfg**, **adcnnav.xml** and the **adcnnav.cfg** files from the **D:\SPF_Training\Saved_Schemas** folder to **C:\Program Files\Common Files\Intergraph\EF Schema\03.08** folder. This is the location of all schema and component schema xml files.

Load the Modified Project Schema file into SPF

1. Under the SPF root directory (D:\SmartPlant\Foundation\2007) double click the file **SmartPlantSchemaLoader.exe**.
2. Verify the login values are correct in the dialog.
3. Click the **Connect** button to login
4. Use the **File name** ellipses button to query for the file **adc.xml** provided for this lab. Enable the *Specific schema revision* radio button and enter **adc**.
5. Click the **Load** button to load the project schema data.
6. Select **ALL** in the option dialog which will appear and click **OK**.
7. Wait for the schema load to complete successfully. The **Load** status message will show *Succeeded*.

Load the Navigation Schema file into SPF

8. Next, use the **File name** ellipses button to query for the file **adcnnav.cfg** provided for this lab.
9. Click the **Load** button to load the navigation schema data.
10. Select **ALL** in the option dialog which will appear and click **OK**.
11. Wait for the navigation schema load to complete successfully. The **Load** status message will show *Succeeded*.
12. Once the schemas have been loaded, **Close** the *SmartPlant Schema Loader* and reset the IIS services to reload the cached model information.

2

C H A P T E R

Schema Object and System Administration

2. Schema Object Administration

To work with the SmartPlant schema in SmartPlant Foundation, you must understand the SmartPlant model. The model used in conjunction with the SmartPlant schema is a class/interface/relationship model. The Schema Object Administration module is used by system administrators to perform the necessary tasks to create and maintain needed definitions and security access for SmartPlant objects.

In this chapter, functions for viewing and modifying class, interface, and property definitions and enumerated lists using the Schema Object Administration utility will be covered. These operations will build on the knowledge already covered in the Introduction to SmartPlant Foundation and Administration I course and the SPF Modeling and Mapping course.



Schema Object Administration

The topics to be covered in discussing the Schema Object Administration and also some additions to SPF System Administration are as follows:

- Class Definitions**
- Interface Definitions**
- Property Definitions**
- Enumerated Lists**
- Component Schema**
- Edge definitions**
- Edge definition security (filter edge in display)**

Also the System Administration topics of *Access Control*, *Email Configuration* and *Vault Replication* will be covered in this chapter.



Schema Object Administration

Schema Object Administration topics (continued):

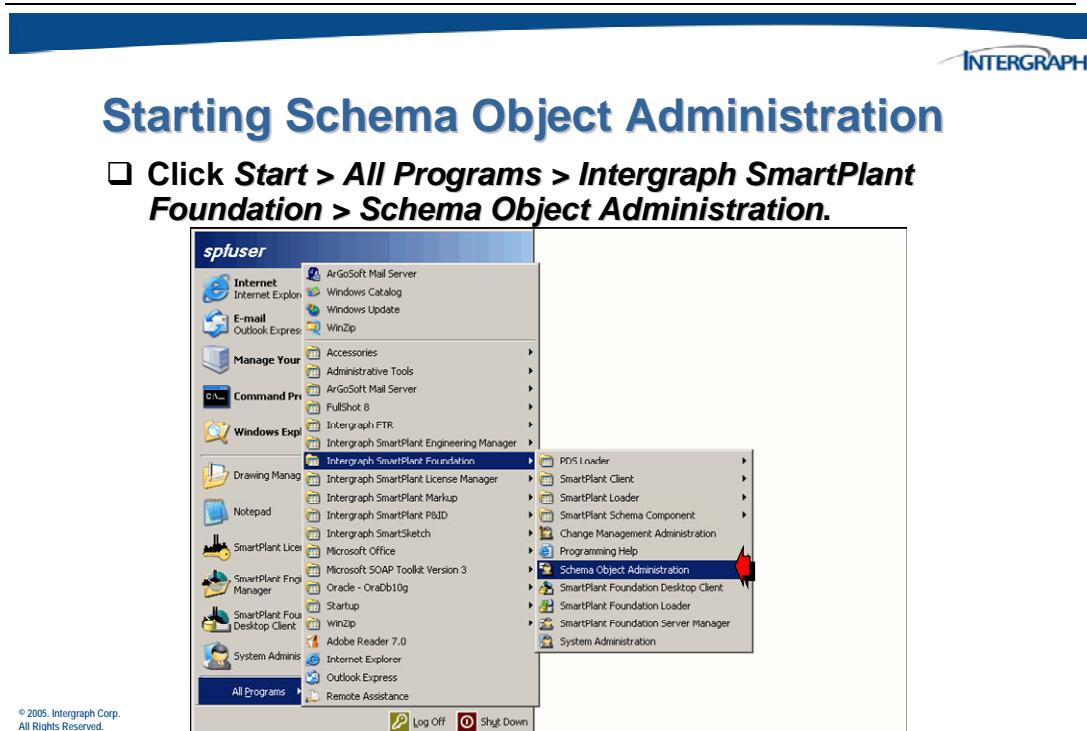
- Graph definitions**
- View definitions**
- Class view maps (security)**
- RelDef Access Security (filter edge in display)**

System Administration topics to be covered

- Access Control**
- Configuring Email and Email Digest**
- Vault Replication**

2.1 Starting Schema Object Administration

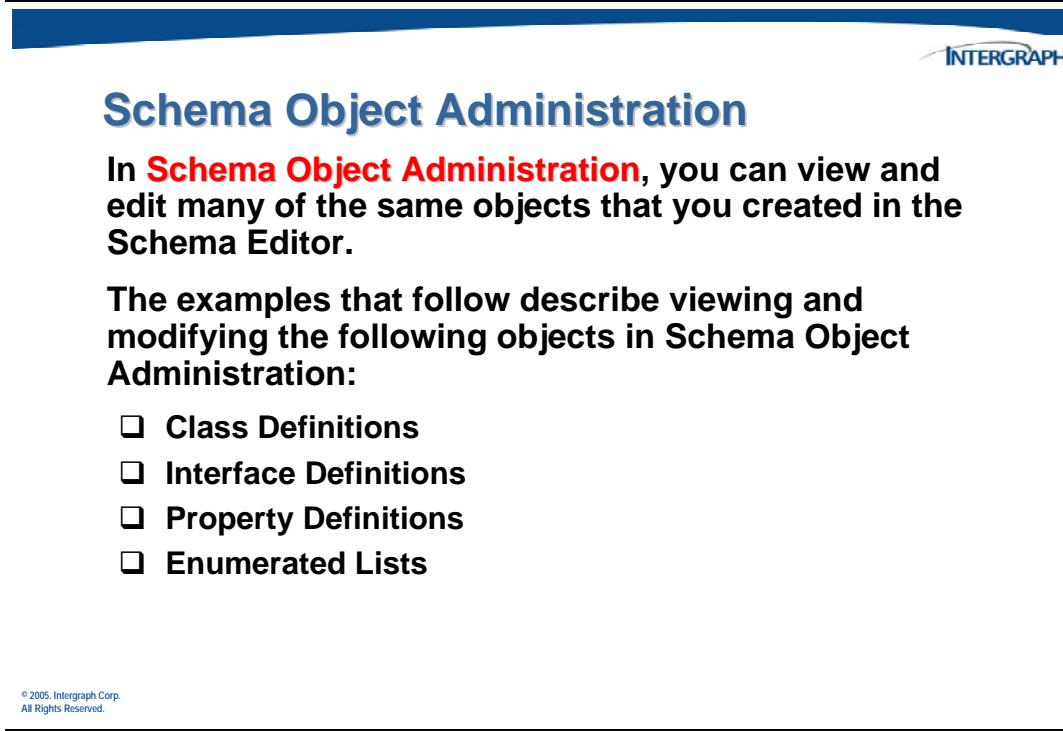
You can use **Schema Object Administration** to view and modify *Class Definitions*, *Interface Definitions*, *Property Definitions* and *Enumerated Lists* in the SPF administration database. This utility can also be used to manipulate *View Definitions*, *Graph Definitions*, and *Edge Definitions* in SPF. You can also attach workflows and methods to interface definitions and convert class definitions in the SmartPlant schema into SPF forms. These functions will be covered in later chapters.



The Schema Object Administration *Logon Information* dialog box appears.



Enter your user name in the **User name** box and your password in the **Password** box in the *Logon Information* dialog box.



2.2 Class Definitions

A class definition is a named description of a set of objects that support or realize the same interface definitions and share the same property definitions and relationships. In the schema, class definitions can represent physical things, such as pumps, or conceptual things, such as projects.



Class Definitions

Class definitions are related to interface definitions, which provides a set of properties for the class.

Importing class definitions to SPF adds the following properties:

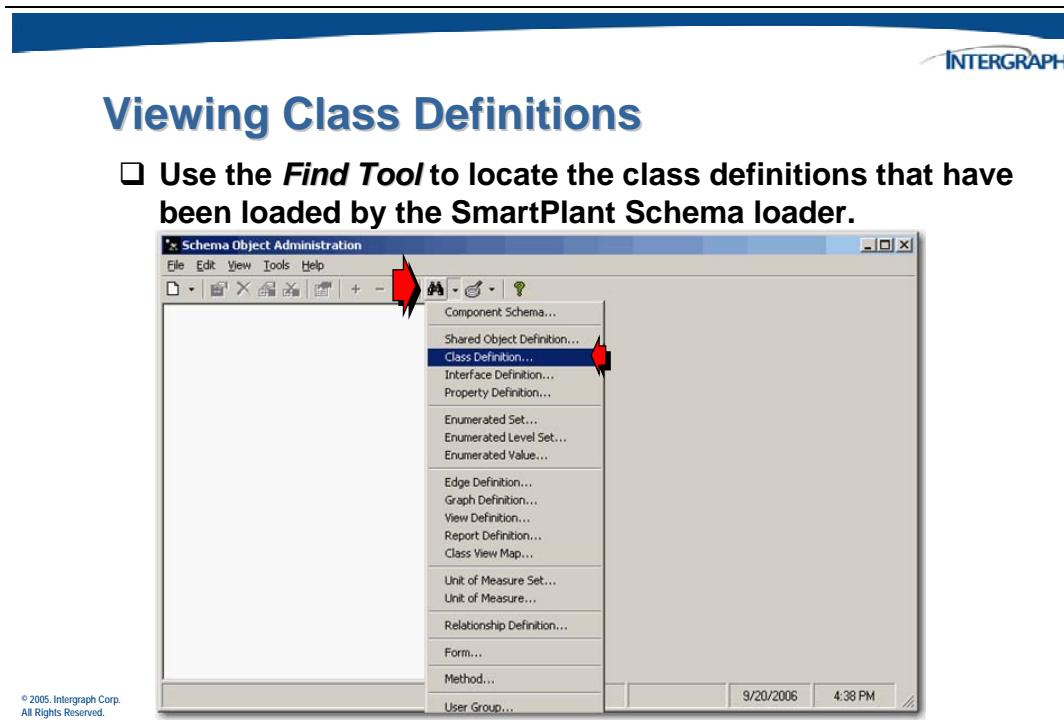
- Configuration Dependency** - ensure an object of this class must be created with at least this level of configuration set
- UniqueKey** - Overrides the pattern normally defined on the EFOObject business object itself

Class Definitions are comprised of Property Definitions and can share Property Definitions, but not the data associated with the properties. Interface Definitions expose the Property Definitions for class definitions by combining the Property Definitions in a named collection group called an interface definition.

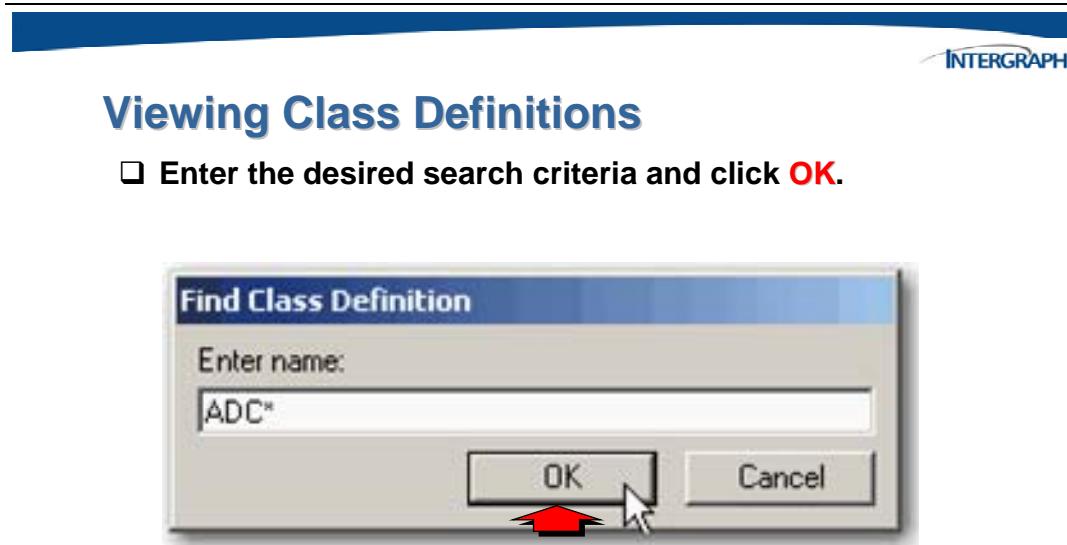
2.2.1 Viewing a Class Definition

The SmartPlant Schema Editor was used to define any necessary class definitions that are needed. Once the schema has been modified, the definitions are then loaded into the SPF administration database using the SmartPlant Schema Loader (SmartPlantSchemaLoader.exe).

You can use the Schema Object Administration utility to verify that the new definitions have been properly loaded into SPF.

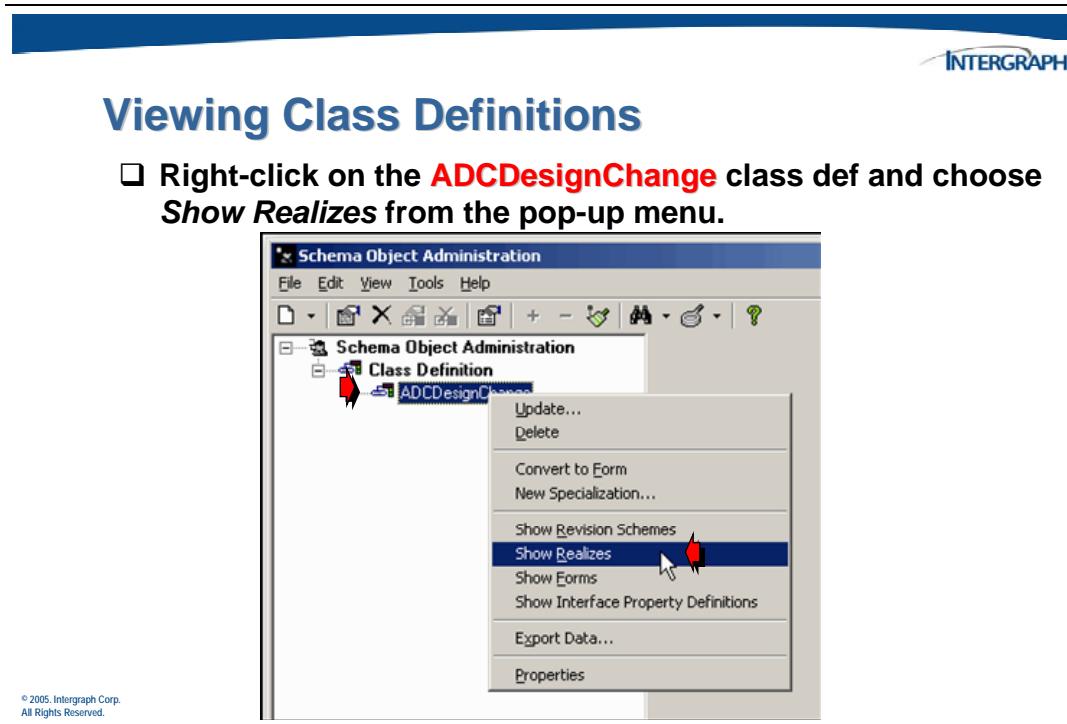


A *Find Class Definition* search dialog will display.



© 2005, Intergraph Corp.
All Rights Reserved.

The related *Interface Definitions* can also be viewed in the tree.

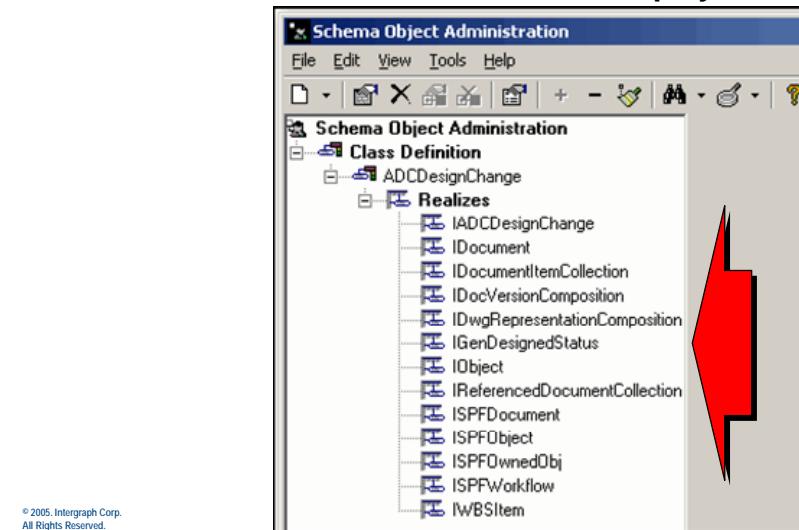


© 2005, Intergraph Corp.
All Rights Reserved.



Viewing Class Definitions

The realized InterfaceDefs will be displayed.



2.2.2 Update a Class Definition

Class definitions that have been loaded from the schema can be modified using Schema Object Administration. Keep in mind that only certain characteristics can be changed. Class definition names CAN NOT be updated.

When you load class definitions from the schema, SmartPlant Foundation adds the following properties:

- Configuration Dependency** - Used to ensure an object of this class must be created with at least this level of configuration set (normally, Plant or Project).
- Unique Key** - Keys provide information for the system to determine uniqueness of an object. The key overrides the pattern normally defined on the EFOBJECT object itself.

Unique key fields along with their values are shown in the examples below from Schema Object Administration.,

The screenshot shows a software interface titled "Class Definition Unique Key". It displays a table of class definitions with the following columns: Class Definition Name, Class Definition Description, M, Configuration Dependency, and Unique Key. The table lists numerous classes, many of which have "Project" as their Configuration Dependency and "CN,NAME" as their Unique Key. A red arrow points to the "Unique Key" column. The bottom left corner of the dialog box contains the copyright notice: "© 2005, Intergraph Corp. All Rights Reserved."

Class Definition Name	Class Definition Description	M	Configuration Dependency	Unique Key
ADCDesignChange	ACME Design Change Note	1		
AnalyzerBrowser		1		
AutoWiringRoutingTaskBr...		1		
BlockBrowser		1		
CableBrowser		1		
CableScheduleInstallation...		1		
CableScheduleInstallation...		1		
CableSetBrowser		1		
CalibrationResultsBrowser		1		
CalibrationSettingsBrowser		1		
ChangesLogBrowser		1		
CircuitRelatedElectricalTa...	Circuit-related electrical tags browser	1		
Contract		1	Project	CN,NAME
ControlSysIO	Control System I/O	1		
ControlSystemTagBrowser		1		
ControlValveBrowser		1		
ConventionalTagBrowser		1		
CS Cabinet Rack		1		
CS Controller		1		
CS Control SysIO		1		
CS DCSPanel		1		
CS Document		1		
CS Fieldbus IO Card		1		
CS Fieldbus Device		1		

INTERGRAPH

Class Definition Unique Key

Class Definition Name	Class Definition Description	M	Configuration Dependency	Unique Key
• DIMInstrument	Dimensional Instrument	1		
• DIMPipingPort	Dimensional Instrument Inlet/Outlet/Pr...	1		
• Document		1		DM,NAME
• DocumentRevision		1		
• DocumentVersion	Document Version	1		DV,NAME
• DrawingBrowser		1		
• DrawingDocument	Drawing Document	1		
• DrawingItem_F	Drawing Item on Facility Model	1		
• DrawingItem_PM	Drawing Item for Planned Material model	1		
• DrawingRepresentation	Drawing Representation	1		
• DrawingSummaryBrowser		1		
• ELEAutoTransformer		1		
• ELEBatteryBank		1		
• ELEBatteryCharger		1		
• ELECable		1		
• ELECablePlug		1		
• ELECableSchedule	Cable Schedule	1		
• ELECapacitor	Capacitor	1		
• ELECircuit		1		
• ELECircuitBreaker		1		
• ELEContactor		1		
• ELEControlSystIO		1		
• ElectricalPowerElementBr...		1		
• ElectricalTagBrowser		1		

© 2005, Intergraph Corp.
All Rights Reserved.

INTERGRAPH

Class Definition Unique Key

Class Definition Name	Class Definition Description	M	Configuration Dependency	Unique Key
• EQDTubularReactor		1		
• EQDTurbine		1		ET,Name
• EQDVacuumPump		1		
• EQDVapor		1		
• EQDVerticalCentrifugalPu...		1		
• EQDWeighingEquipment		1		
• EQLDocument	Equipment List Document	1		
• EQLEquipment	Equipment List	1		
• EquipmentBrowser		1		
• FieldbusTagBrowser		1		
• FieldbusTagNumberListBr...		1		
• File	A File	1		
• FlowInstrumentBrowser		1		
• FunctionalArea	Plant Functional Area	1		FA,Plant,NAME
• FunctionalUnit	Plant Functional Unit	1		FU,Plant,FunctionalArea,NAME
• FunctionRequirement		1		
• GeneralPanelBrowser		1		
• GeneralProcessDataBrow...		1		
• HookUpBrowser		1		
• HubCabinetBrowser		1		
• HubEquipmentBrowser		1		
• IMLDocument		1		
• IMInstrument	Instrument List	1		
• INDXDocument	Instrument Index	1		

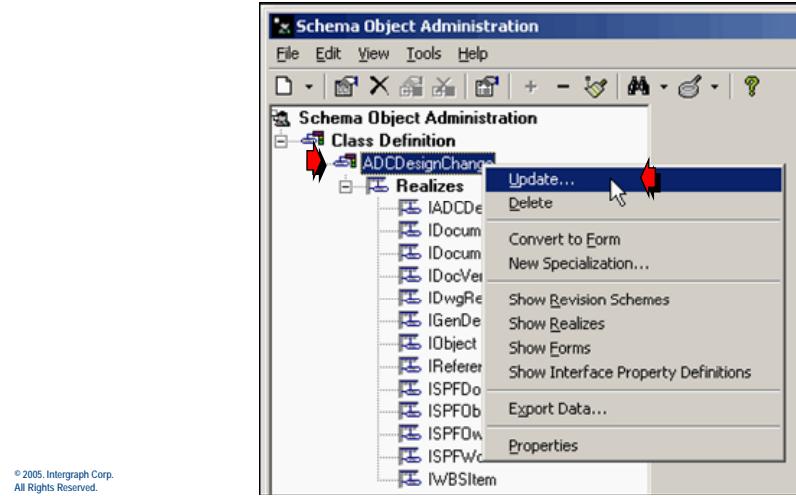
© 2005, Intergraph Corp.
All Rights Reserved.

The **Unique Key** value does not get loaded by the Schema Loader and must be added by updating the ClassDef information.



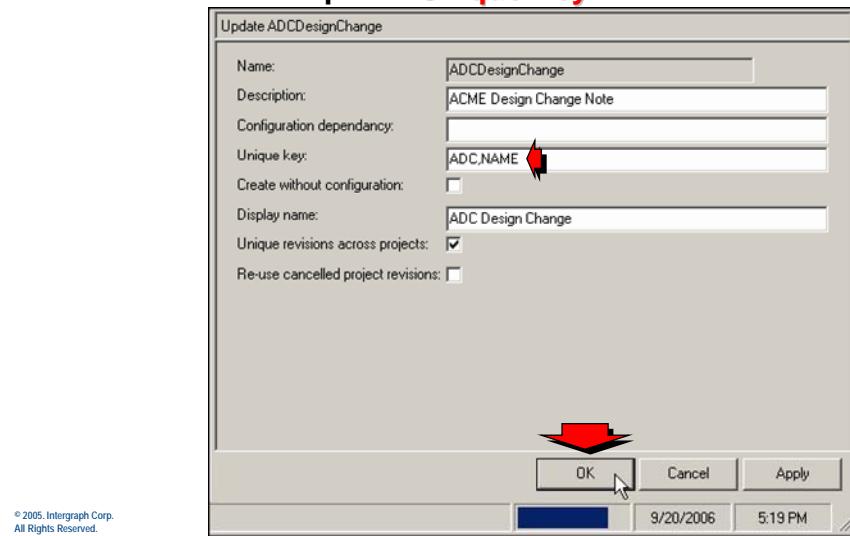
Updating Class Definitions

- ❑ Right-click on the **ADCDesignChange** class def and choose *Update* from the pop-up menu.



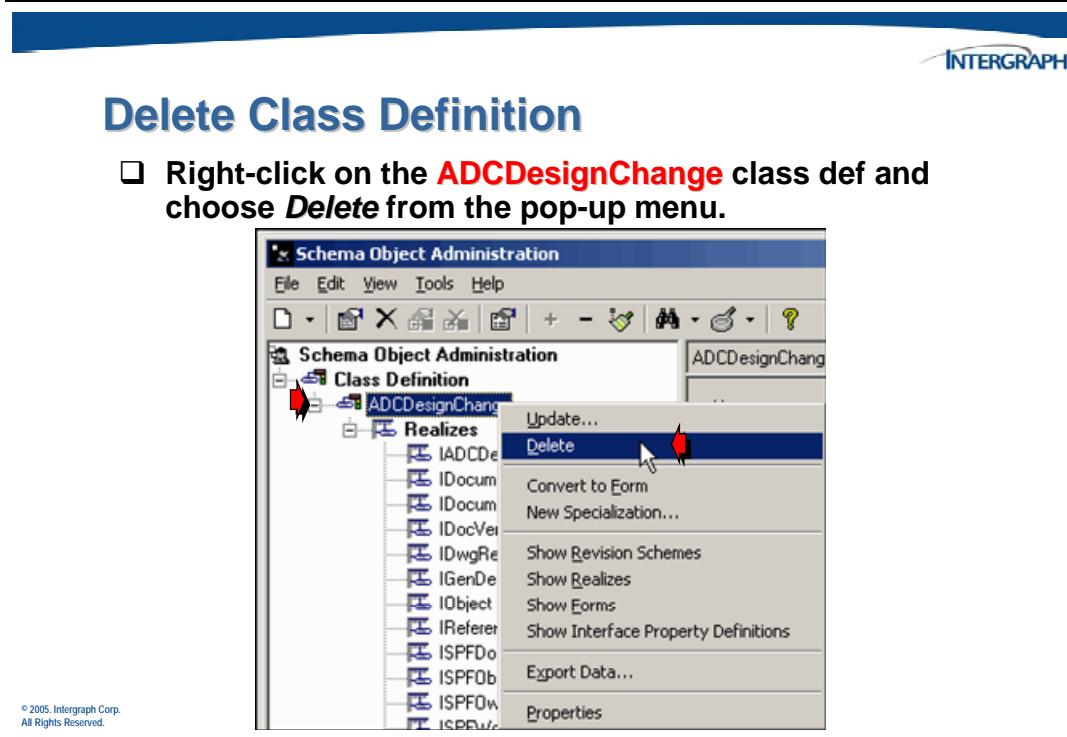
Updating Class Definitions

- ❑ Add the required **Unique Key** information to this class.

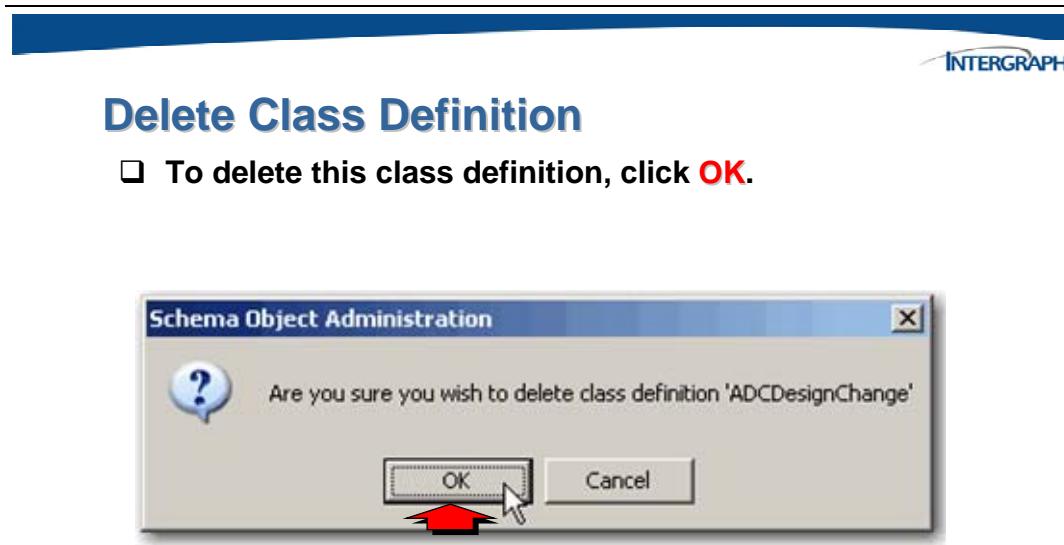


2.2.3 Delete a Class Definition

Class Definitions can be deleted from the administration database but extreme caution must be taken into consideration. The delete operation does not remove the class definition information from the schema file.

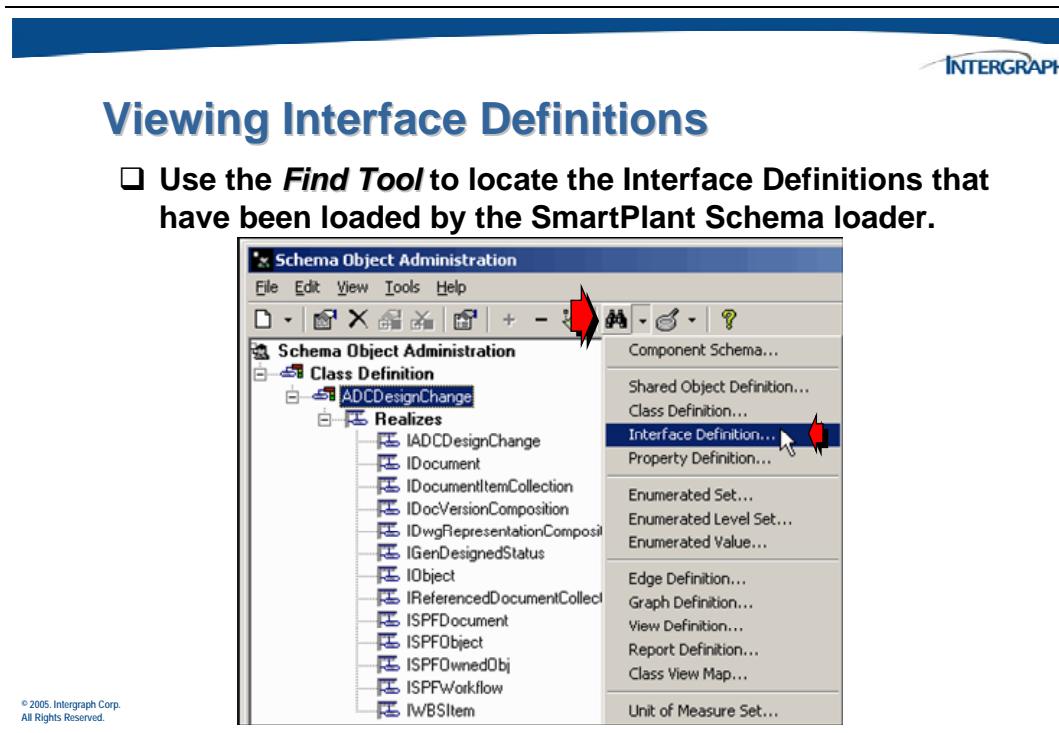


A delete confirmation dialog will display.



2.3 Interface Definitions

An **Interface Definition** (InterfaceDef) is the most important concept in the meta schema. An interface definition is a named collection of property definitions. Interface definitions expose the property definitions for class definitions. Every interface definition is realized by one or more class definitions. By sharing specific interface definitions, class definitions can also share property definitions, but not the data associated with the properties. Just as in the real world, the role defines both the properties and relationships of an object. Some interface definitions are defined to carry properties. Some are defined to carry relationships. Others may be defined merely to indicate a role.



2.3.1 Viewing an Interface Definition

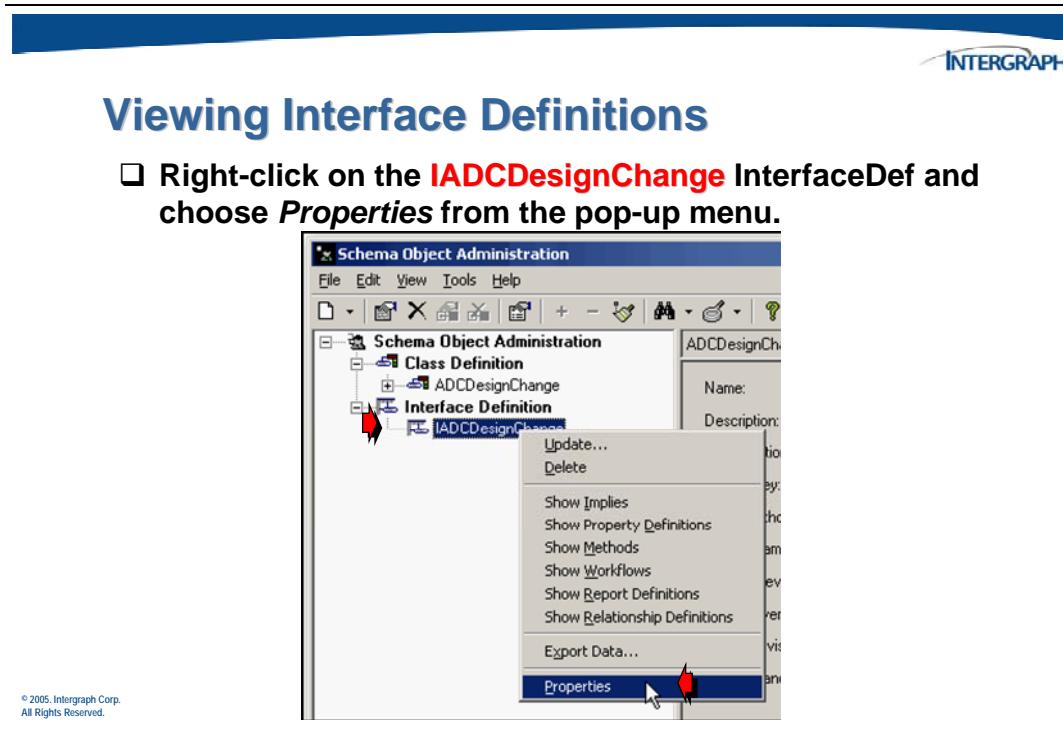
Use the *Find* dialog to search for the interface definitions that have been loaded.

Viewing Interface Definitions

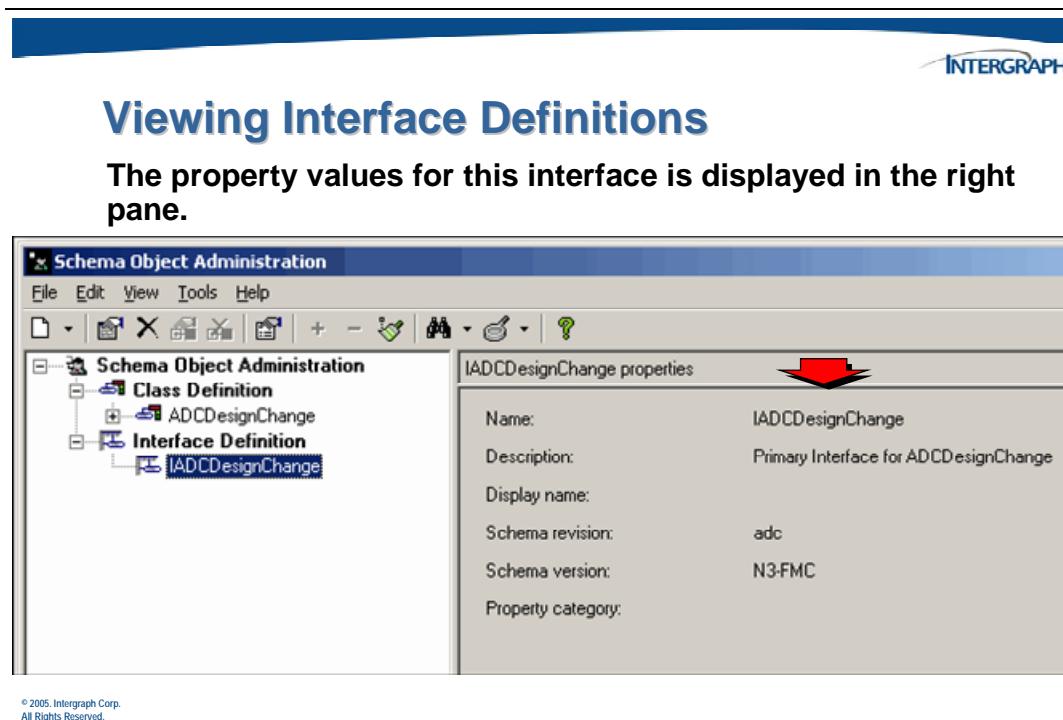
- Enter the desired search criteria and click **OK**.



The resulting *Interface Definitions* are shown in the tree view.

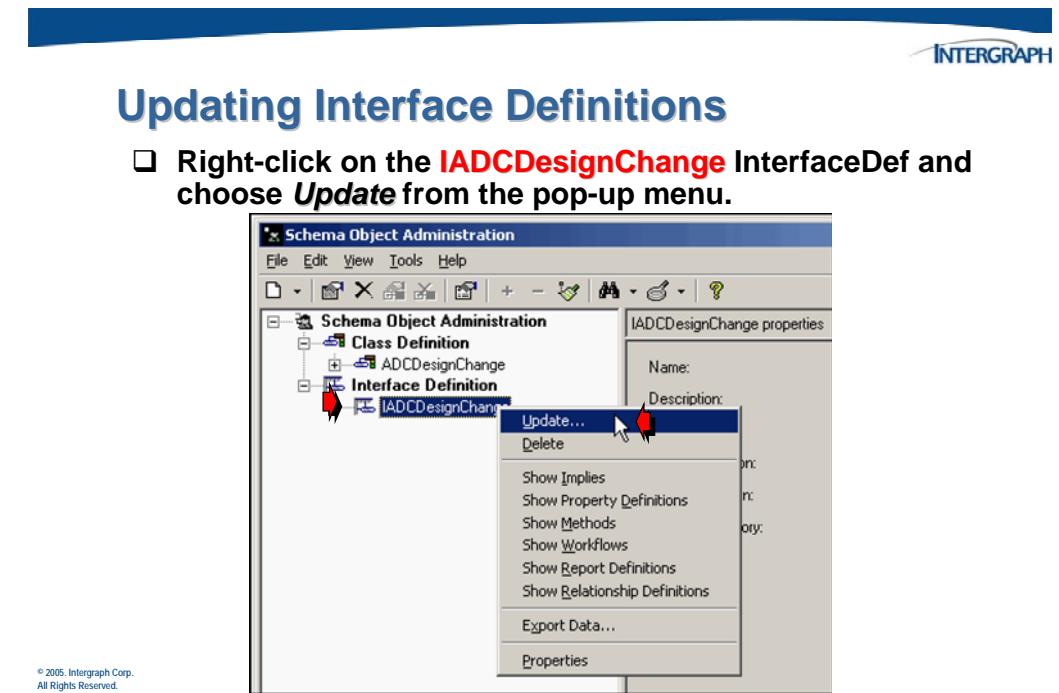


View the properties of any of the interface definitions in the right pane.

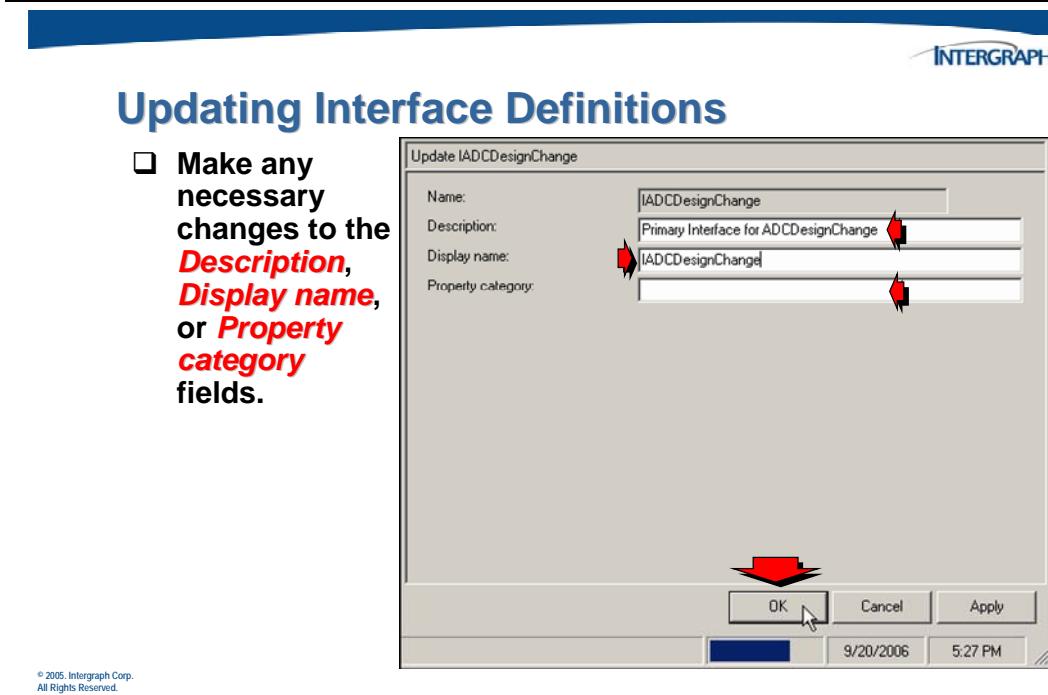


2.3.2 Update an Interface Definition

Like class definitions, certain properties of the interface definition can be changed.

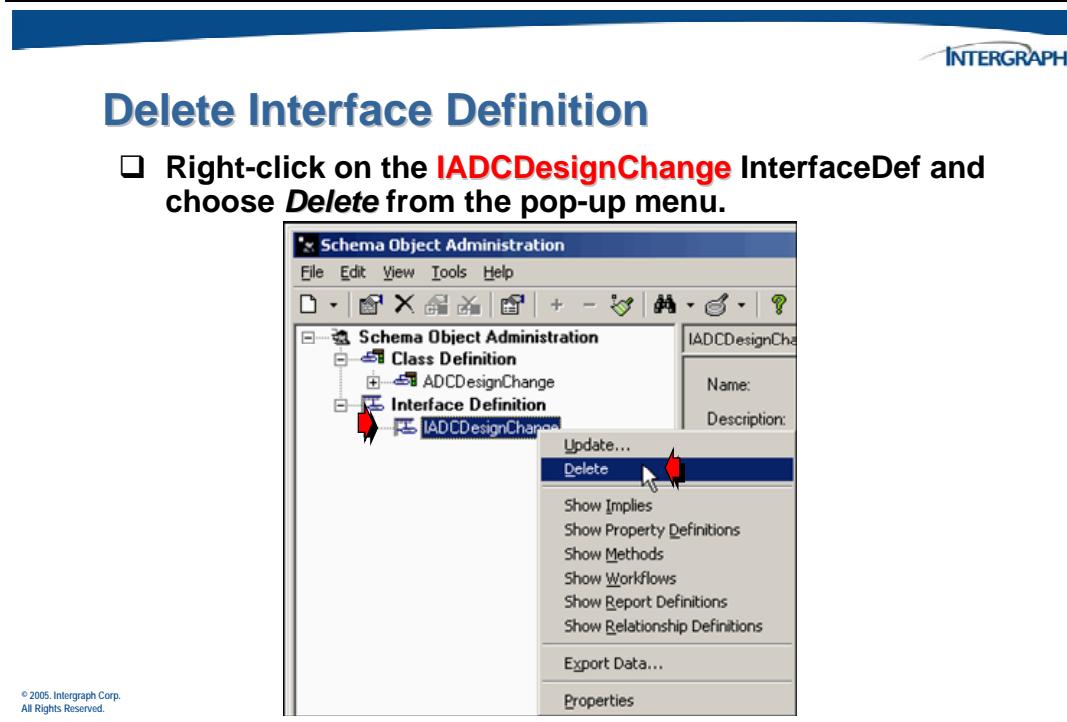


The *Update* form will display in the right pane.

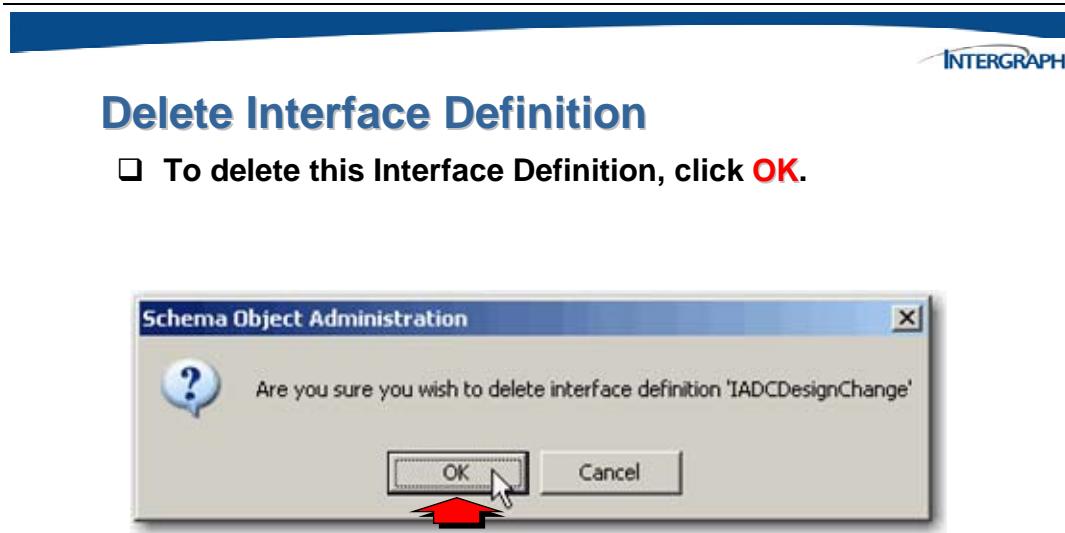


2.3.3 Delete an Interface Definition

Like class definitions, interface definitions can be deleted from the administration database but extreme caution must be taken into consideration. The delete operation does not remove the interface definition information from the schema file.



A delete confirmation dialog will display.



2.4 Property Definitions

Property definitions are components of class definitions and are exposed to the SmartPlant Foundation interface through interface definitions.



Property Definitions

A Property Definition is a characteristic used to describe a class.

Property definitions for class definitions in SmartPlant Foundation contain some additional properties that are not part of the schema.

Property information can be used to determine the property type.

© 2005, Intergraph Corp.
All Rights Reserved.

All property definitions for an object are exposed through its interface definitions and never directly by the object. The property definitions that apply to a particular interface definition are defined by the **Exposees** relationship between objects of type **InterfaceDef** and objects of type **PropertyDef**. A particular property definition for a given class definition is typically exposed by one, and only one, interface definition.

For example, the **IObject** interface definition exposes the following property definitions:

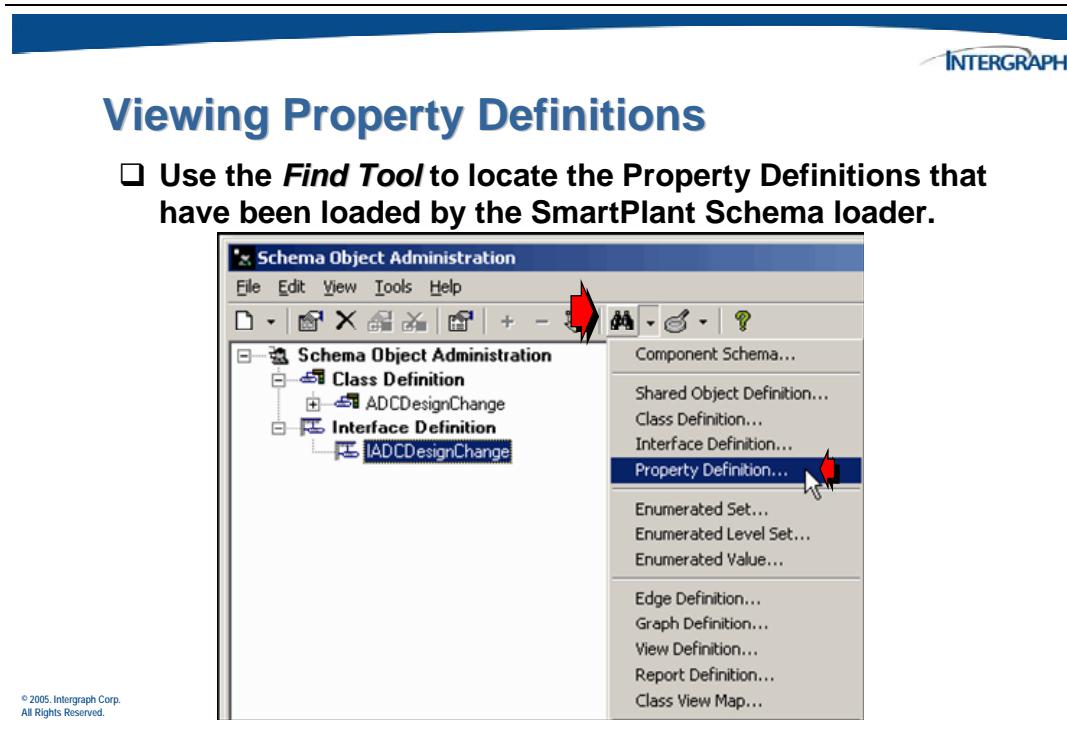
- UID** - Unique identifier for the object. This identifier is only required to be unique within the SmartPlant Schema.
- Name** - Name of the object
- Description** - Description of the object

Property information can be used to determine the property type or, in other words, the possible values for that property definition. Standard schema property types include *Boolean, integer, double, and string*.

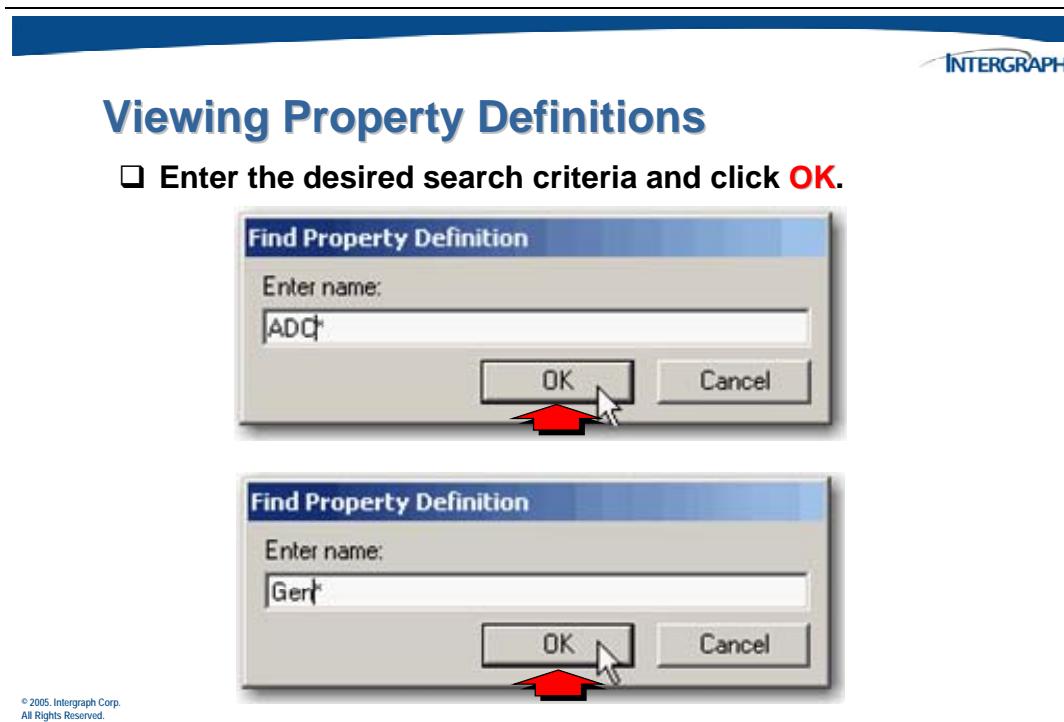
Property definitions of the enumerated property type have a list of possible string property values defined for them. A property definition of this type must match an entry in the list of enumerated property values defined for the property type.

2.4.1 Viewing a Property Definition

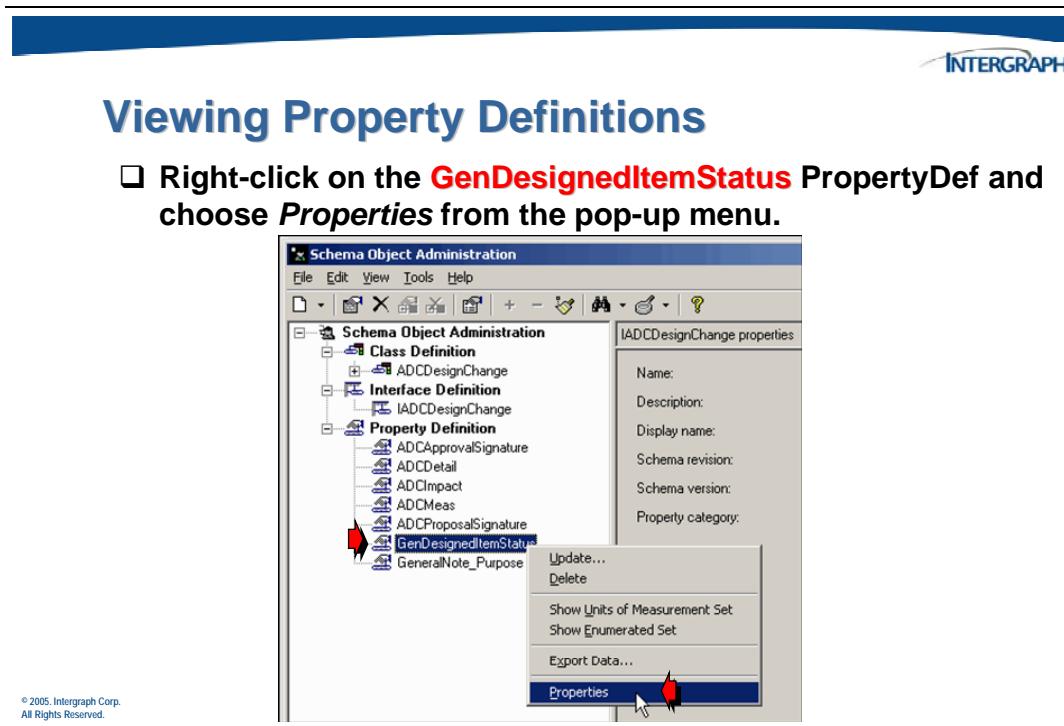
To view and verify any of the properties that were loaded using the schema loader, use Schema Object Administration to perform a search.



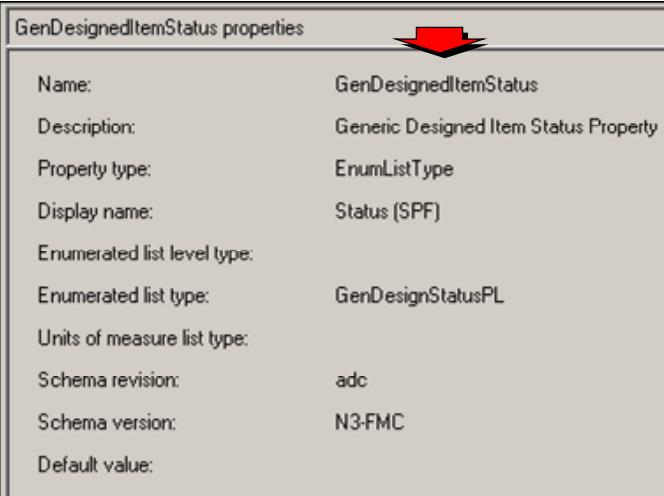
Use the *Find* dialog to search for any property definitions that have been loaded.



The resulting *Property Definitions* are shown in the tree view.



View the properties of any of the property definitions in the right pane.

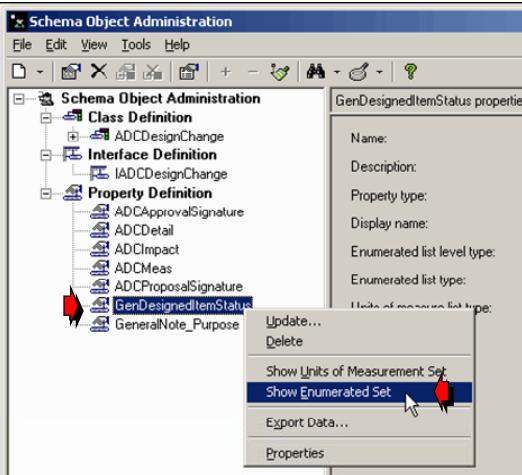


The screenshot shows the "GenDesignedItemStatus properties" dialog box. A red arrow points from the text above to the title bar of this dialog. The dialog lists the following properties:

Name:	GenDesignedItemStatus
Description:	Generic Designed Item Status Property
Property type:	EnumListType
Display name:	Status (SPF)
Enumerated list level type:	
Enumerated list type:	GenDesignStatusPL
Units of measure list type:	
Schema revision:	adc
Schema version:	N3-FMC
Default value:	

© 2005, Intergraph Corp.
All Rights Reserved.

This property has values supplied by an enumerated list (picklist).



The screenshot shows the "Schema Object Administration" interface with the "Property Definition" node selected in the tree view. A red arrow points from the text above to the "GenDesignedItemStatus" property definition in the tree. A context menu is open over this item, with a red arrow pointing to the "Show Enumerated Set" option. Other options in the menu include "Update...", "Delete", "Show Units of Measurement Set", "Export Data...", and "Properties".

© 2005, Intergraph Corp.
All Rights Reserved.



Viewing Property Definitions

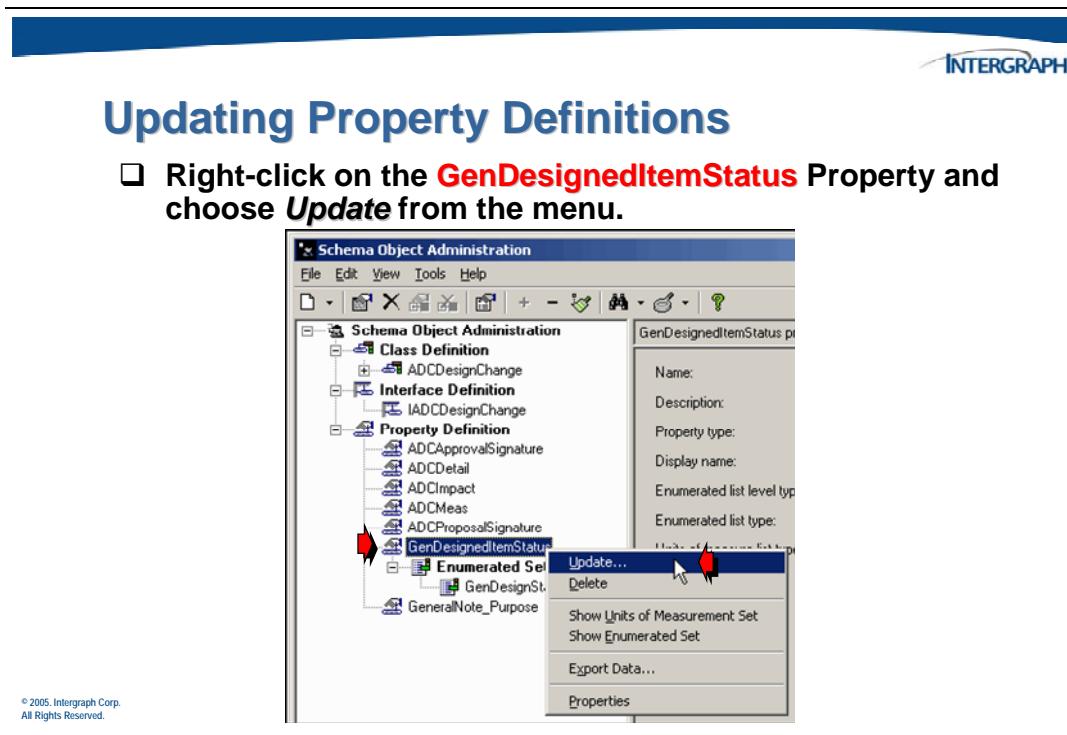
The tree is expanded to show the associated enumerated set (list) for this property.

The screenshot shows the 'Schema Object Administration' application interface. On the left, a tree view displays various schema objects: 'Schema Object Administration', 'Class Definition', 'Interface Definition', 'Property Definition', and an 'Enumerated Set'. Under 'Property Definition', several items are listed, including 'ADCDesignChange', 'IADCDesignChange', 'ADCAccuracySignature', 'ADCDetail', 'ADCImpact', 'ADCMes', 'ADCPoposalSignature', 'GenDesignedItemStatus', and 'GenDesignStatusPL'. A red arrow points from the text 'The tree is expanded to show the associated enumerated set (list) for this property.' towards the 'GenDesignStatusPL' node. On the right, a details pane titled 'GenDesignedItemStatus properties' lists the following information:

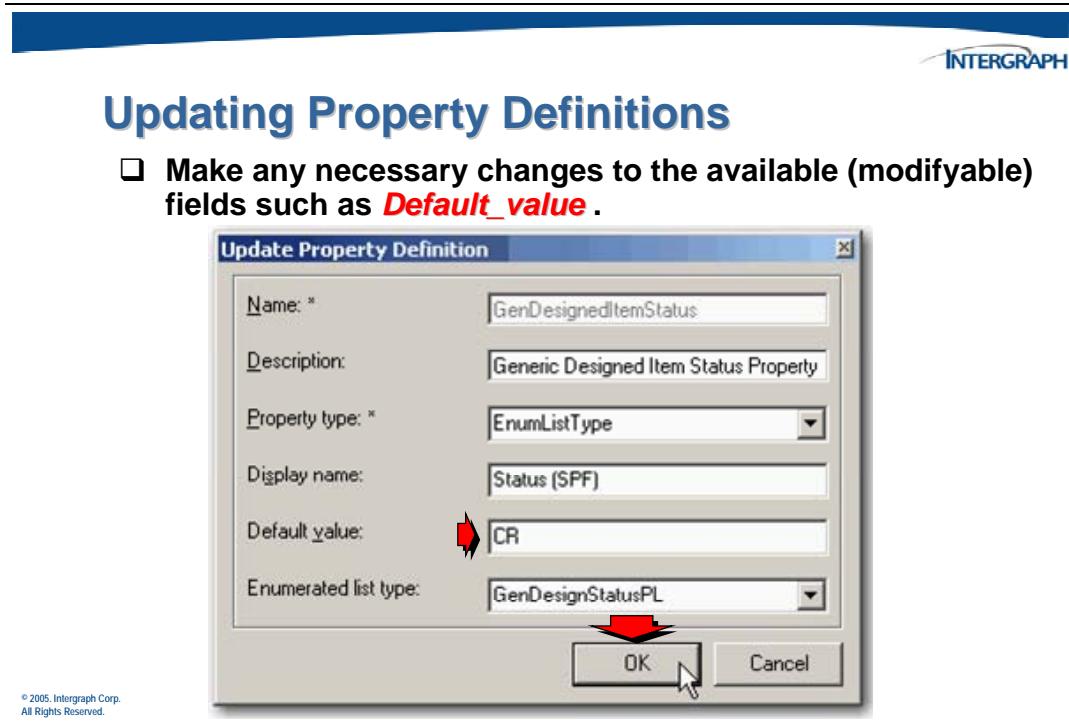
GenDesignedItemStatus properties	
Name:	GenDesignedItemStatus
Description:	Generic Designed Item Status Property
Property type:	EnumListType
Display name:	Status (SPF)
Enumerated list level type:	
Enumerated list type:	GenDesignStatusPL
Units of measure list type:	
Schema revision:	adc
Schema version:	N3-FMC
Default value:	

2.4.2 Update a Property Definition

Some properties of the property definition can be changed such as the *Display name* or the *Default_value*.

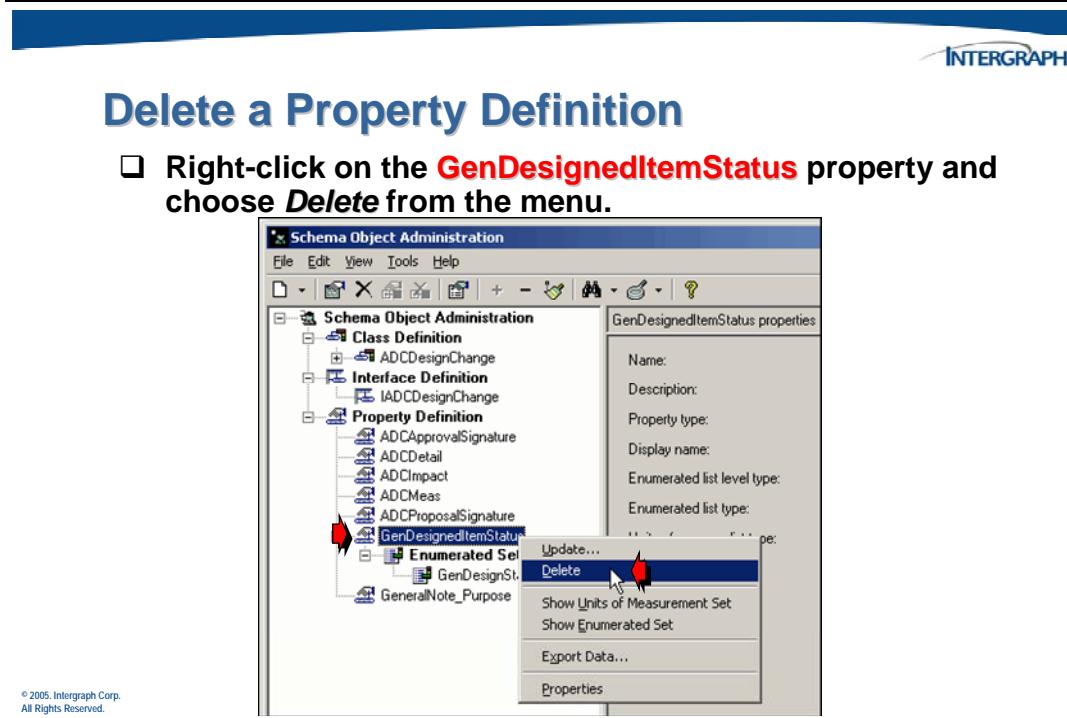


The *Update* form will display in the right pane.



2.4.3 Delete a Property Definition

Property definitions can be deleted from the administration database but again, extreme caution must be taken into consideration. The delete operation does not remove the property definition information from the schema file.



A delete confirmation dialog will display.

Delete a Property Definition

- To delete this Property Definition, click **OK**.



2.5 Enumerated List Sets

In the SmartPlant schema, some property definitions are of the enumerated property type.



Enumerated List Sets

If a property has a list of possible string properties defined for it, it is called an enumerated list.

Enumerated lists can contain all the possible values for a particular property.

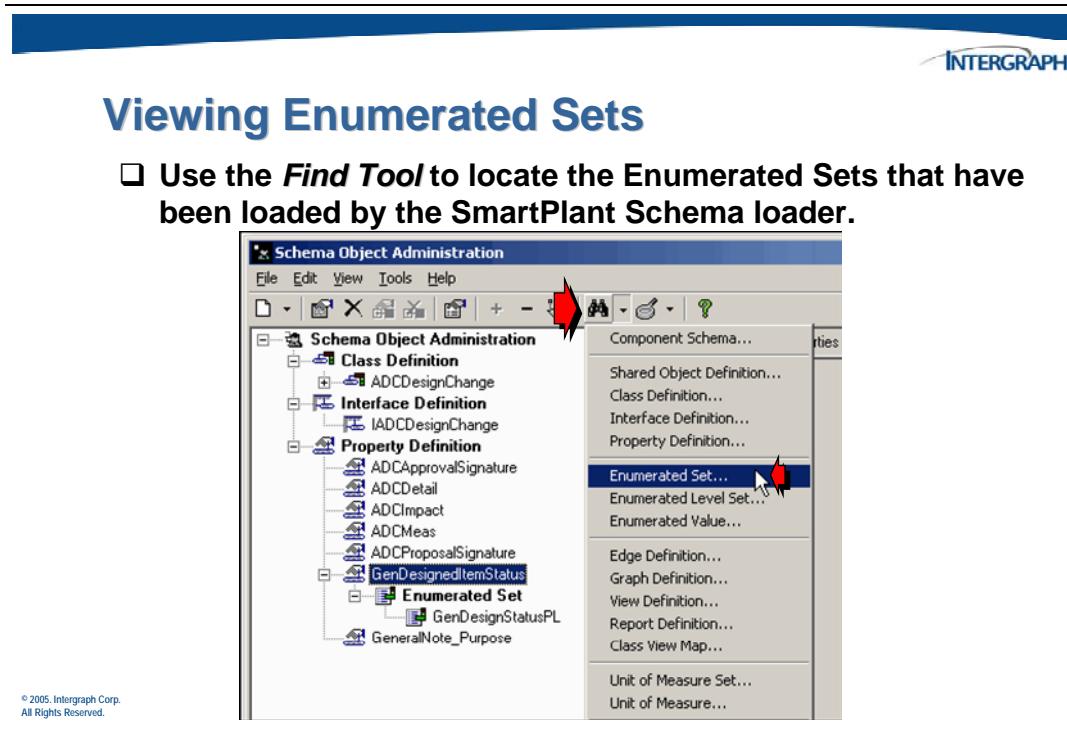
Enumerated List Sets are referred to in the Schema Editor as Enumerated List Types.

© 2005, Intergraph Corp.
All Rights Reserved.

You can also combine enumerated lists by creating child lists that are referenced by other enumerated lists (nested enumerated lists).

2.5.1 Viewing Enumerated Sets

The enumerated lists or sets that were loaded using the schema loader can be viewed and verified. First, use Schema Object Administration to perform a search.



Use the *Find* dialog to search for any property definitions that have been loaded.

Viewing Enumerated Sets

- Enter the desired search criteria and click **OK**.

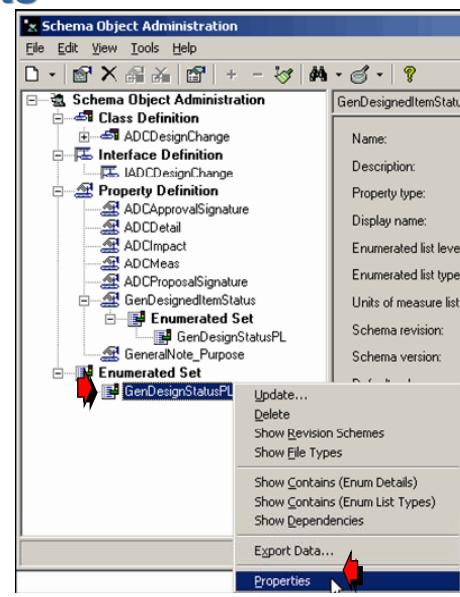


© 2005, Intergraph Corp.
All Rights Reserved.

The resulting *Enumerated Set* is shown in the tree view.

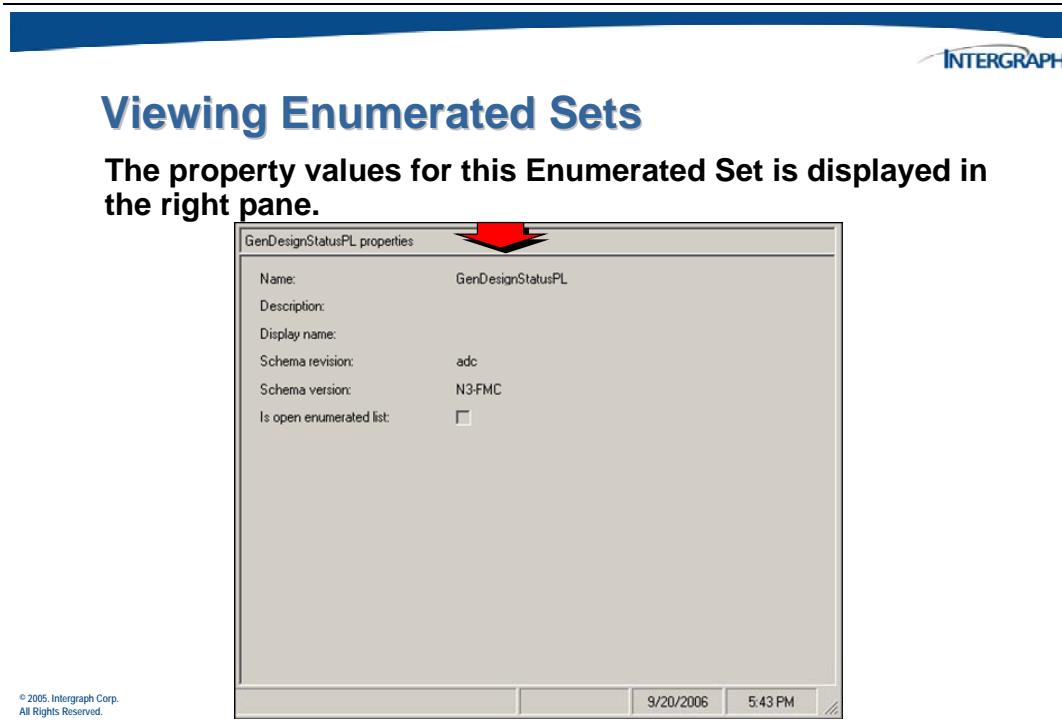
Viewing Enumerated Sets

- Right-click on the **GenDesignStatusPL** Enumerated Set (list) and choose *Properties* from the menu.

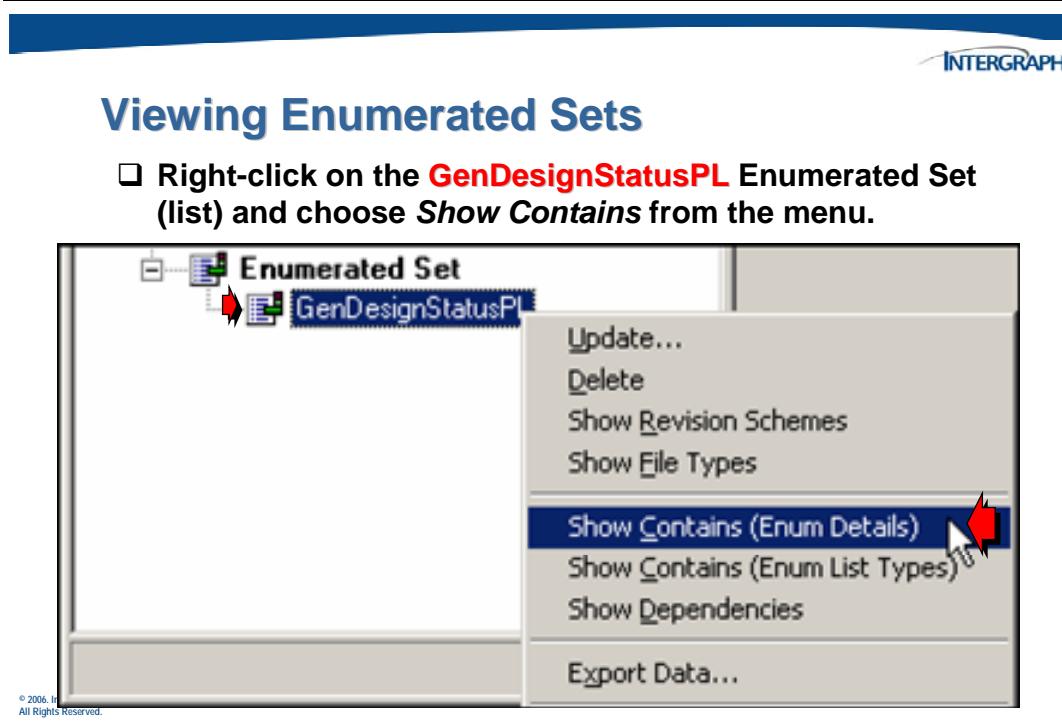


© 2005, Intergraph Corp.
All Rights Reserved.

View the properties of any of the enumerated set in the right pane.



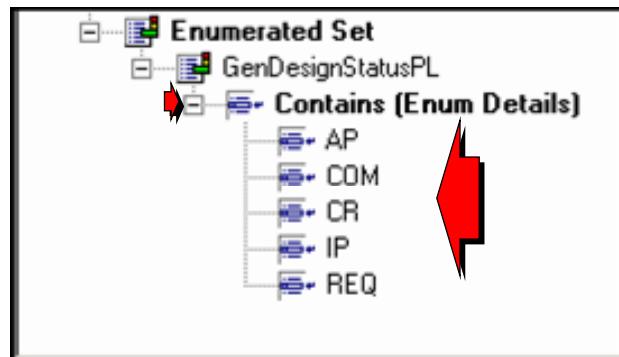
Expand the tree to see the values contained in the enumerated set.





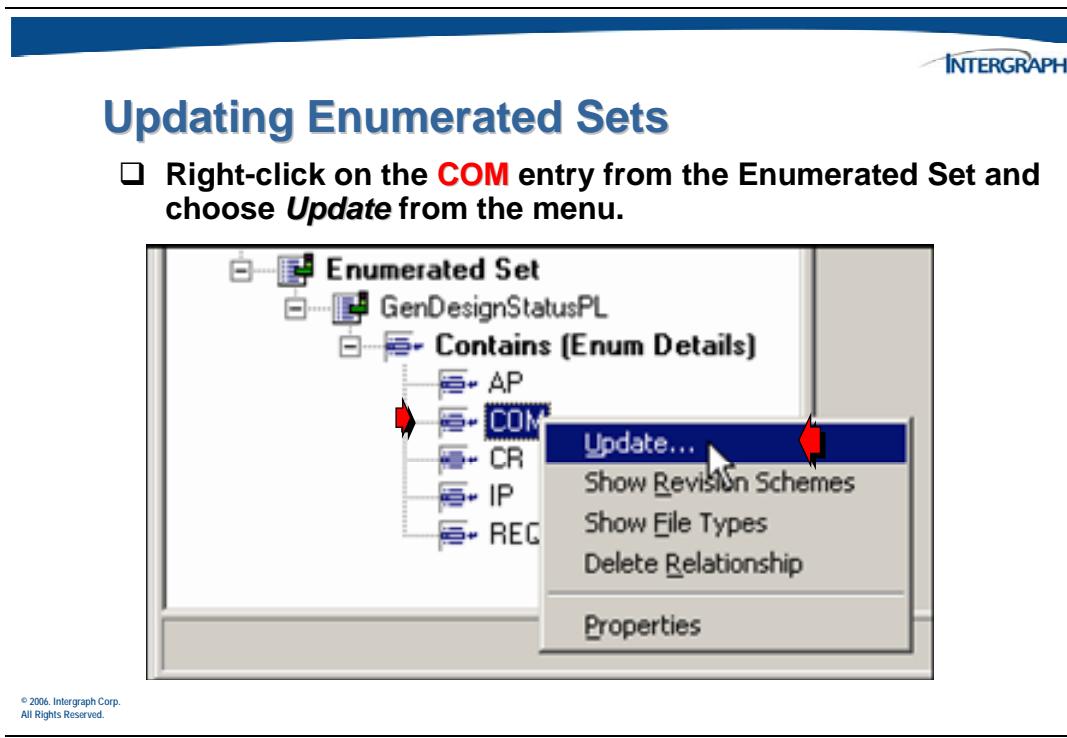
Viewing Enumerated Sets

The contents of the Enumerated Set is shown in the expanded tree (Enum Details).

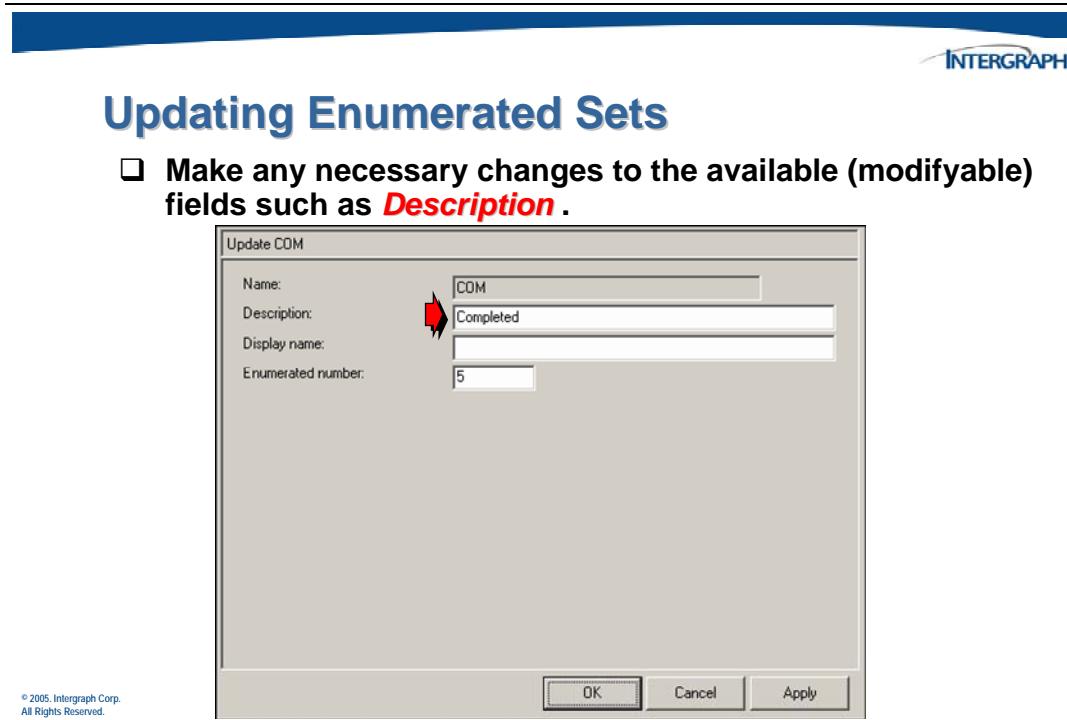


2.5.2 Update an Enumerated Set

Only certain properties of the enum details can be changed such as the *Description* or the *Display name*.

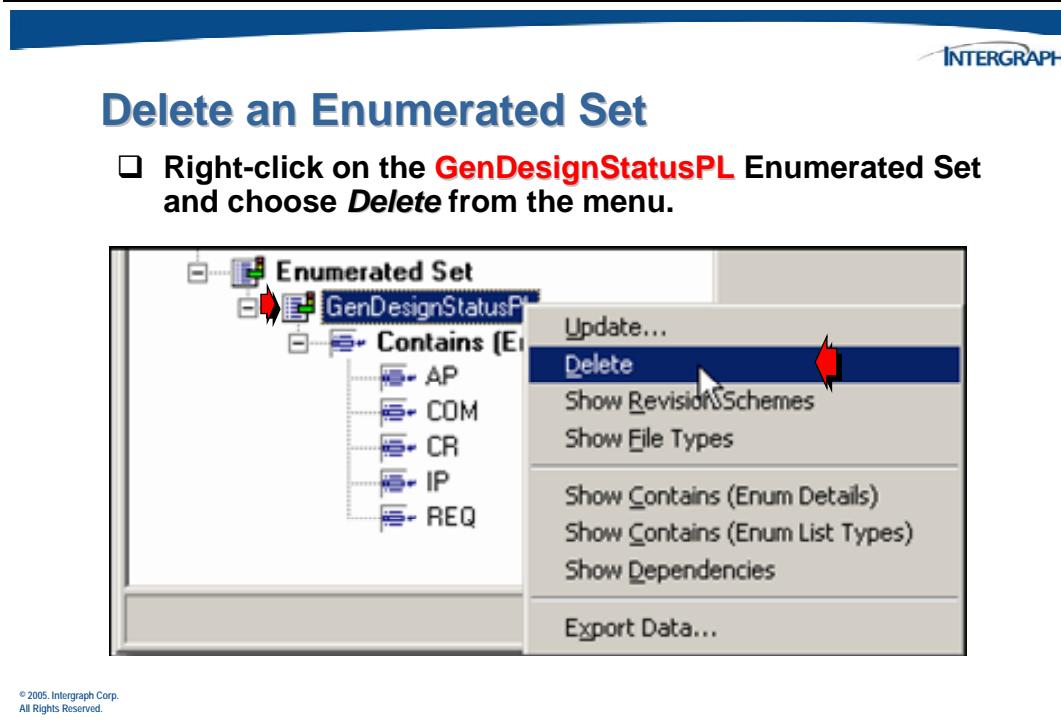


The *Update* form will display in the right pane.

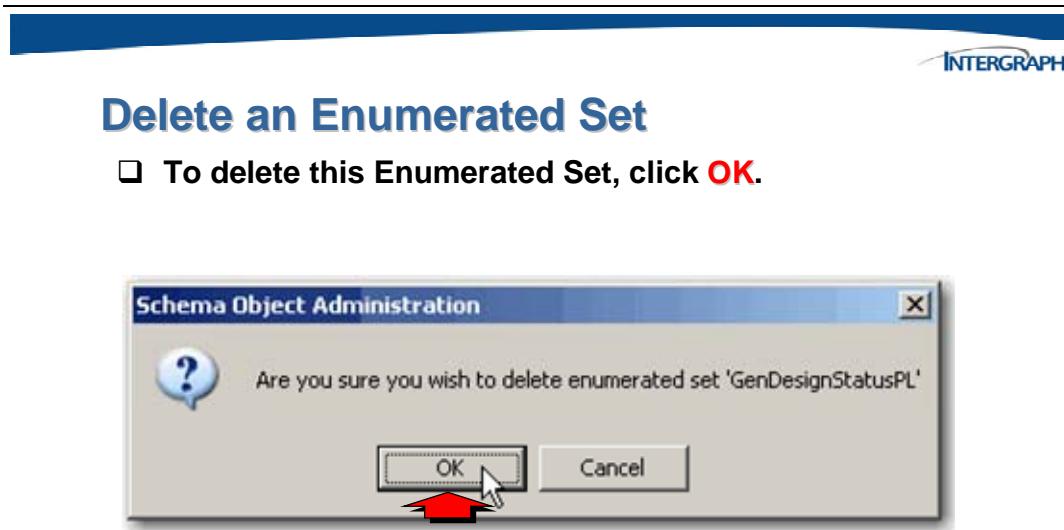


2.5.3 Delete an Enumerated Set

An enumerated set that is in error or no longer needed can be deleted from the administration database. The delete operation does not remove the enumerated set information from the schema file.



A delete confirmation dialog will display.



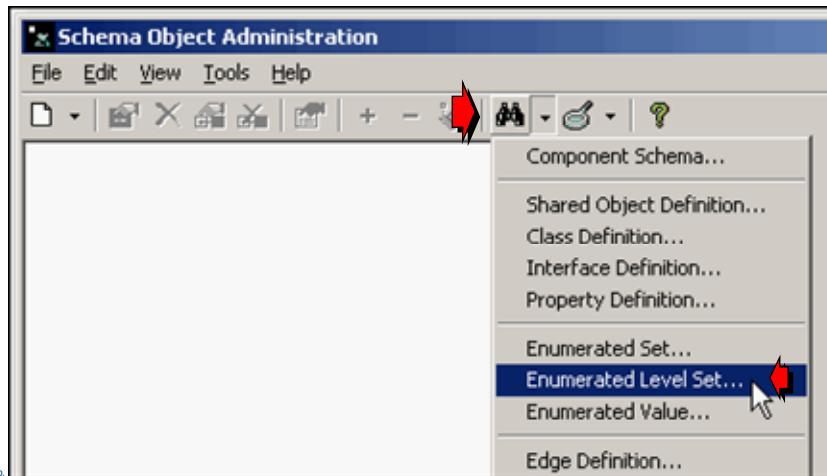
2.5.4 Viewing Enumerated Level Sets

Some enumerated lists contain other enumerated lists as entries. This gives you the ability to have subsets of list entries. These lists are called *Enumerated Level Sets*. To view these kinds of enumerated sets, search for and view these in Schema Object Administration.



Viewing Enumerated Level Sets

- Use the *Find Tool* to locate the Enumerated Level Sets in the administrative database.



Note: Enumerated Level Sets are referred to in the Schema Editor as **Enumerated List Level Types**.

Use the *Find* dialog to search for the enumerated level sets in the administration database.

Viewing Enumerated Level Sets

- ❑ Enter the desired search criteria and click **OK**.

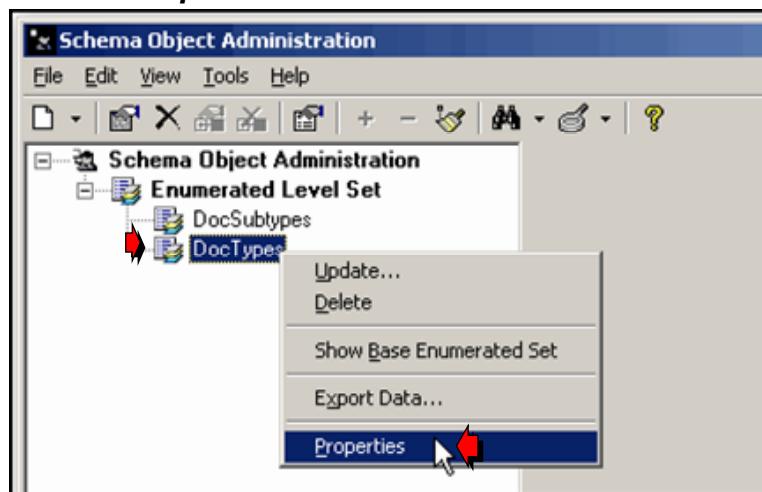


© 2005, Intergraph Corp.
All Rights Reserved.

The resulting *Enumerated Level Set* is shown in the tree view.

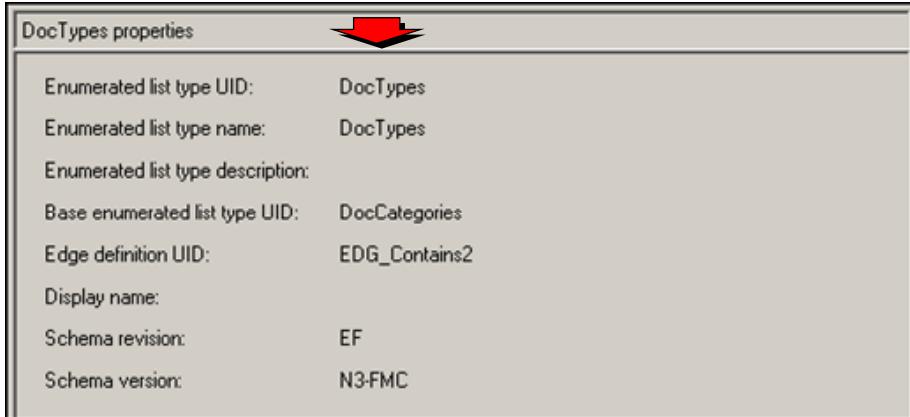
Viewing Enumerated Level Sets

- ❑ Right-click on the **DocTypes** Enumerated Level Set and choose *Properties* from the menu.



© 2005, Intergraph Corp.
All Rights Reserved.

View the properties of any of the enumerated level set in the right pane.

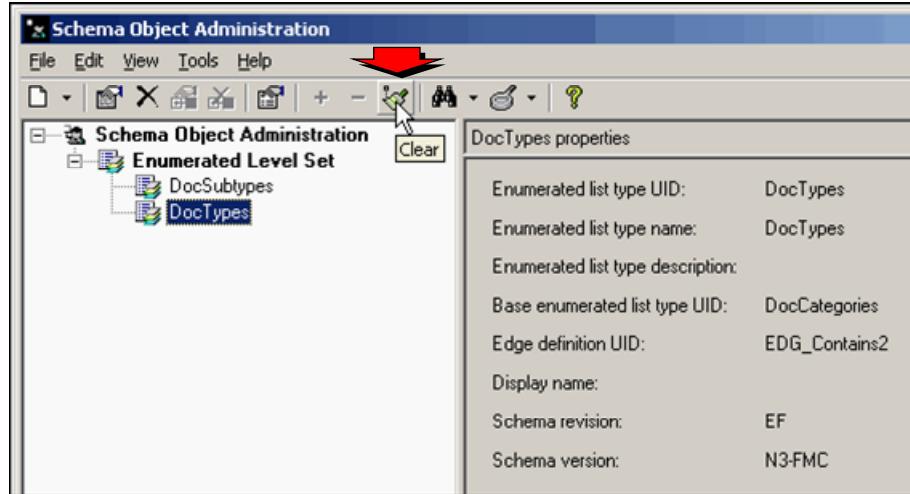


The screenshot shows the 'DocTypes properties' dialog box. A red arrow points from the text above to the title bar of this dialog. The properties listed are:

Enumerated list type UID:	DocTypes
Enumerated list type name:	DocTypes
Enumerated list type description:	
Base enumerated list type UID:	DocCategories
Edge definition UID:	EDG_Contains2
Display name:	
Schema revision:	EF
Schema version:	N3-FMC

© 2005, Intergraph Corp.
All Rights Reserved.

You can see in this example that the enumerated list type **DocTypes** is a subset of the Base enumerated list type (parent) **DocCategories**.



The screenshot shows the 'Schema Object Administration' window. A red arrow points from the text above to the 'Clear' button in the toolbar. The left pane shows a tree view with 'Schema Object Administration' expanded, showing 'Enumerated Level Set' and its sub-items 'DocSubtypes' and 'DocTypes'. The right pane shows the 'DocTypes properties' dialog box, identical to the one in the previous screenshot. A small copyright notice is visible at the bottom of the interface.

2.6 Viewing and Modifying Navigation Objects

The objects that are used for navigating and viewing information are *Edge Definitions*, *Graph Definitions*, and *View Definitions* and *Class View Maps*.



Navigation Objects in Schema Object Admin

The examples that follow describe viewing and creating the following objects in Schema Object Administration:

- Edge Definitions
- Graph Definitions
- View Definitions
- Class View Maps

2.6.1 Edge Definitions in Schema Object Administration

Edge definitions, graph definitions, and view definitions are used to provide different ways of viewing and traversing the data in SmartPlant Foundation.

Edge definitions are single or multiple relationship definitions with direction. An edge definition is used to traverse from a starting object to related objects.



Edge Definitions

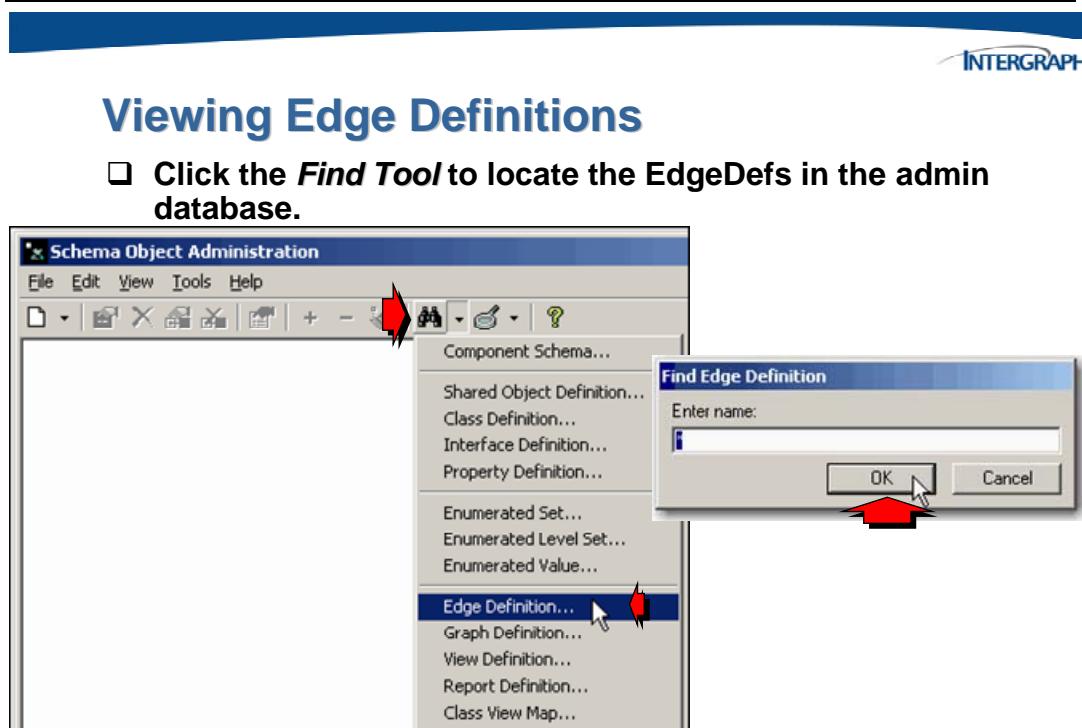
Edge Definitions are single or multiple relationship definitions with direction and are used to traverse from a starting object to related objects.

Each **Edge Definition** consists of the following:

- A starting interface definition
- A path definition that originates at that starting interface definition
- Discrimination criteria that can be applied to objects at the end of the path
- Position criteria that can select a specific object from the set of objects at the end of the path that satisfy the discrimination criteria

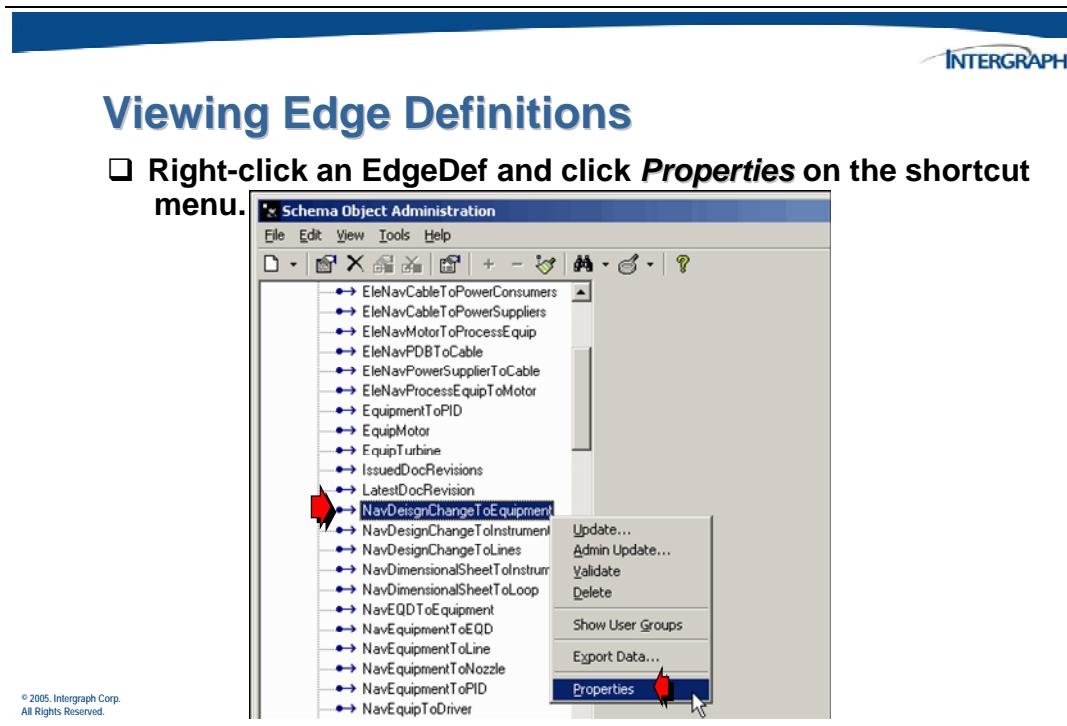
2.6.2 Viewing Edge Definitions

Schema Object Administration can be used to view existing *EdgeDefs*. Use the **Find** command to specify search criteria to be used in locating existing EdgeDefs. These EdgeDefs may have been created with the SmartPlant Schema Editor and imported into SPF with the SmartPlant Schema Loader.

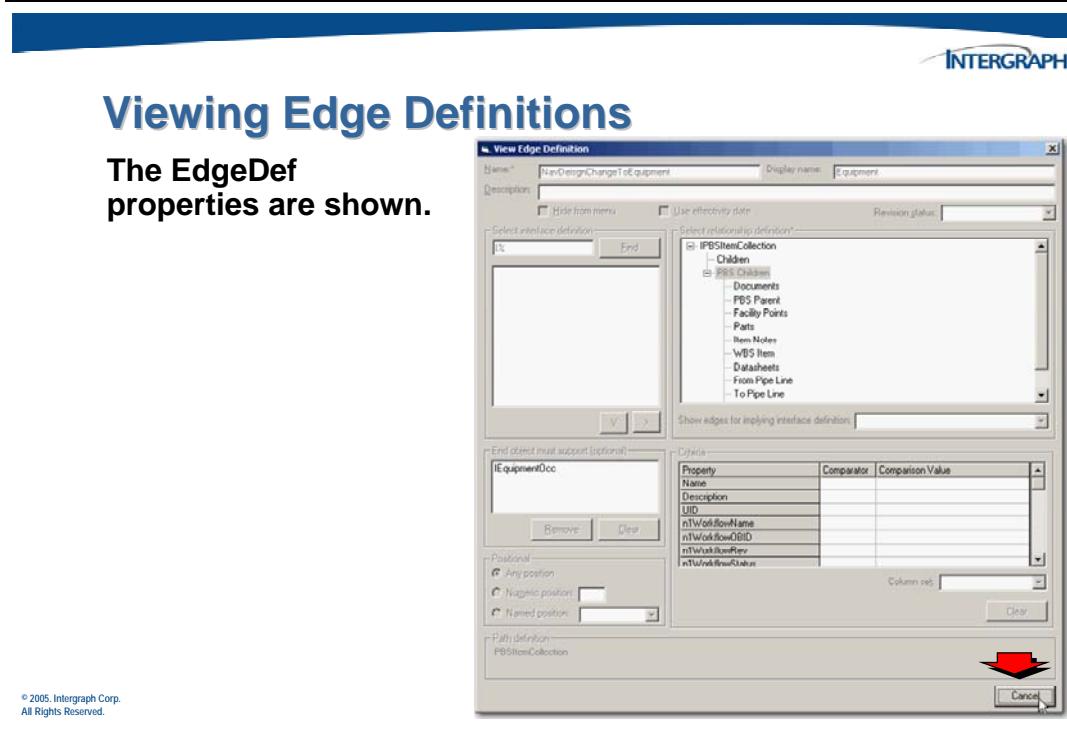


When the *Find* dialog box appears, enter the search criteria to be used in finding and displaying EdgeDefs.

The results of the Find will display in the tree view.



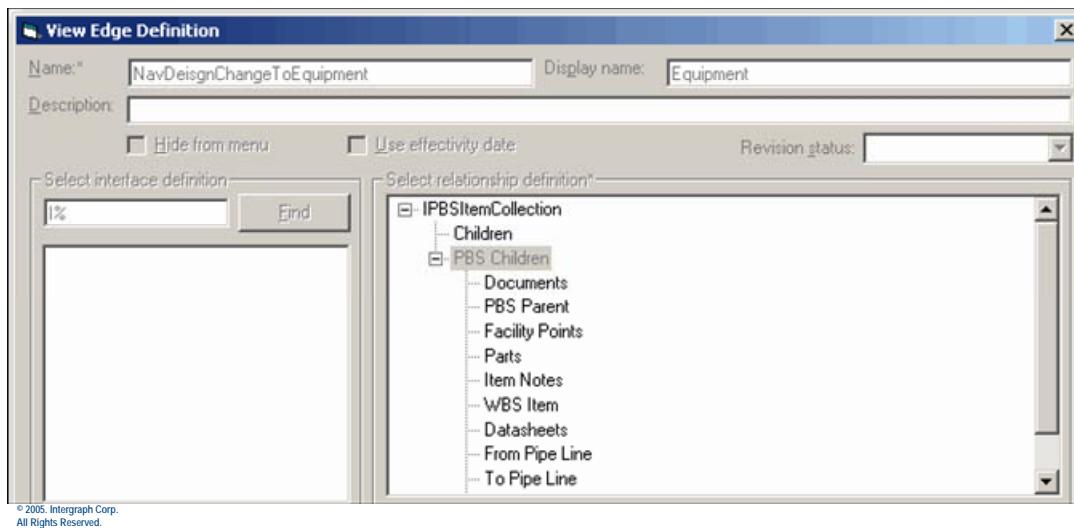
The *View Edge Definitions* dialog box appears.





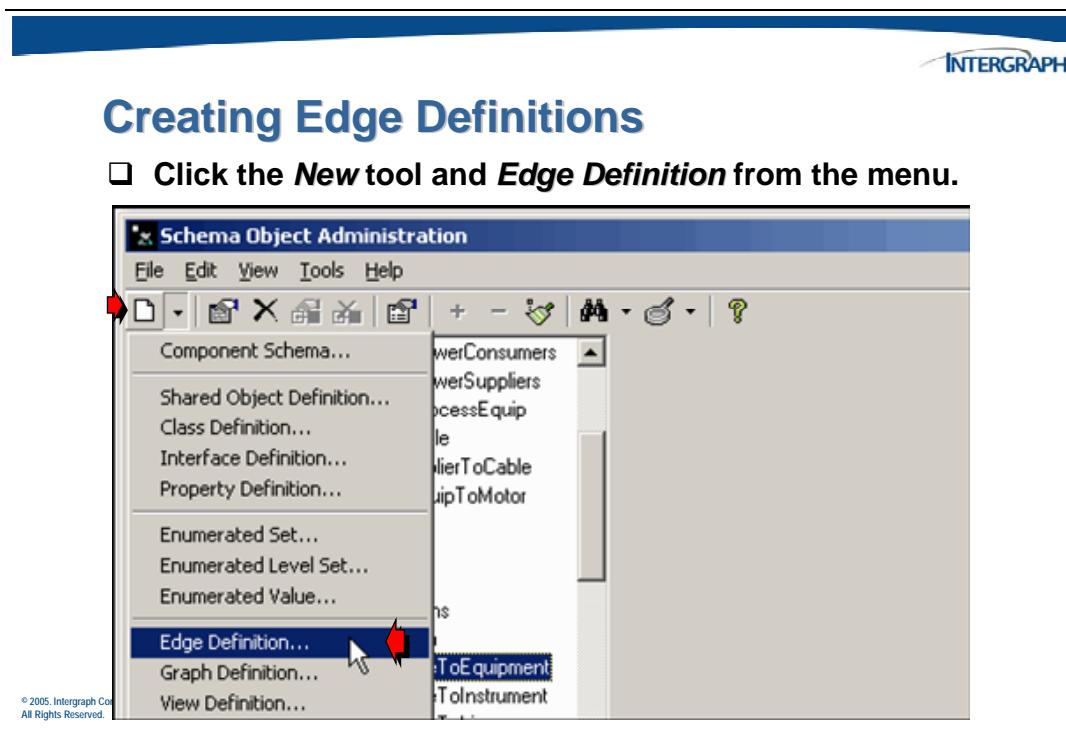
Viewing Edge Definitions

Zoomed in view of the *Edge Definition* properties.



2.6.3 Creating Edge Definitions

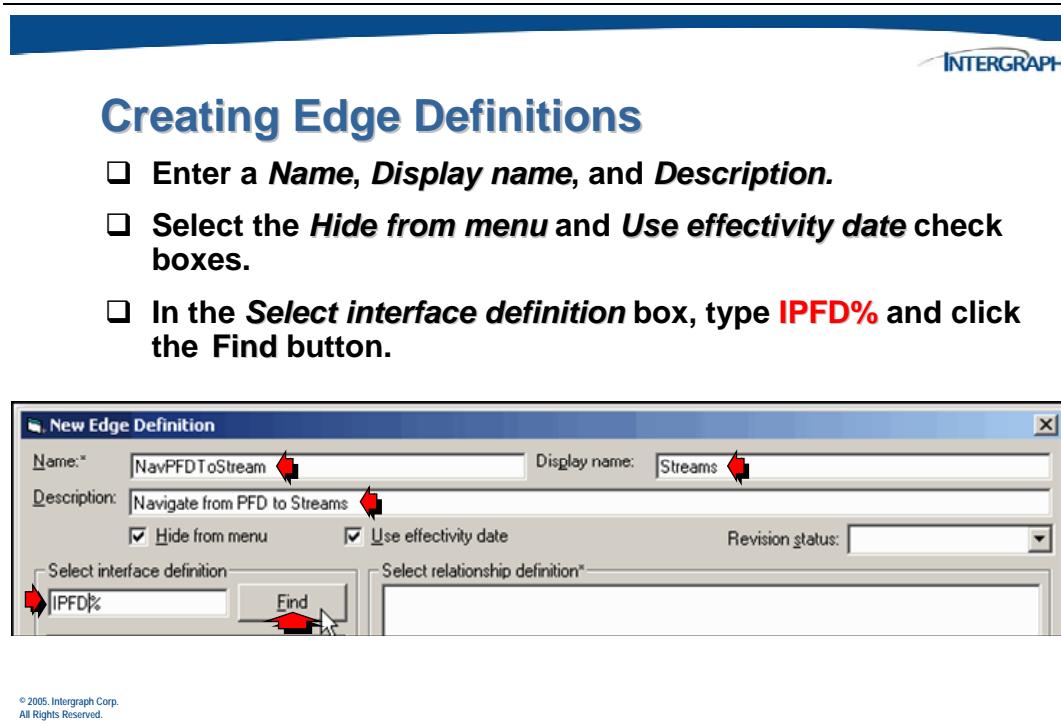
When you create a relationship definition in *Schema Object Administration*, two edge definitions are created: one for traversing from **end 1** of the relationship to **end 2** and the other for traversing from **end 2** to **end 1**. Each edge definition is given a UID that is the UID of the relationship definition followed by **_12** for the edge definition going from **end 1** to **end 2** or by **_21** for the edge definition going from **end 2** to **end 1**. The name for each edge definition is the corresponding role name defined for the relationship definition. The path definition is either + (end 1 to 2) or – (end 2 to 1) followed by the UID for the relationship definition. These edge definitions do not have discrimination or position criteria, and they are not persisted in the SmartPlant schema.



You can explicitly define edge definitions for a variety of purposes in the SmartPlant Schema Editor and in Schema Object Administration. Explicit edge definitions may be used to traverse across multiple relationships, to select only certain objects from the destination end, or to do multiple relationship traversal and selective discrimination.

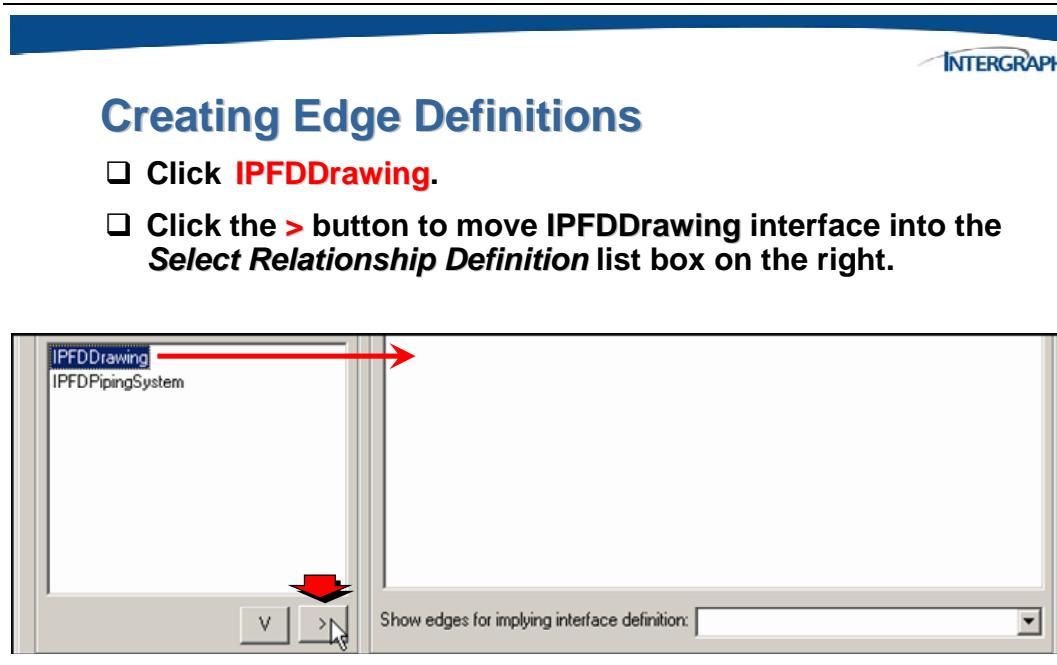
In SmartPlant Foundation, edge definitions are custom relationship expansions that appear on the shortcut menu when a user right-clicks an object that exposes the relationship referenced by the edge definition. Instead of requiring the user to go directly from **Object A** to **Object B** in the tree view, an edge definition can be created to allow the user to traverse through multiple intermediate objects and display an object at the end of the relationship that meets specific criteria. Without the EdgeDef, you would have to traverse **A** to **B**, **B** to **C**, **C** to **D** but with the EdgeDef you would only traverse from **A** directly to **D**.

The *New Edge Definition* dialog box appears.



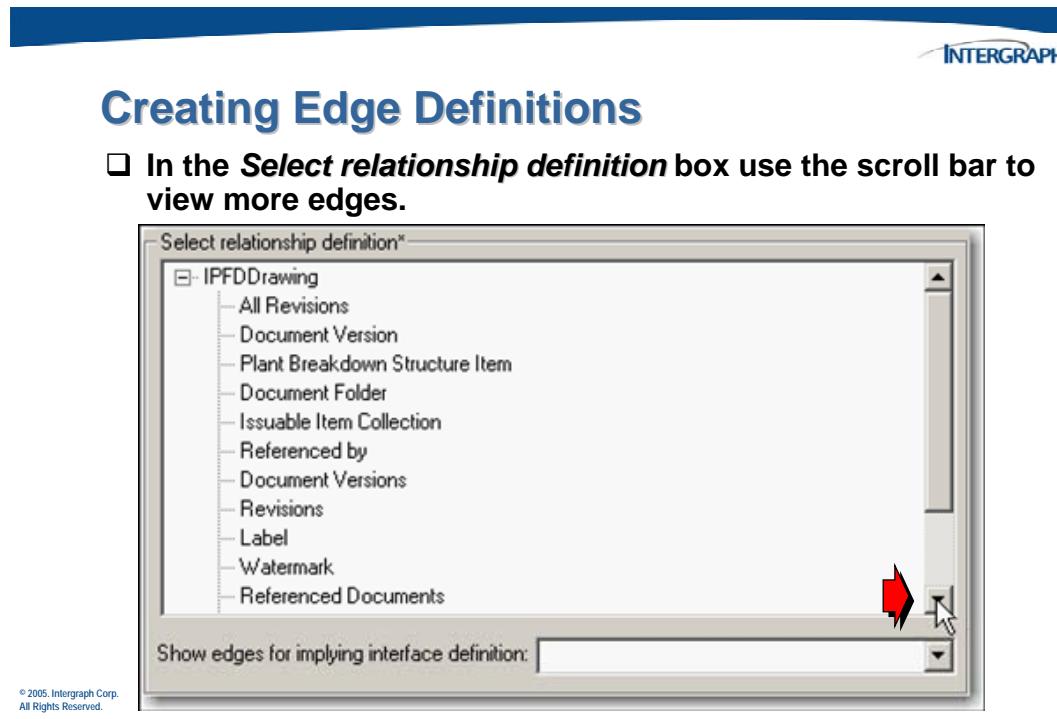
In SPF Schema Object Administration, you can set a flag on the edge definition to determine whether or not all users have access to the edge definition. If you only want the edge definition to appear on shortcut menus for users whose user group has been associated with the edge definition, select **Hide from menu**. Otherwise, the edge definition appears on all shortcut menus for appropriate objects, regardless of the user's user group.

This displays a list of *InterfaceDefs* matching the name specified.



© 2005, Intergraph Corp.
All Rights Reserved.

The **IPFDDrawing** will be the starting InterfaceDef for the edge definition. Click the + to expand the IPFDDrawing InterfaceDef in the *Select Relationship Definition* box.

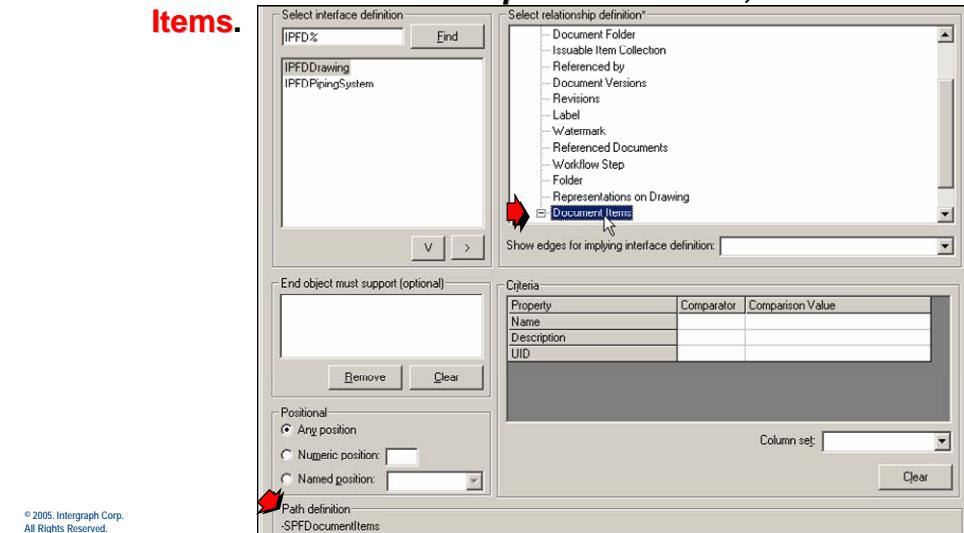


© 2005, Intergraph Corp.
All Rights Reserved.



Creating Edge Definitions

- In the **Select relationship definition** field, click **Document Items**.

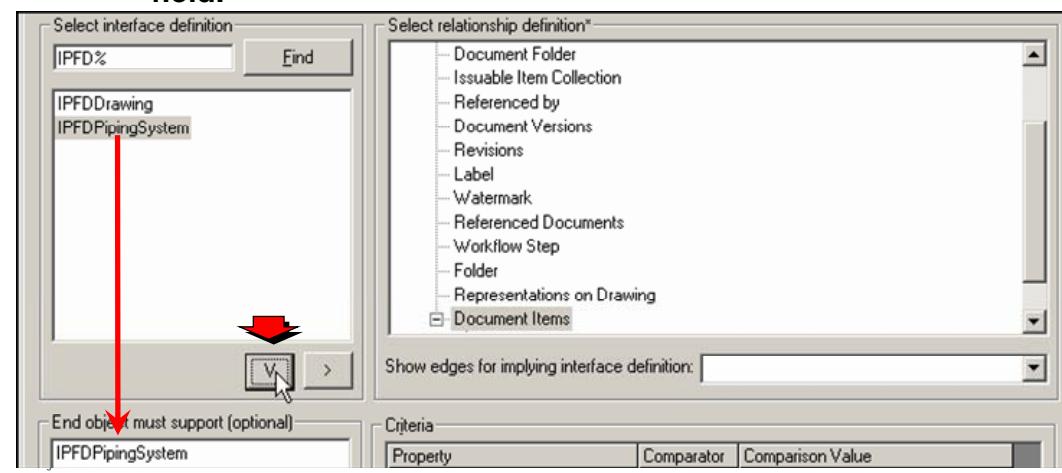


The path portion of the Edgedef is now defined.

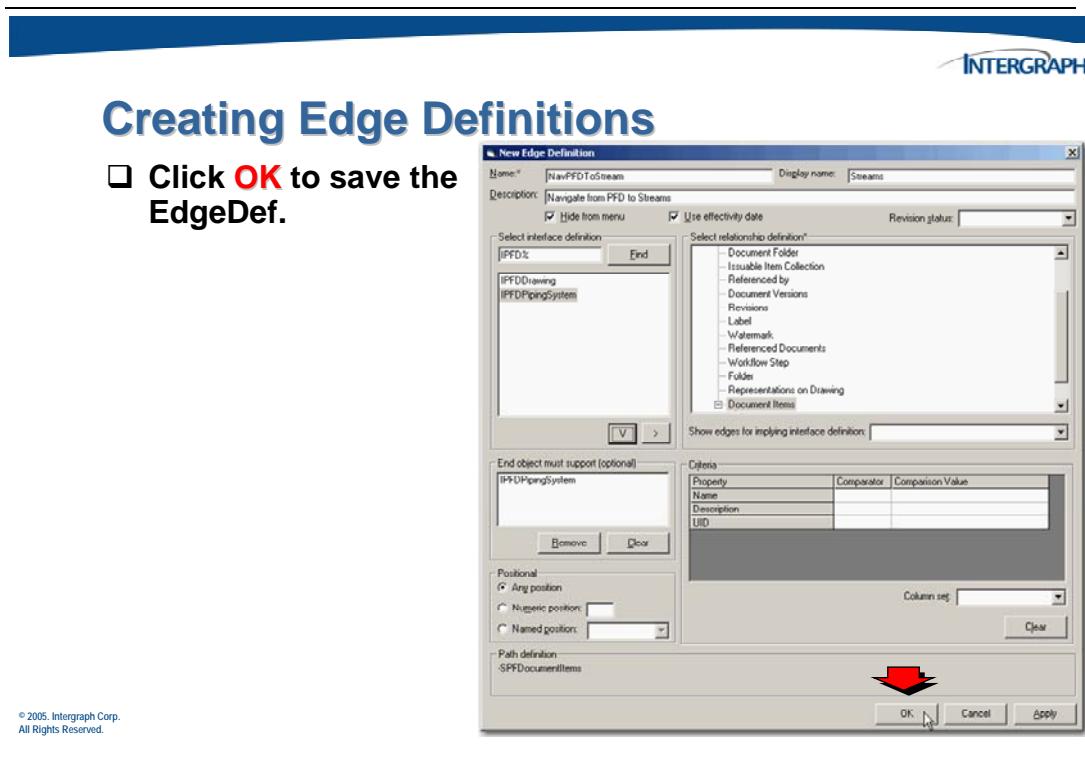


Creating Edge Definitions

- Click **IPFDPipingSystem** and click the **V** button to move the InterfaceDef into the **End object must support (optional)** field.



This sets the condition that only objects on the other end of the edge that have the IPFDPPipingSystem will be shown.



2.6.4 Setting Edge Definition Security

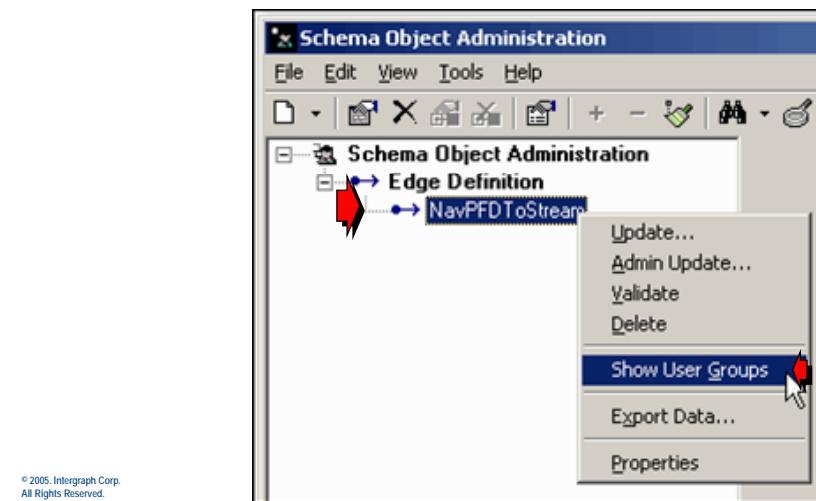
Associating a user group with an edge definition means that only those users who are members of the user group can see the selected relationship in the SPF client if the **Hide from menu** is turned on for the edge definition.

You will define access security for EdgeDefs using the same procedure as defining access security for any of the other objects in SPF.

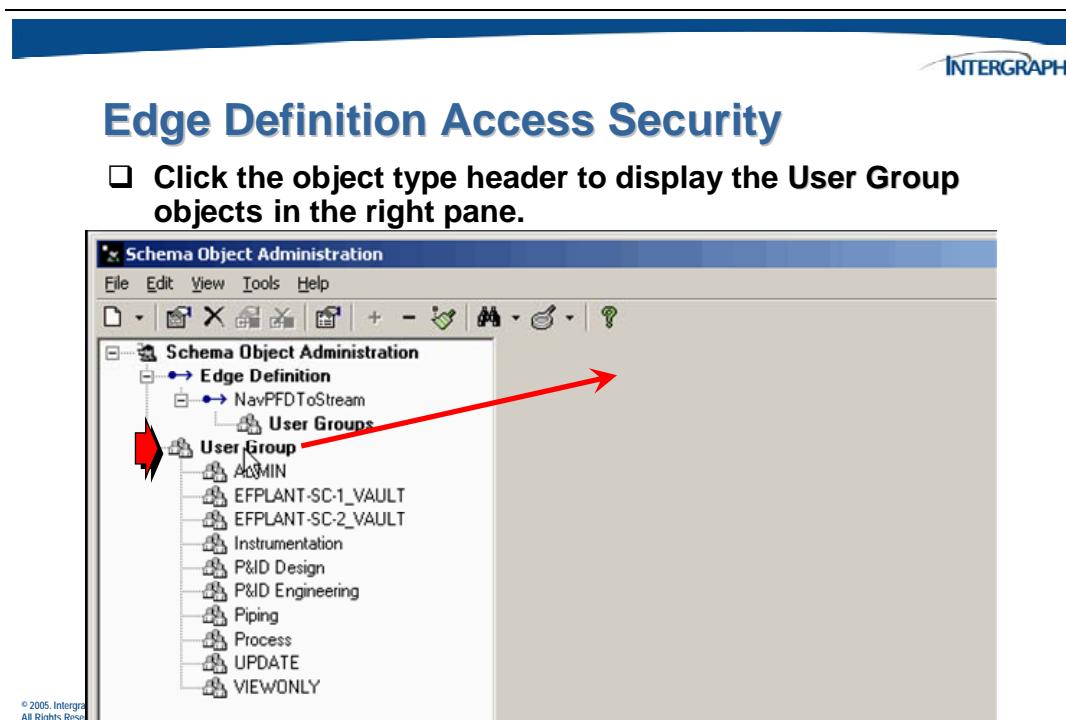
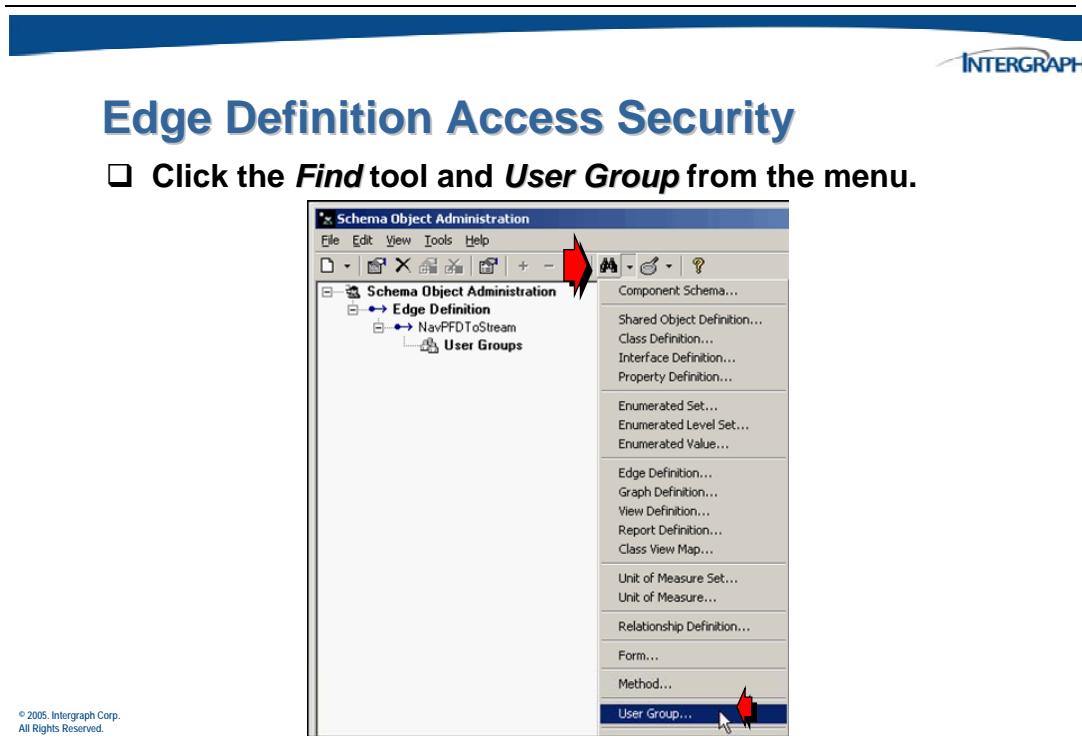


Edge Definition Access Security

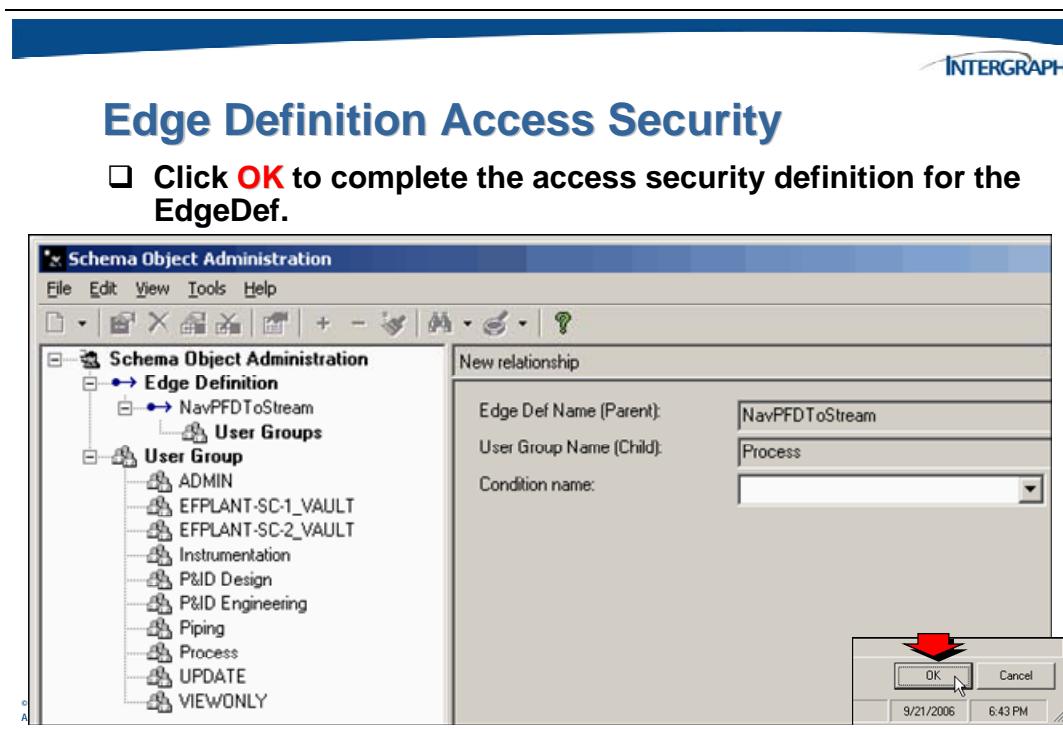
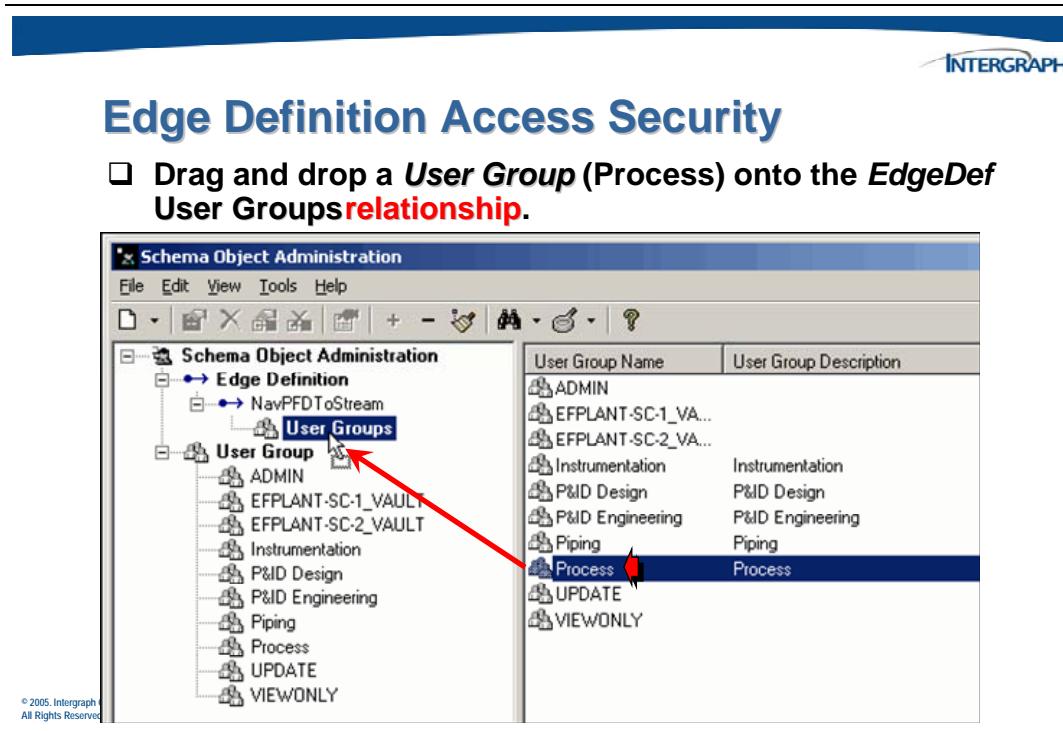
- Right-click the new EdgeDef and click **Show User Groups** on the shortcut menu.



Perform a search to locate the **User Group** objects.

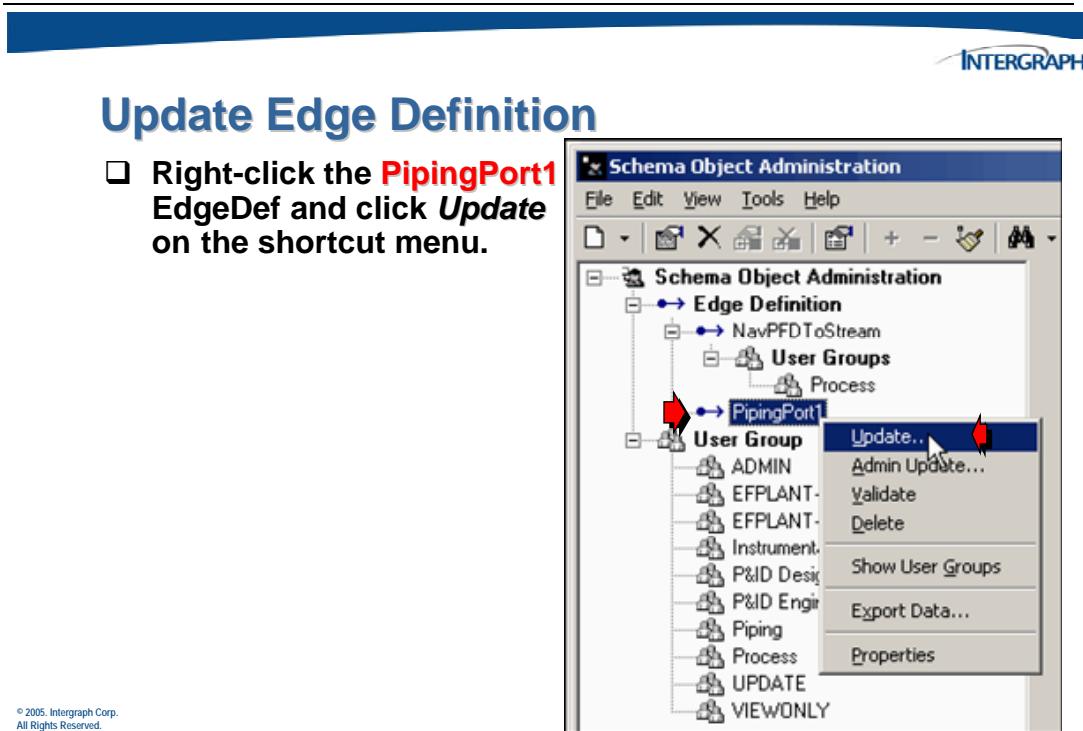


Locate the **Process** user group and drag it into the tree view.



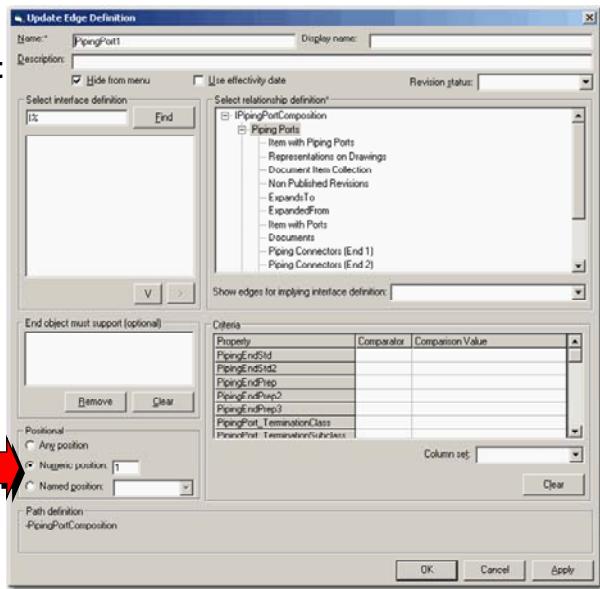
2.6.5 EdgeDefs with Conditions

You can update EdgeDefs and add conditions based on the properties.



© 2005, Intergraph Corp.
All Rights Reserved.

Notice the Numeric Position set to 1 in the bottom left corner.



The Numeric Position is set to 1 in the bottom left corner. This creates an **EdgeDef** that selects the first object on the other end of the selected relationship definition.

Update Edge Definition

Name: IPipingPort1 Display name:

Description:

Select interface definition: IPipingPortComposition

Relationship definition: IPipingPortComposition Hide from menu Use effectiveness date

Show edges for implying interface definition:

End object must support (optional):

Positional: Any position Numeric position: Named position:

Criteria:

Property	Comparator	Comparison Value
PipingEndStd		
PipingEndStd2		
PipingEndTyp		
PipingEndTyp2		
PipingEndTyp3		
PipingPort_TerminationClass		
PipingPort_TerminationObjectClass		

Path definition: IPipingPortComposition

OK Cancel Apply

© 2005, Intergraph Corp.
All Rights Reserved.

Notice the *properties* in the **Criteria** table. It is also possible to create conditions based on all of these properties. You could, for example, create an EdgeDef from a pump to its outlet nozzle if you knew the outlet nozzle was always named “Outlet” (for example, SmartPlant P&ID always names inlet “1” and outlet “2”).

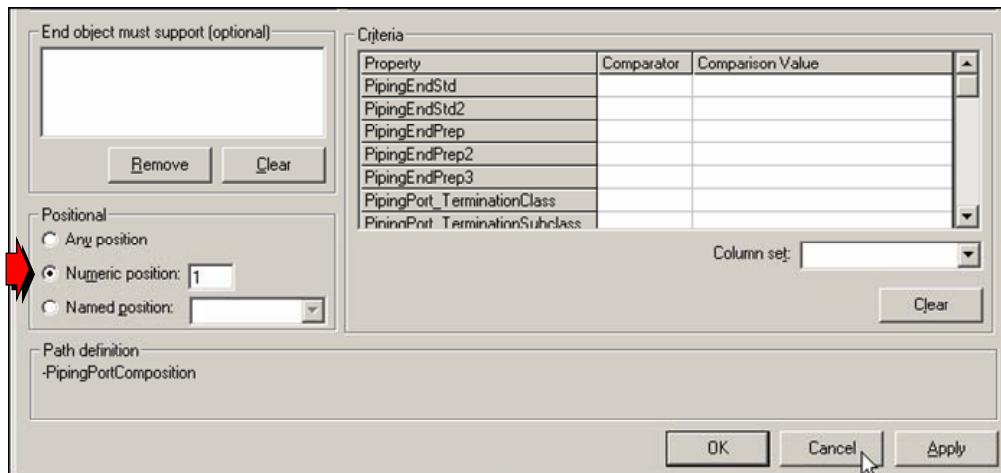
Numeric Position is simply the position of the related object across the relationships originating from the source object. Obviously Numeric Position is only meaningful across “to many” relationships.

For example, the relationship definition between a piece of equipment and its nozzles is 1:M (one to many). Therefore, an edge definition that has Numeric Position=1 will get the first nozzle across a relationship of that type, an edge definition with Numeric Position=2 will get the second object and so on.



Update Edge Definition

Zoomed in view of the *Update Edge Definition* dialog.



© 2005. Intergraph Corp.
All Rights Reserved.

2.6.6 Graph Definitions in Schema Object Administration

Graph definitions are a set of edge definitions with structure. A directed graph definition is used to define directed graphs.



Graph Definition

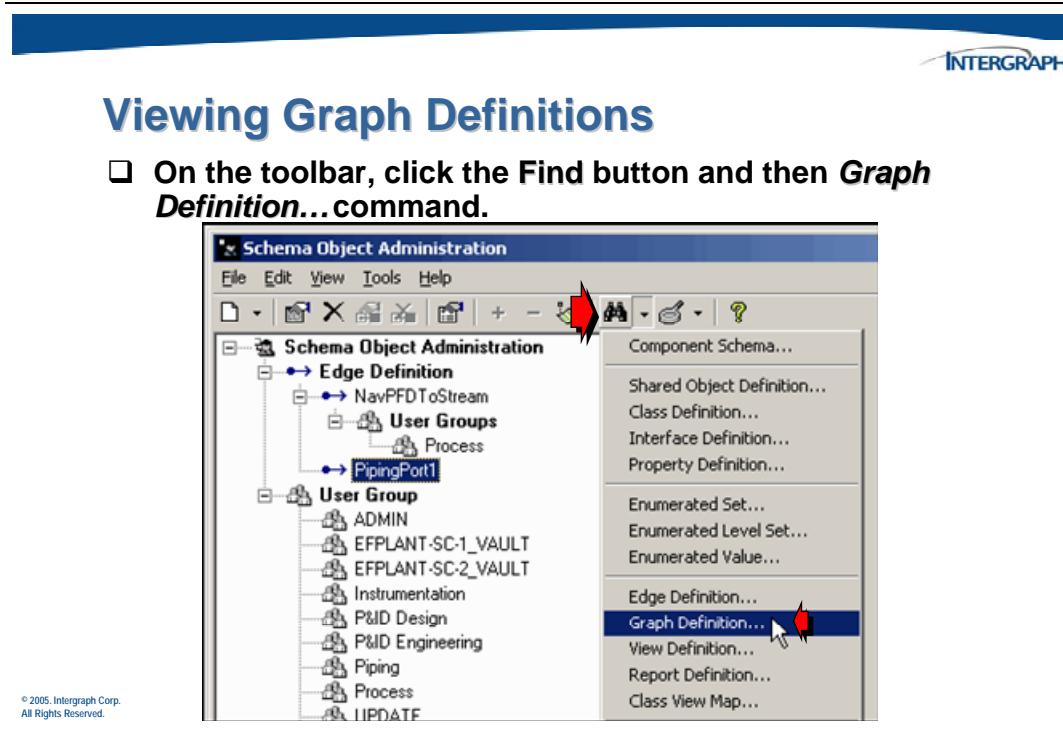
Graph Definitions are a set of edge definitions with structure. Any edge that is tied to the starting interface definition or to any interface definition implied by that interface definition can be used as part of the GraphDef.

Graph Definitions include the following:

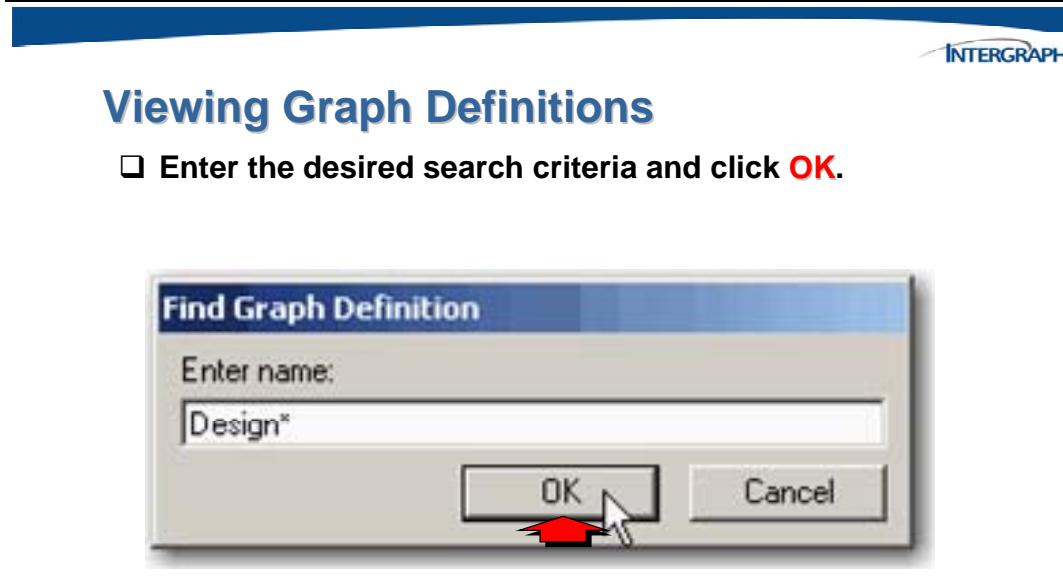
- A relationship to its starting interface definition
- A definition of the directed graph

In SmartPlant Foundation, graph definitions are used to define default expansions and alternate expansions for class definitions. They are also used to define datasheets for SmartPlant. If the name of a class definition is the same as the name of a graph definition, the graph definition is automatically used to define the default expansion for the class definition.

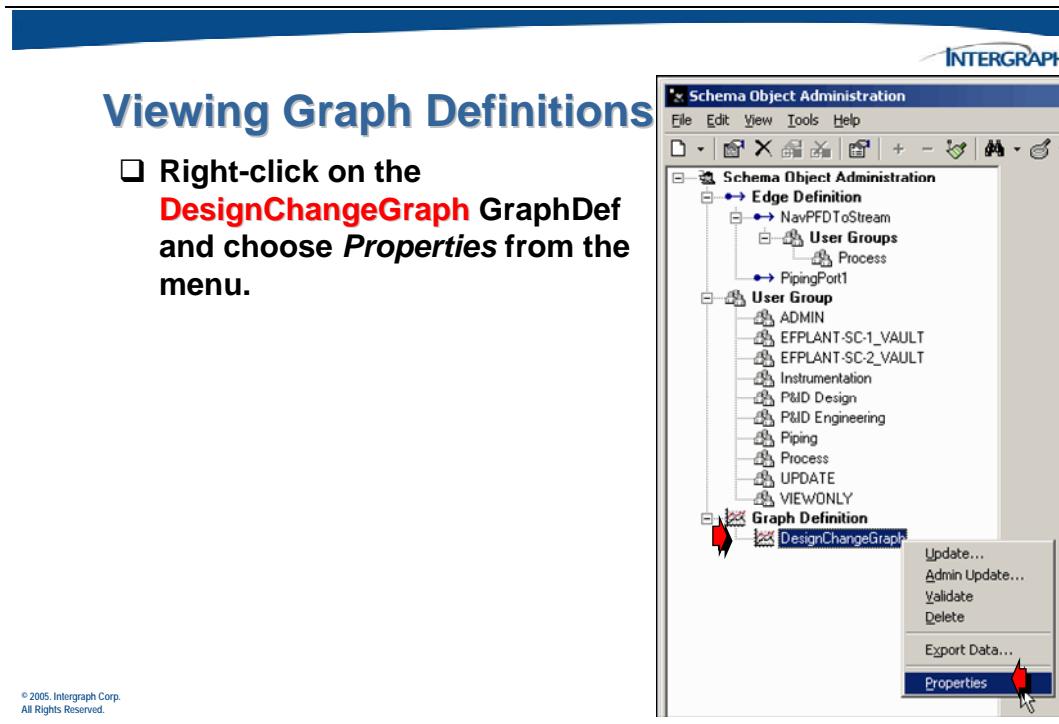
To view the edge definitions currently in the SPF system, use Schema Object Administration to perform a search.



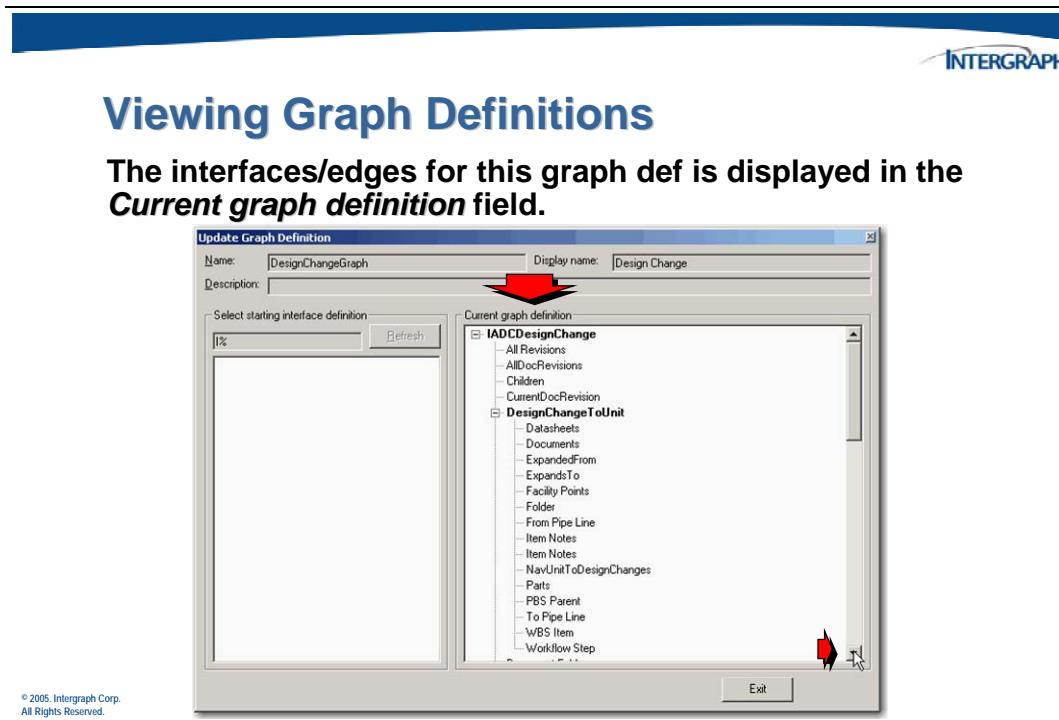
The *Find Graph Definition* dialog will display.



The resulting *Graph Definitions* are shown in the tree view.



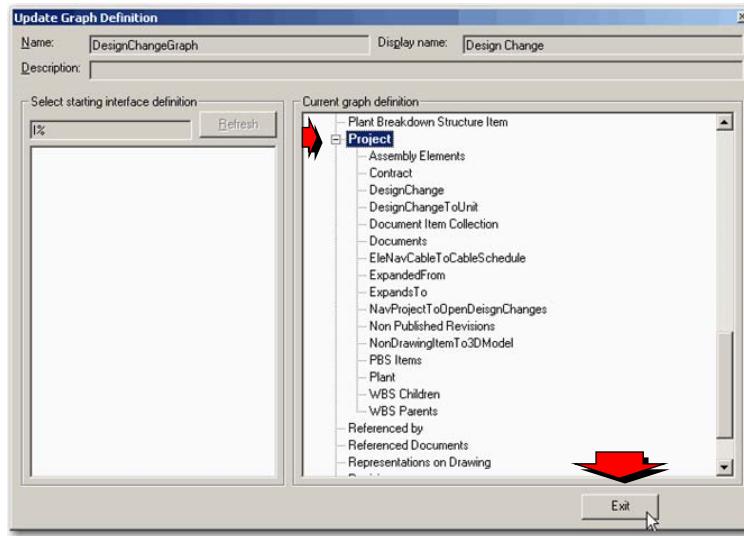
View the properties of any of this graph definition in the *Update Graph Definition* dialog.





Viewing Graph Definitions

Select the **Exit** button to close the **Graph Definition** dialog.



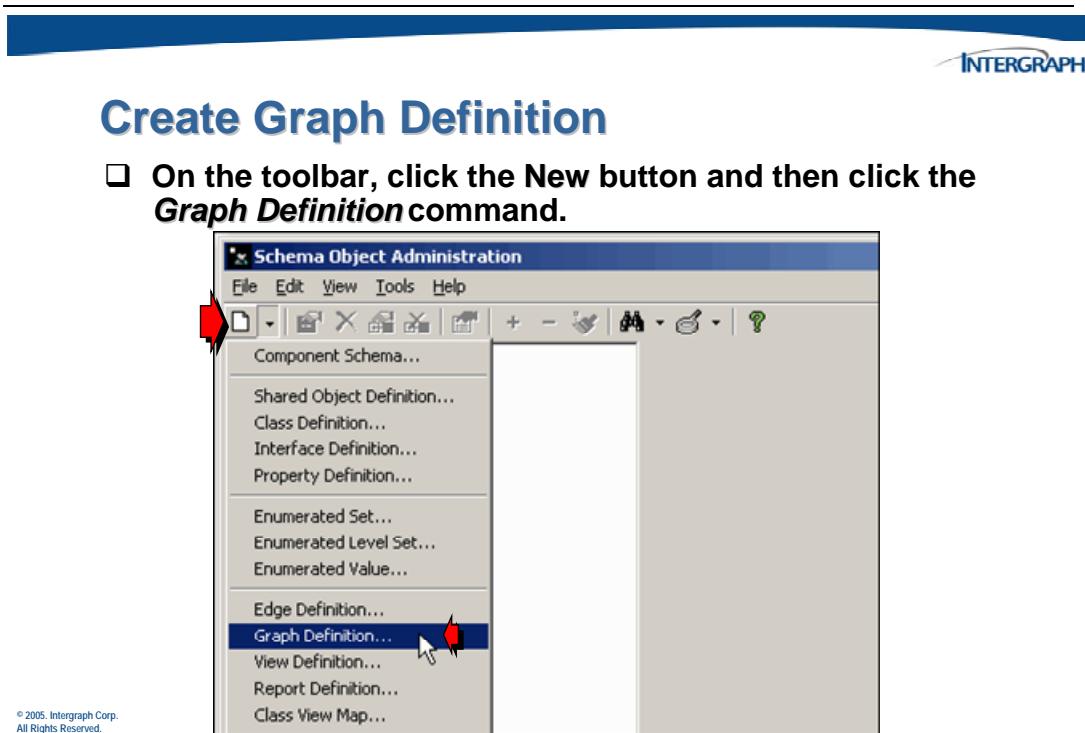
© 2005. Intergraph Corp.
All Rights Reserved.

2.6.7 Creating Graph Definitions

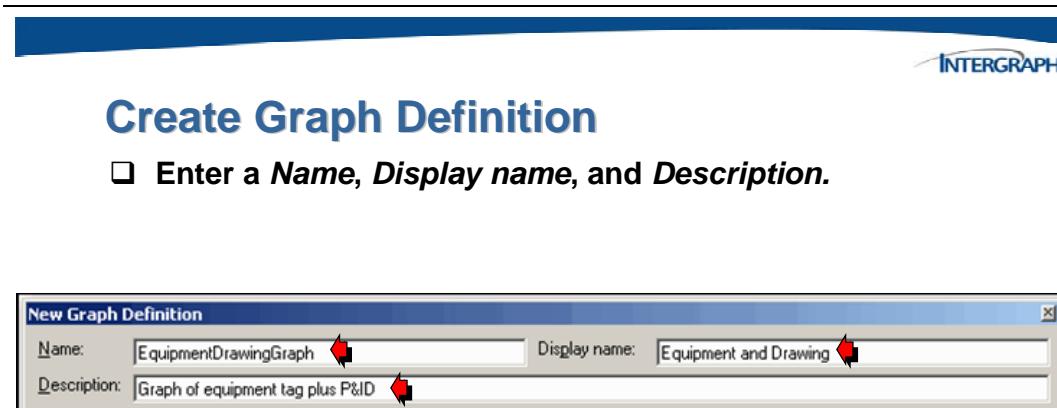
Each graph definition starts at an interface definition and branches out from that interface definition to related interface definitions. Any edge definition that is tied to the starting interface definition or to an interface definition implied (either directly or indirectly) by that interface definition can be used as part of the graph definition.

Likewise, when an edge definition is traversed as part of a graph definition, the ending interface definition (and its implied interface definitions) for that edge definition can also be used to add edge definitions to the directed graph definition. The directed graph definition is, therefore, a connected network of edge definitions starting at a particular interface definition.

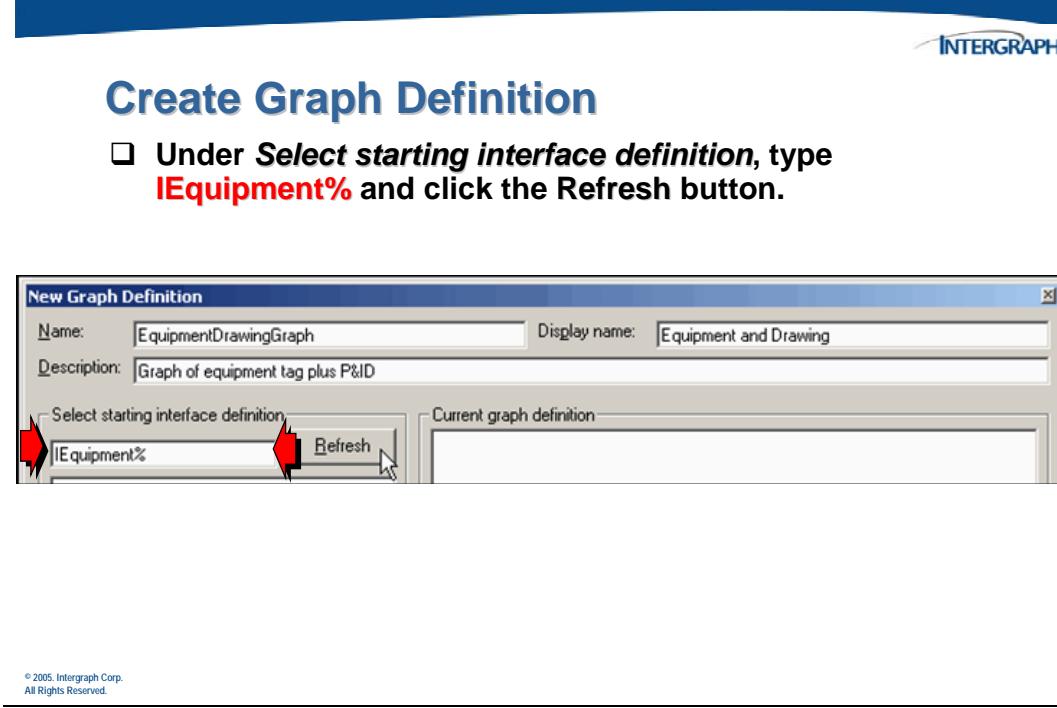
Below is an example of creating a GraphDef.



The *New Graph Definition* dialog box appears.



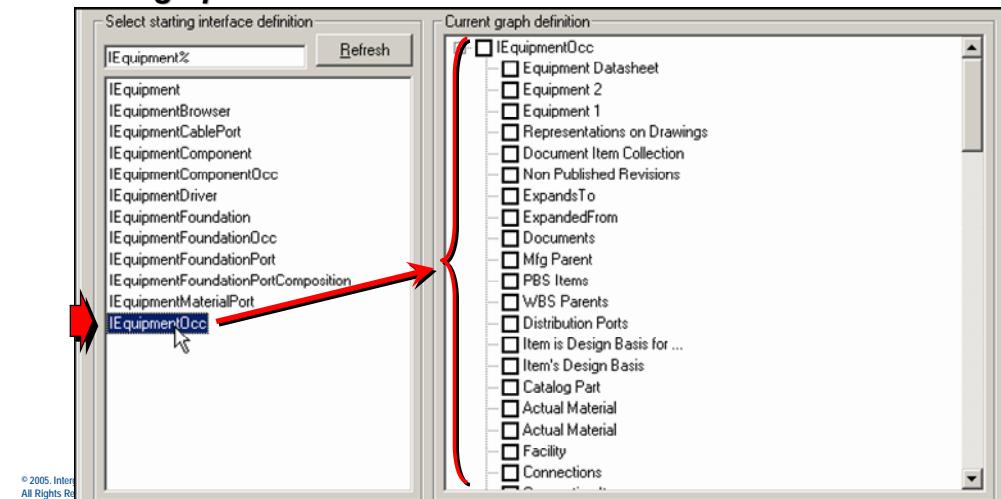
If you know all or part of the name of the interface that you want to use as the starting point for the graph definition, type your search criteria in the box. In the list, select the interface that you want to use as the starting point for the graph definition.





Create Graph Definition

- Click **IEquipmentOcc** to display the edges in the **Current graph definition** list.

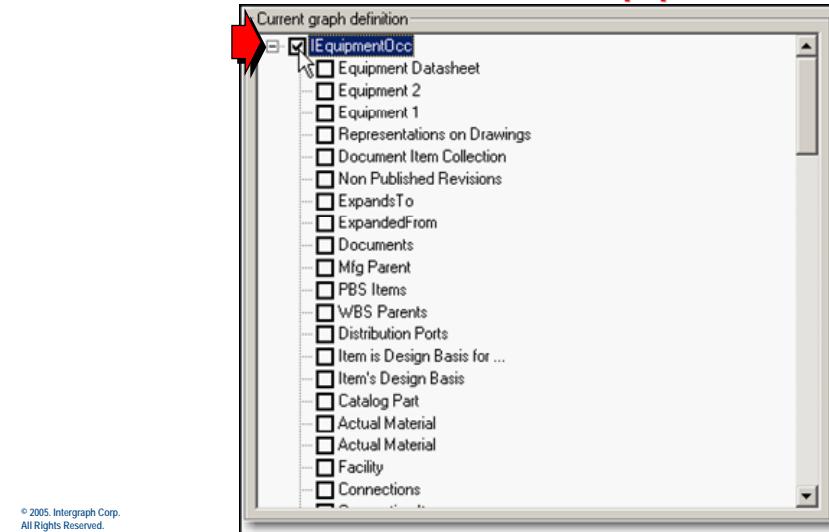


Under **Current graph definition**, select the check boxes beside the edge definitions that you want to include in this graph definition.

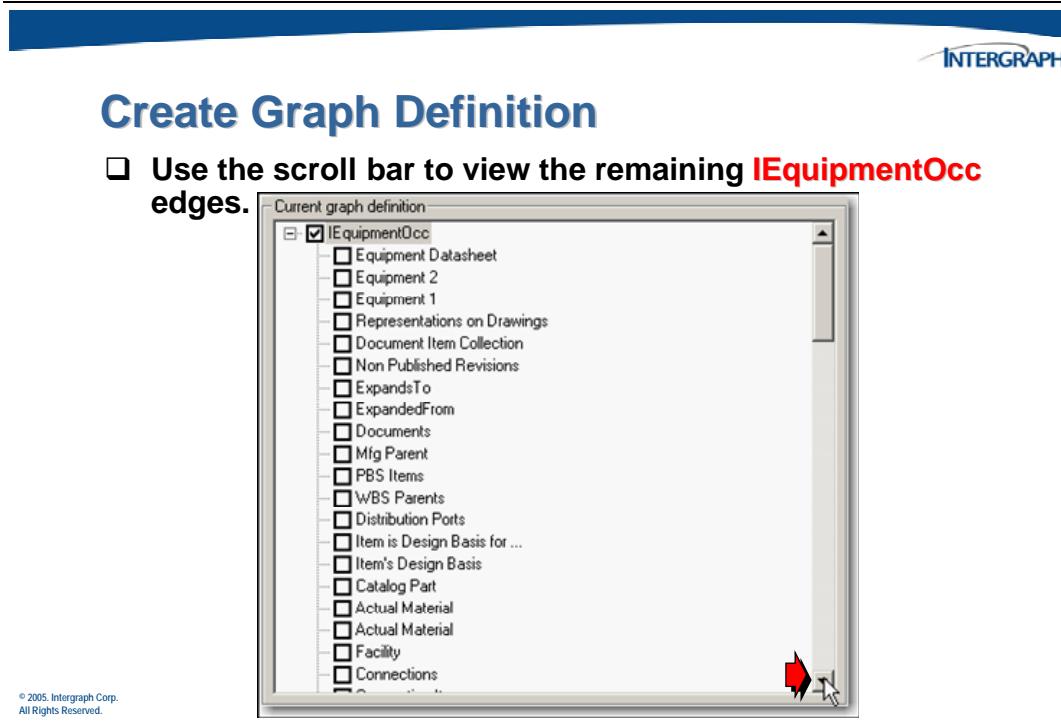


Create Graph Definition

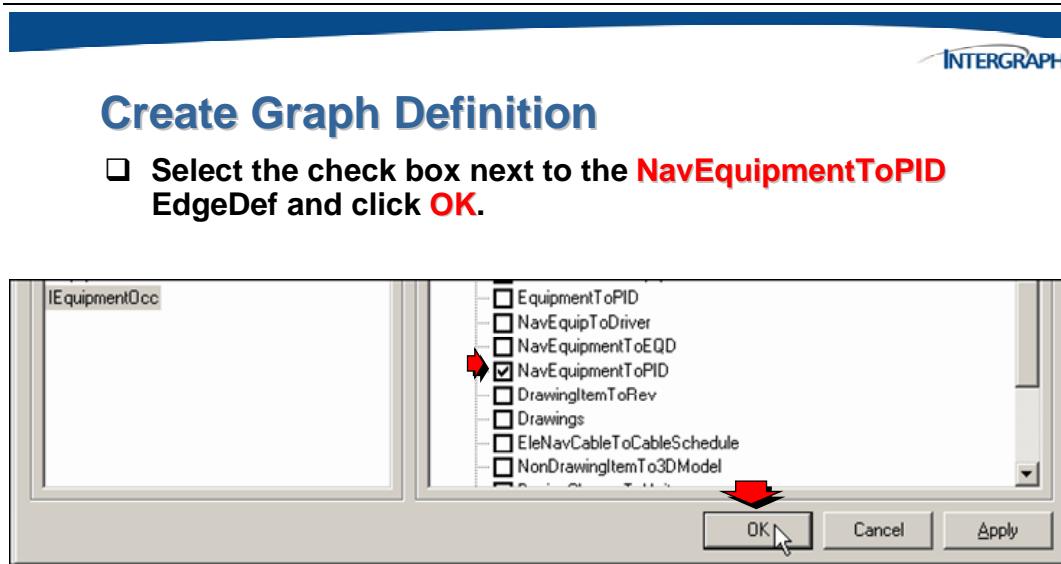
- Select the check box next to **IEquipmentOcc**.



Scroll down to see the rest of the edge definitions in the *Current graph definition* box.



Graph definitions that you create in Schema Object Administration are not copied back into the SmartPlant schema. They are only used in SmartPlant Foundation.



2.6.8 View Definitions in Schema Object Administration

View definitions are specific properties extracted from all the possible properties that a graph definition exposes. A view definition is used to provide a different view of data from that provided by the underlying schema. While schema-based depictions of the data are useful to users who are familiar with the underlying schema, other users need more user-oriented views of the data for the view to be meaningful.

View definitions are SmartPlant equivalent of relational database views. A relational database view is a combination of joins and projections where the joins are used to retrieve the desired objects and the projects are used to determine which properties (columns) to include and what to call those properties. SmartPlant equivalents of relational database joins are directed graph definitions. The projection of this graph data in SmartPlant is accomplished using view definitions.



View Definition

View definitions (ViewDef) are properties extracted from a list of possible properties exposed by a Graph Definition. A view definition is used to provide a different view of data from that provided by the underlying schema.

SmartPlant View Definitions consist of the following:

- A relationship to its starting interface definition**
- An identification of the directed graph definition that applies**
- A definition of the projection of property definitions from the directed graph definition**

A view definition is based on a directed graph definition and, therefore, like the directed graph definition, has a relationship to its starting interface definition. In actuality, this interface definition is always the same interface definition that the directed graph definition contains. The directed graph definition for the view definition defines the set of edge definitions that will be traversed when the view corresponding to the view definition is created.

2.6.9 Creating View Definitions

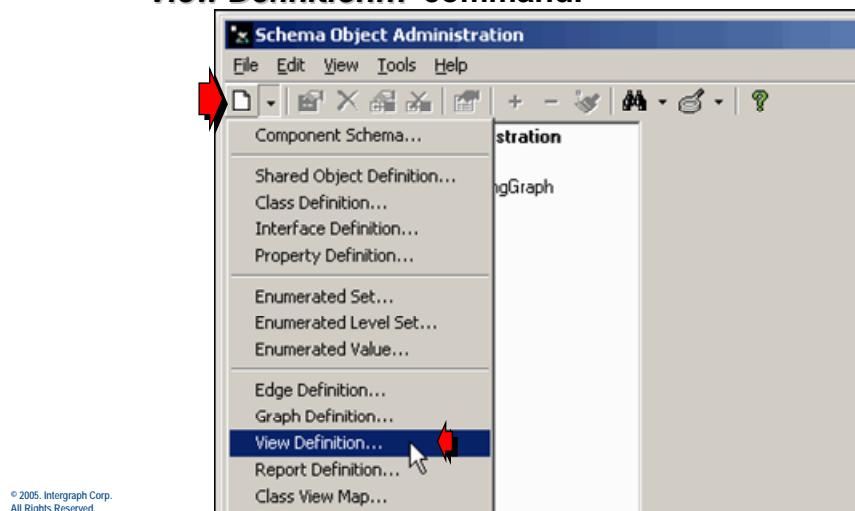
In SPF, view definitions are used to define what users see in the **Properties** window when they select an object in the client. User access to view definitions is defined by creating a class view map. ViewDefs also form the basis for Report Definitions.

Below is an example of creating a ViewDef.

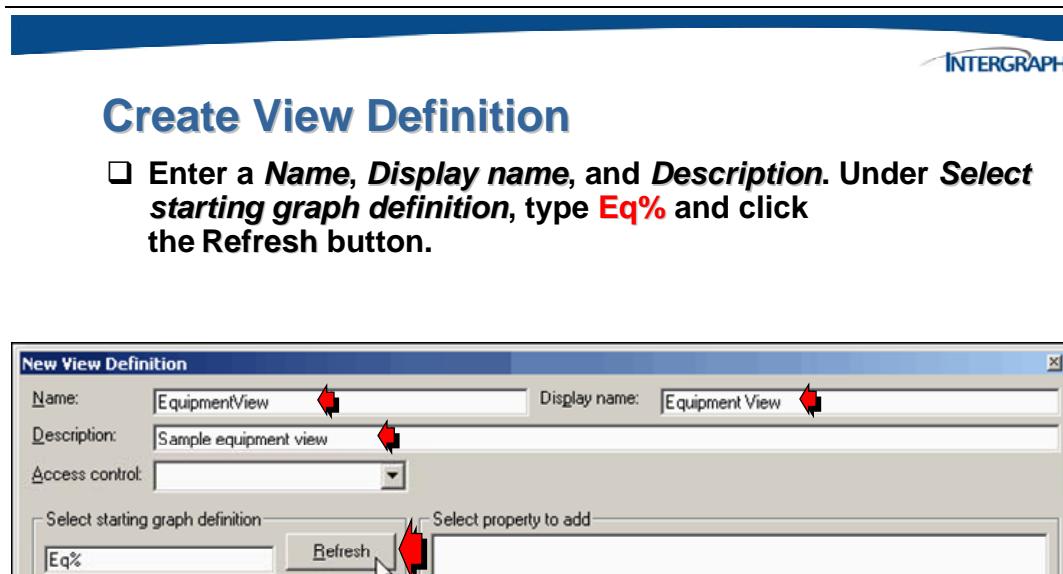


Create View Definition

- ❑ On the toolbar, click the New button and then click the **View Definition...** command.

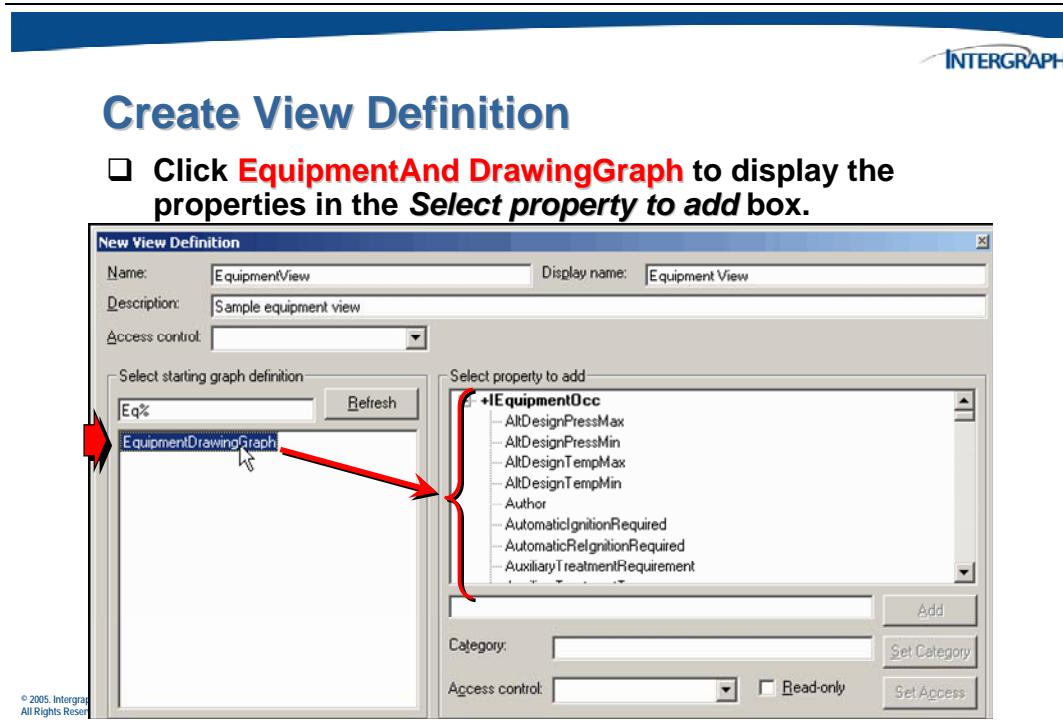


The *New View Definition* dialog box appears.



© 2005, Intergraph Corp.
All Rights Reserved.

If you know all or part of the name of the graph definition that you want to use as the starting point for the view definition, type your search criteria in the box.



Select a property that you want to include in your view definition.

The screenshot shows the 'Create View Definition' dialog box. On the left, there's a 'Select starting graph definition' dropdown set to 'Eq%' and a 'Refresh' button. The main area is titled 'Select property to add' and contains a tree view of properties from the selected graph definition. The 'Name' property is selected, highlighted with a red arrow. Below the tree view is a text input field labeled 'Tag No.' with a red arrow pointing to it. To the right of the input field are buttons for 'Add' (with a red arrow), 'Set Category', and 'Set Access'. At the bottom of the dialog are buttons for 'Category' and 'Access control'.

In the **Select property to add** box, click **Name**, then type **Tag No.** in the text box just to the left of the Add button.

Click the **Add** button.

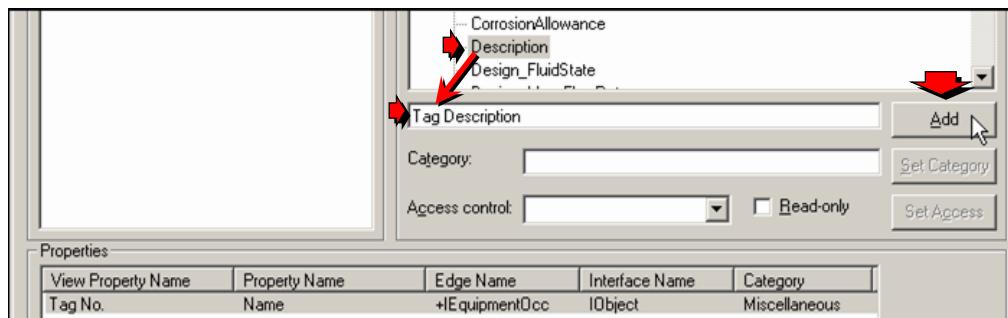
If you want to change the name of the property as it will be displayed in the SPF client, type the new name for the property in the box beneath the **Select property to add** tree view.

You have now added the *Name* property from the IEquipmentOcc InterfaceDef to the ViewDef, where it will be shown as **Tag No.**. Repeat this procedure for all properties from the selected graph definition that you want to include in your view definition.



Create View Definition

- In the **Select property to add** box, click **Description**, and then type **Tag Description** in the text box.



- Click the **Add** button.

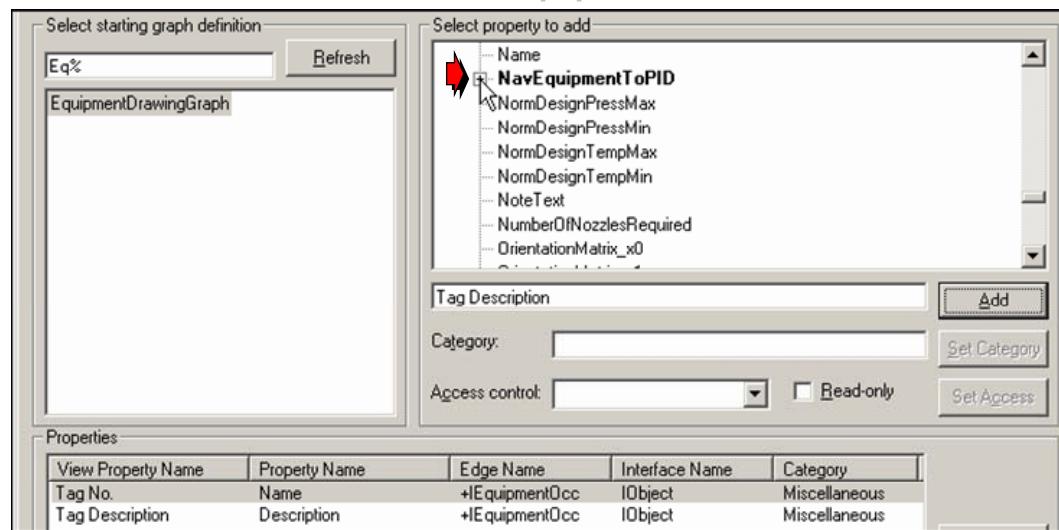
© 2005, Intergraph Corp.
All Rights Reserved.

Use the scroll bar to the right to find the desired properties to be added to the view.



Create View Definition

- Click to expand the **NavEquipmentToPID EdgeDef**.





Create View Definition

- ❑ Use the scroll bar to view the listed properties.

IEquipmentOcc



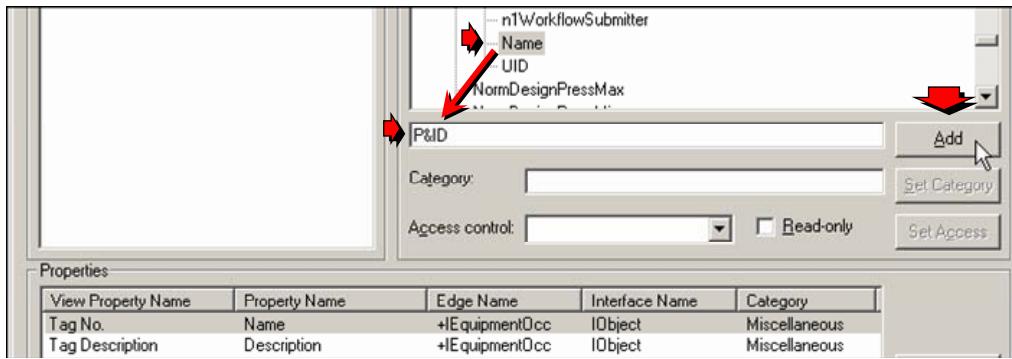
© 2005, Intergraph Corp.
All Rights Reserved.

Now add the name of the related P&ID (NavEquipmentToPID-Name) to the ViewDef, where it will be displayed as *P&ID*.



Create View Definition

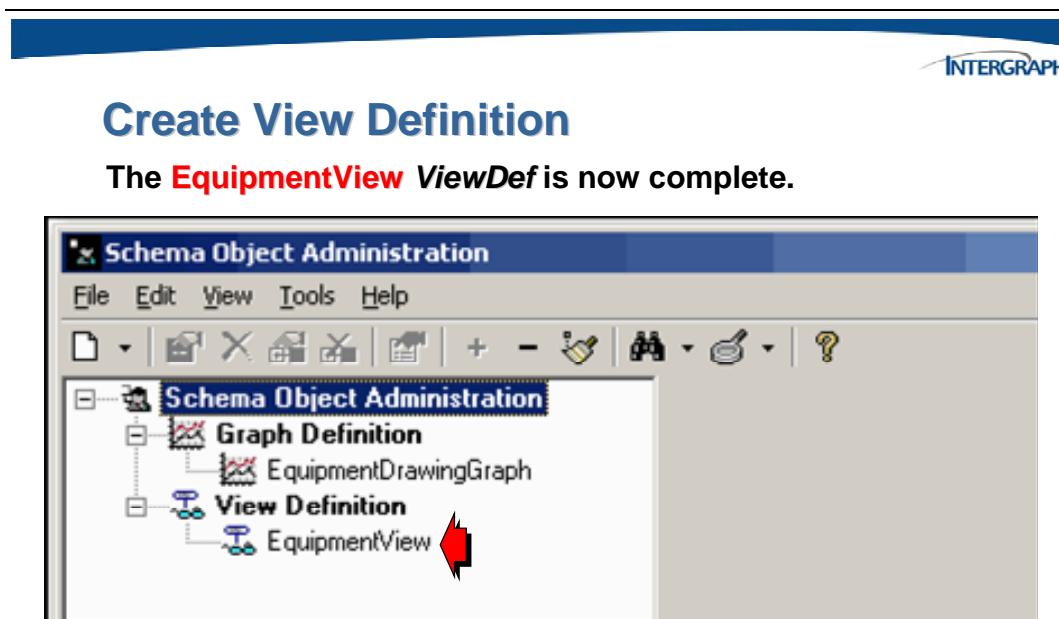
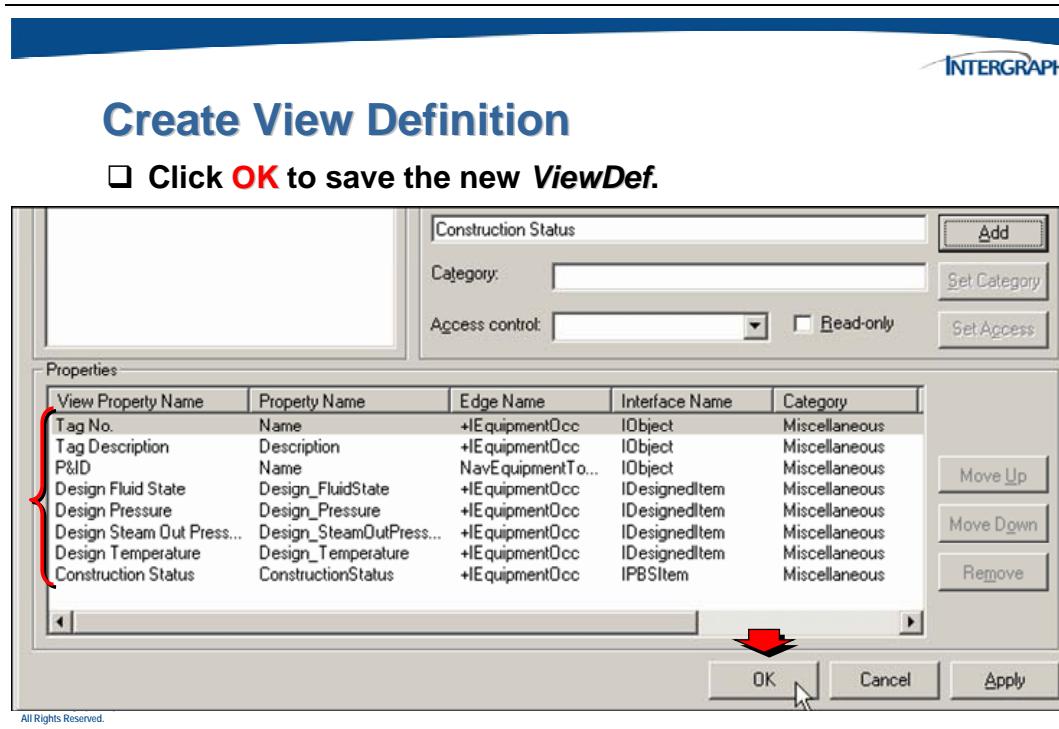
- ❑ In the **Select property to add** box, click **Name**, then type **P&ID** in the text box.



- ❑ Click the Add button.

© 2005, Intergraph Corp.
All Rights Reserved.

Add any additional properties needed for display using this new view def.



2.7 Class View Maps

SPF has a three-stage algorithm for determining the display of an item in the **Properties** window.

First, SPF looks for a **Class View Map** associated to the current user's user group. Second, if a Class View Map isn't found, SPF looks for a ViewDef with the same name as the ClassDef of the object and uses that in the **Properties** window. Finally, if neither a Class View Map nor a ViewDef is found, SPF creates a dynamic view based on all interfaces and properties that have been published by all tools.

The class view map specifies a set of class definitions and the default view definition that should be used for each one. Class view maps are defined in the SmartPlant schema using the SmartPlant Schema Editor or in the SPF system administration database using Schema Object Administration; a default set of class view maps is delivered with the SmartPlant schema. Then, you can associate user groups with the class view map in Schema Object Administration to define which view the software uses when a particular user selects a class definition in the SmartPlant Foundation client.

You can also use view definitions to create alternate views for classes.



Class View Map

A **Class View Map** is a collection of **ClassDefs** and **ViewDefs** in the schema that define the default **ViewDefs** used by the software for a set of **ClassDefs**.

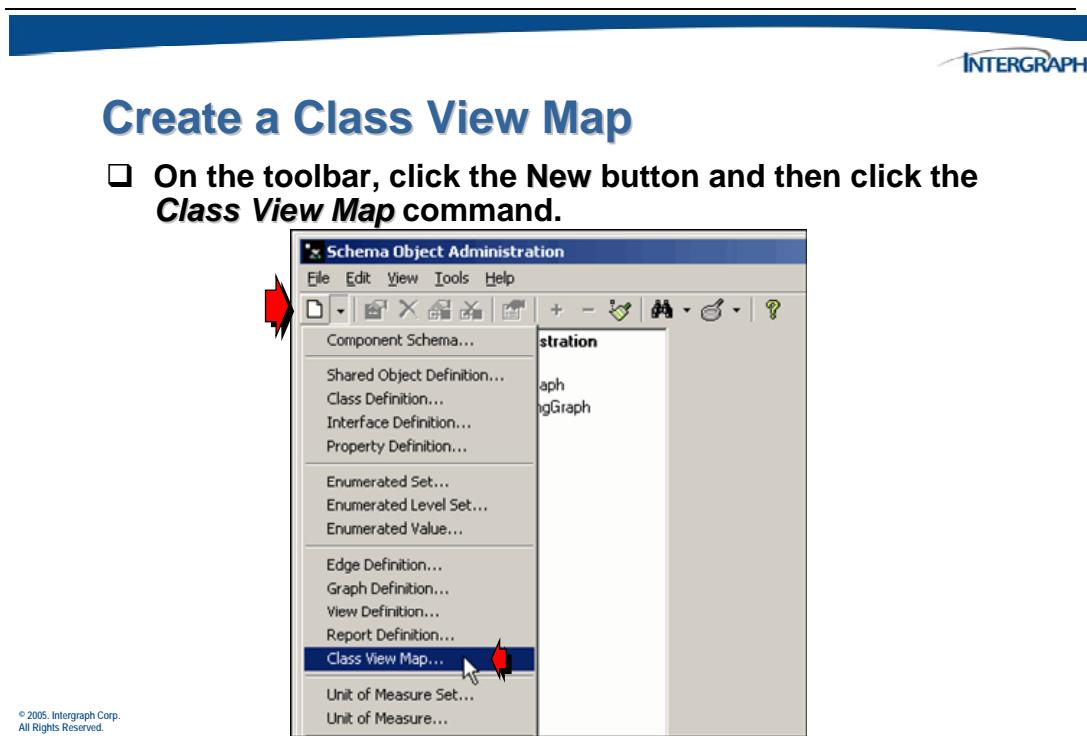
To create an alternate view for a class, you must do the following:

- Create a **ViewDef** (along with any needed graphs or edges)
- Create a **Class View Map** and map the ViewDef to the **ClassDef**
- Add a **User Group** to the Class View Map

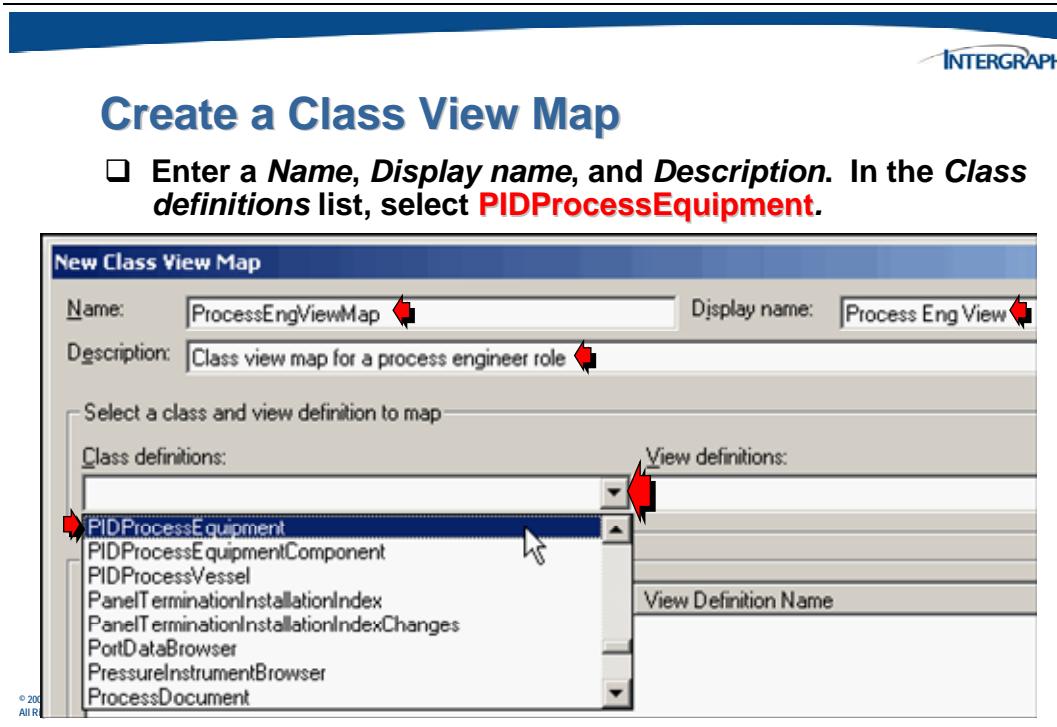
2.7.1 Creating Class View Maps

Defining a class view map is especially important for shared class definitions because it allows you to use the same view definition for multiple classes, including those objects that are shared across tools. By using the same view definition for multiple classes, users see the same presentation of information whether they are looking at the originally published class or a class from another tool for the shared object. For example, if SmartPlant P&ID publishes an object of the **PIDProcessEquipment** class and Zygad publishes an object of the **EQDCentrifugalPump** class, the shared object will have a different class in SmartPlant Foundation. To allow users to see the same view of the shared object regardless of which tool published it, the same view definition could be associated with both the class definitions.

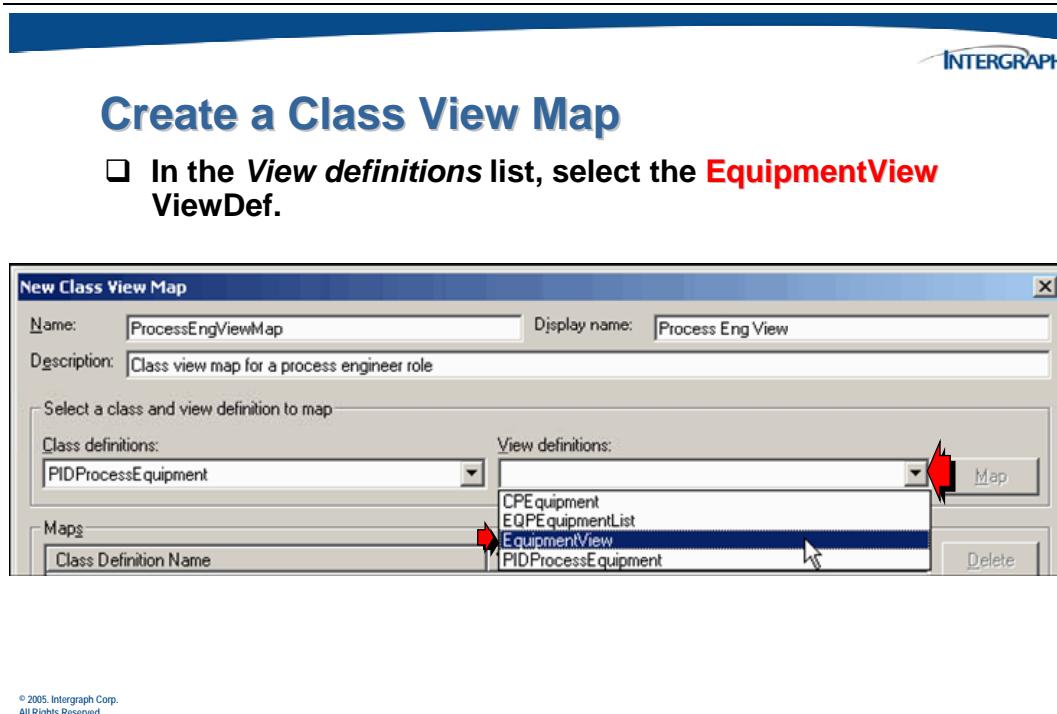
Below is an example of creating a Class View Map.



The *New Class View Map* dialog box appears. In the **Class definitions** list, select the class definition for which you want to create a class view map.



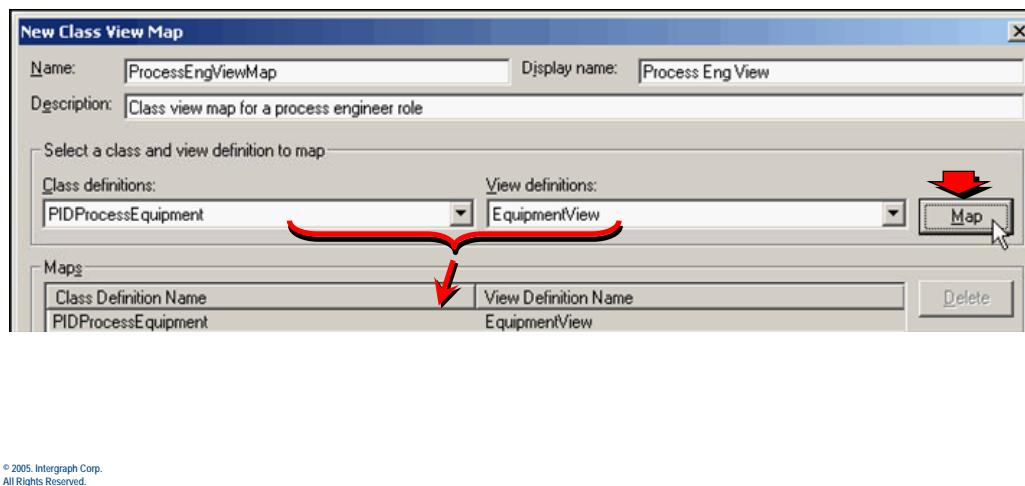
In the **View definitions** list, select the view definition that you want to use for the selected class definition.





Create a Class View Map

- Next, click the Map button.



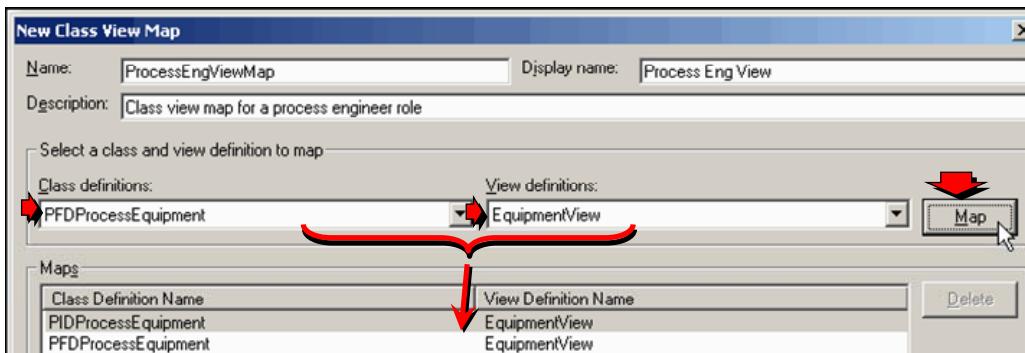
© 2005, Intergraph Corp.
All Rights Reserved.

After user access is assigned to the class view map, this class view map will specify that anytime a user in the *Process Engineer* user group clicks an object that is of ClassDef **PIDProcessEquipment** the software uses the *EquipmentView* ViewDef to display properties for the object in the **Properties** window. Map the *PFDProcessEquipment* ClassDef to the *EquipmentView* in the same way.



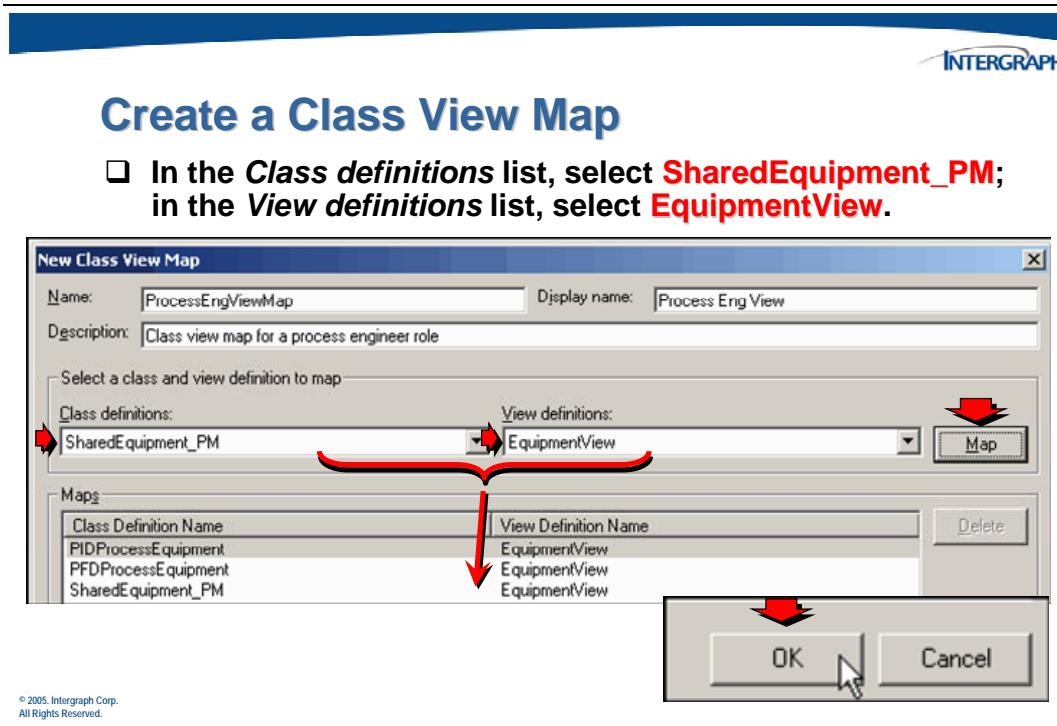
Create a Class View Map

- In the **Class definitions** list, select **PFDProcessEquipment**; in the **View definitions** list, select **EquipmentView**.



© 2005, Intergraph Corp.
All Rights Reserved.

Select the *SharedEquipment_PM* SharedObjDef in the **Class definitions** box and map it to the *EquipmentView* ViewDef.

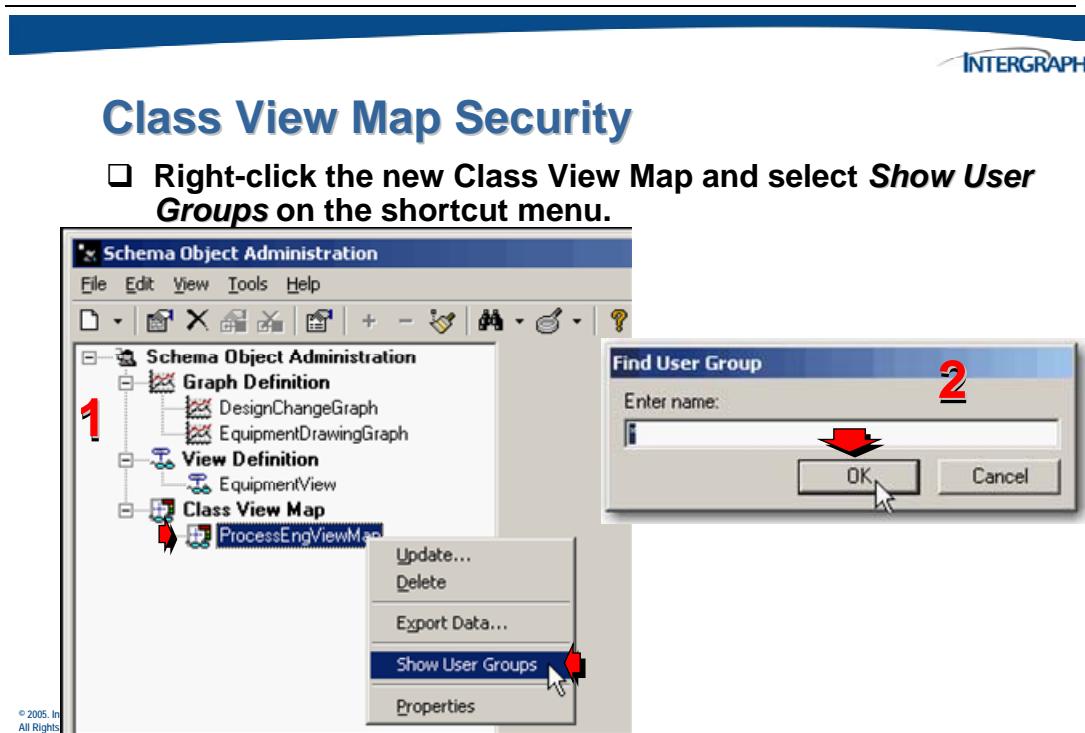


All the ways a piece of equipment may appear in a SmartPlant installation that has SmartPlant P&ID and Aspen Zygad have now been mapped.

2.7.2 Setting Class View Map Security

Associating a specific user group with a Class View Map means that those views are applied to members of the user group who can see the selected relationship in SPF.

You will define access security for Class View Maps using the same procedure as defining access security for EdgeDefs.



Locate the **Process User Group** and drag it into the tree view.

Class View Map Security

Drag and drop a *User Group* onto the *Class View Map User Groups relationship*.

User Group Name	User Group Description
ADMIN	
EFPLANT-SC-1_VA...	
EFPLANT-SC-2_VA...	
Instrumentation	Instrumentation
P&ID Design	P&ID Design
P&ID Engineering	P&ID Engineering
Piping	Piping
Process	Process
UPDATE	
VIEWONLY	

You have completed a ClassViewMap intended for a process engineer's view.

Class View Map Security

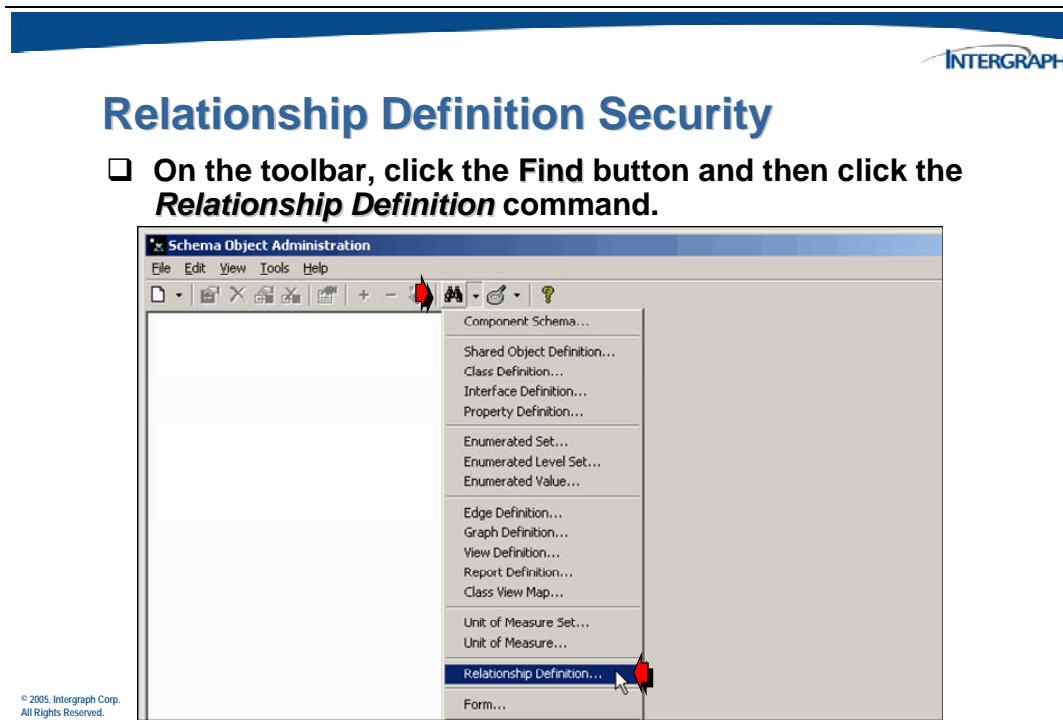
The **ProcessEngViewMap Class View Map** now has access security defined.

User Group Name	User Group Description
ADMIN	
EFPLANT-SC-1_VA...	
EFPLANT-SC-2_VA...	
Instrumentation	Instrumentation
P&ID Design	P&ID Design
P&ID Engineering	P&ID Engineering
Piping	Piping
Process	Process
UPDATE	
VIEWONLY	

2.8 RelDef Security

Associating a user group with a relationship definition means that only those users who are members of the user group can see the selected relationship in the SPF client if the **Filter out edge in display 1** and **Filter out edge in display 2** flags are set to false for the relationship definition.

You will define access security for RelDefs using the same procedure as defining access security for EdgeDefs.



Perform a search to locate a *RelDef* such as **PlantProjects** and review its properties.



Relationship Definition Security

- ❑ Enter the desired search criteria and click **OK**.

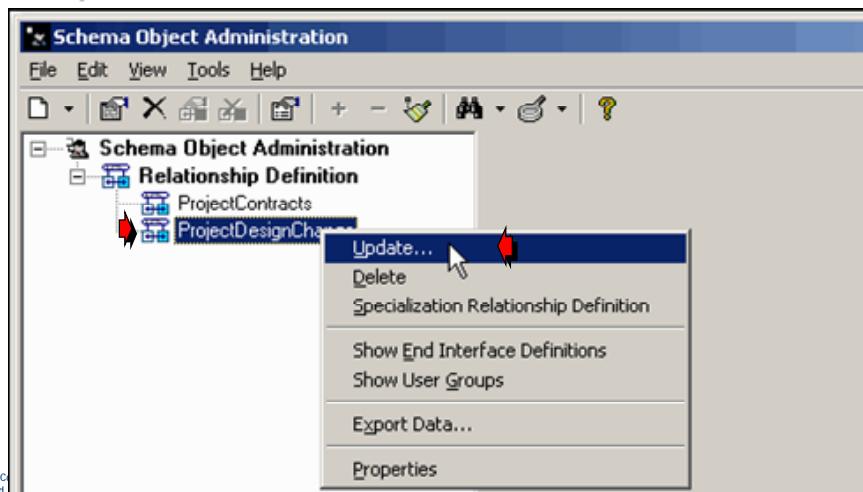


© 2005, Intergraph Corp.
All Rights Reserved.

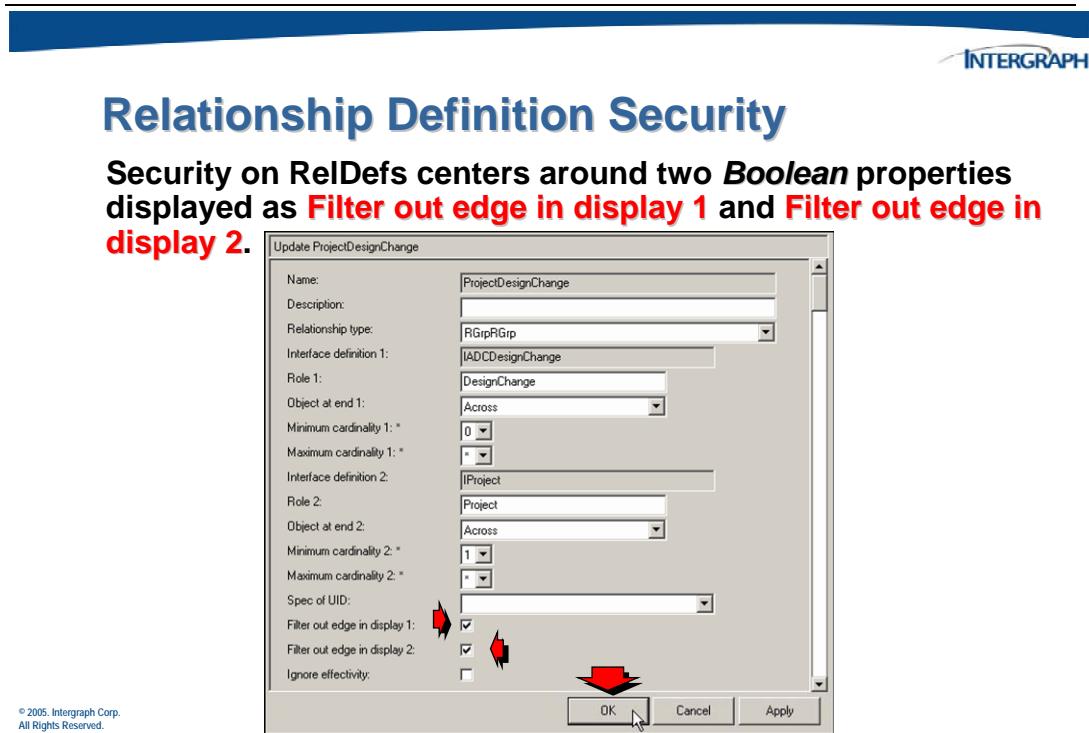


Relationship Definition Security

- ❑ Right-click the **ProjectDesignChange** RelDef and click **Update** on the menu.

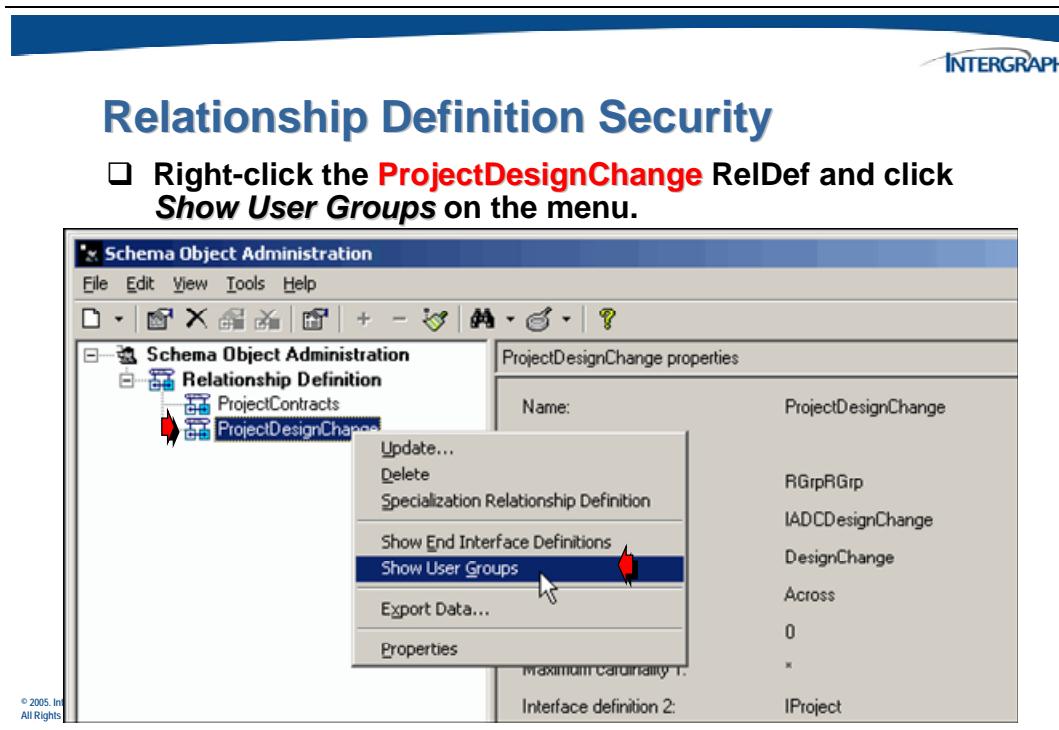


The first Boolean property administers security on the forward edge and the second flag administers security on the reverse edge.

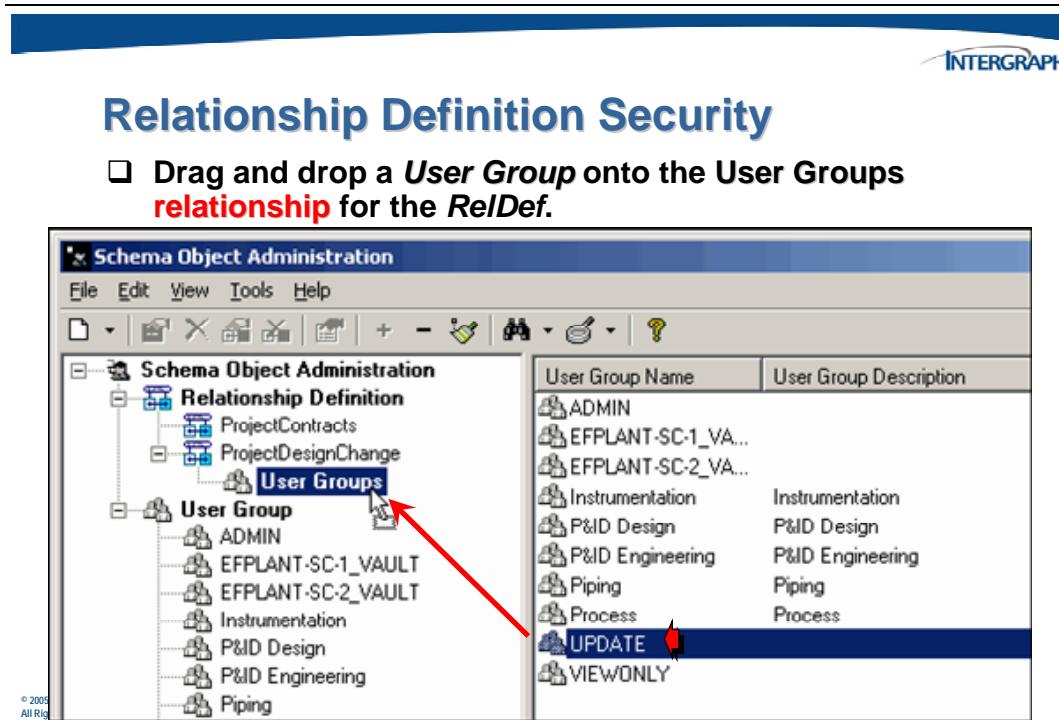


If a filter edge in display property is false, it means DO NOT apply security, show it to all users when they use right-click an object. If it is true, it means honor the user group settings on the RelDef.

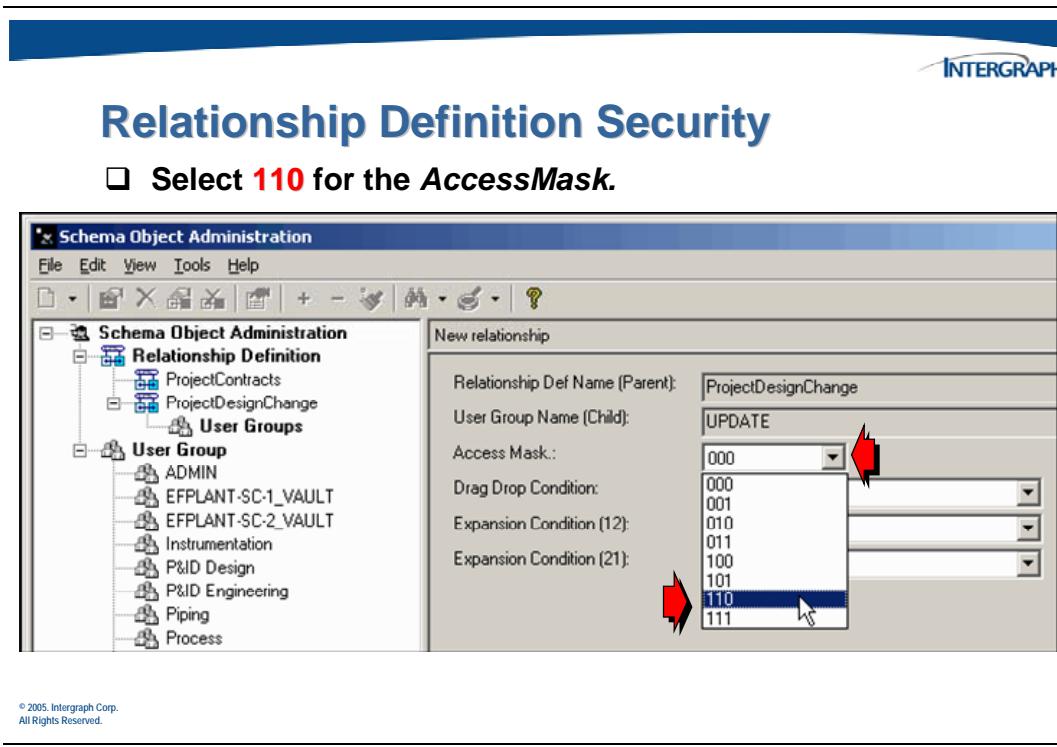
As an example, give access to this RelDef to the *UPDATE* user group. Find the *UPDATE* user group and drop it on the **User Groups** node in the tree view.



Perform a search to find the **User Group** objects.



The *New relationship* form appears.



The *access mask* consists of a **three-digit** number. Each digit defines the type of access users in the user group should have under specific circumstances.

The **first digit** indicates whether the relationship definition appears if the user expands the first interface definition on the associated relationship definition. A **1** means that the relationship definition should appear under these circumstances; a **0** indicates that it should not appear.

The **second digit** indicates whether the relationship definition appears if a user expands the second interface definition on the associated relationship definition. A **1** means that the relationship definition should appear under these circumstances; a **0** indicates that it should not appear.

The **third digit** indicates whether the user can perform a drag-and-drop operation on the relationship definition. A **1** indicates that a drag-and-drop operation can be performed; a **0** indicates that it cannot.

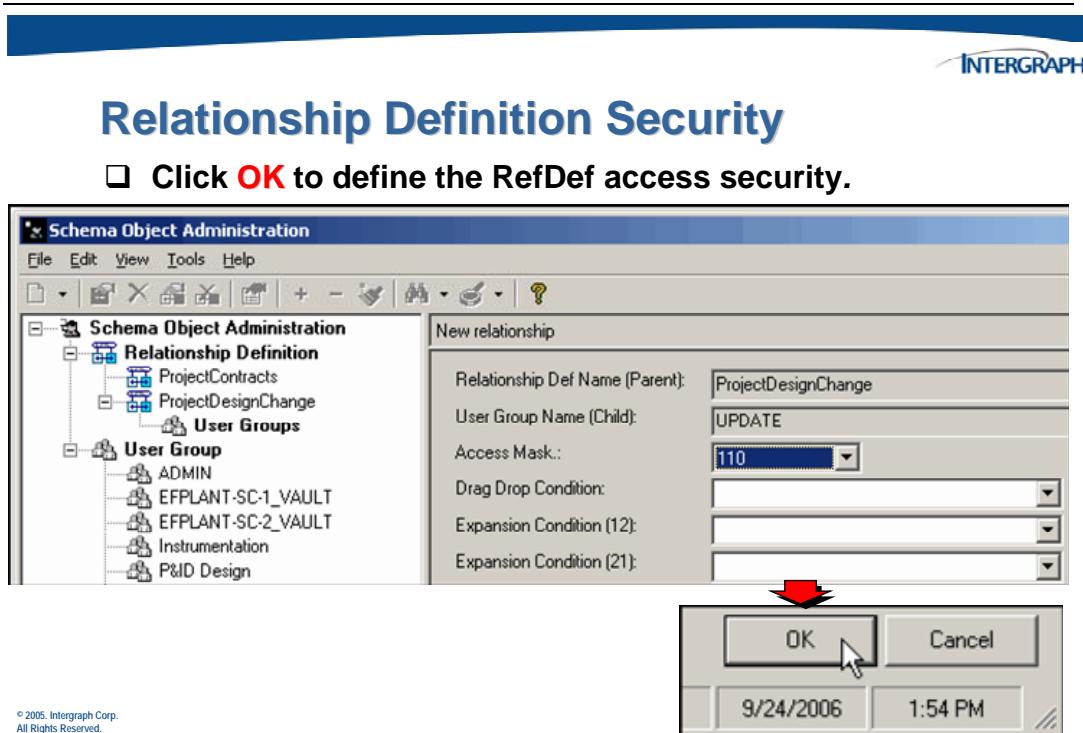
For example, **000** in the **Access Mask** box means that users in this group have no access to the relationship definition, while **110** means that users in the user group can expand the relationship from both interface definitions but cannot perform drag-and-drop operations on the relationship definition.

The **Access Mask** and **Drag Drop Condition** options serve to further define the security for this RelDef. The three bits in the AccessMask correspond to the following:

- Allow the User Group to browse the forward edge

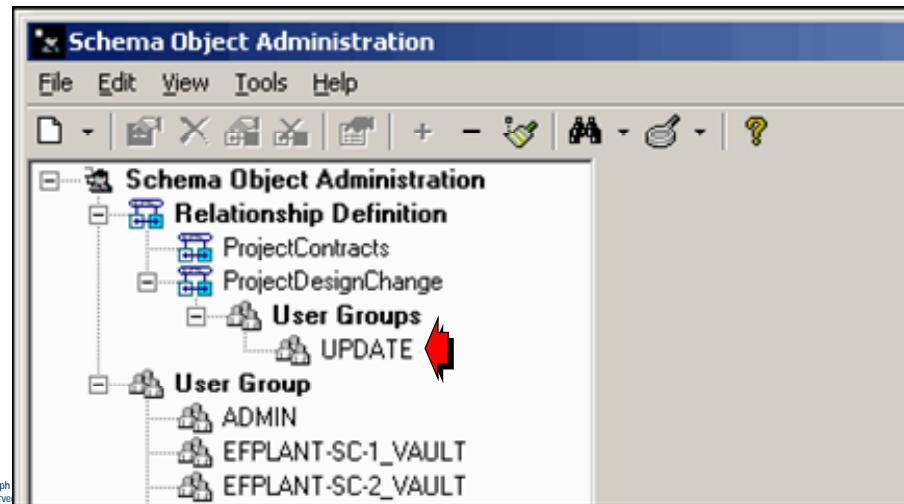
- Allow the User Group to browse the reverse edge
- Allow the User Group to create this RelDef through a drag-drop operation

The condition is applied to two objects to further determine if drag-drop creation is allowed.



Relationship Definition Security

The **ProjectDesignChange RelDef** now has additional access security defined.



The *UPDATE* user group should now be able to browse from Plants to Projects and vice versa.

2.9 Access Control

Access Controls are used to control user access to individual display items on the create and update dataforms in SmartPlant Foundation. Display items do not have access control defined, which means that any user has access to that display item. However, if access control is applied, then a user only has access to the display item if they are a member of one of the user groups defined in the access control list.



Access Control

Access controls are used to control who has access to data fields such as those on a form.

The **Access Control** characteristics are:

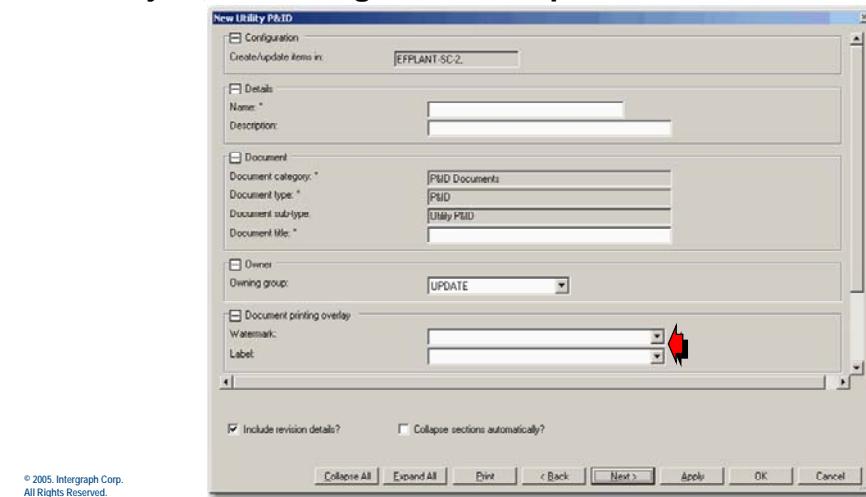
- Access control name
- Access control description

User Groups are associated to **Access Controls** using a drag and drop operation



Access Control

Form field security can be applied using an Access Control object, restricting access to specific members of a User Group

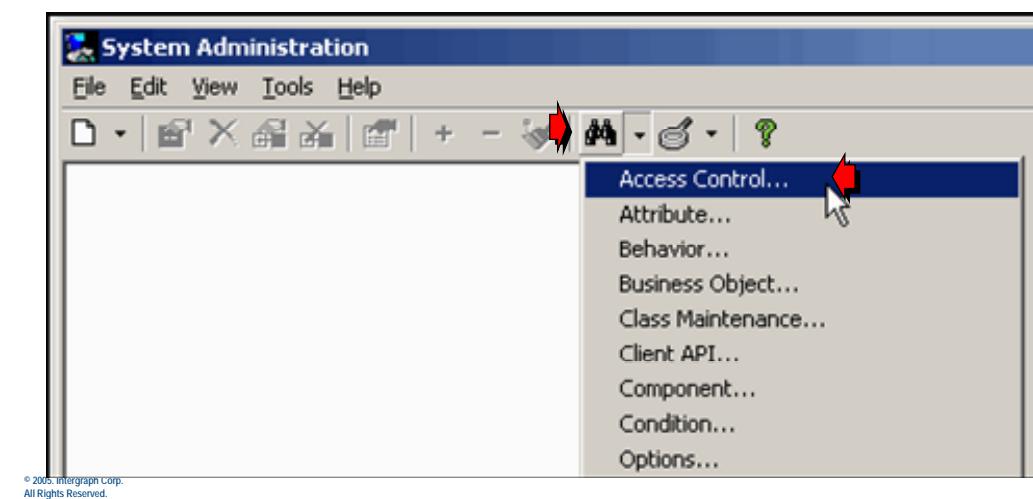


To review the existing *Access Control* objects, select the **Find** tool from the toolbar and then **Access Control** from the menu.



Access Control

- On the toolbar, click the **Find** button and then click the **Access Control...** command.





Access Control

- Enter the desired search criteria and click **OK**.



© 2005, Intergraph Corp.
All Rights Reserved.

Click **OK** to perform the query.



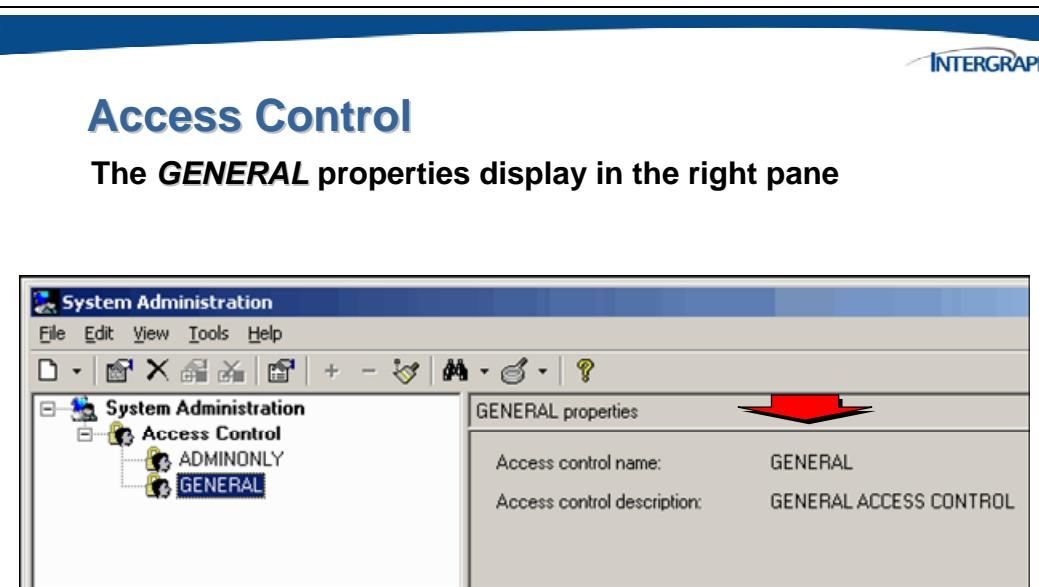
Access Control

- Right click on an **Access Control** object and select **Properties** from the pop up menu



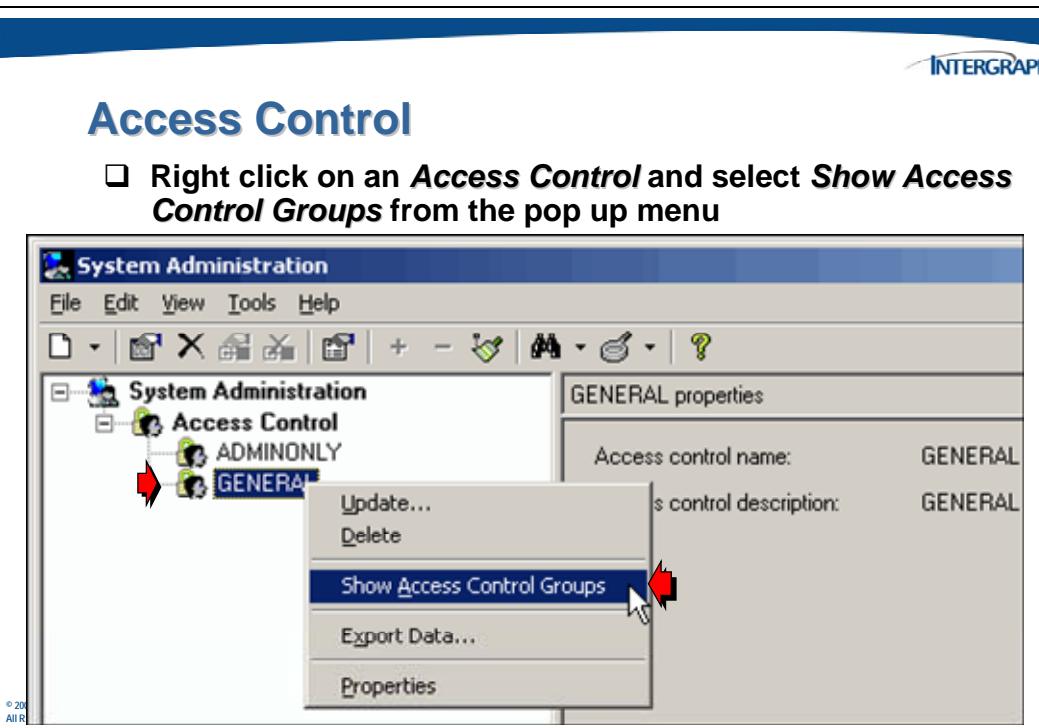
© 2005, Intergraph Corp.
All Rights Reserved.

This will allow you to review the *Access Control* properties.



© 2005, Intergraph Corp.
All Rights Reserved.

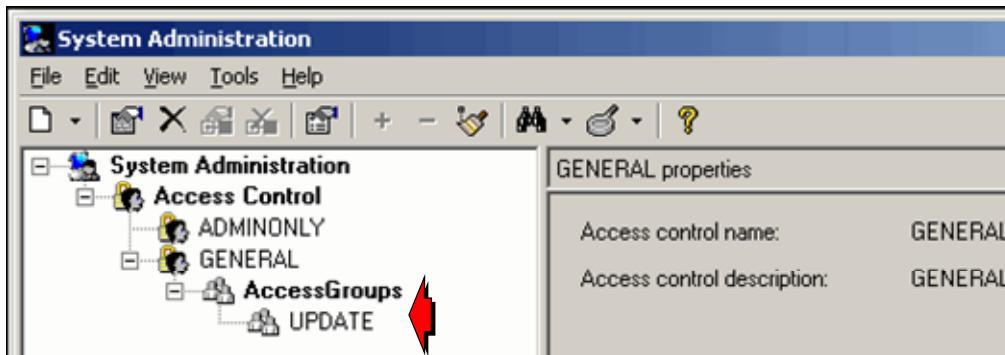
Continue to review the *Access Control* relationships.





Access Control

The **AccessGroups** relationship is displayed in the **Tree View**



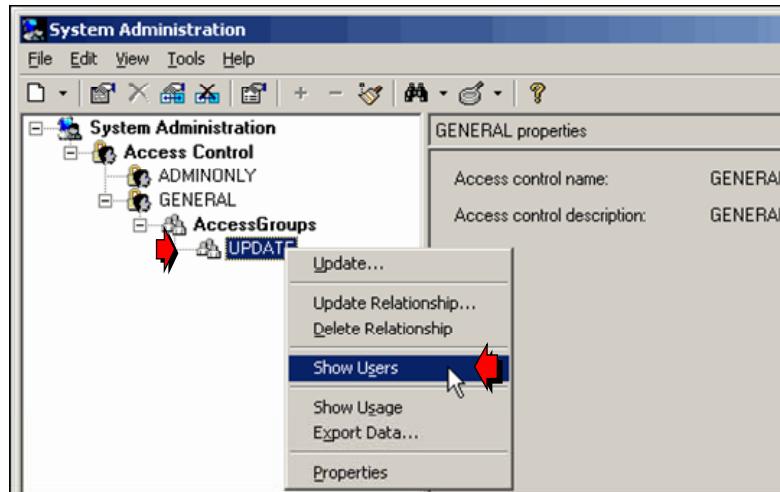
© 2005, Intergraph Corp.
All Rights Reserved.

Next, review the users related to the **AccessGroup**.



Access Control

- Right click on an **AccessGroup** and select **Show Users** from the pop up menu

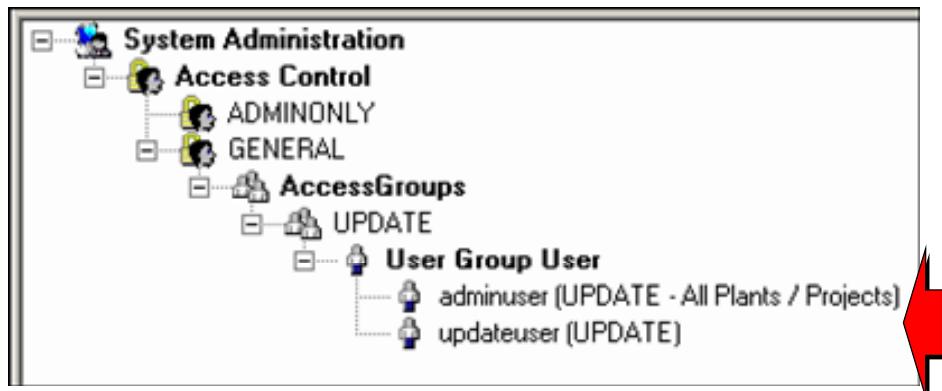


© 2005, Intergraph Corp.
All Rights Reserved.



Access Control

The **User Group User** relationships are displayed in the **Tree View**



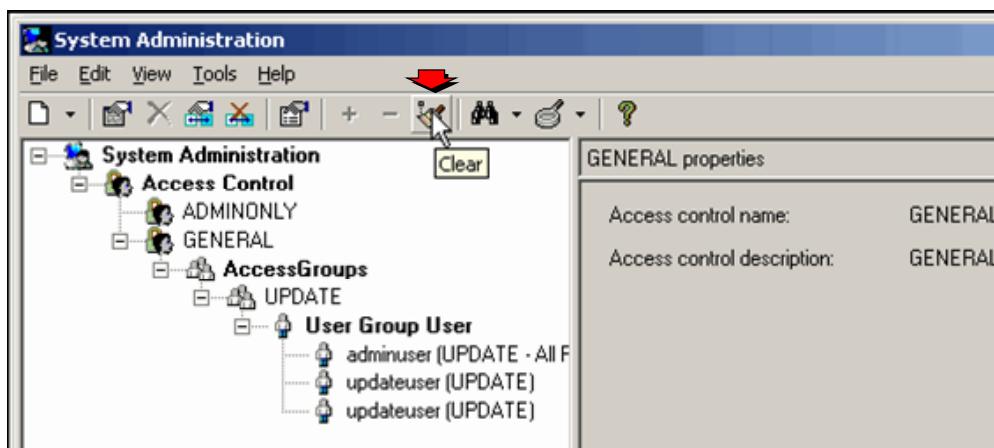
© 2005, Intergraph Corp.
All Rights Reserved.

Before creating any new *Access Control* objects you may want to clear the Tree View display.



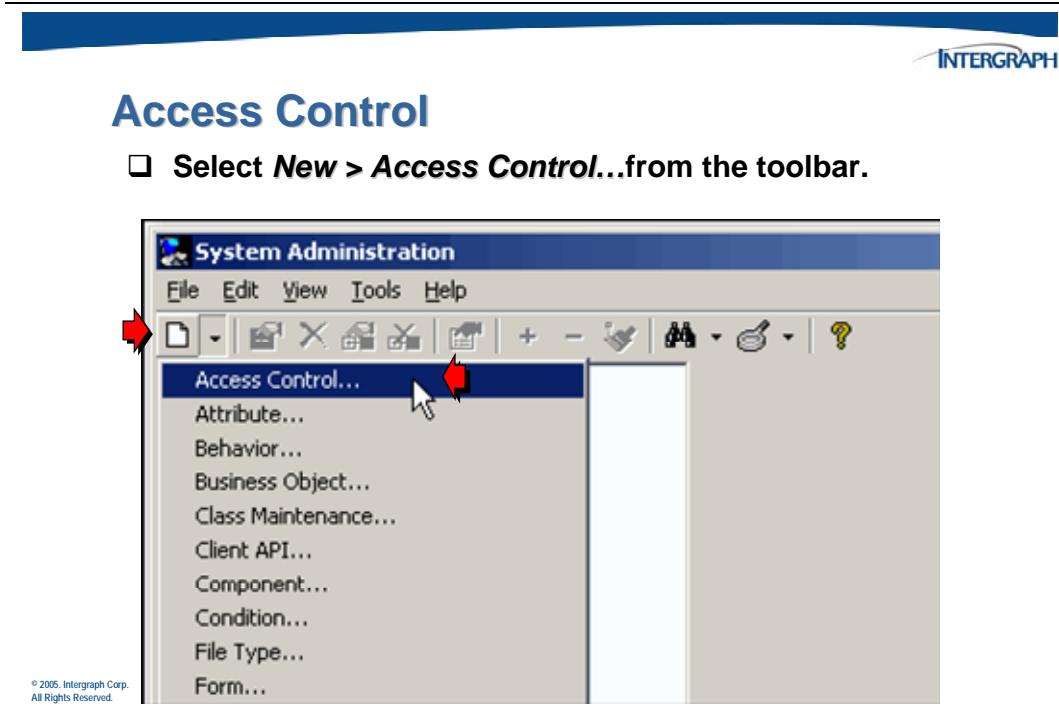
Access Control

- Select the **Clear Treeview tool** to reset the **Tree View**

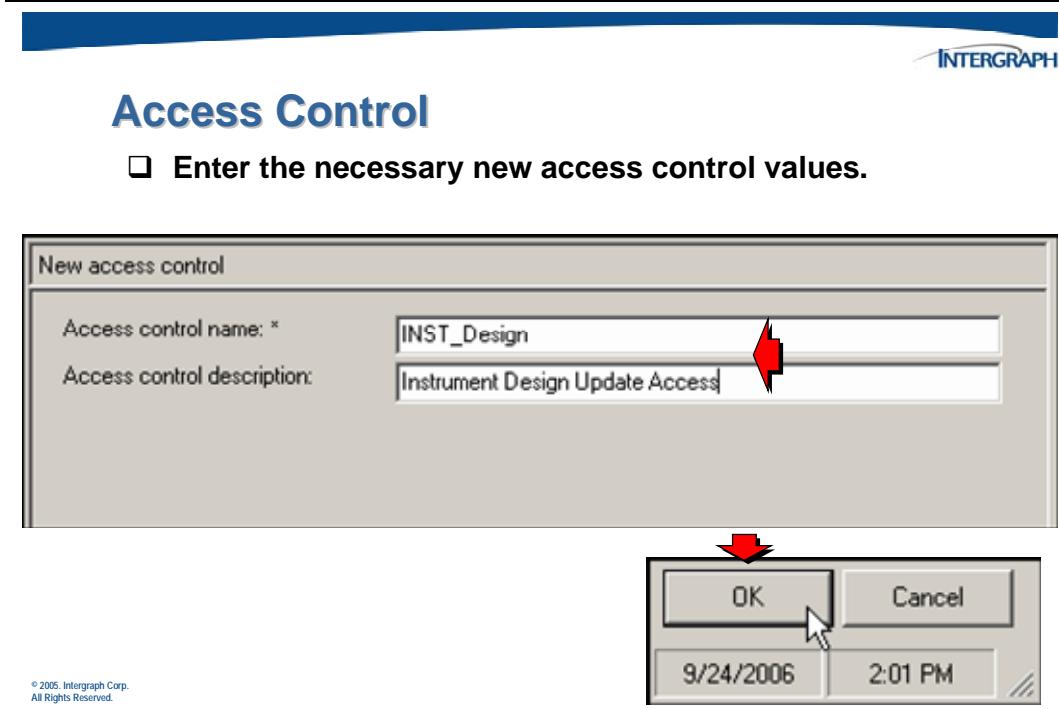


© 2005, Intergraph Corp.
All Rights Reserved.

These objects are used by forms to refer to access controls. To create a new *Access Control* object:



The *New access control* form will display in the right pane.

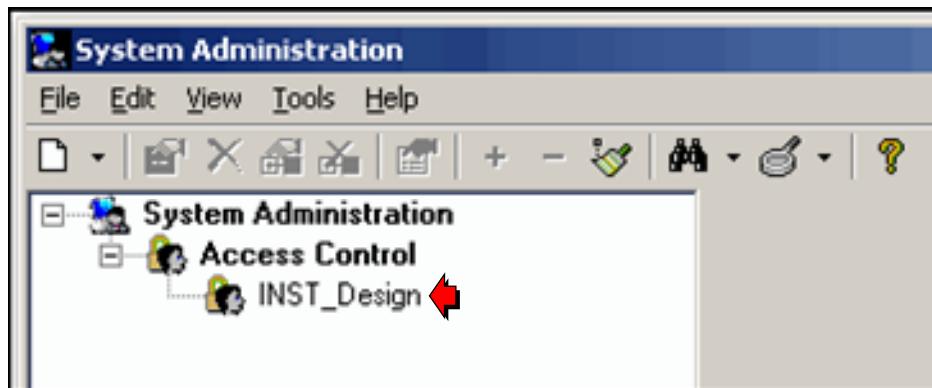


Click **OK** to create the new access control object.



Access Control

- The new access control will be added to the **Tree View**



© 2005, Intergraph Corp.
All Rights Reserved.

Now add *access control groups* to the new access control object.



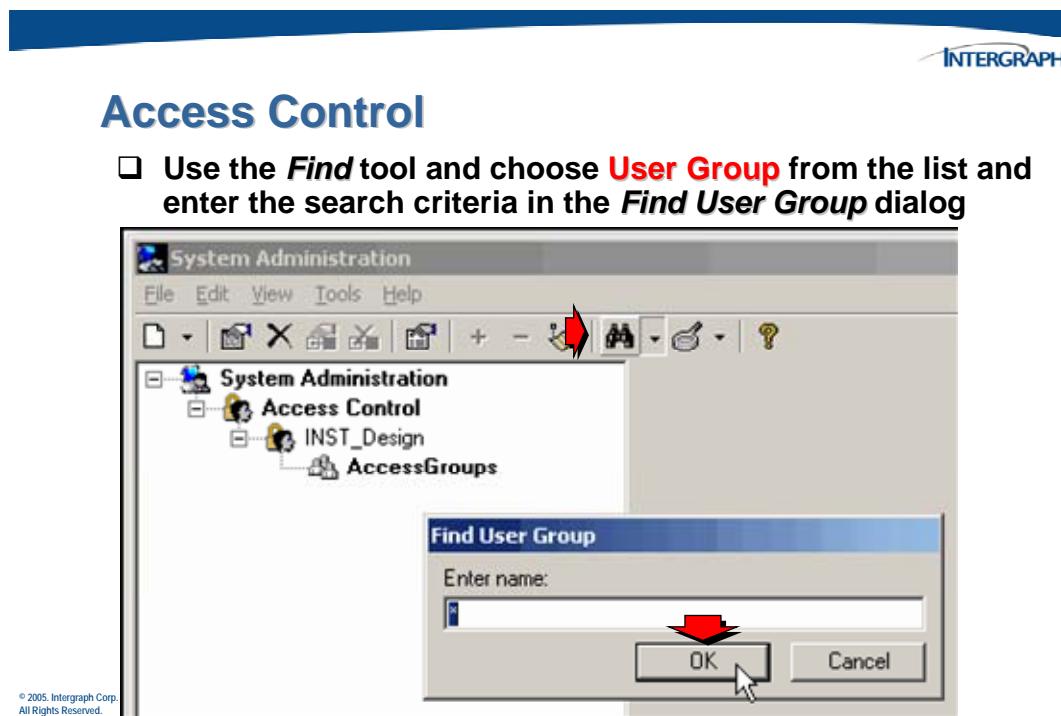
Access Control

- Right click on the access control name and select **Show Access Control Groups** from the menu

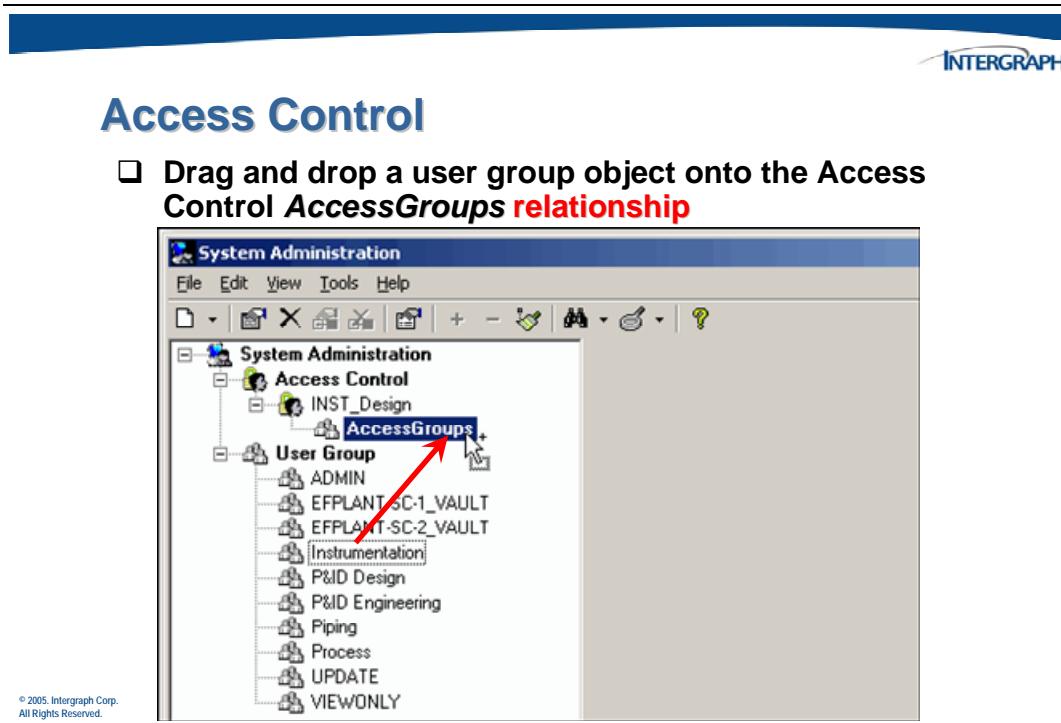


© 2005, Intergraph Corp.
All Rights Reserved.

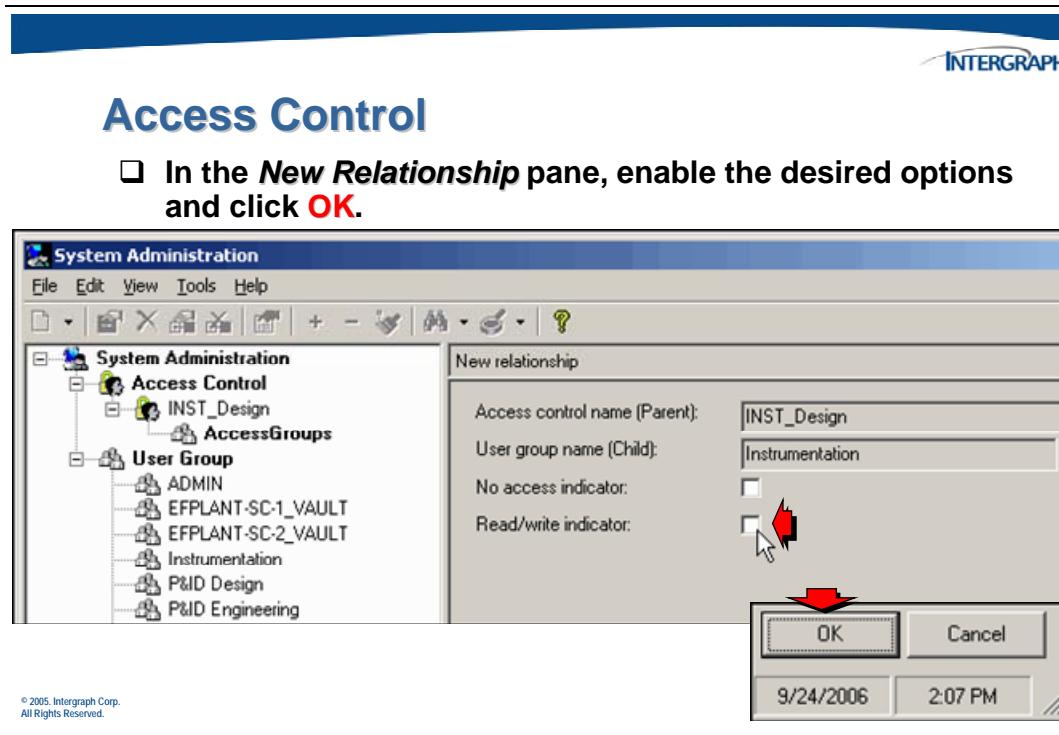
Perform a search to locate the existing user groups. The *Find User Group* dialog will display.



Select a user group to add to the access control object.



The *New relationship* form will display.

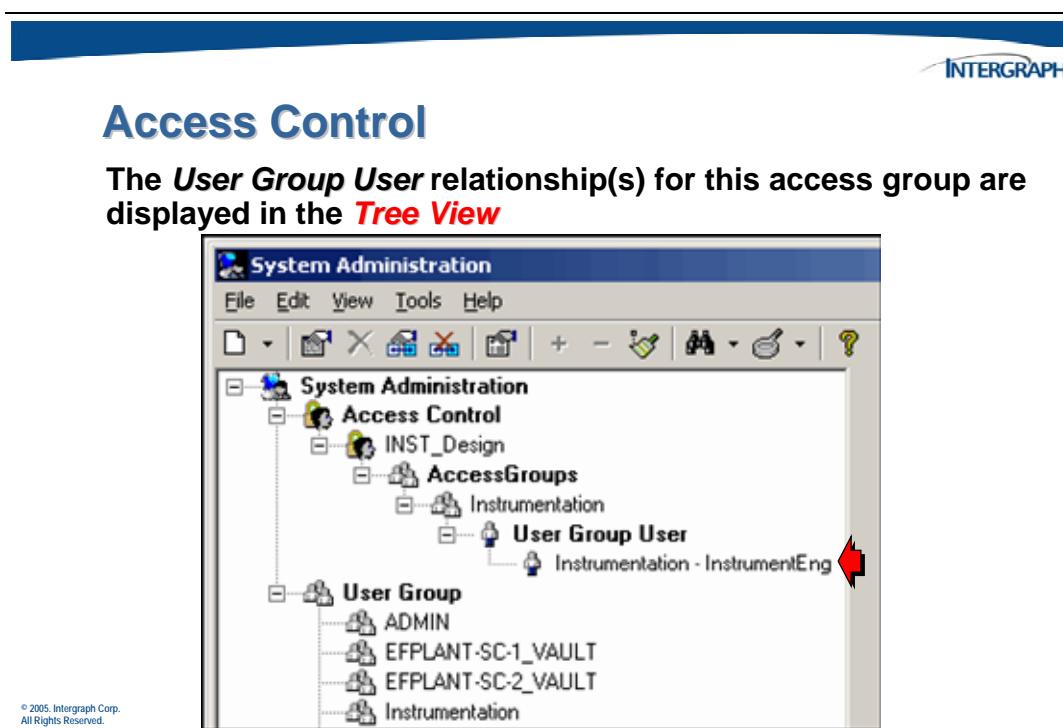
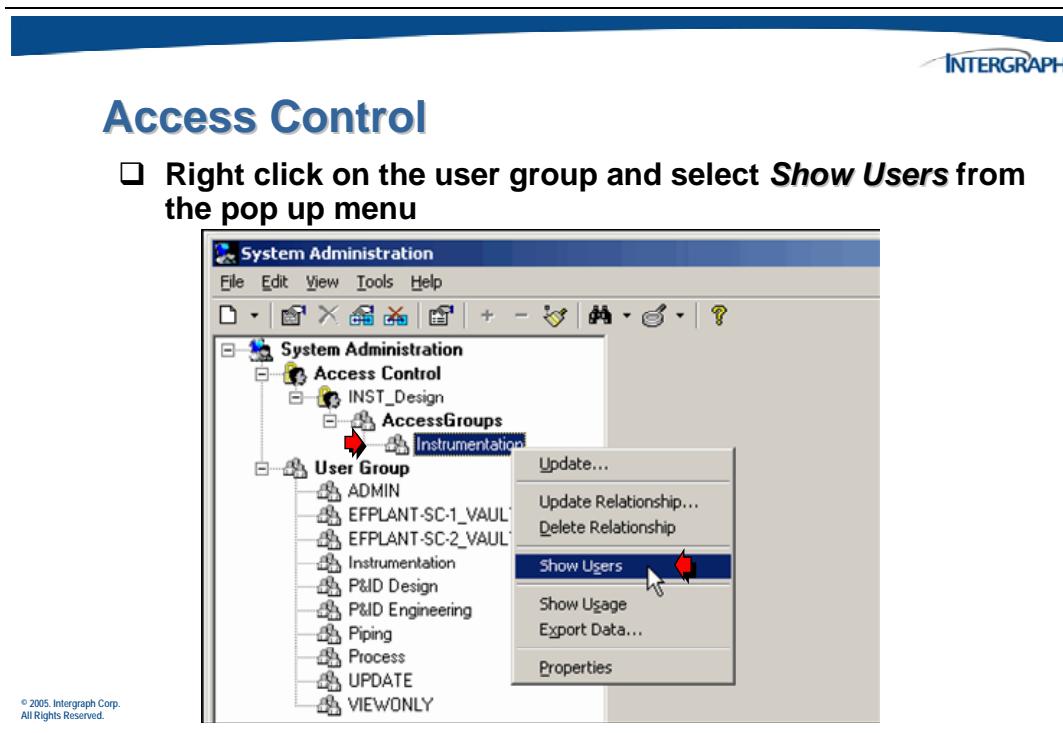


No access indicator - Indicates that users who are members of the selected user group will not have access to any object that has this access control applied to it. For example, a display item with this access control would not be visible to users in the selected user group.

Read/write indicator - Specifies how users in the selected user group see data associated with this access control if the *No access indicator* check box is not selected. If the Read/write indicator check box is selected, users in the user group have write access to objects. If the option is not selected, users see objects with the access control as read-only. For example if the Read/write indicator option is turned on and the display item is added to the section with read/write access, users have read and write access to the item; otherwise, the user will have read-only access.

Note: If both the *No access indicator* and *Read/write indicator* options are turned off, users have read-only access to the object.

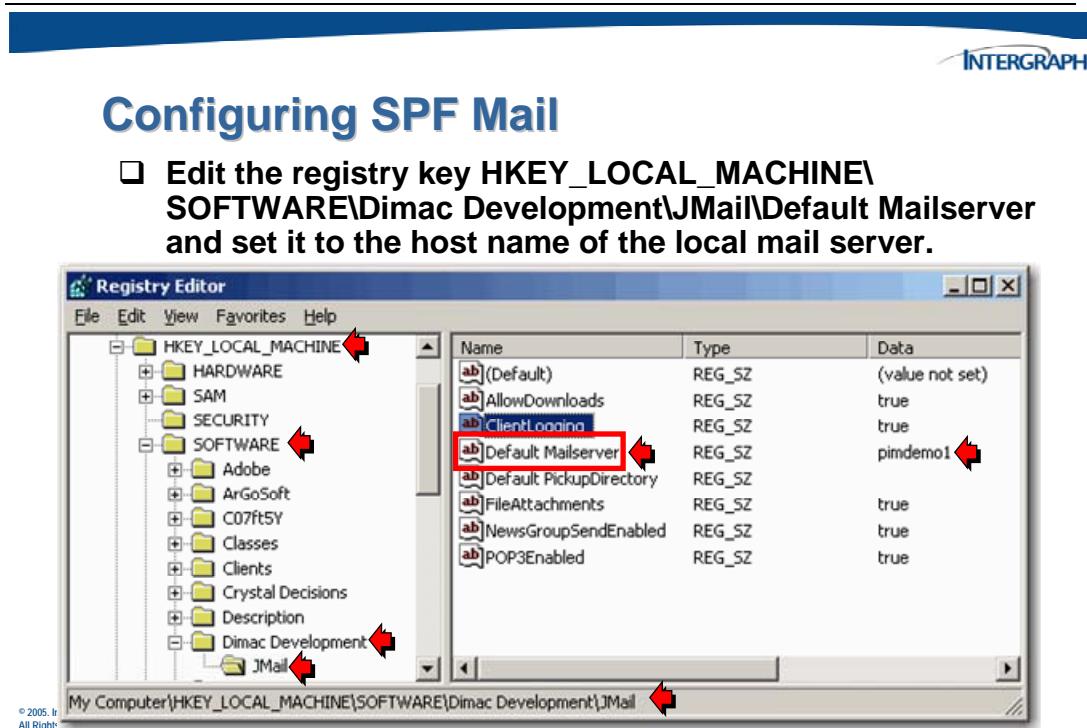
Review the users that are already associated to the user group.



2.10 Configuring SmartPlant Foundation E-mail

SmartPlant Foundation can be configured to use either the mail component JMAIL from Dimac Corporation (<http://www.dimac.com>) or Collaboration Data Objects for Windows NT Server (CDONTS).

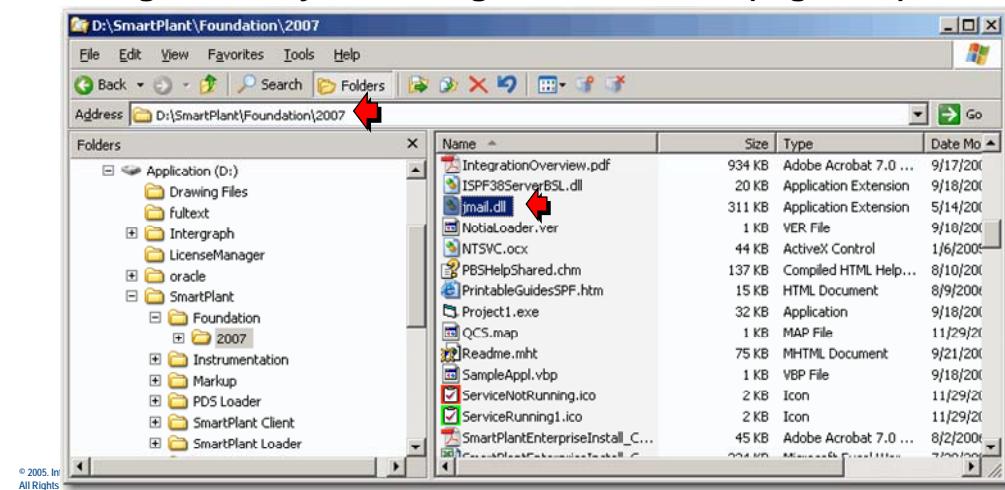
Set a *default* mail server for JMAIL:





Configuring SPF Mail

JMAIL requires the jmail.dll be available in the folder path registered by the dll registration server (regsvr32).

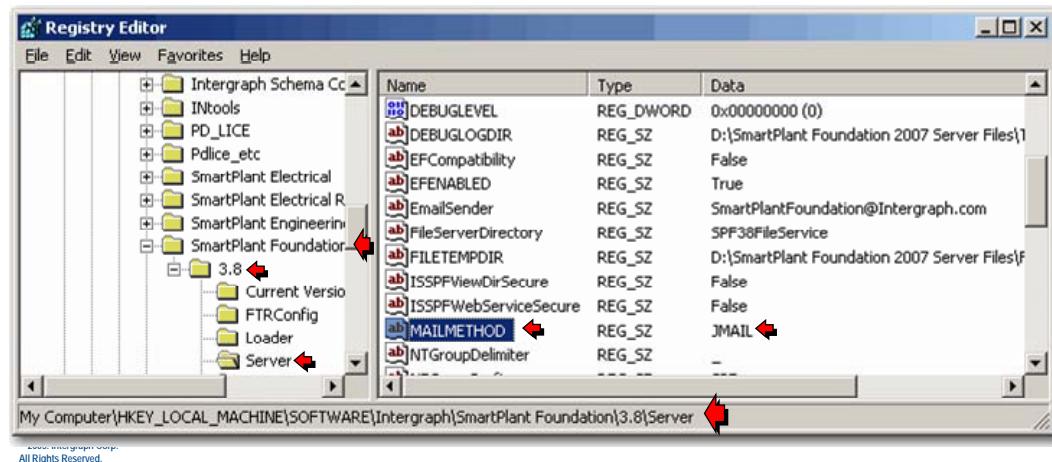


To change the mail transport method for SmartPlant Foundation, change the following registry key: HKEY_LOCAL_MACHINE\SOFTWARE\Intergraph\SmartPlant Foundation\<Version>\Server\MAILMETHOD.



Configuring SPF Mail

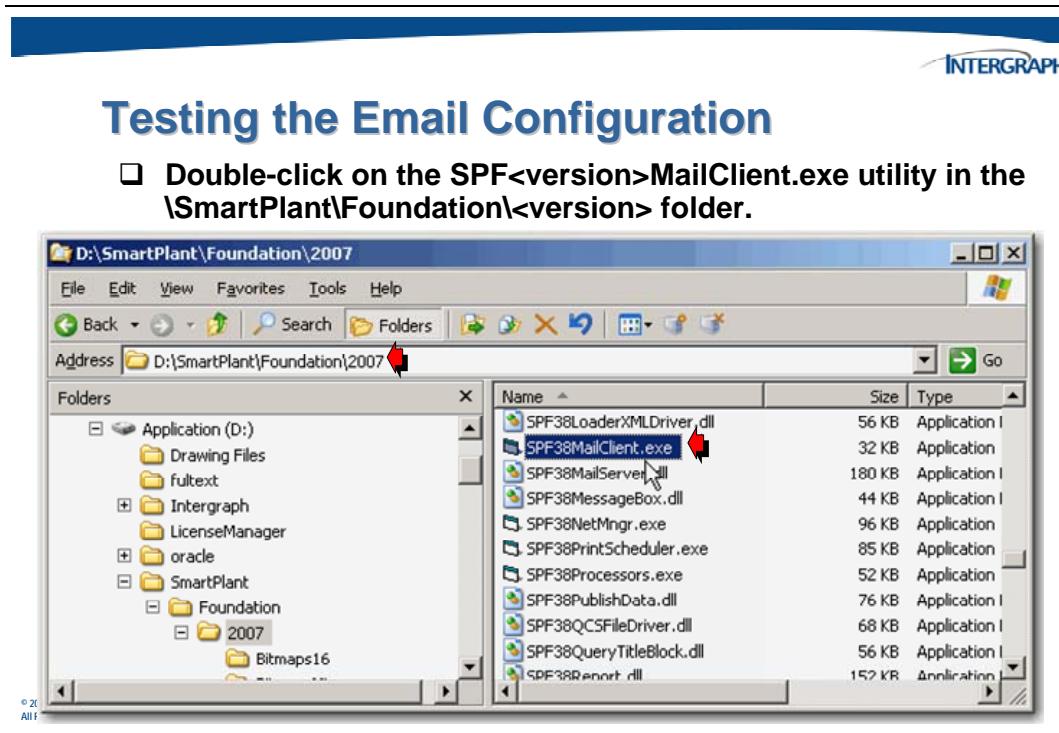
- Configure the following registry key
HKEY_LOCAL_MACHINE\SOFTWARE\Intergraph\SmartPlant Foundation\<Version>\Server\MAILMETHOD



Values are JMAIL for JMAIL and CDO for CDONTS (JMAIL is the default value).

2.10.1 Testing the E-mail Configuration

A stand-alone test harness is supplied to test the SmartPlant Foundation e-mail interface. To run the stand-alone test harness to test e-mail in SmartPlant Foundation, do the following:

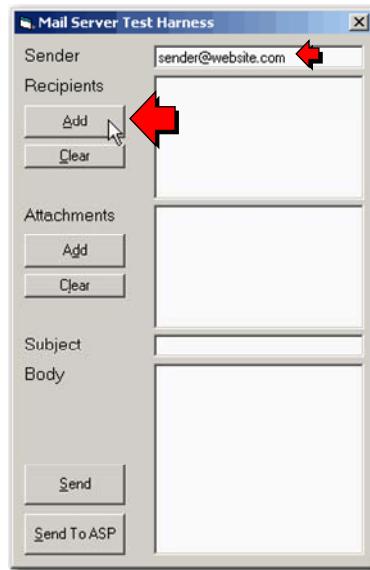


Note: The Simple Mail Transport Protocol (SMTP) must be installed and running on the mailserver.



Testing the Email Configuration

- Enter an optional Sender address and then click Add to add a recipient.

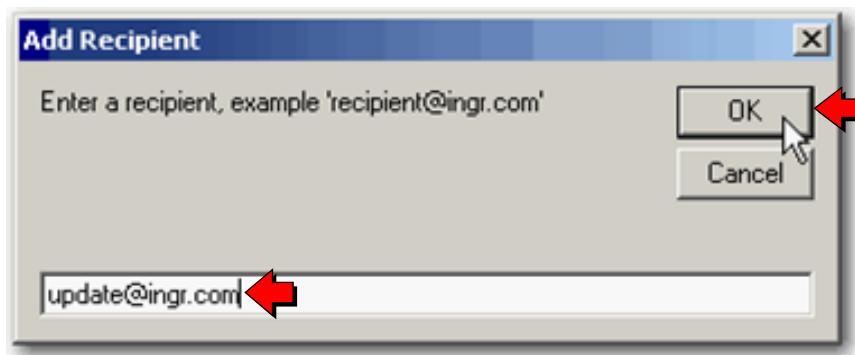


© 2005, Intergraph Corp.
All Rights Reserved.



Testing the Email Configuration

- Enter the recipient's email address, and click OK.

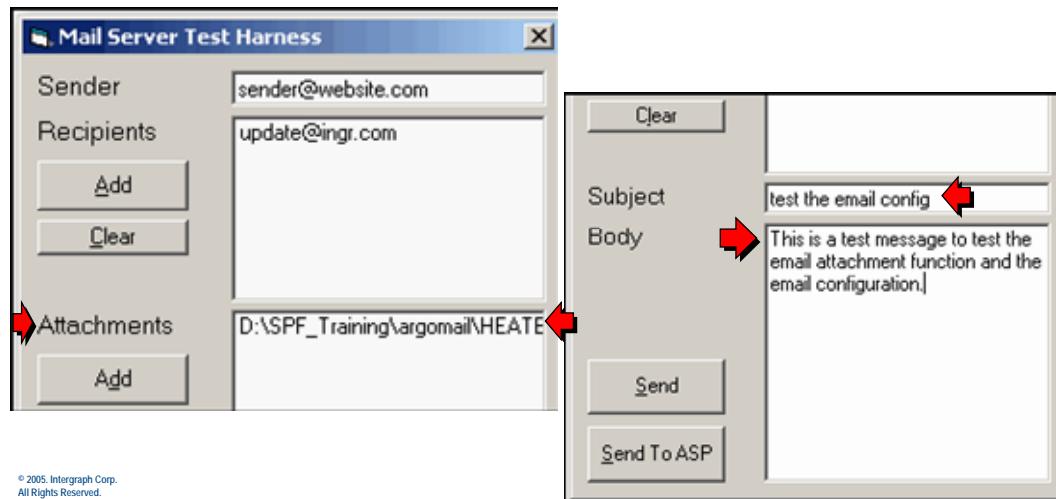


© 2005, Intergraph Corp.
All Rights Reserved.



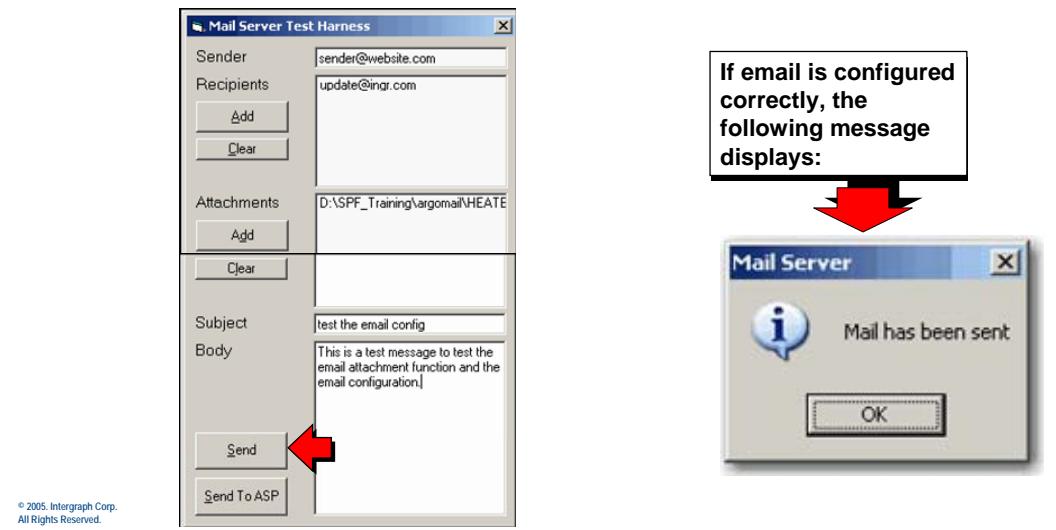
Testing the Email Configuration

- Add an optional Attachment, a *Subject* and *Body* text.



Testing the Email Configuration

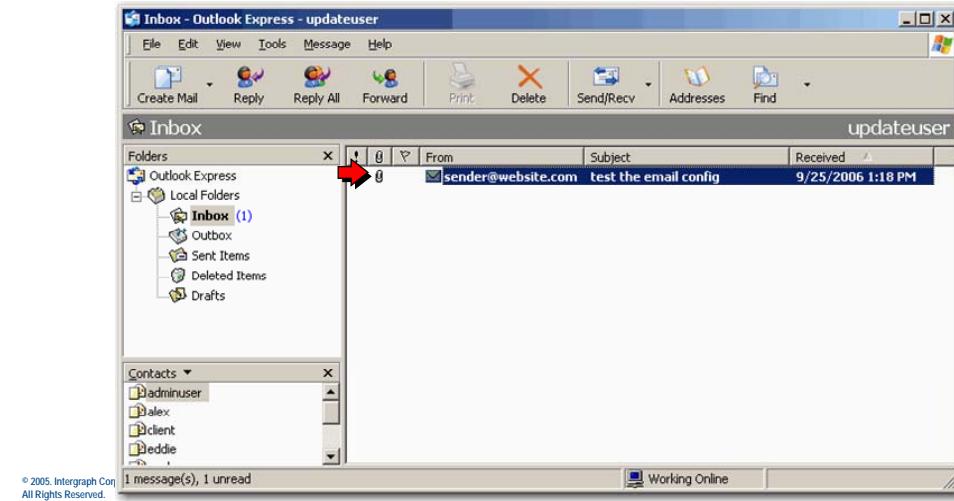
- Click Send to test the email.





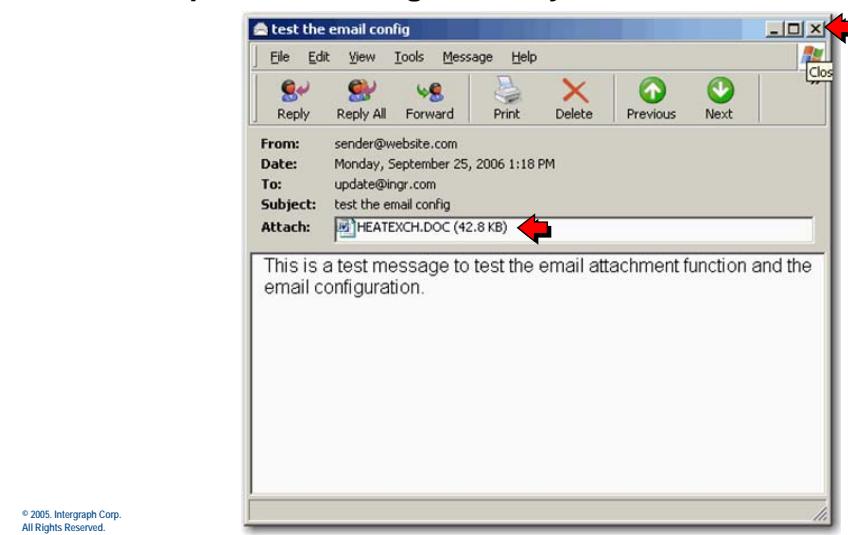
Testing the Email Configuration

- Open the email client to verify that the test message was sent.



Testing the Email Configuration

- Open test message to verify the contents and the attachment.



2.11 Email Digest

Users can now receive e-mails related to workflow steps, SmartPlant notifications, and subscription notification in e-mail digests, instead of as individual e-mails



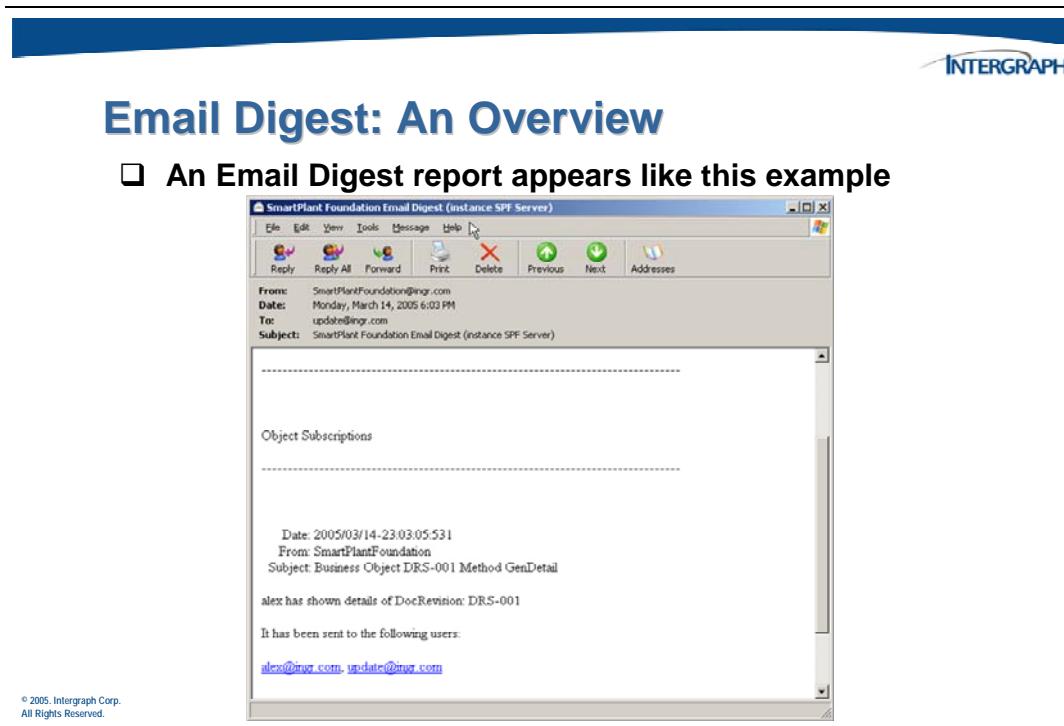
Email Digest

Users can receive emails related to workflow steps, Loader notifications, and subscription notification in email digests.

SPF will generate emails from multiple sources

- Workflow task notifications
- Notifications from object subscriptions
- SmartPlant Loader notifications
- Various administrative emails (token report, overdue workflow, etc.)

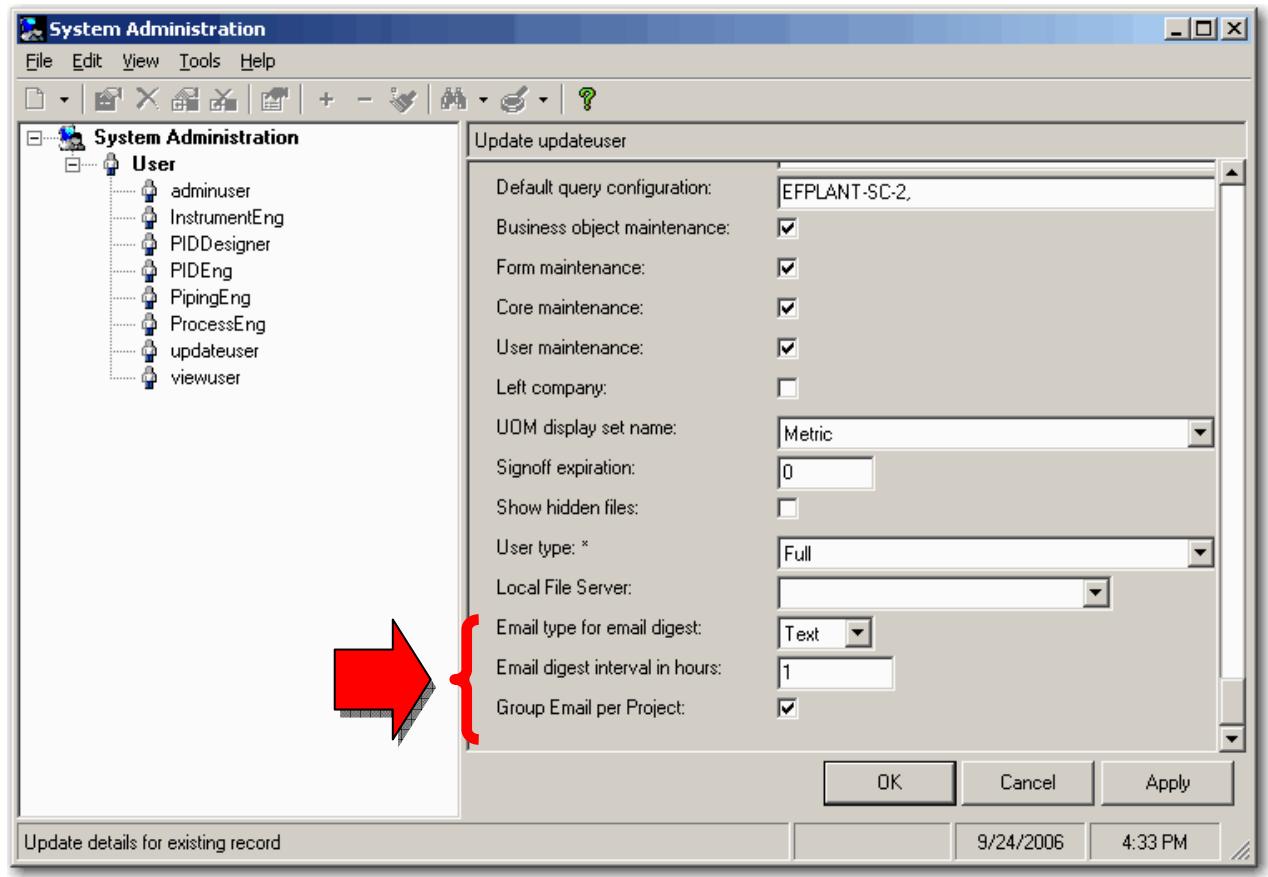
The following is an example of a Email Digest which is sent to the user.



2.11.1 Email Digest Setup

The ability to use email digests are configured when an SPF user account is created.

When creating a *User* within System Administration, there are fields at the bottom of the create dialog that configure the **Email Digest** for that user.





Email Digest Setup

The following fields are used to configure the email digest:

- Email type for email digest** – allows you to specify the format of email digests, either HTML or Text.
- Email digest intervals in hours** – allows you to set the interval (in hours) at which the system generates and sends email digests to the user.
- Group Email per Project** – allows you to select whether the user receives emails regarding workflow tasks, subscriptions, and SmartPlant Loader notifications as individual emails or grouped into an email digest per project. A check denotes that the digest is enabled.

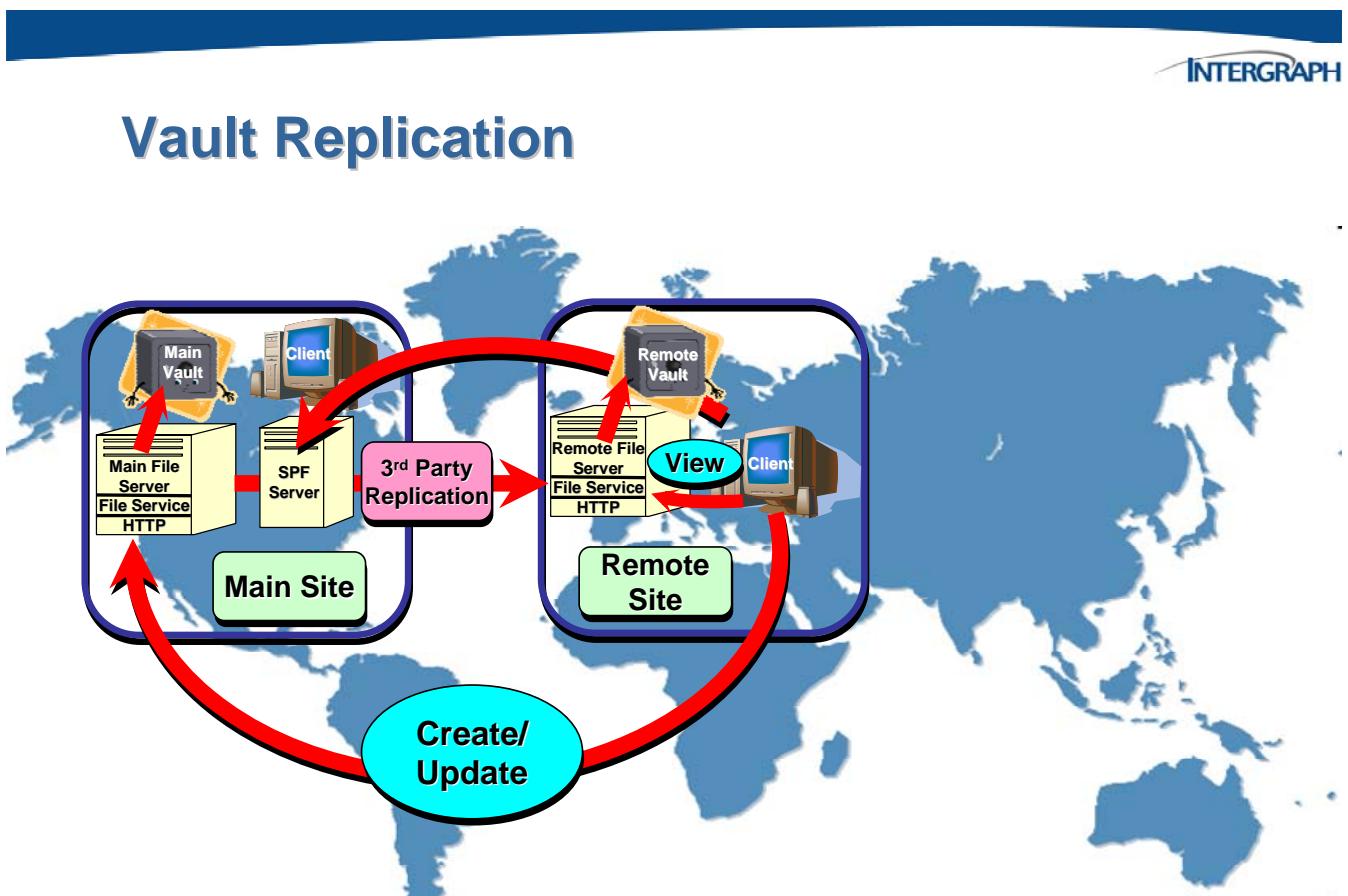
2.12 Introduction to Vault Replication

With the introduction of direct file transfer in SPF 3.6.2 it now also possible to configure file (vault) replication.

The following diagram shows how a remote SmartPlant Foundation Client access the local replicated vault for view functions.

All Create/Update functions go back to the “**Main File Server**”. Viewing operations can access the replicated vault.

Note: The replication must replicate the files to the same physical location as the source vault as all the replicated vaults use the same vault information and hence need to be located in the same location.



2.12.1 Setting up a standalone File Server

The pre-requisites for setting up a local file server require the following

- IIS 5 or 6
- .Net Framework 1.1 (2003)

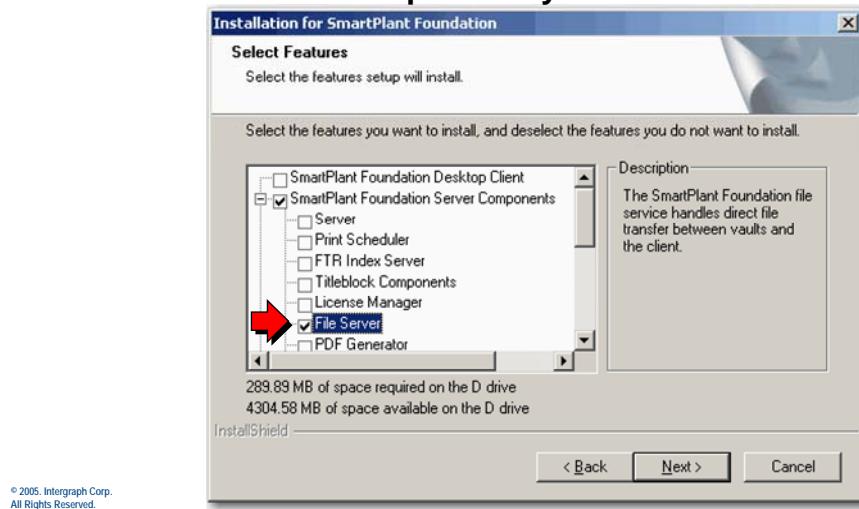
Optional

- SmartPlant Markup 3.4 (required for reading reference file headers)

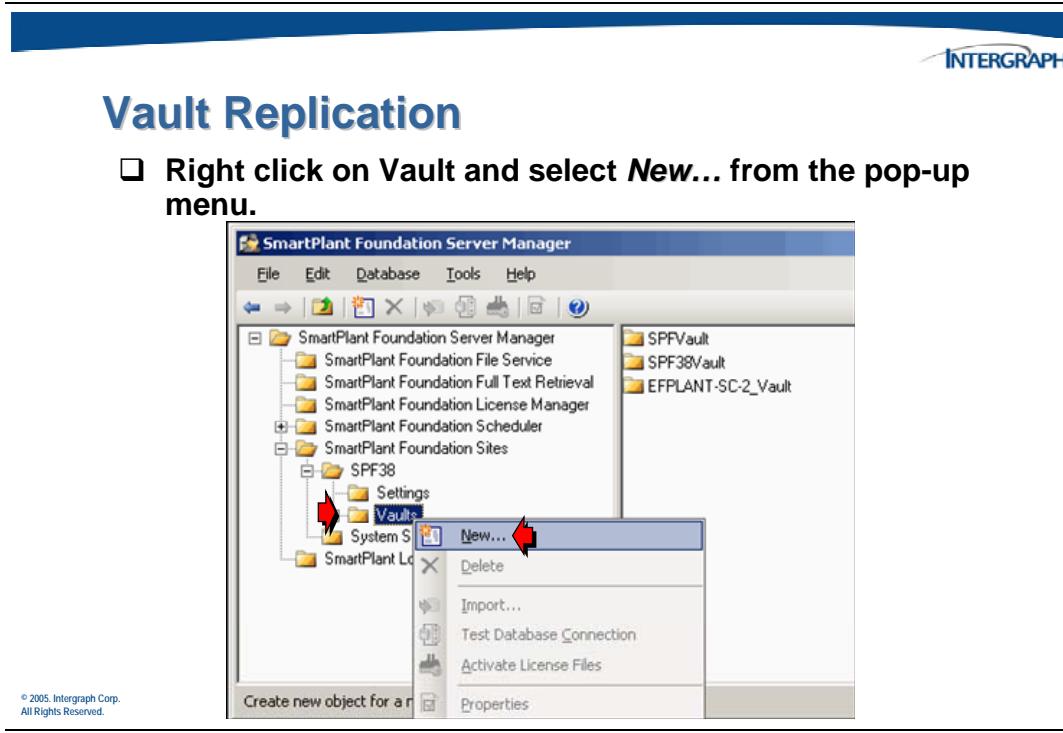


Vault Replication

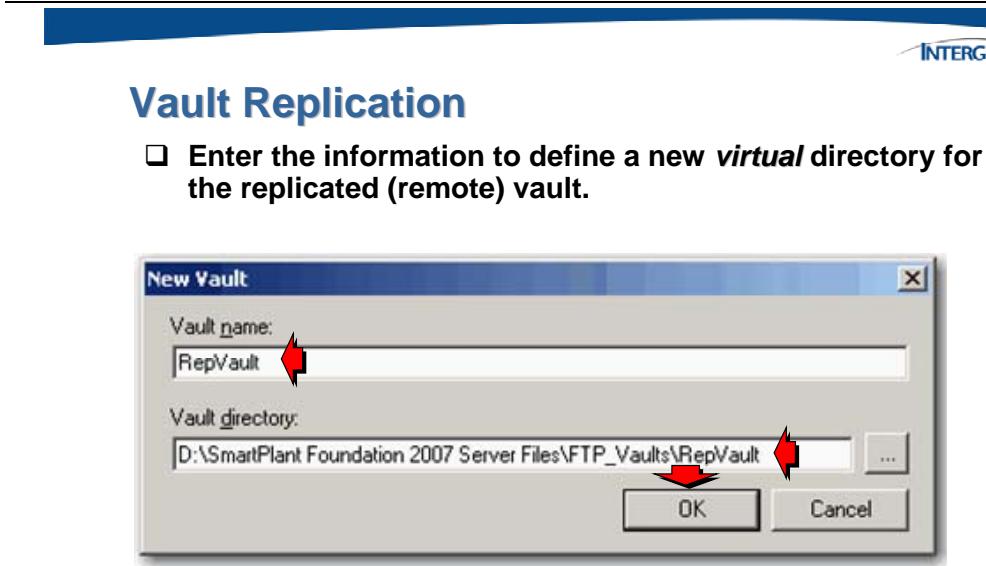
- Execute the SmartPlant Foundation installation and select the **File Server** option only.



Next, use *SmartPlant Foundation Server Manager* to create the vault “virtual” directory on the remote file server.



The *New Vault* dialog will display.

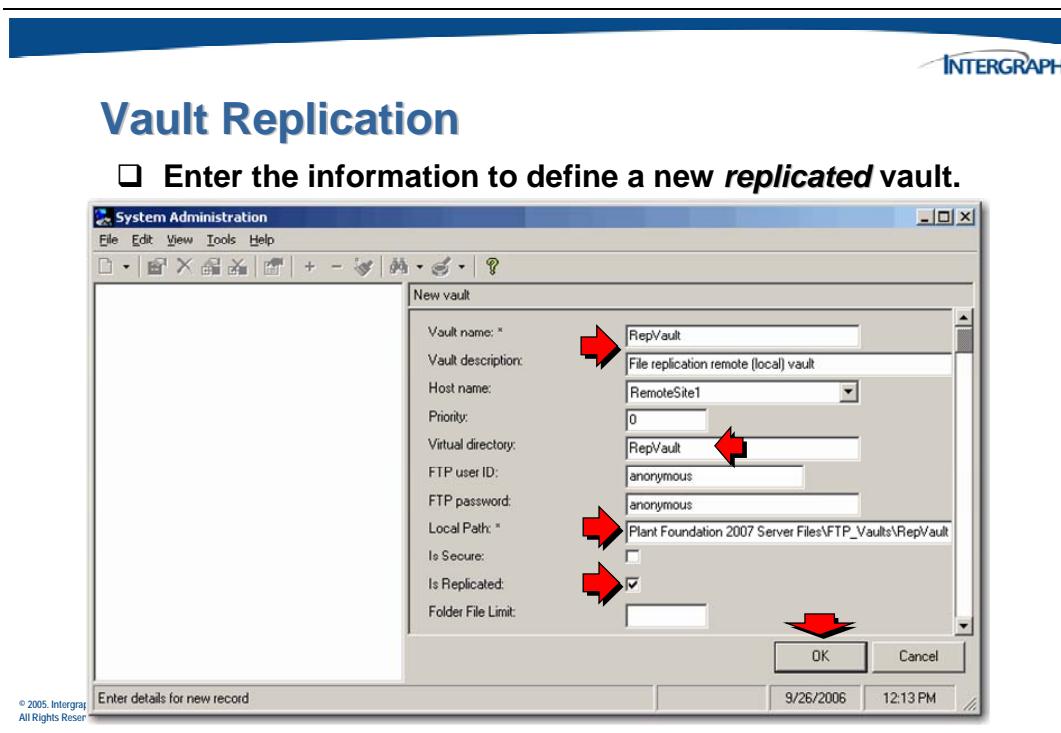


This is the same functionality that is used in the Internet Information Services Manager wizard to add a virtual directory.

2.12.2 Setting up Vaults for Replication

Using the *SmartPlant Foundation System Administration* utility enable the “**IsReplicated**” flag for the Vault object instance.

From the menu, select **File> New> Vault**. The *New Vault* dialog will display.

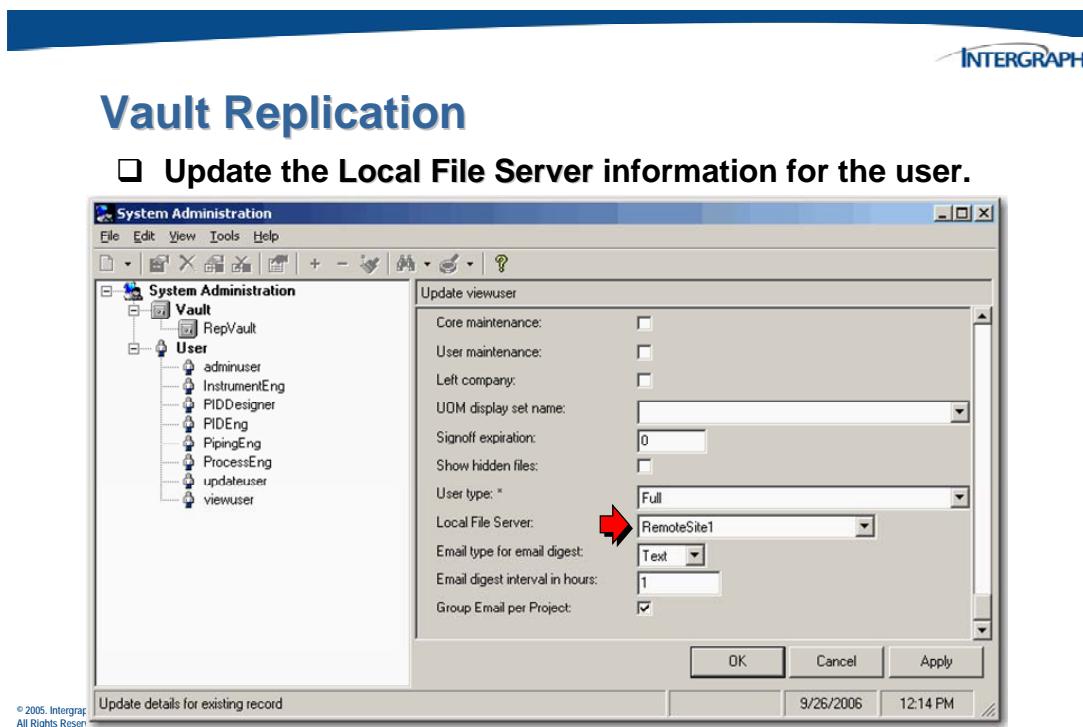


Click **OK** to create the new replicated “local” vault.

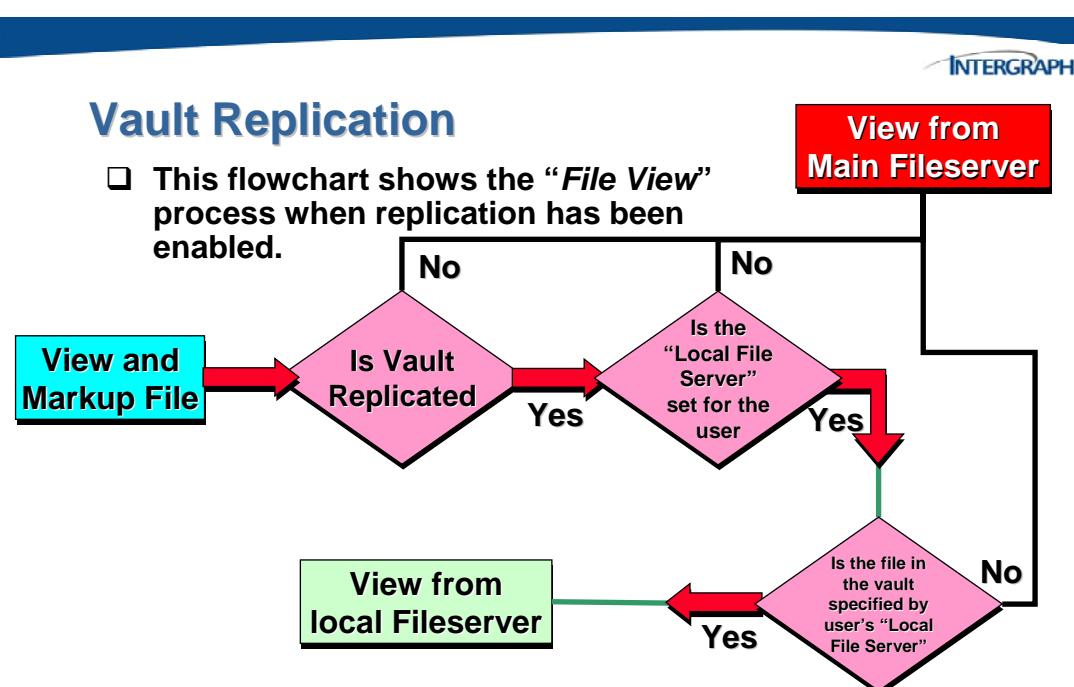
2.12.3 Setting up Users for Replication

In the *System Administration* utility set the “**Local File Server**” to the hostname of the SmartPlant Foundation Fileserver which is local to the SPF user. If no value is specified then the “**Main**” Fileserver is assumed.

From the menu, select **Tools>Find>User**. Highlight the user from the tree, right-click on the user object and choose **Update** from the pop-up menu. The *Update user* dialog will display in the right pane.



SmartPlant Foundation Replication only applies to the process of **viewing**. All the other file operations occur directly back to the “Main Fileserver”. The following flowchart shows the “File View” process when replication has been enabled.



© 2006, Intergraph Corp.
All Rights Reserved.

2.12.4 File Replication Mechanisms

Replication is achieved by using a third party application. For our testing we used **Robocopy** as the replication mechanism.

Note: The replication must replicate the files to the same physical location as the source vault as all the replicated vaults use the same vault information and hence need to be located in the same location

The following example demonstrates full replication where **All** data is copied from the source to the destination with two types of synchronization:

- Constant Synchronization - data is constantly synchronized.
- Periodic Synchronization - Data is periodically synchronized to reduce network bandwidth.

2.12.5 Using Robocopy for SPF Replication

Below is an example of a file replication package (there are numerous packages on the market that can be used).

Robocopy (Robust File Copy Utility) is a 32bit command-line tool used for file replication and is included in *Microsoft® Windows® Resource Kit*.

The version used here is version XP010 which requires Microsoft® Windows® Server 2003, Microsoft® Windows® XP, Microsoft® Windows® 2000, or Microsoft® Windows NT® version 4.0.

To run Robocopy, use the following syntax at the command prompt:

ROBOCOPY source destination [file [file]...] [options]

The following table defines these syntax elements.

Variable	Meaning	Comments
<i>source</i>	Source directory	You can use <i>drive:\path</i> or \\server\share\path
<i>destination</i>	Destination directory	You can use <i>drive:\path</i> or \\server\share\path
<i>File</i>	Names of files to act upon	You can use wildcard characters (?) and (. If no files are listed, Robocopy defaults to all files (*).
<i>options</i>	Command-line options you wish to use	Available options are described later in this document.

Example:

Parameter	Meaning
<i>Source</i>	Source directory (can be UNC)
<i>destination</i>	Destination directory (can be UNC)
<i>File</i>	Names of files to act upon
<i>Options</i>	--o--
<i>/S</i>	Copies Subdirectories (excluding empty ones)
<i>/MOT:1</i>	Synchronize changes for each change
<i>/MIR</i>	Mirrors a directory tree (will delete at destination)
<i>/NDL</i>	Display only file changes
<i>/LOG+:<logfile></i>	Appends to logfile
<i>/TEE</i>	Display to screen as well as logfile

i.e.

```
robocopy \spfserv\vaults \fileserv\vaults c:\apd\dir1 . /S /MOT:1 /MIR /NDL /TEE
```

This would replicate every minute as long as a file has changed.

2.12.6 Constant Synchronization

This is achieved by setting one on the monitoring options to 1

i.e. /MON:1 or /MOT:1

The command:

```
robocopy \\spfserv\\vaults \\fileserv\\vaults c:\\apd\\dir1 . /S /MOT:1 /MIR /NDL /TEE
```

This would replicate every minute along as a file has changed.

2.12.7 Periodic Synchronization

Robocopy copy can be controlled to run in several different ways.

The following option can be used to **Constant interval:**

i.e. **/MON:30**

- Will make robocopy run every 30 minutes

Restricted hours:

i.e. **/RH:2400:0400**

- Will restrict robocopy to only run between the hours of 24:00 and 04:00

You can also control the following:

Bandwidth Restriction:

i.e. **/IPG:100**

- Will put a 100 millisecond delay between each robocopy packet (normally 64 kilobyte)

Batch jobs - the robocopy could be run as a batch job (AT or “Scheduled Tasks”)

File Size:

i.e. **/MAX:<1000000>**

- Will exclude files of size larger than 1000000 bytes (1MB).

2.12.8 Recovery options

In case on failure of a remote file location, robocopy can be made to wait and retry before failing . (The default is a million retries of 30 seconds about a year).

/R:n specifies number of retries

/W:secs specifies the wait in seconds

2.12.9 Installation as a Service

Robocopy can be installed as a service using another *Microsoft® Windows® Resource Kit* utility - “svrany”.

2.13 Activity – Schema Object Administration

The goal of this activity is to familiarize you with the tasks that will need to be performed using the Schema Object Administration interface.

1. Log on to your operating system as *spfuser* with no password (if not already logged in).
2. Click **Start > All Programs > Intergraph SmartPlant Foundation > Schema Object Administration** to start *Schema Object Administration*.
3. When the *Login* dialog box appears, type the *user name* **adminuser** with no password and click **OK**.

Creating Edge Definitions

4. Create a new **Edge Definition** using Schema Object Administration.
 - Click **File > New > Edge Definition**.
 - In the **Name** box, type the name of the new edge definition - **EquipmentToList**
 - If you want the edge definition to have a different name when it is displayed in the SPF user interface, type the name in the **Display name** box – **Equipment To List**.
 - Type a description in the **Description** box – **Navigate from Equipment to List**.
 - If you only want the edge definition to appear on shortcut menus for users whose user group has been associated with the edge definition, select the **Hide from menu** check box.
 - If you want to use the creation date for a document revision as the effectiveness date for displaying objects associated with a document master when you start from a document revision and expand the master document, select the **Use effectiveness date** check box.
 - In the *Select interface definition* box, locate the **IEquipmentOcc** Interface. Type the criteria *IEquip%* and click **Find** to view a list of interfaces.
 - In the list, select the interface that you want to use as the starting point for the edge definition. Select the **IEquipmentOcc** interface and then click .
 - Under *Select relationship definition*, select the **EquipmentComponents** (first listed) RelDef, which is the relationship definition that you want to include in the edge definition.

- In the *Select interface definition*, locate the **IPipingConnectorComposition** interface. Type the criteria *IPip%* and click **Find** to view a list of interfaces.
- You want the end object in the relationship to support the **IPipingConnectorComposition** interface. Select that interface in the **Select Interface Definition** list, and then click .
- Click **OK** to save the new edge definition.

Note: Edge definitions that you create in Schema Object Administration are not copied back into the SmartPlant schema. These definitions are only used in SPF.

5. You are now going to add a user group to the EdgeDef relationship to allow browse functionality.
 - Right-click the new EdgeDef name and click **Show User Groups** to view the relationship.
 - Drag and drop the *Process* user group object to the EdgeDef relationship object.
 - **Process** to *EquipmentToList*

In the *tree view* pane, verify that a relationship between the user group and the EdgeDef has been created.

Creating a Graph Definition

6. Create a **Graph Definition** to traverse by default when you expand a ClassDef in the SPF tree view. The GraphDef must have the same name as the ClassDef.
 - Click **File > New > Graph Definition**.
 - When the *New Graph Definition* dialog box appears, in the **Name** box, type the name of the new graph definition – **PIDProcessEquipment1**
 - If you want the graph definition to have a different name when it is displayed in the SPF user interface, type the name in the **Display Name** box – **Process Equipment**.
 - Type a description in the **Description** box – **Graph of equipment tag plus P&ID**.
 - In the *Select starting interface definition* box, locate the **IEquipmentOcc** Interface. Type the criteria *IEquip%* and click **Refresh** to view a list of interfaces.

- In the list, select the interface that you want to use as the starting interface definition. Select the **IEquipmentOcc** interface.
- Under *Current graph definition*, select the check boxes beside the **Equipment Components** edge definition. This is the one that you want to include in the graph definition.
- Click **OK** to save the new graph definition.

Note: Graph definitions that you create in Schema Object Administration are not copied back into the SmartPlant schema. These definitions are only used in SPF.

Create a View Definition

7. Create a new **View Definition** using Schema Object Administration.
 - Click **File > New > View Definition**.
 - When the *New View Definition* dialog box appears, in the **Name** box, type the name of the new view definition - **ViewProcessEquipment**
 - Give the view definition a different name when it is displayed in the SPF user interface. Type the name in the **Display Name** box – **PIDProcessEquipment1**.
 - Type a description in the **Description** box – **Lab PID equipment view**.
 - Type PID% and click **Refresh** to view a list of all graph definitions.
 - Under *Select starting graph definition*, select the **PIDProcessEquipment1** graph definition that you want to use as the starting point for the view definition.
 - In the *Select property to add* box, scroll down and select the **Name** property that you want to include in your view definition.
 - To change the name of the property as it will be displayed in the SPF client, type the new name **Tag No.** for the property in the box under the *Select property to add* tree view and click **Add**.
 - In the *Select property to add* list, scroll up and select the **Description** property that you want to include in your view definition and click **Add**.
 - In the *Select property to add* list, scroll down and select the **Name** property (in the **Equipment Components** tree) that you want to include in your view definition. You may need to expand this tree to see the property.
 - To change the name of the property as it will be displayed in the SPF client, type the new name **P&ID** for the property in the box under the *Select property to add* tree view and click **Add**.

- Click **OK** to save the new view definition.

Note: View definitions that you create in Schema Object Administration are not copied back into the SmartPlant schema. These definitions are only used in SPF.

Create a Class View Map

8. Create a new **Class View Map** using Schema Object Administration.
 - Click *File > New > Class View Map*.
 - When the *New Class View Map* dialog box appears, in the **Name** box, type the name of the new view definition - **EquipmentClassViewMap**
 - Give the view map a different name when it is displayed in the SPF user interface. Type the name in the **Display Name** box –**Equipment View**.
 - Type a description in the **Description** box – **Equipment View Map for a process engineer**.
 - In the *Class definitions* list, select the **PIDProcessEquipment** class definition for which you want to create a class view map.
 - In the *View definitions* list, select the **ViewProcessEquipment** view definition that you want to use for the selected class definition and click **Map**.
 - In the *Class definitions* list, select the **PFDProcessEquipment** class definition for which you want to create a class view map.
 - In the *View definitions* list, select the **ViewProcessEquipment** view definition that you want to use for the selected class definition and click **Map**.
 - Click **OK** to save the new class view map.
9. You can define which users see the view definitions defined by the class view map when they select objects of the specified classes by associating user groups with the class view map.
 - Right-click the new Class View Map name and click **Show User Groups** to view the relationship. You may need to search for User Groups first.
 - Drag and drop the user group object to the Class View Map relationship object.
 - **Process to EquipmentClassViewMap**

- Verify in the *tree view* pane that a relationship between the user group and the Class View Map has been created.

Associating User Groups with Relationship Definitions

10. Use **Schema Object Administration** to associate a user group with a relationship definition:
 - Click **Tools > Find > Relationship Definition**.
 - In the *Find Relationship Definition* dialog box, enter your search criteria, **PlantProjects**, and then click **OK**.
11. Right-click the relationship definition, **PlantProjects**, with which you want to associate a user group, and click **Show User Groups** on the shortcut menu.
12. Click **Tools > Find > User Group**.
 - In the *Find User Group* dialog box, enter your search criteria, **UPDATE**, and then click **OK**.
13. Drag the **UPDATE** user group onto the *User Groups* relationship under the **PlantProjects** relationship definition.
14. In the *New Relationship* form, select the access that you want users to have to the relationship definition in SmartPlant Foundation in the Access Mask box, **110**.
15. Click **OK** to save the new relationship.
16. The final change needs to be with the Class Definition for the ADCDesignChange. Because the object instance which is to be created will need a **Unique Key**, edit the Unique Key field and ADD the following:

Unique Key: **ADC,NAME**
17. **Exit Schema Object Administration**.

18. Once you have exited from Schema Object Administration, you may take a short break until the other students have finished this activity.

C H A P T E R

3

Creating SmartPlant Foundation Forms

3. SmartPlant Forms

SmartPlant Foundation uses forms to provide support for creation, update, query, clone, and information display of all objects such as Tag, Asset, Models, and Documents.

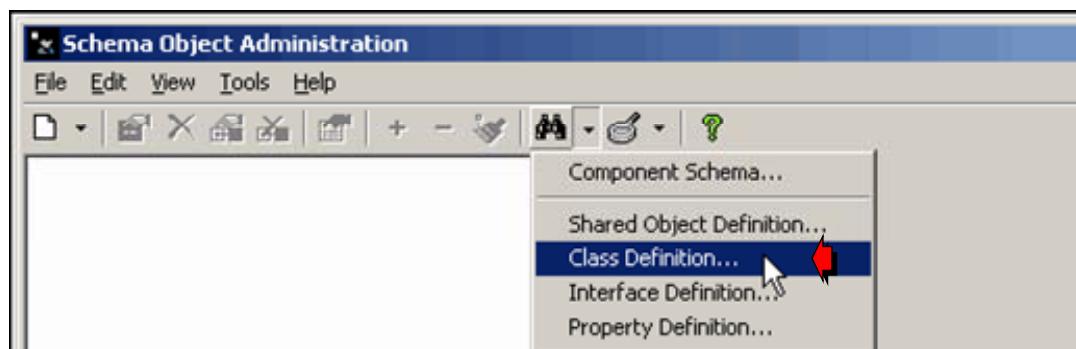
3.1 Converting to a Form

In Schema Object Administration, you can generate forms for class definitions contained in the SmartPlant schema. These forms contain all the property definitions that are exposed by the interface definitions realized by a particular class definition. The forms you generate in Schema Object Administration are used in the SPF client applications.



Generating Forms

- Click the **Find** tool and select **Class Definition...** from the drop down menu.



Enter your search criteria in the **Find Class Definition** dialog box.



Generating Forms

- Type **ADC*** in the *Find Class Definition* dialog box and click **OK**.



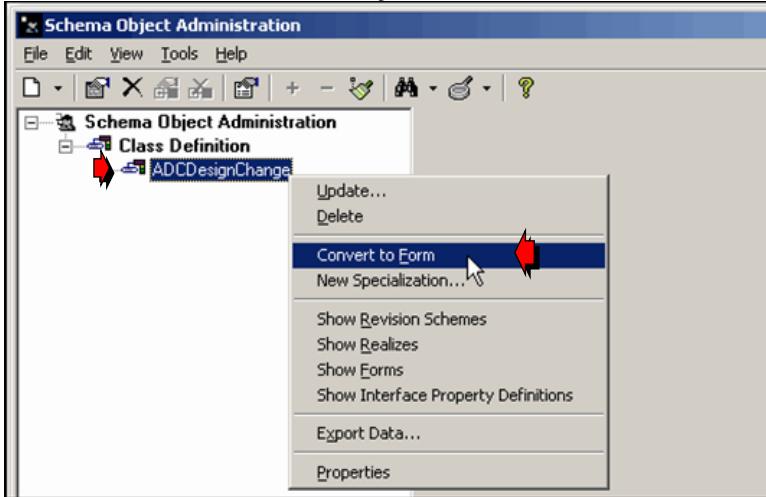
© 2005, Intergraph Corp.
All Rights Reserved.

Right-click the class definition for which you want to generate a SmartPlant Foundation form.



Generating Forms

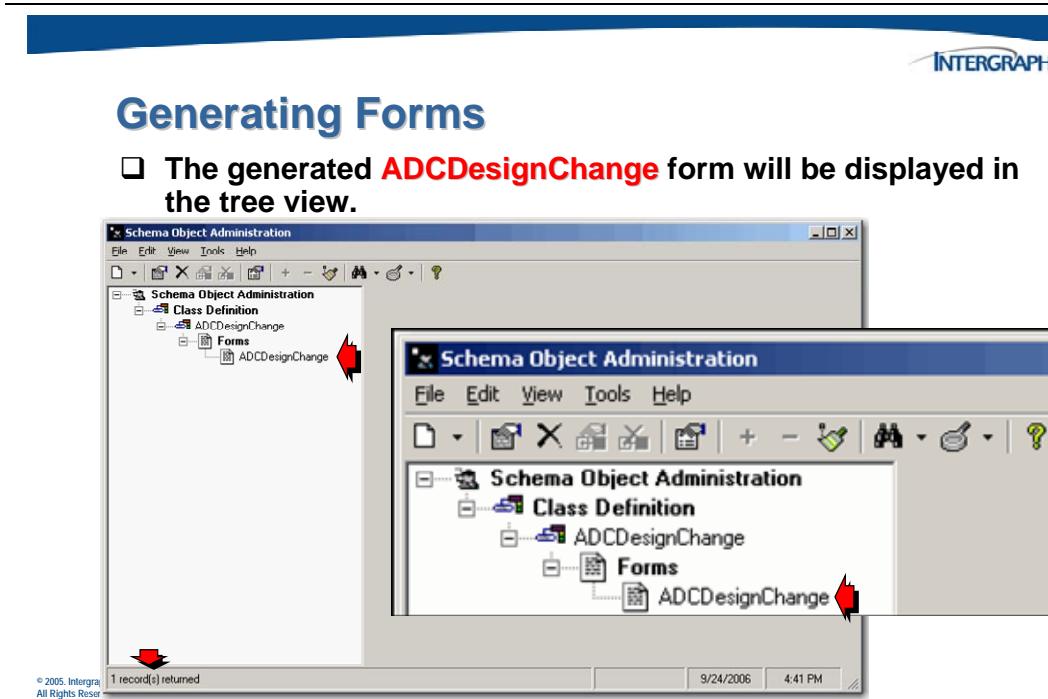
- Right-click the **ADCDesignChange** ClassDef and click **Convert to Form** on the dynamic menu.



© 2005, Intergraph Corp.
All Rights Reserved.

To see a list of all property definitions that the forms will contain, right-click the class definition, and then click **Properties** on the shortcut menu.

Schema Object Administration creates a form containing all the property definitions exposed by the interface definitions realized by the selected class definition.



Editing a Form

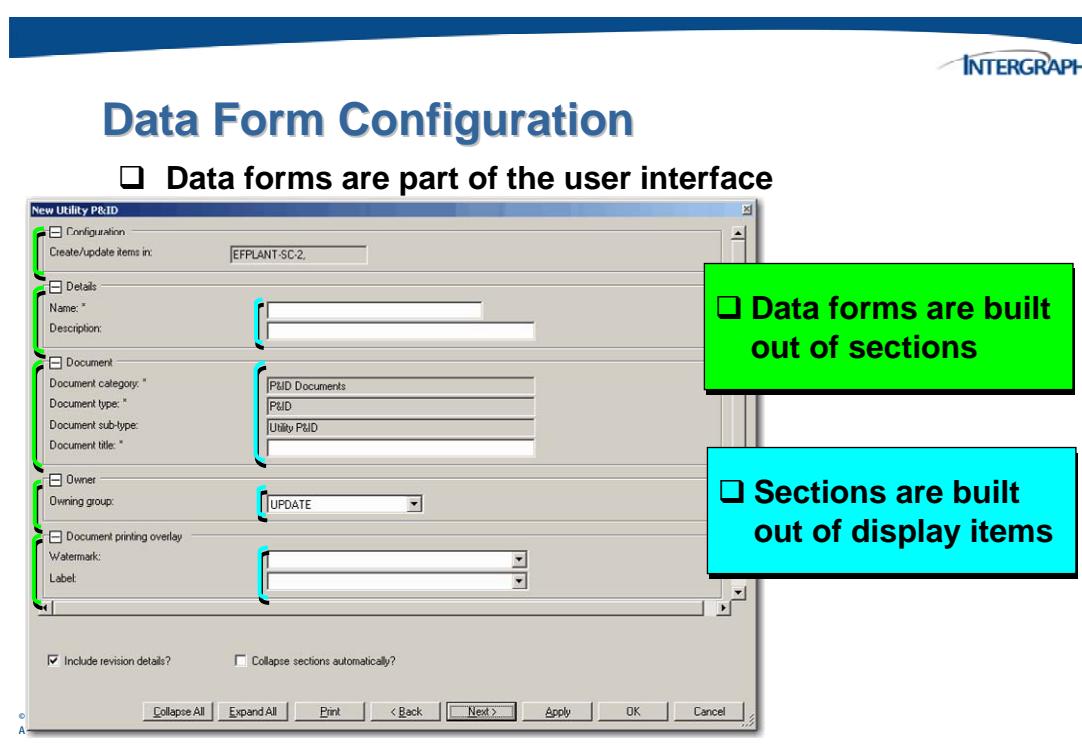
The best way to visualize the form that was converted from the schema definitions that were loaded into the admin database is as follows:

- ClassDef = Form**
- Interfaces = Sections**
- Properties = Display Items**

3.2 Form Builder Editor

The SPF configuration can vary widely from customer to customer. The type and format of data also varies and must be displayed dynamically. To accomplish this, **dataforms** can be created and modified using the *Form Builder Editor*.

- Create, Update, Query, GetInfo (viewing), Copy and Revise forms functionality is available



3.2.1 Form Editing

Before changing a dataform, determine the purpose and content of the form.

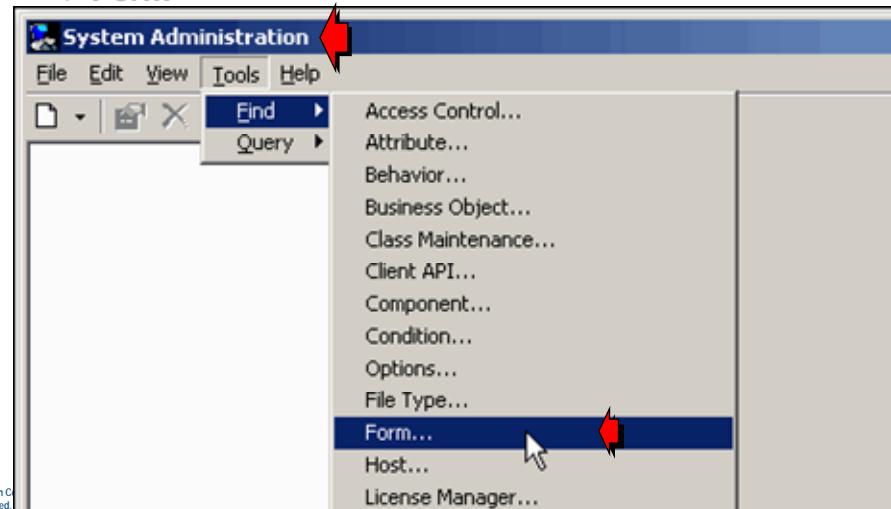
In order to modify the forms that were converted using the Schema Object Administration utility, you must open the *Form Builder Editor* using the SPF System Administration utility.

Once inside the SPF System Administration utility, perform a search to locate the form that was generated (converted) using the Schema Object Administration utility.



Editing a Form

- From the **System Administration** menu, select **Tools > Find > Form**.



A *Find Form* dialog will display.

Editing a Form

- ❑ Type **ADC*** in the *Find Form* dialog box and click **OK**.

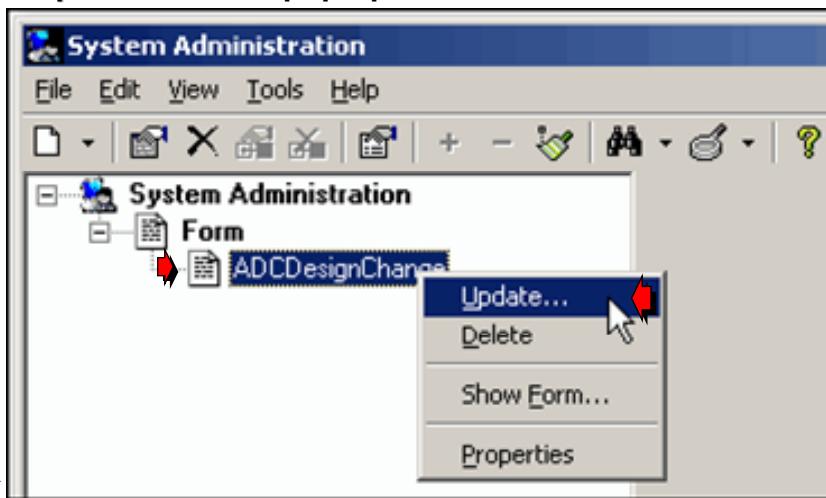


© 2005, Intergraph Corp.
All Rights Reserved.

Use the right mouse button to open the Form Builder Editor and make any necessary changes to the converted form.

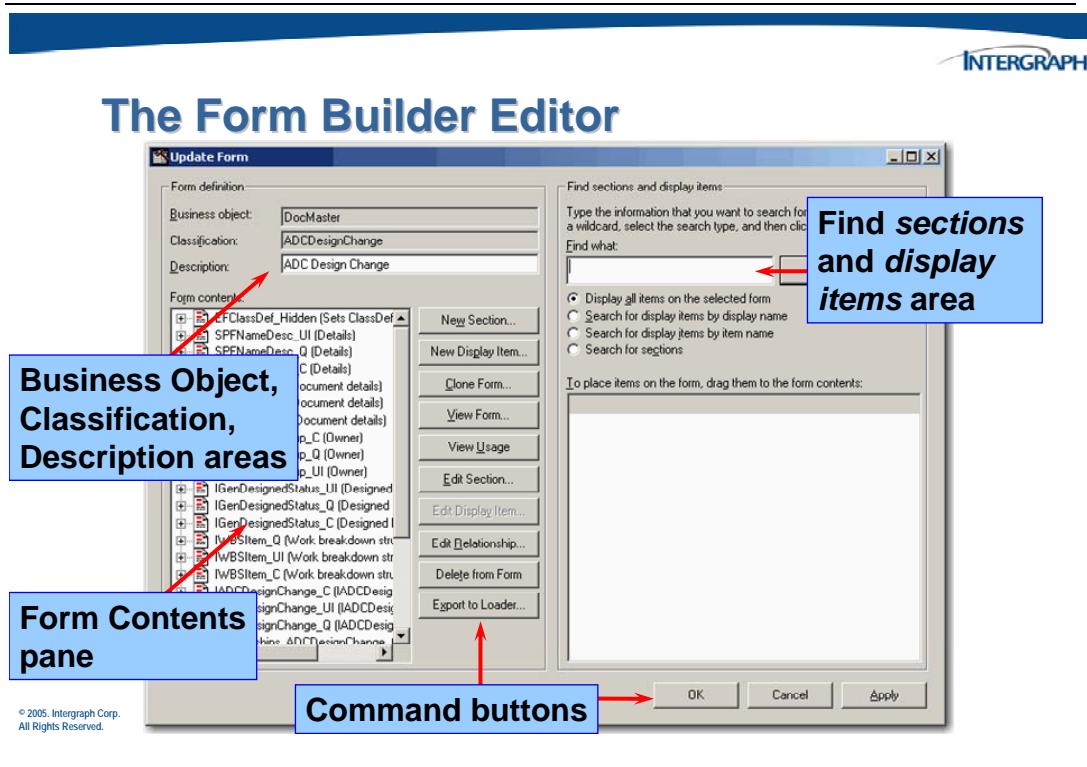
Editing a Form

- ❑ Right-click the **ADCDesignChange** form and choose **Update** from the pop-up menu.



© 2005, Intergraph Corp.
All Rights Reserved.

The Form Builder Editor window will display containing the contents of the generated form.



The *Form Builder Editor* dialog window contains the following areas:

- Business Object** - this field displays the name of the business object that uses the data form.
- Classification** - this field displays the class definition of the object.
- Description** - this field allows you to enter a brief description of the purpose of the form.
- Form Contents** (left pane) – initially blank but for converted forms it will represent the *sections* and *display items* currently contained in the dataform.
- Find Sections and Display Items** (right pane) - this section of the form allows searches for existing *display items* and *sections* that can be added to the dataform.
- Command Buttons** - allows administrators to perform operations on the selected items.

3.2.2 Modifying Data Form Sections

Sections define a collection of *Display Items* (properties or relations). A section can be attached to multiple data forms with any changes to that section automatically updated on all data forms.



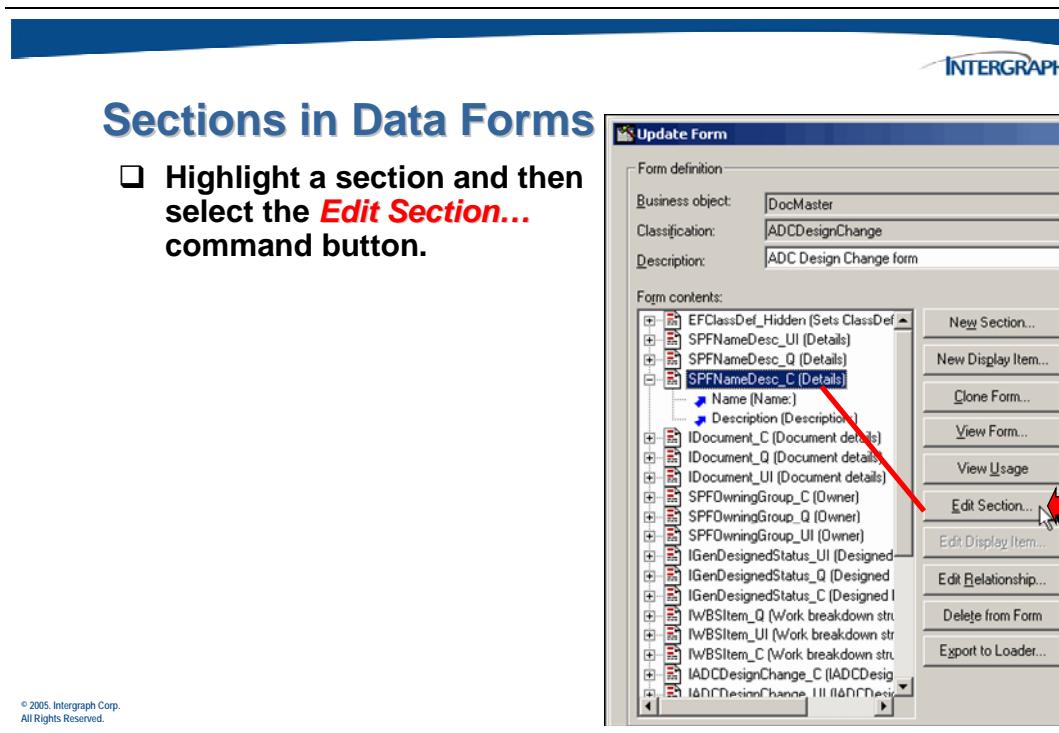
Sections in Data Forms

- A Section can be used in multiple data forms**
- Section properties include name and description (title of the section in the form)**
- Relationship between form and section defines:**
 - sequence of section
 - purpose (create, update, query, information, revise and copy)
 - user access control (read-write, read-only or property level)
- Sections are defined from the **Convert to Form** command in Schema Object Administration**

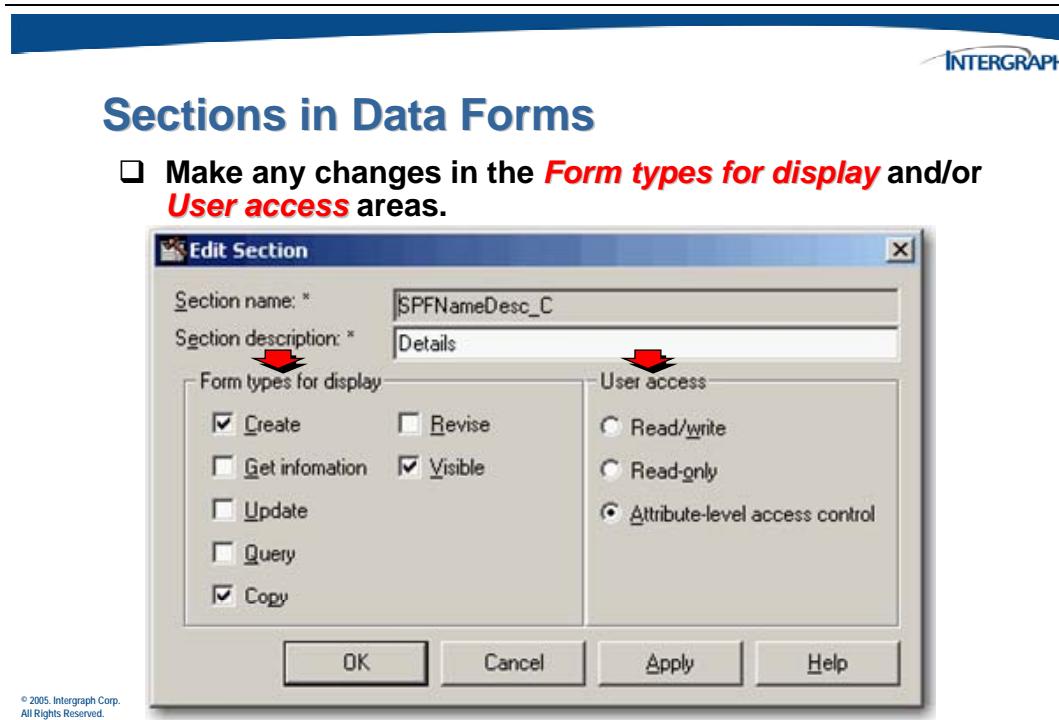
© 2005, Intergraph Corp.
All Rights Reserved.

Sections may be either created or re-used from the database. You can create sections using the *New Section* dialog window. After a section is created, *display items* can be added to the section. Using the **Find Now**, you can add existing sections to a data form to create a relationship.

Use the **Edit Section...** command to review the sections of a converted form.



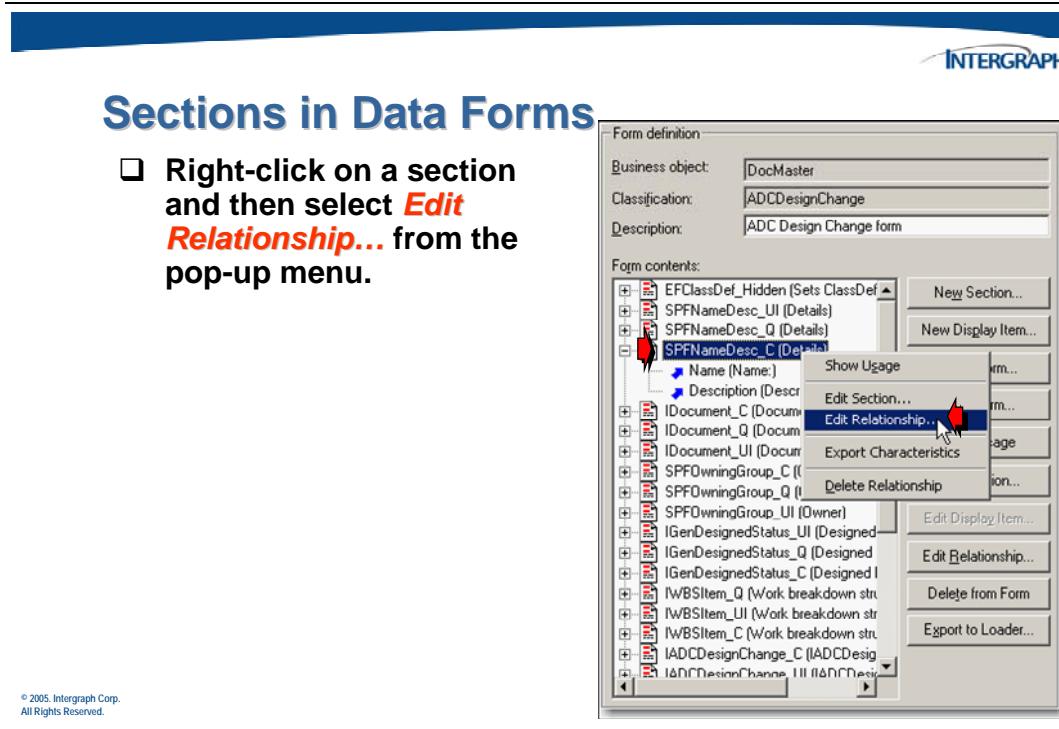
The *Edit Section* dialog will display.



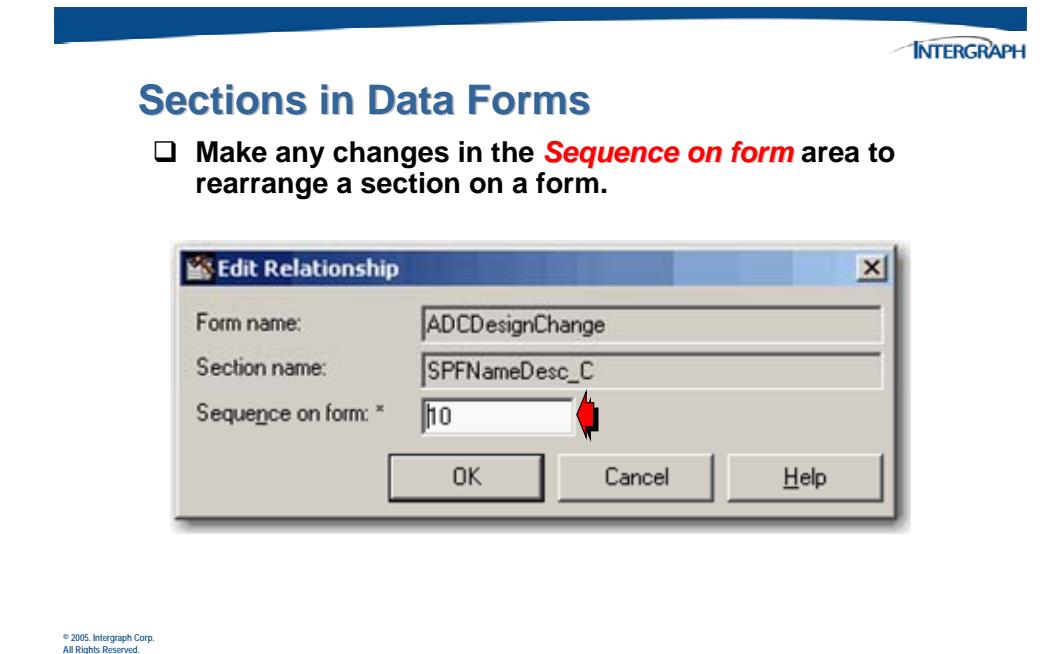
The *Edit Section* dialog window contains the following fields:

- Section Name** - specifies the name of the section and is read-only
- Section Description** - specifies a brief description of the section and is required
- Form Types For Display** - displays the appropriate section behavior check boxes
 - **Create** - used in the forms for create and clone functions
 - **Get Information** - used in the forms for viewing. All display items are presented as *read-only* so the same section can be used for **Update** and **GetInfo**.
 - **Update** - used in the forms for the update function
 - **Query** - used in the forms for query functions
 - **Copy** - used in the forms for copy functions. When an object is copied the display items listed in this section will only be copied.
 - **Revise** - Allows you to view only those sections that are part of the Revise form for the business object.
 - **Visible** - Specifies that the section is to be displayed only on the forms selected on the other check boxes. If left unchecked, then the section is hidden from the selected forms.
- User access** - displays the appropriate user access rights options radio buttons
 - **Read/Write** - forces all display items in the section to be editable regardless of their individual settings
 - **Read-Only** - forces all display items in the section to be read only regardless of their individual settings
- Attribute-level access control** - indicates that the display items in the section have individual access control

Sections are associated to the data form by a relationship. To view and modify the relation properties, use the **Edit Relationship...** command.



The *Edit Relationship* dialog will display.



Change the sequence of the section on the data form in the *Edit Relationship* dialog. It is recommended that the initial sequence numbers should be defined in increments of **10**.

The sequence value is not defined for the section itself, but for the relationship between the section and the data form.

3.2.3 Modifying Display Items

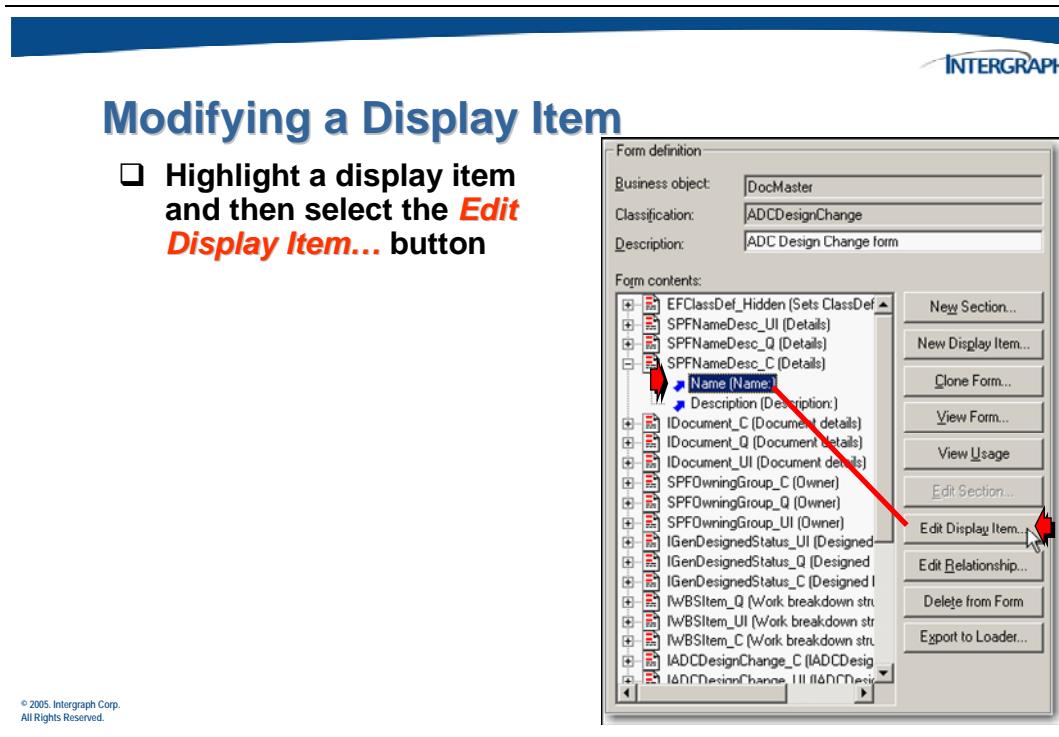
Display items are used to present data or relationships on a data form. Properties or relations are assigned for a display item from a relational table. New display items can be created or display items can be re-used from the database. After a display item is created, the display item can then be added to a section for inclusion on a data form. Once created, the new display item icon appears in the right pane. The display item can then be added to an existing section. Using the **Find Now** button from the *Update Form* dialog, existing sections can be added to a data form to create a relationship.



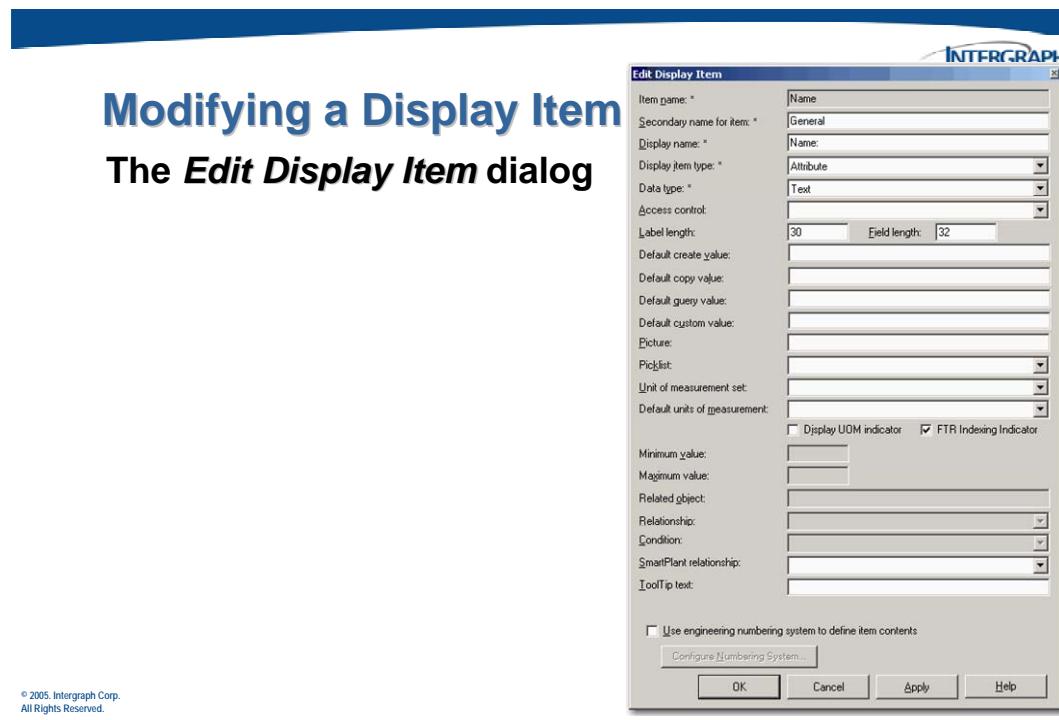
Display Items in Data Forms

- Display items can be used in multiple sections**
- Relationship between section and display item defines:**
 - **sequence number** in the section
 - **top and left position** in the section
 - **required indicator**
 - **read-only or read/write flag**
- All the display items are automatically set to read-only mode by when the form is used for viewing**

Use the *Update Form* dialog to modify existing **Display Items**.

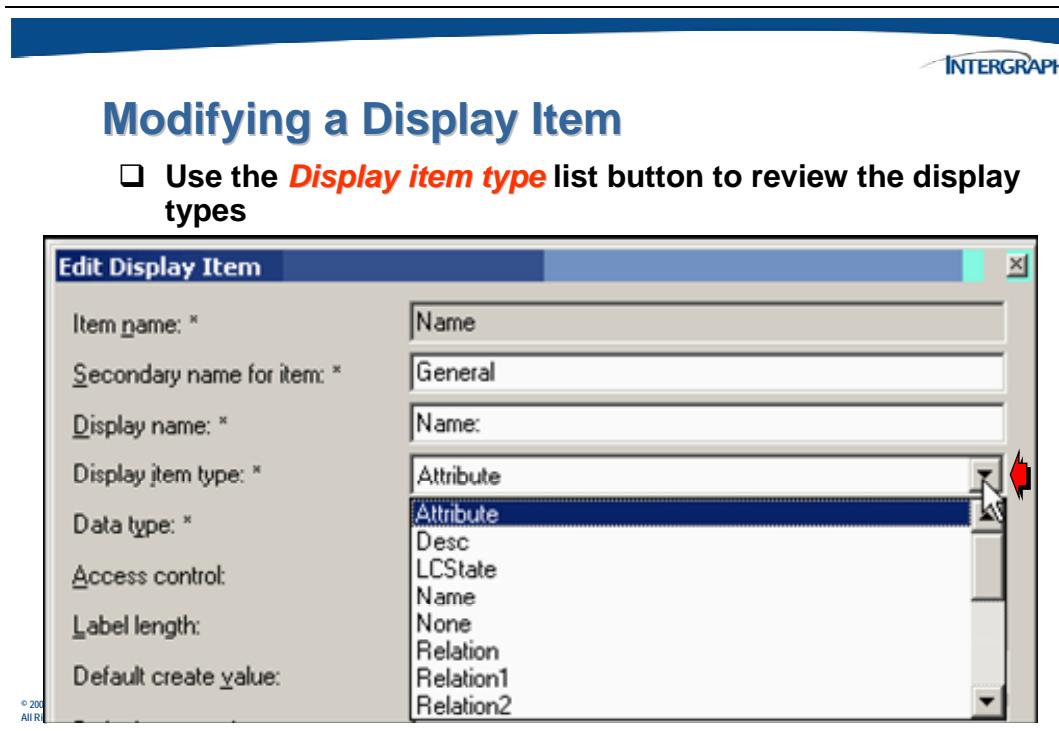


The *Edit Display Item* dialog will display.



The *Edit Display Item* dialog window contains the following fields:

- Item name** - specifies the display item name, is required, and is a read-only field.
- Secondary name for item** - this name provides access to the same data stored in **Item name** but access controls can be applied here to control user access to this data. This is a required field.
- Display name** - specifies the display item name as it should appear on the data form and is required
- Display item type** - use the  button to select a display item type. This is a required field.





Display Item Types

- Attribute** - used to display all data except relationships and the key object data, such as its **Name**. The Attribute display item is used to store and display additional user data against the object.
- Desc** - used for the description of the object in a BO model. Typical values are **Tag** description and **Asset** description.
- LC State** - allows the definition of a display item to handle the POSC/Caesar Life Cycle Statuses. LCState is a special type of display item that has specific handling functionality built into SPF.

Life Cycle Statuses are:

- Planned
- Actual

© 2005, Intergraph Corp.
All Rights Reserved.



Display Item Types

- Name** - used for the names of objects. Name is a special type and is used in conjunction with the object key to define the object's unique ID.
- None** - used to define label text for the dataform. The None display item type allows the definition of headers, column headers, and row headers within the data form.
- Relation** - used to define a display item where the value is derived from another object, for example, **System** or **Area**. The data displays as a code and a description in two separate boxes. The code can be typed or selected from a list. This display item type **is not used** for schema object models.

© 2005, Intergraph Corp.
All Rights Reserved.



Display Item Types

- EFRelation1** - usually defined as a read-only display item, which is used to relate two objects together in a parent/child way. For example, when you want to create a new item such as transmittal section and relate it to a transmittal, you would normally accomplish this from the shortcut menu on the transmittal. Since this type of display item is read-only, it is placed in a section that has the "visible" indicator set to false (invisible). This display item type can be used only with schema objects and relations.
- EFRelation2** - is used to define a display item where the value is derived from another object, for example, contract. The data displays in a combo box. This display item type can be used only with schema objects and relations.

© 2005, Intergraph Corp.
All Rights Reserved.



Display Item Types

- MultiRel** - used on query forms as a list box for multiple selections. Selections are collected for queries by using the logical **OR** operator. An example of using this type of display item would be on a system item where the user can select multiple values and SPF will query against all selected values. This is the multi-selection version of **Relation**. The data displays on the form in a box containing 4 lines of data for multi-selection.
- Object** - used for validation items that cannot be defined as Relation. For example, the display item type **Object** would be used in a situation where validation against a tag contained a list of all tags that is too large to display. This field type allows you to type in the name of the object (tag number), then SPF displays the description after validating the name.

© 2005, Intergraph Corp.
All Rights Reserved.

When you select the **display item type** to use in the *Edit Display Item* dialog, the selection determines the additional fields that are activated according to the selection.

The *Edit Display Item* dialog window fields (con't):

- Data Type** - specifies which *Data Type* can be selected from the list and is a required field. If type *Attribute* is chosen, then the valid selections are:
 - **Text** - used for text
 - **Quantity** - used when value and unit of measure are both used
 - **Boolean** - displays a check box
 - **DateTime** - used for date and time display. The picture definition can be used to control the format.
 - **Count** - used for the values without unit of measure.
- Access control** - limits the user's access to this display item. Select an access control type from the **Access Control** list.



Modifying a Display Item

- Use the **Access control** list button to select the access control list to apply to this item (field) in the form.

The screenshot shows the 'Edit Display Item' dialog box. On the right side, there is a dropdown menu labeled 'Access control'. A red arrow points to the bottom-right corner of this dropdown, where a mouse cursor is hovering over a small square icon. The dropdown menu lists three options: 'ADMINONLY', 'GENERAL', and 'INST_Design'. The rest of the dialog box contains other fields like 'Item name', 'Display name', 'Display item type', 'Data type', 'Label length', and 'Default create value'.

The *Edit Display Item* dialog window fields (con't):

- Label length** - specifies the "Display name" length and contains a numeric value
- Field length** - specifies the input field size and contains a numeric value. The input is truncated to the specified size before it is stored in the database.



Additional Display Item Definitions

- Label and field sizes**
- Tool tip text**
- UOM indicator (where applicable)**
- Relationship (where applicable)**
- Minimum and/or maximum value (where applicable)**
- ENS (Engineering Numbering System)**

© 2005, Intergraph Corp.
All Rights Reserved.

- Picklist** - used to select a pre-defined list to attach to the display item. This value is only valid for the *Display Item Types*: *Name*, *LCState*, and *Attribute* with a *Data Type* of *Quantity* or *Text*.
- Picture** - defines a data format value for the field. The format can either be *U* for *Uppercase* or any picture mask as defined in *Picture Mask Types*. This option is available for all *Display Item Types* except *Attribute* and *None*.



Modifying a Display Item

- Enter an optional picture mask type.**

The dialog box has the following fields:

Item name: *	Name
Secondary name for item: *	General
Display name: *	Name:
Display item type: *	Attribute
Data type: *	Text
Access control:	
Label length:	30
Field length:	32
Default create value:	
Default copy value:	
Default query value:	
Default custom value:	
Picture:	U

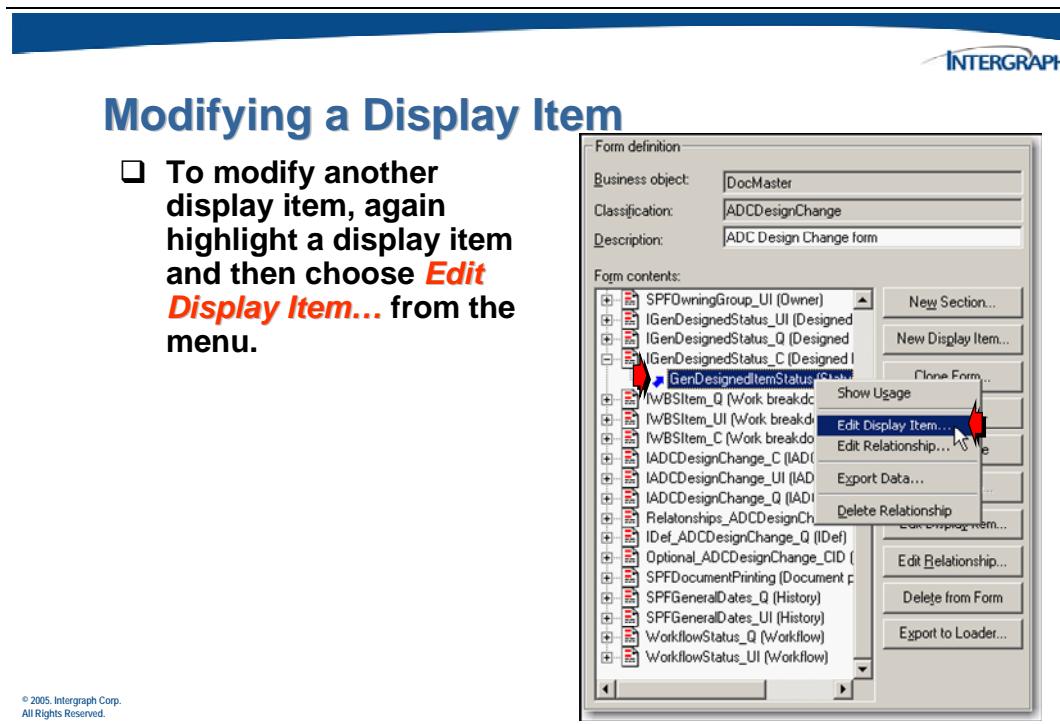
- Click OK to update this *Display Item***

© 2005, Intergraph Corp.
All Rights Reserved.

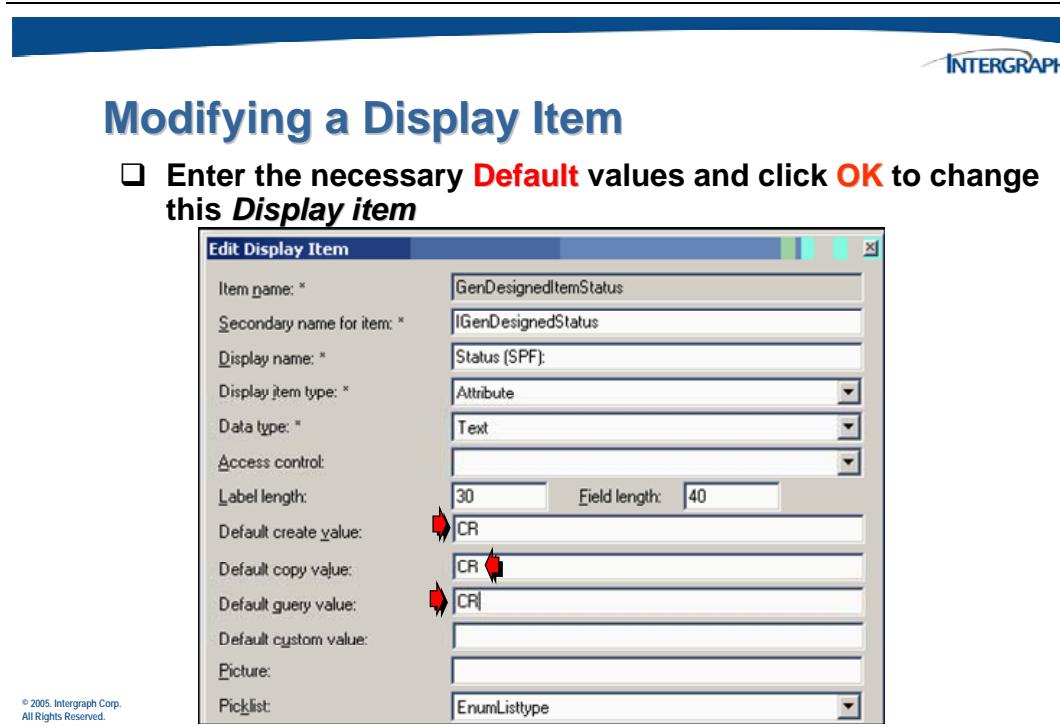
The picture mask can contain one or more of the following characters:

Character	Description
U	Uppercase.
#	Digit placeholder.
.	Decimal placeholder. The actual character used is the one specified as the decimal placeholder in international settings. This character is treated as a literal for masking purposes.
,	Thousands separator. The actual character used is the one specified as the thousands separator in international settings. This character is treated as a literal for masking purposes.
:	Time separator. The actual character used is the one specified as the time separator in international settings. This character is treated as a literal for masking purposes.
/ or -	Date separator. The actual character used is the one specified as the date separator in international settings. This character is treated as a literal for masking purposes.
\	Treat the next character in the mask string as a literal, allowing the '#', '&', 'A', and '?' characters to be included in the mask. This character is treated as a literal for masking purposes.
&	Character placeholder. Valid values for this placeholder are ANSI characters in the following ranges: 32-126 and 128-255.
>	Convert all the characters that follow to uppercase.
<	Convert all the characters that follow to lowercase.
A	Alphanumeric character placeholder (entry required). For example: a – z, A – Z, or 0 – 9.
a	Alphanumeric character placeholder (entry optional).
9	Digit placeholder (entry optional). For example: 0 – 9.
C	Character or space placeholder (entry optional). This operates exactly like the '&' placeholder, and ensures compatibility with Microsoft Access.
?	Letter placeholder. For example: a – z or A – Z.
Literal	All other symbols are displayed as literal.

Review and modify any of the existing display items.



In this example, setting some of the defaults is shown.



The *Edit Display Item* dialog window fields (con't):

- Default create value** – specifies the value the field will default to during a create function
- Default copy value** - specifies the value the field will default to during a copy function
- Default query value** - specifies the value the field will default to during a query function
- Default custom value** – Not Used

Special default values are available for use with *Display Items* to access data from related items and the environment session. The following default values are available:

ENV.PLANTNAME - inherits the name of the plant from the selected create configuration.

ENV.PROJECTNAME - inherits the name of the project from the selected create configuration.

PARENT.NAME – is used on a relationship display item to inherit the name of the related object.

PARENT.DESCRIPTION - is used on a relationship display item to inherit the description of the related object.

PARENT.CLASS - is used on a relationship display item to inherit the business object class name of the related object.

PARENT.CLASSIFICATION - is used on a relationship display item to inherit the classification of the related object.

PARENT.ID1 - is used on a relationship display item to inherit the manufacture ID of the related manufacture.

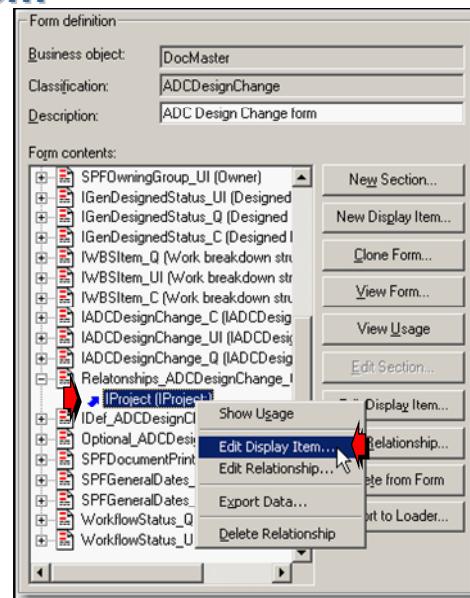
PARENT.REL.RelationshipName - is used to first retrieve the parent object, then navigate across a specified relationship to retrieve the name object related by this relationship.

NOW - is used with date display items, and defaults to a current date and time.



Modifying a Display Item

- To modify the Project display item, right-click on the IProject display item and then choose **Edit Display Item...** from the menu.



© 2005, Intergraph Corp.
All Rights Reserved.



Modifying a Display Item

- The ENV.PROJECTNAME default value inherits the name of the project from the selected create configuration.

Edit Display Item	
Item name: *	Rel_ProjectDesignChange
Secondary name for item: *	IProject
Display name: *	Project:
Display item type: *	ERelation2
Data type: *	
Access control:	
Label length:	30
Field length:	40
Default create value:	ENV.PROJECTNAME

© 2005, Intergraph Corp.
All Rights Reserved.



Modifying a Display Item

SPF uses the relationship specified in the SmartPlant relationship field to create the relationship between a new Design Change document and the current active project.

Minimum value:

Maximum value:

Related object:

Relationship:

Condition:

SmartPlant relationship: DYNEDG_ProjectDesignChange_21::DesignChange

ToolTip text:

© 2005, Intergraph Corp.
All Rights Reserved.

Units of Measurement (UOMs) in SmartPlant Foundation are set up to identify the units in which the particular property is recorded. UOM sets are groups of UOMs that measure the same property, but in different units. A UOM set can be attached to a dataform display item to provide a picklist of valid UOMs.



Modifying a Display Item

- To add a UOM to a display item, highlight the display item and then choose the **Edit Display Item...** button

Form definition

Business object: DocMaster

Classification: ADCDesignChange

Description: ADC Design Change form

Form contents:

- SPFOwningGroup_Q (Owner)
- SPFOwningGroup_UI (Owner)
- IGenDesignedStatus_Q (Designed)
- IGenDesignedStatus_C (Designed)
- IwBSItem_Q (Work breakdown str)
- IwBSItem_UI (Work breakdown str)
- IwBSItem_C (Work breakdown str)
- IADCDesignChange_C (IADCdesig
- ADCDetail (Change Detail)
- ADCMeas (Change UOM)
- ADCImpact (Change Imp)
- ADCPossibleSignature (
- ADCApprovalSignature (
- IADCDesignChange_UI (IAC
- IADCDesignChange_Q (IAD
- Relationships_ADCdesignC
- IDef_ADCDesignChange_Q (
- Optional_ADCDesignChange_CID (
- SPFFDocumentPrinting (Document

© 2005, Intergraph Corp.
All Rights Reserved.

In this example, configuring a Unit of Measurement (UOM) is shown.

Modifying a Display Item

Configure or change the desired **Unit of measurement set** and **Default units of measurement**.

- Unit of measurement set** - used to select a Unit of Measurement (UOM) from the list to associate to the display item. The list is defined by the units of measurement that are created within the System Administration module. For example, the *Length* list contains values ft, cm, etc. This option is only available for the **Display Item Types**: *Name*, *Desc*, *LCState*, and *Attribute* with a **Data Type** of *Quantity* or *Text*.
- Default units of measurement** - used to select a default UOM value to apply to the display item during create operations. This option is only available for a **Display Item Type** of *Attribute* with all **Data Types** except *Boolean*.
 - **Display UOM Indicator** - check box used to display UOM. This option is only available for a **Display Item Type** of *Attribute* with all **Data Types** except *Boolean*.
- FTR Indexing Indicator** - used to indicate that the display item will be processed by the FTR index server for FTR searching. This option is only available for a **Display Item Type** of *Attribute* with a **Data Type** of *Text*.
- Minimum value** - defines the low value for the display item in the field. This only applies to a **Display Item Type** of *Attribute* with a **Data Type** of *Quantity*.
- Maximum value** - defines the high value for the display item in the field. This only applies to a **Display Item Type** of *Attribute* with a **Data Type** of *Quantity*.
- Related object** - defines the object filter to limit the number of related objects listed. For example, value **Plant** would limit the values only to those belonging to

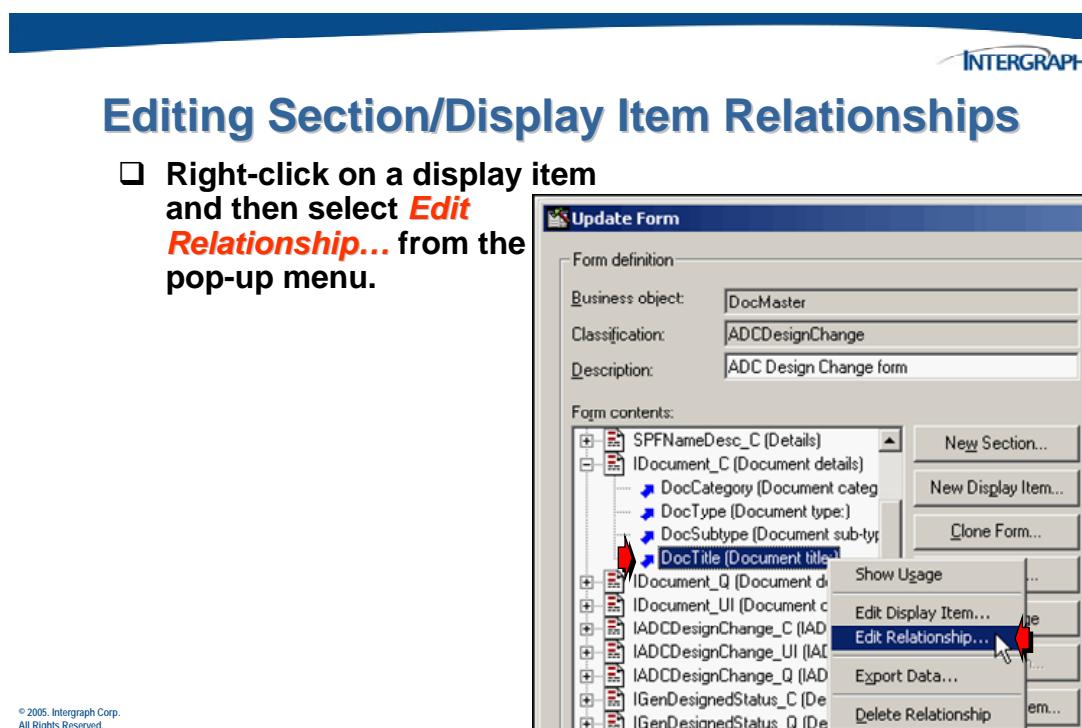
active plant. The filter only applies to the following **Display Item Types**: *Relation, Relation1, Relation2, MultiRel, MultiRel2*, and *Object*.

- Relationship** – defines a relationship to filter its values to a picklist for this field.
Display Item Types: *Relation, Relation1, Relation2, MultiRel, MultiRel2*
 - SmartPlant relationship** - defines the relationship name to use to retrieve the values for the display item. The relationship name only applies to the following
Display Item Types: *Attribute, LCState, EFRelation1*, and *Object*.
 - ToolTip text** - specifies the text to be displayed when the user pauses the cursor over the icon.
 - Use Engineering Numbering System to define item contents** - automatically sets up the field based on other display items. This option is available for **Display Item Types**: *Name, Desc*, and *Attribute* with a **Data Type** of *Text*.
-

Note: The ENS will be covered in detail in a later section of this chapter.

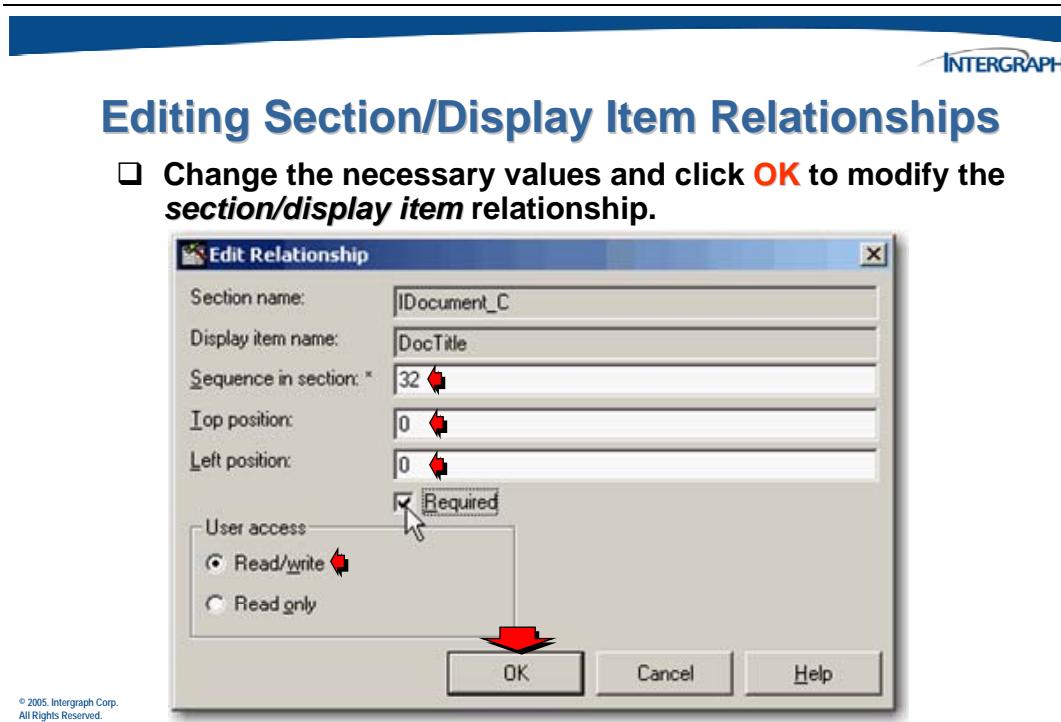
3.2.4 Display Item to a Form Relationships

Dragging and dropping display items onto a dataform section will create a relationship between the section and the display item. The defined display item relationship can be viewed and/or edited using the *Edit Relationship...* command from the *Update Form* dialog.



The display item order is determined by the sequence number entered in the relationship dialog.

The *Edit Relationship* dialog will display.

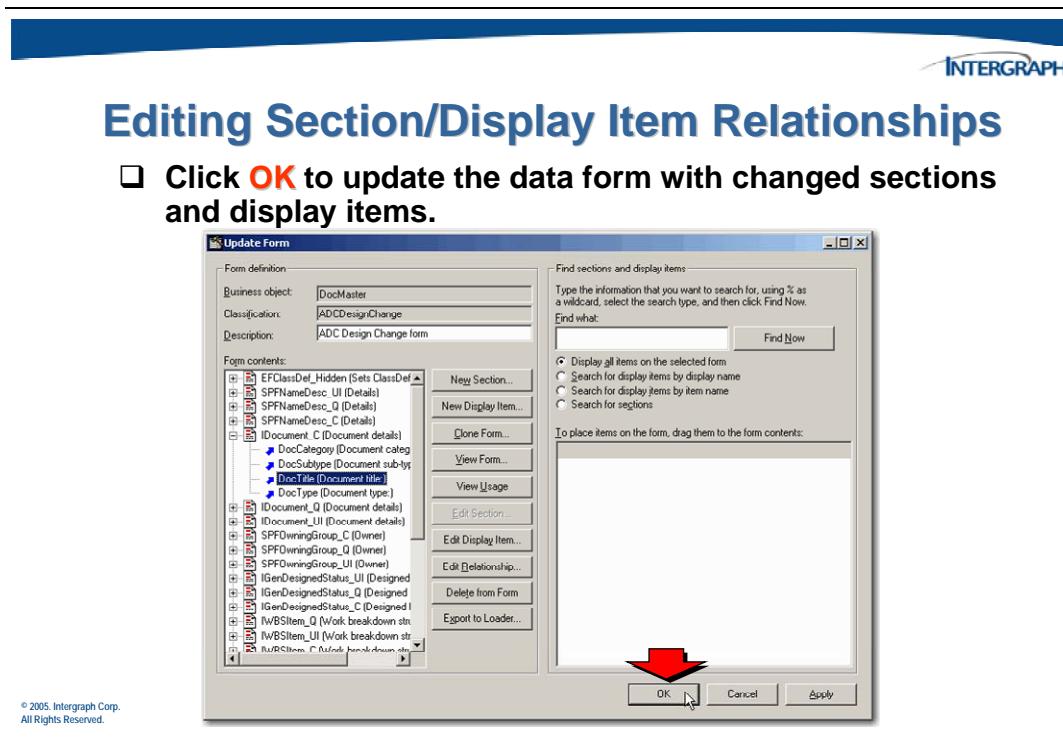


The *Edit Relationship* dialog window contains the following fields:

- ❑ **Section name** - this field displays the name of the section participating in the relationship. It is a read-only field.
- ❑ **Display item name** - this field displays the name of the display item participating in the relationship
- ❑ **Sequence in section** - defines the display item's sequence position within the section
- ❑ **Top position** - specifies a number that defines the number of rows between this display item and the beginning of the section
- ❑ **Left position** - specifies a number that defines the number of columns from the left of the data form that the display item should be indented
- ❑ **Required** - specifies that this display item contains required values that must be entered during *create* or *update* operations
- ❑ **User access** - defines an option from the list to specify this display item as:
 - **Read/write** – allows users read and write access to this field in the data base
 - **Read only** - prevents the user from executing data storage to the data base

Click **OK** on the *Edit Relationship* dialog to modify the section/display item relationship.

Once all of the section/display items have been modified, you can exit from the *Update Form* dialog.

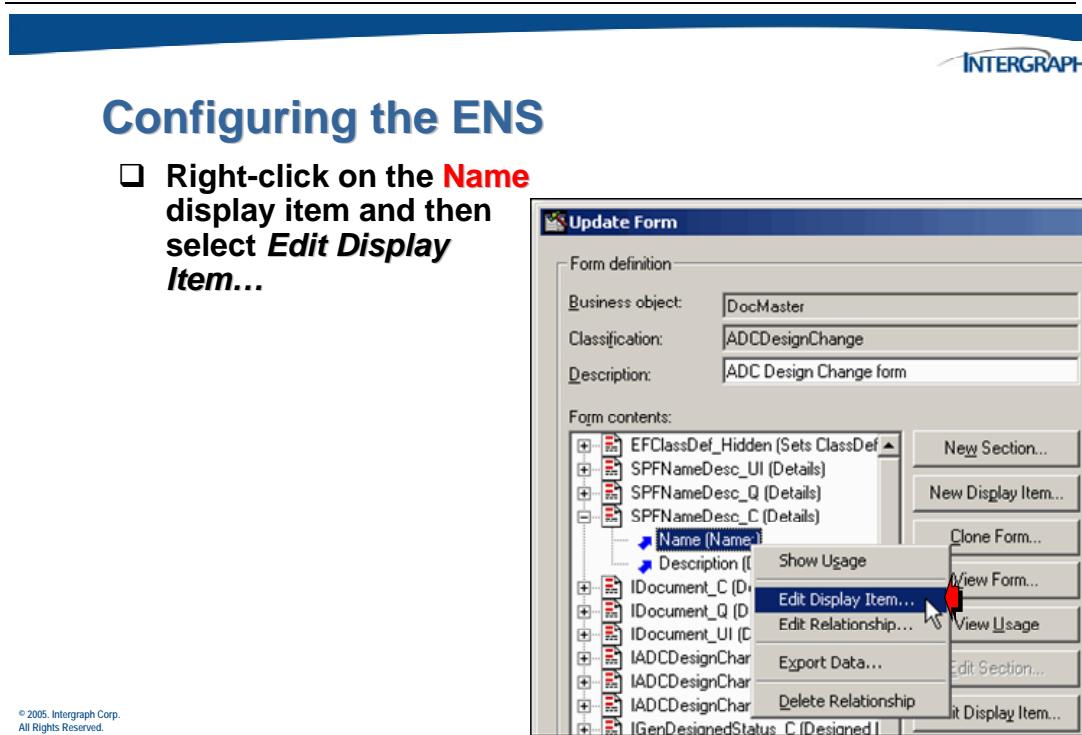


3.3 Configuring the Engineering Numbering System

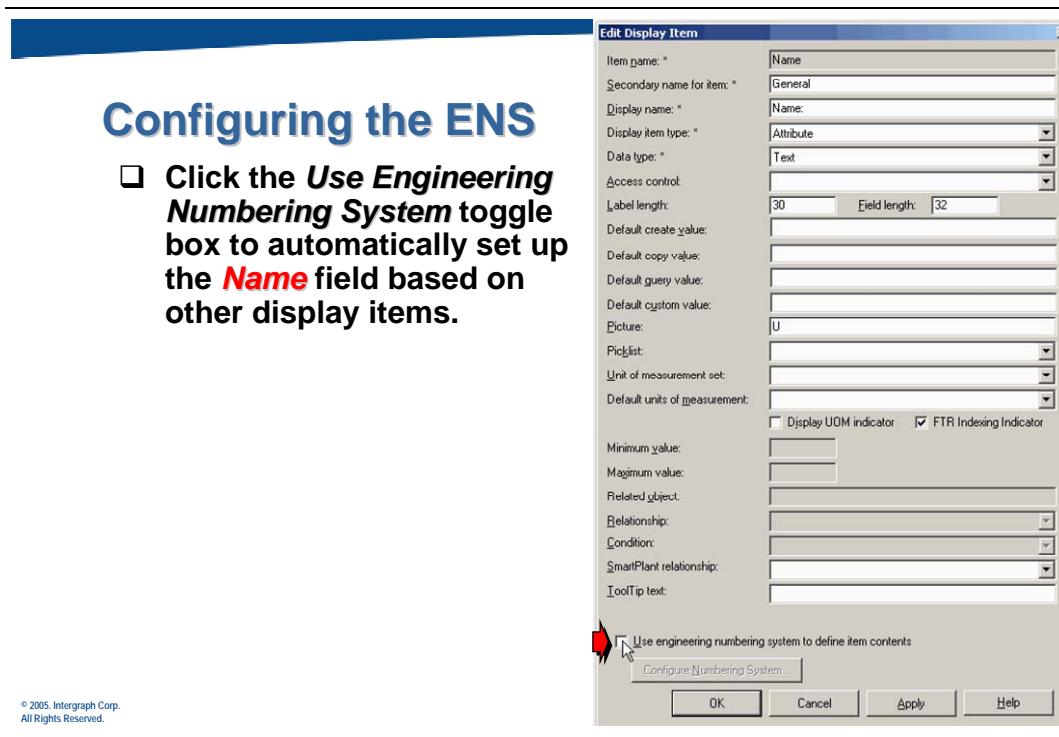
The SPF Engineering Numbering System (ENS) provides an automated object identification facility, which allows project-specific identification formats to be configured as "templates". SPF then uses these templates to guide the user when creating Business Object Identifiers. The **ENS** template is composed of various components that extract data from other display items on the form. These components are configured in sequence with additional delimiter characters to define the required template. This option is available for the following display item types: **Name**, **Desc**, and **Attribute** with a data type of *Text*.

The display item must be attached to a form and the form must be saved before the numbering system can be configured. A form may contain only one ENS field.

In the following example, the *ADCDesignChange* object name will be determined by using the ENS. It will consist of a combination of the *ADCDesignedItemStaus* property, the *IProject* property, and a unique *sequence* property.

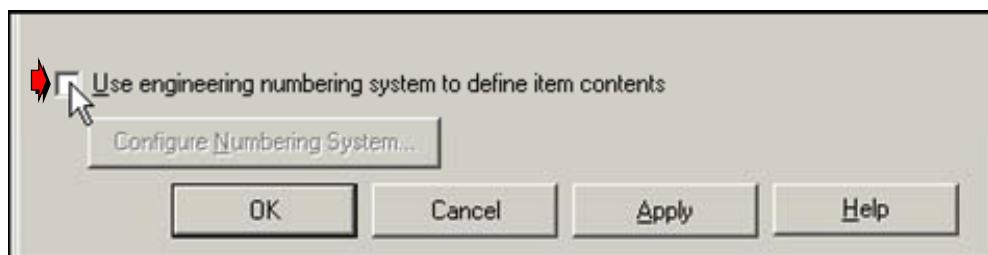


The *Edit Display Item* dialog will display.

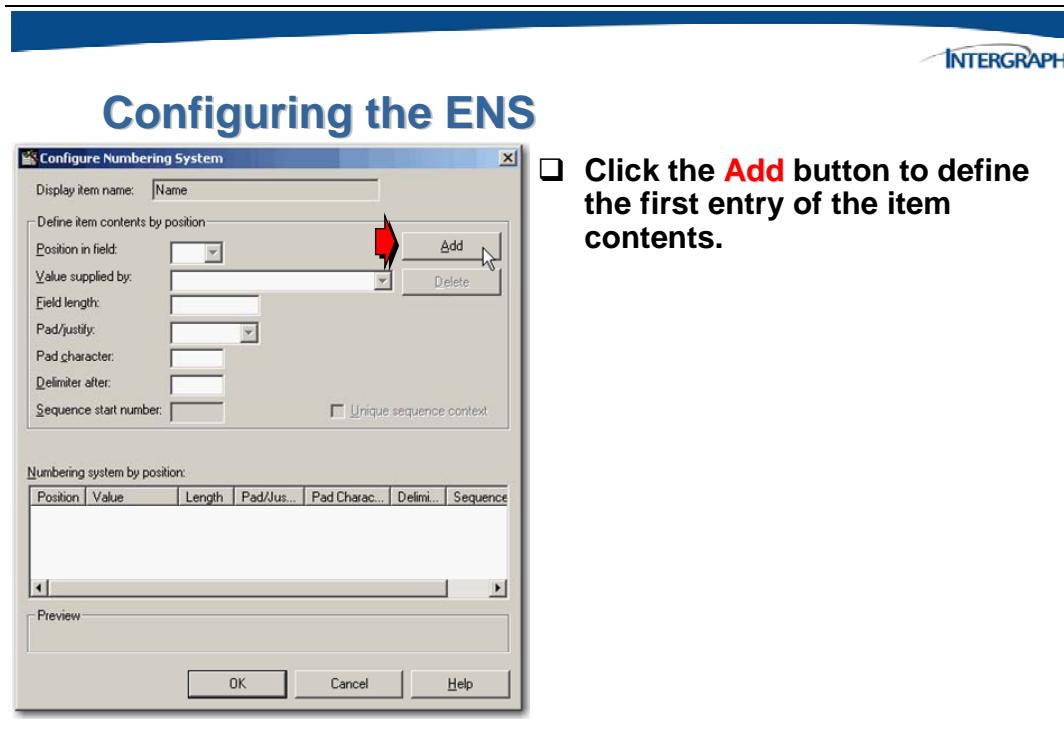


Configuring the ENS

Zoomed in view of the *Edit Display Item* dialog.



The *Configure Numbering System* dialog will be displayed.

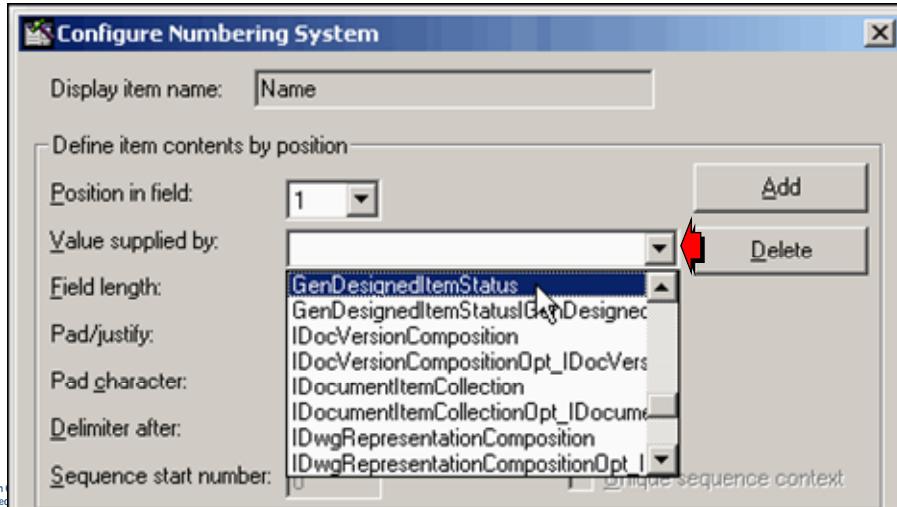


The following describes the fields in the *Configure Numbering System* dialog:

- ❑ **Position in field** - Allows you to define the position in the formatted ENS name for the component being added or edited.

Configuring the ENS

- ❑ In the **Value supplied by** field, click the and select an item from the available list.



The *Configure Numbering System* dialog (con't):

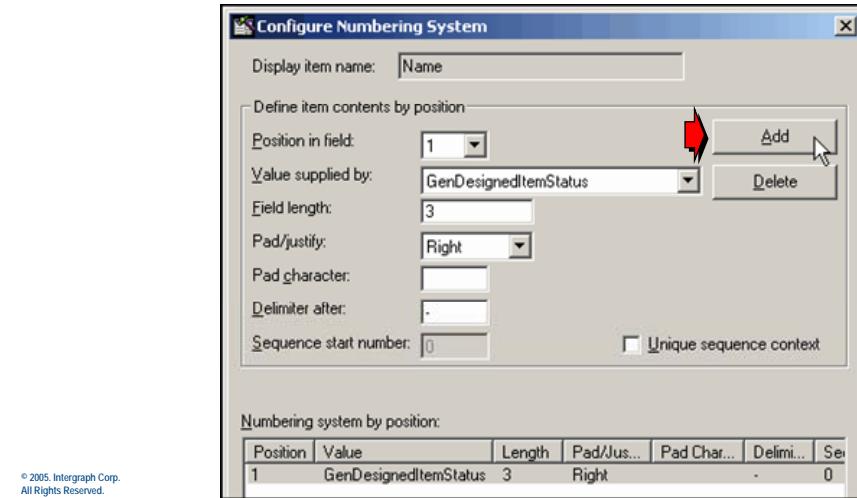
- ❑ **Value supplied by** - allows you to define what other display item should be used to construct the identifier.
- ❑ **Field length** - specifies the minimum overall length of the field. This field can be edited.
- ❑ **Pad/justify** - specifies if the field should be padded on the Left, Right, or contain no padding. This field can be edited.
- ❑ **Pad character** - allows you to enter a pad character for the field if the Pad/justify box is set to a value other than None.
- ❑ **Delimiter after** - allows you to enter a character that is used to separate this field from any that follow it.
- ❑ **Sequence start number** - allows you to enter the start value for the sequence number component of the template.
- ❑ **Unique sequence context** - allows you to define which fields are counted to find the next available unique sequence number.

Once the first identifier component has been added, click the **Add** button to add the next component.



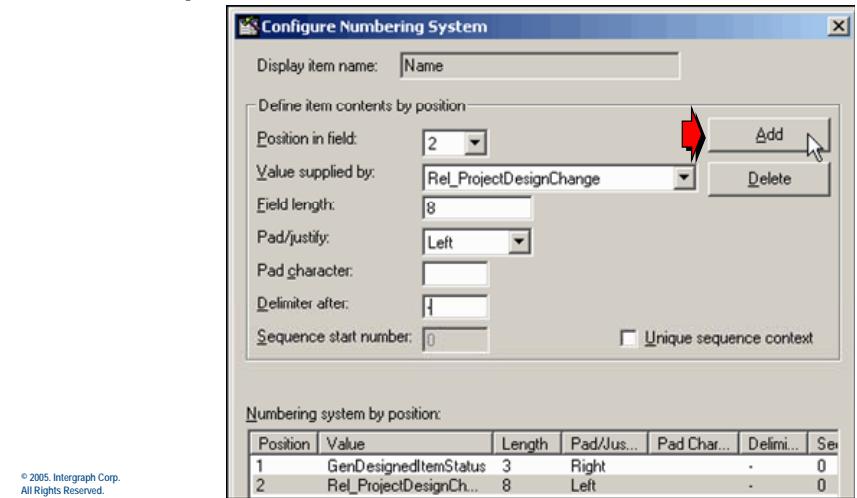
Configuring the ENS

- Repeat the procedure to add the other items that will make up the **ENS** name value.

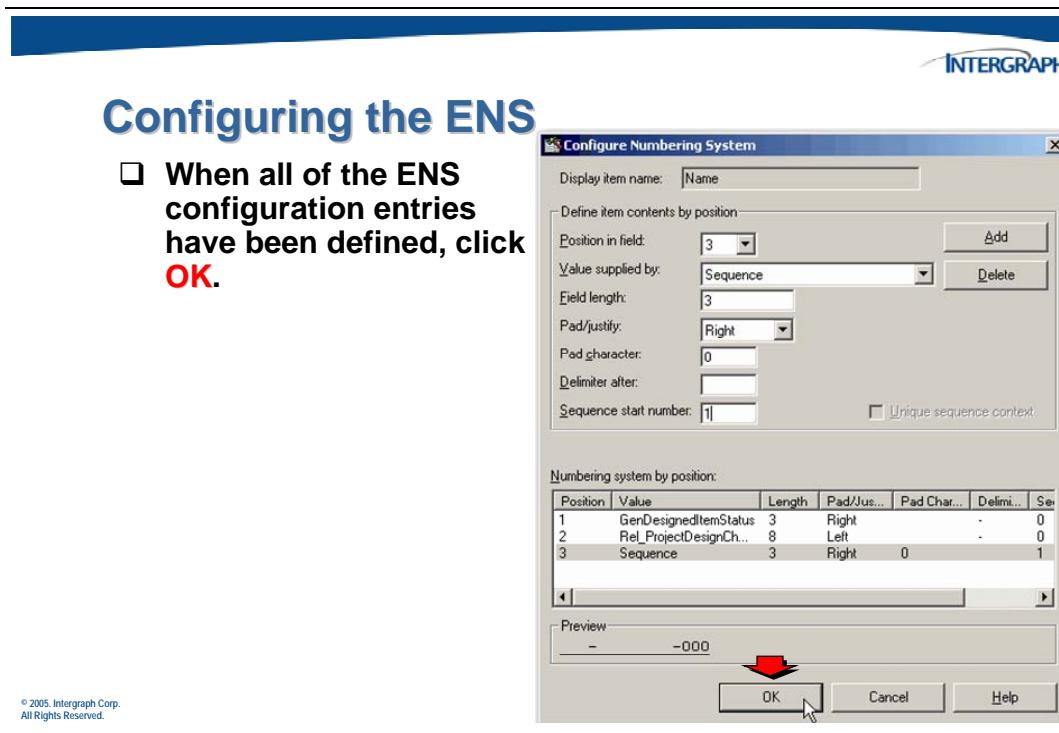


Configuring the ENS

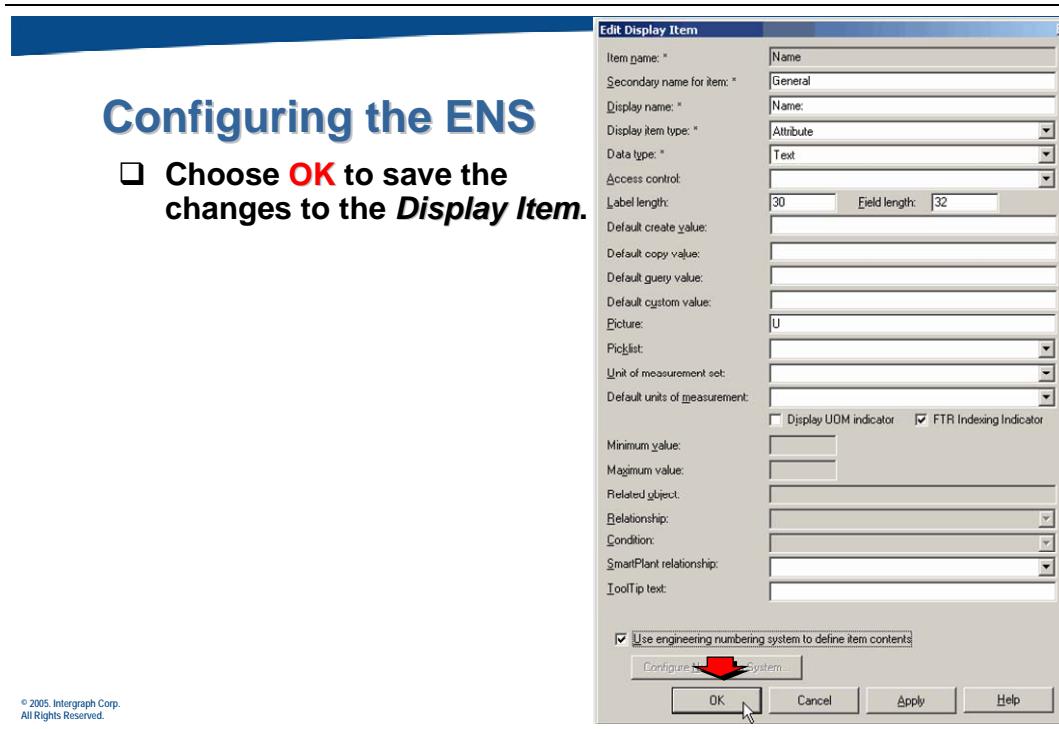
- Repeat the procedure to add the other items that will make up the **ENS** name value.



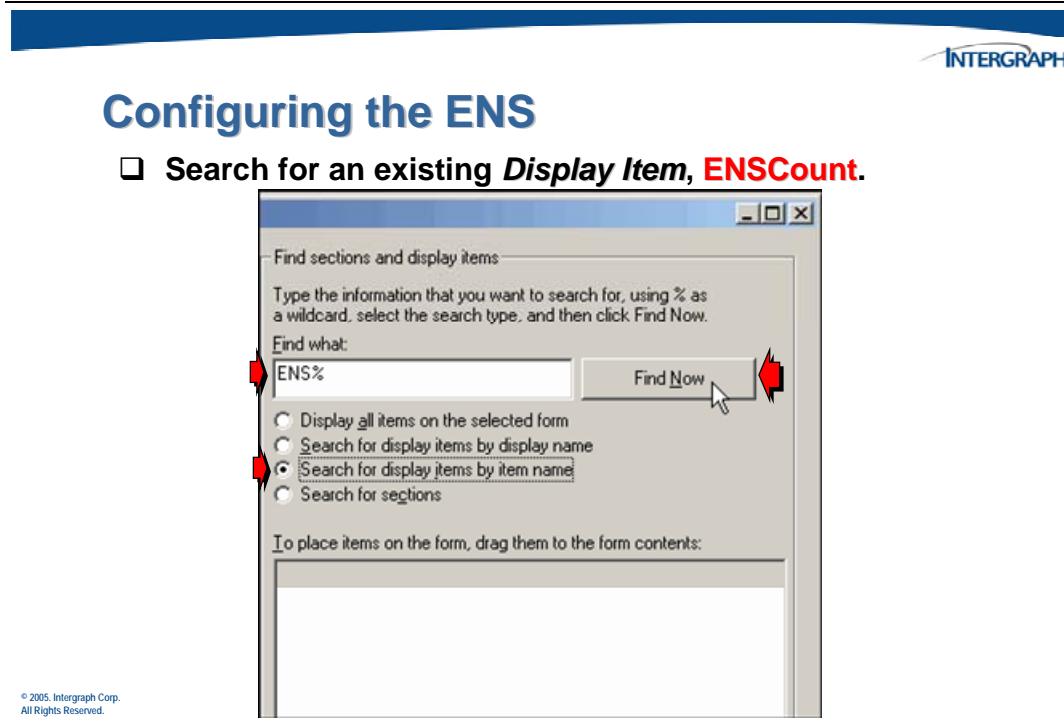
There are now three components that define the ENS identifier.



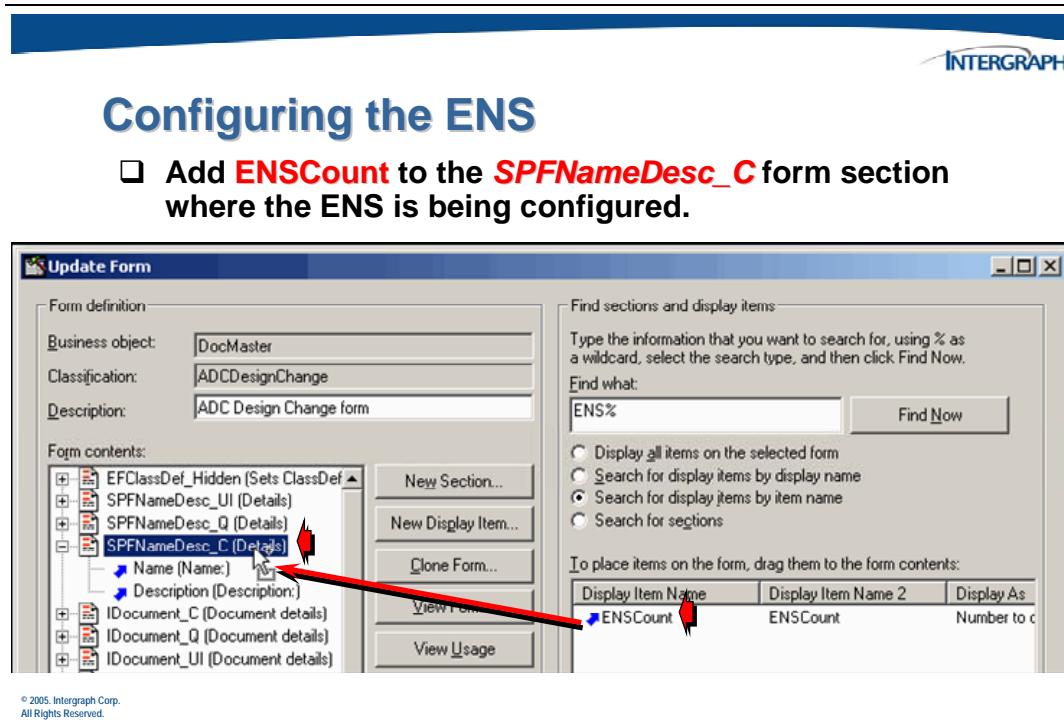
The ENS configuration for the *SPFNameDesc.Name* display item is now complete.



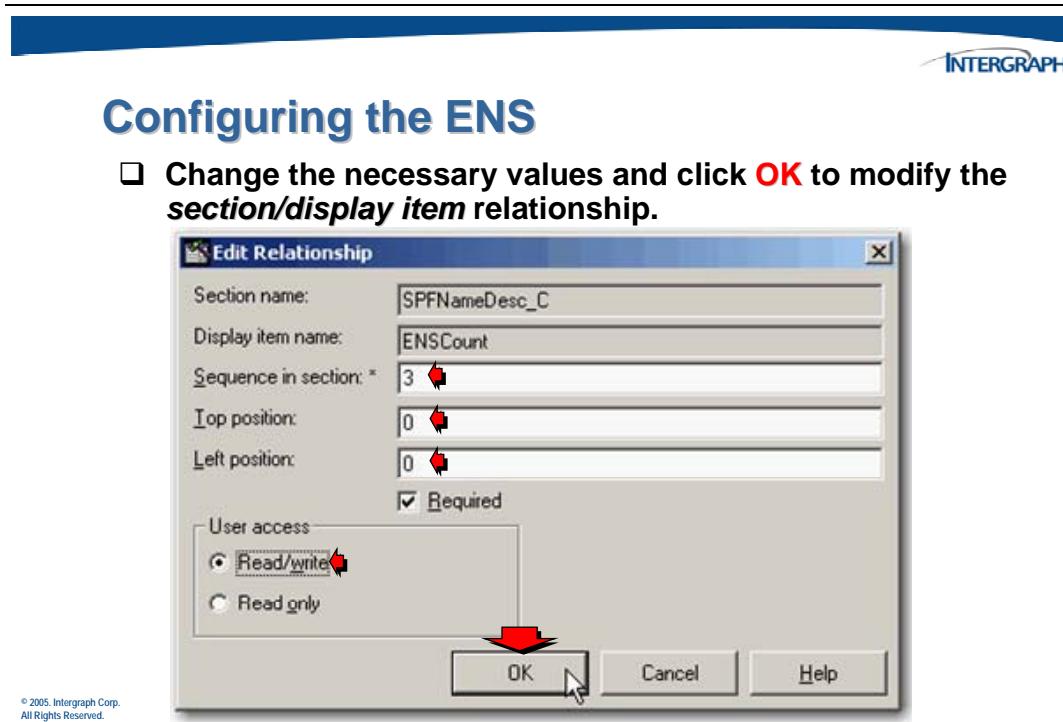
Before the ENS will function, an existing display item called **ENSCount** must be added to the *SPFNameDesc* form section.



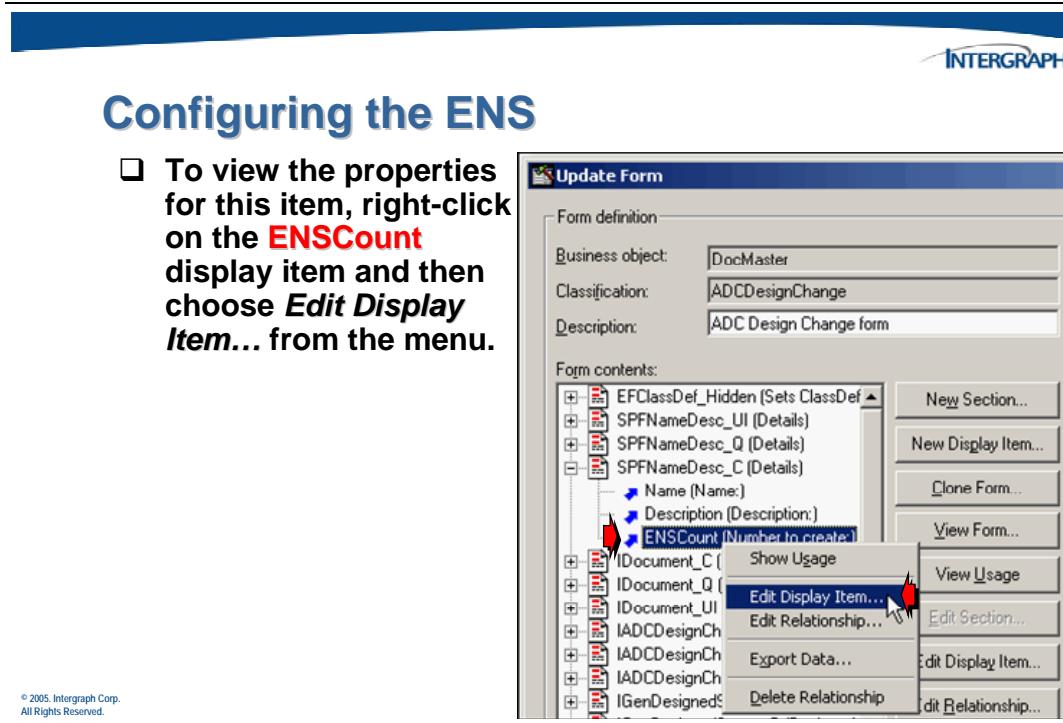
Use a drag and drop to add this display item to the section.



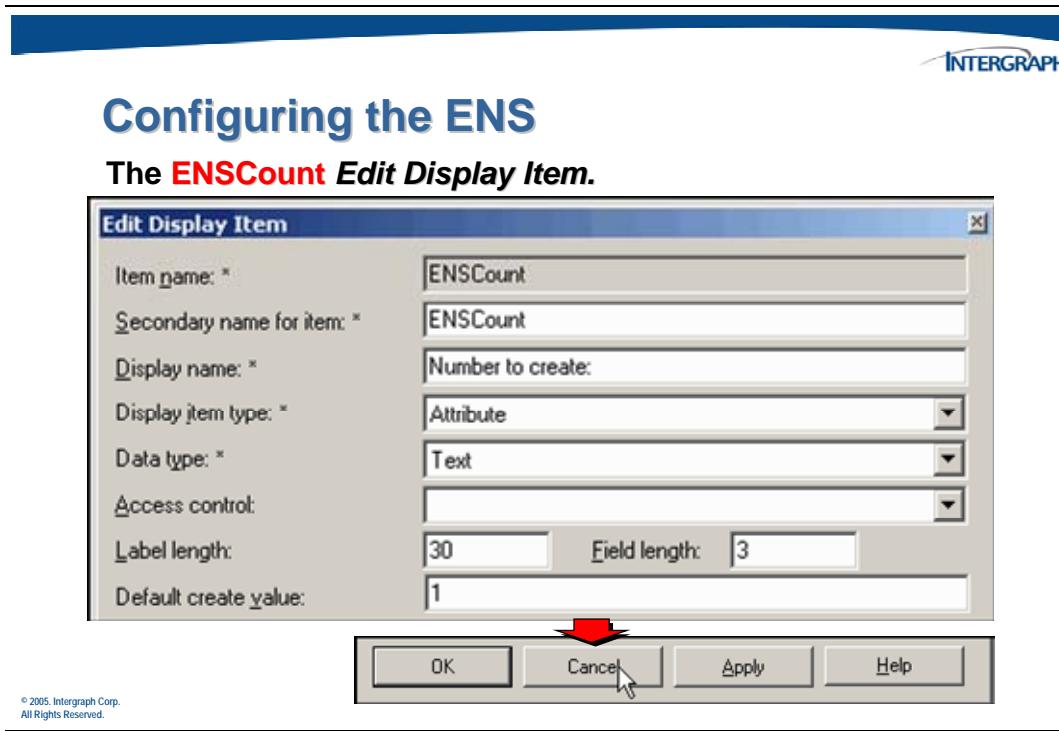
The *Edit Relationship* dialog will display as a result of the drag and drop operation.



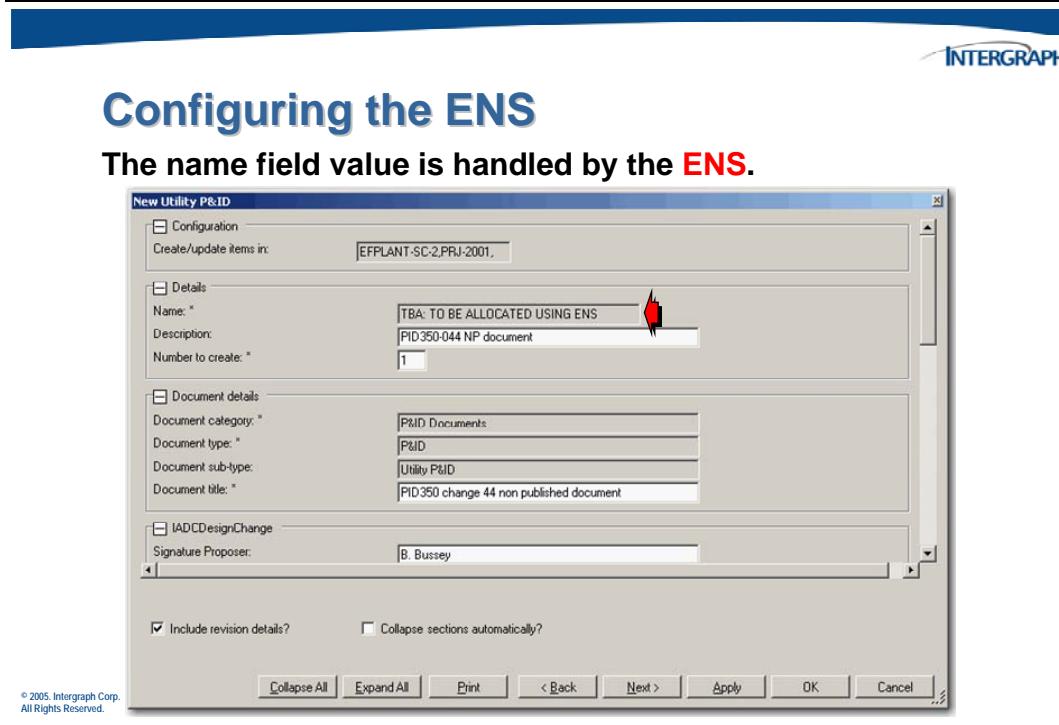
Use the *Edit Display Item* command to review the **ENSCount** display item properties.



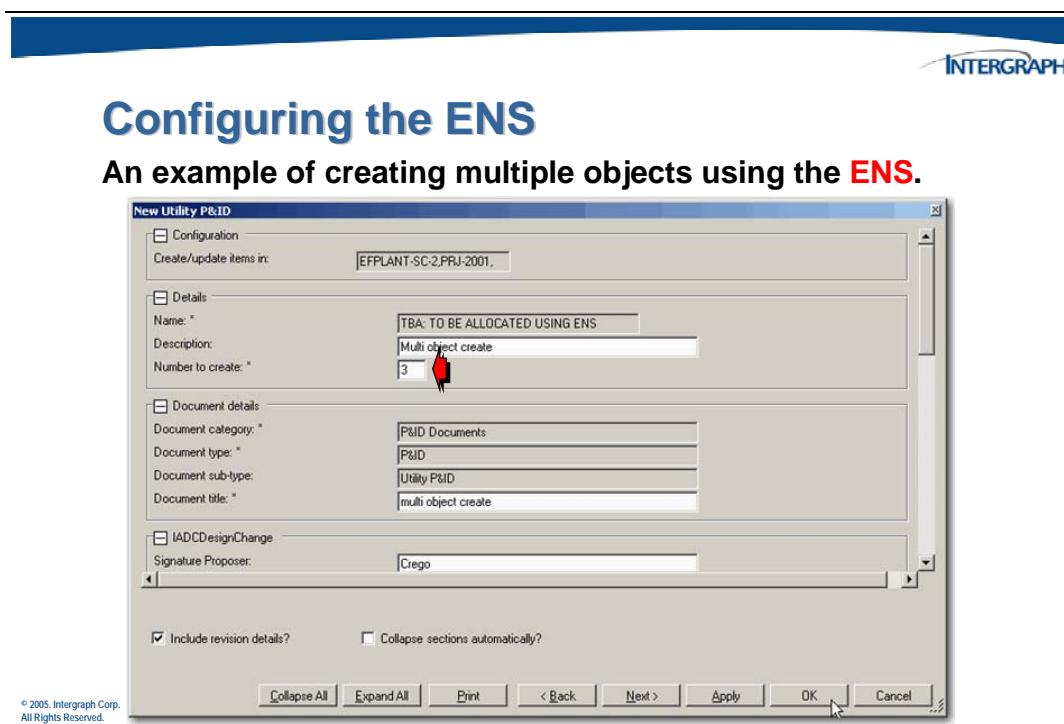
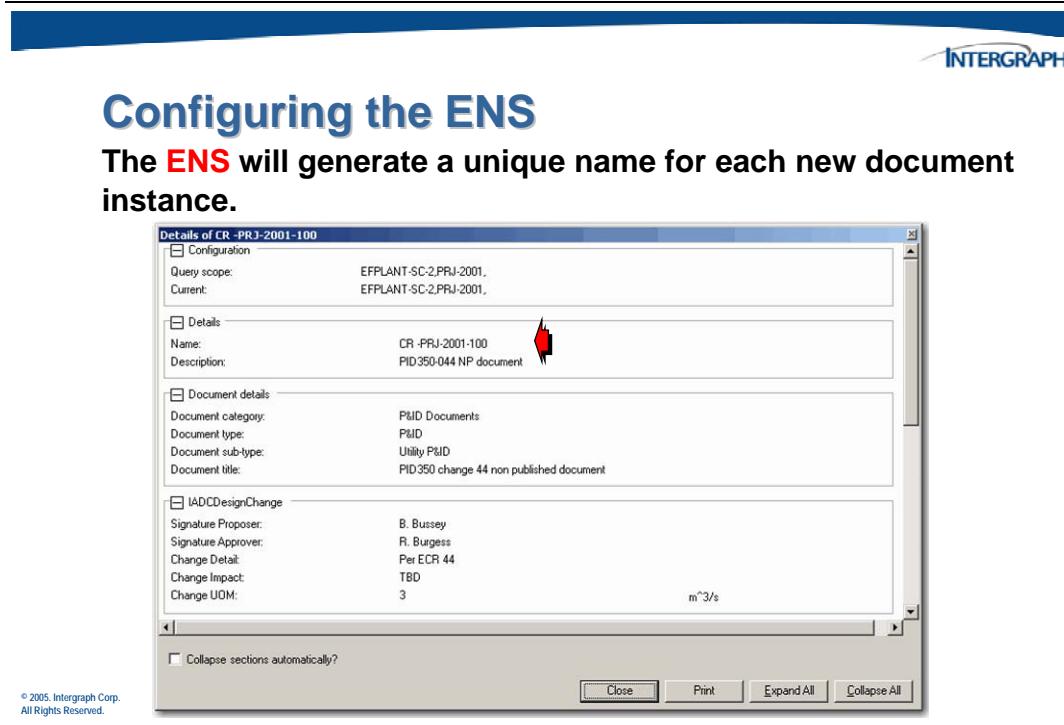
No changes have to be made here so **Cancel** this dialog.



The **Name** field in the form is now unavailable during a create operation – the value will be determined by the *Engineering Numbering System*.



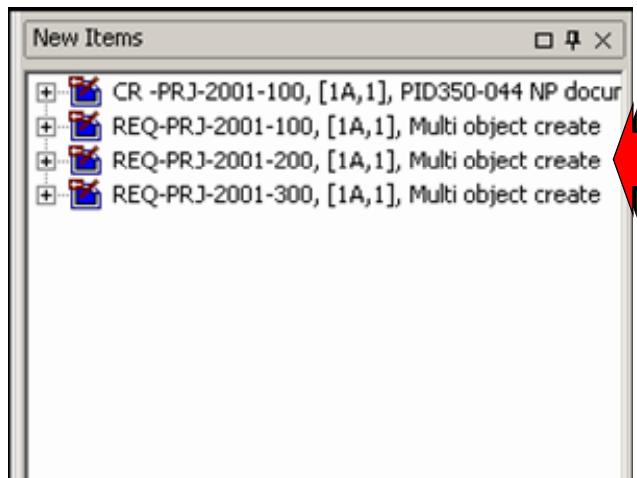
The ENS will take the values from the *Design Status* field (**CR**), the *Project Name* field (**PRJ-2001**) and add a unique *sequence number* starting at the configured value (**1**) to construct the object name.





Configuring the ENS

Again, the **ENS** will generate a unique name for each new object instance.



3.4 Additional Form Builder Functions

The following section will review two utilities included in the *Form Builder* utility, which are useful in creating and working with forms.



Additional Form Builder Functions

- View form**
- Create export file**

© 2005, Intergraph Corp.
All Rights Reserved.

The **View form** utility allows the user to view the format of the forms rows and columns before you save or exit the utility. This utility only views the fields it will not test for relationship or syntax of the entry fields.

The **Export utility** allows the user to export the forms definition into a SPF Loader file so it can be loaded into a different database or reloaded into the same database if needed.

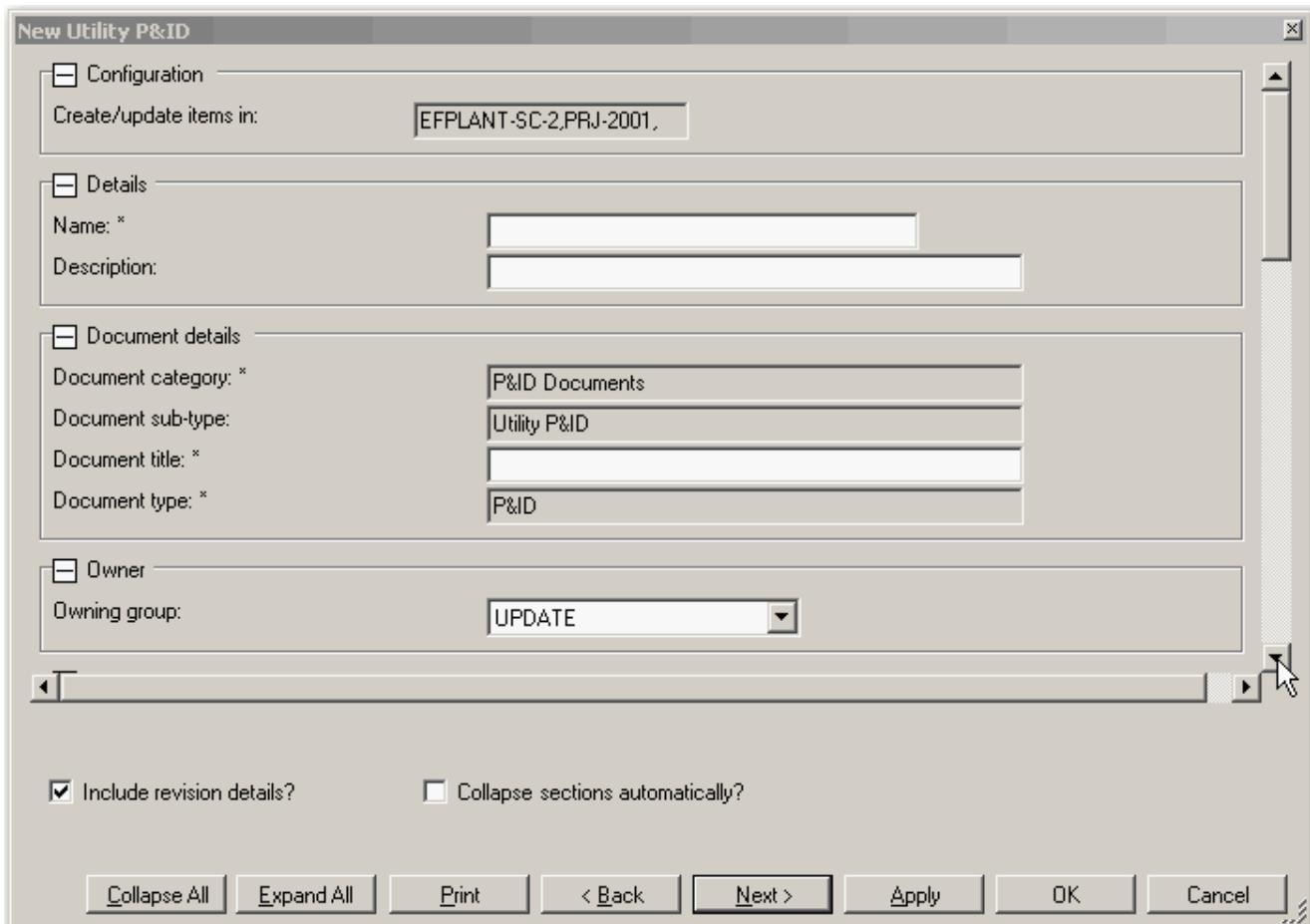
3.5 Activity – Generating and Modifying Forms

The objective of this activity is to generate and modify a data form for the *ADCDesignChange* interface object. You will also use the Edit Relationship command to delete relationships between some of the sections to the form.

If you are not currently logged into your machine:

1. Log on to your operating system (if not already logged in):
spfuser with no password
2. Start the *Schema Object Administration* utility by selecting *Start > All Programs > Intergraph SmartPlant Foundation > Schema Object Administration*.
3. When the *Login* dialog window appears, use the *User name* **adminuser** with no password and click on **OK**.
4. Use the **Find** command to locate and display the new class definition, *ADCDesignChange*.
 - Select the *Tools > Find > Class Definition...* command from the menu.
 - In the *Find Class Definition* dialog, enter “ADC” and the wild card character and click **OK**.
5. Convert this class definition to a dataform.
 - Right-click on the *ADCDesignChange* class definition
 - Select the **Convert to Form** command from the pop-up menu
6. Exit from the *Schema Object Administration* utility and start the *System Administration* utility by selecting *Start > All Programs > Intergraph SmartPlant Foundation > System Administration*.
7. When the *Login* dialog window appears, use the *User name* **adminuser** with no password and click on **OK**.

8. Use the **Find** command to locate and display the new converted form.
 - Select the **Tools > Find > Form...** command from the menu.
 - In the *Find Form* dialog, enter “ADC” and the wild card character and click **OK**.
9. Right-click on the ADCDesignChange form and select **Update** from the pop-up menu. The *Update Form* dialog will display. The following contains an example of the default form created by the **Convert to Form** command.



New Utility P&ID

Designed Item Status
Status (SPF): CR

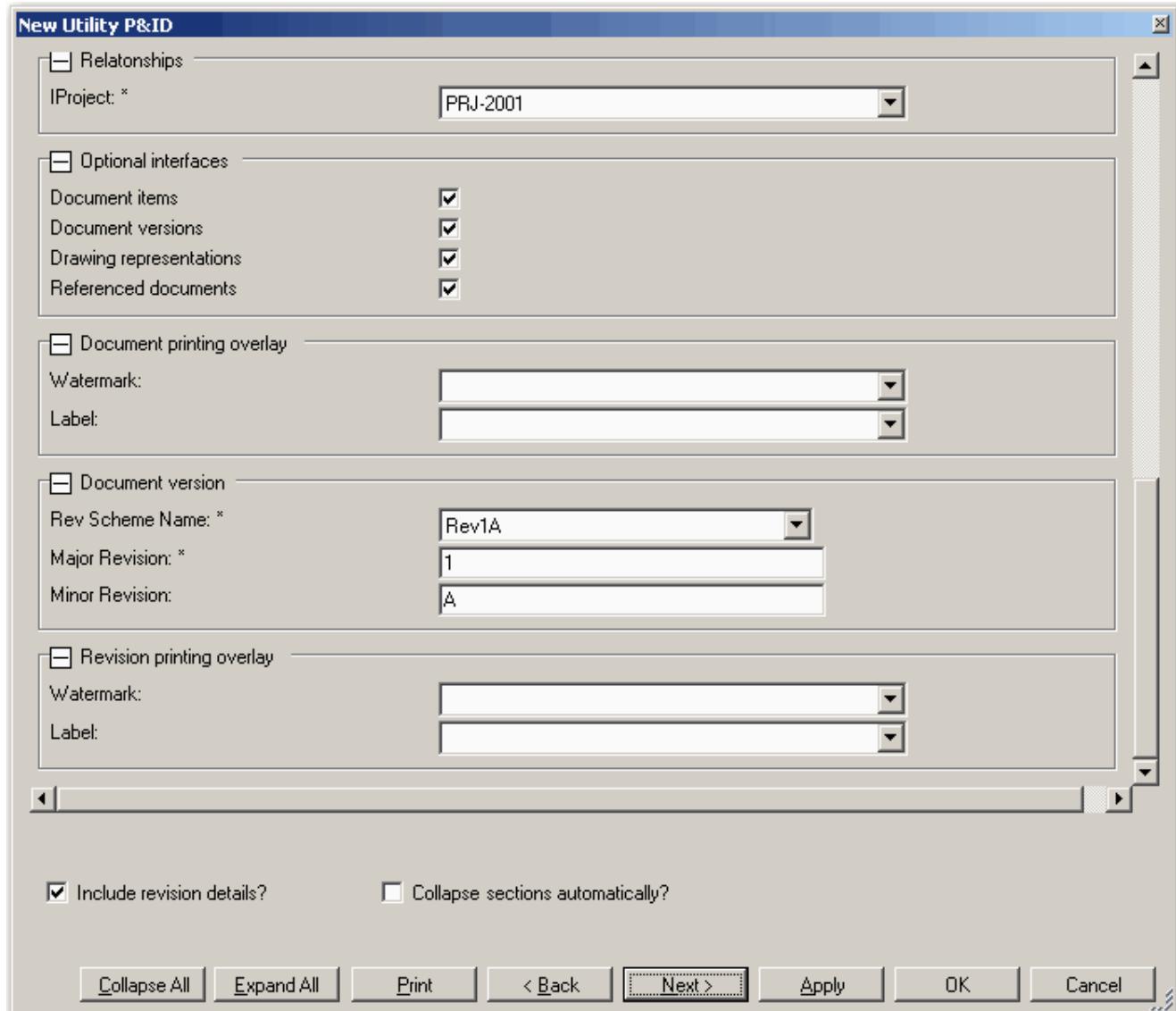
Work breakdown structure item
Purpose:
Type:

IADCDesignChange
Change Detail:
Change UOM: m³/s
Change Impact:
Signature Proposer:
Signature Approver:

Relationships

Include revision details? Collapse sections automatically?

[Collapse All](#) [Expand All](#) [Print](#) [Back](#) [Next >](#) [Apply](#) [OK](#) [Cancel](#)



10. Modify the display items for the *SPFNameDesc_C* section. From the *Update Form* dialog, highlight the **Name** display item and select the **Edit Display Item** button. Change the following values in the *Edit Display Item* dialog:

Picture - U

Click **OK** on the *Edit Display Item* dialog.

11. Modify display items for the *IGenDesignedStatus_C* section. From the *Update Form* dialog, highlight the **GenDesignedItemStatus** display item and select the **Edit Display Item** button. Change the following values in the *Edit Display Item* dialog:

Default create value - CR

- Default query value - **CR**

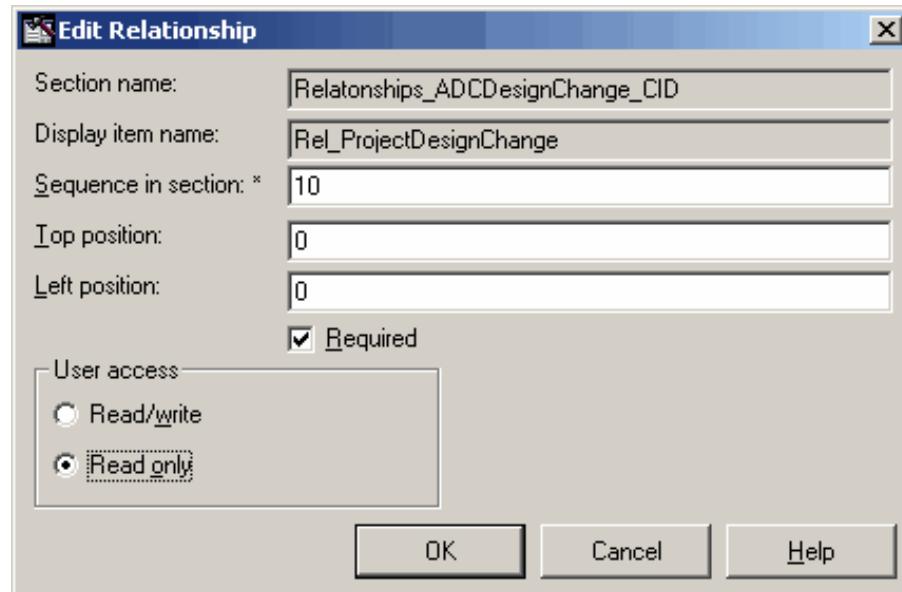
Click **OK** on the *Edit Display Item* dialog.

12. Modify the display name for the *Relationships_ADCDesignChange_CID* section. From the *Update Form* dialog, highlight the **Relationships_ADCDesignChange_CID** section and select the **Edit Section** button. Change the following values in the *Edit Section* dialog:

- Section description - **Configuration scope**

Click **OK** on the *Edit Section* dialog.

13. Edit the Display Item **Rel_ProjectDesignChange(IProject)** under the **Relationships_ADCDesignChange_CID** section. Edit the Relationship on the Display Item and remove the **Required** field indicator. Change the **User access** to *Read only*.



14. Edit the section order on the data form. Have the **IADCDesignChange** section appear after the **IDocument** section in the form. Highlight the section and then select the **Edit Relationship...** button.

- Verify the form name - **ADCDesignChange**
- Verify the Section name - **IADCDesignChange_C**
- Sequence on form * - **17**

Click **OK** on the *Edit Relationship* dialog.

- Verify the form name - **ADCDesignChange**

- Verify the Section name - *IADCDesignChange_Q*
- Sequence on form * - **17**

Click **OK** on the *Edit Relationship* dialog.

- Verify the form name - *ADCDesignChange*
- Verify the Section name - *IADCDesignChange_UI*
- Sequence on form * - **17**

Click **OK** on the *Edit Relationship* dialog.

15. Edit the display item order on the data form. Change the order of the *Document type*, *Document title* and *Document sub-type* display items in the **IDocument** sections. Highlight the display items and then select the **Edit Relationship...** button.

- Verify the Section name - *IDocument_C, Q, UI*
- Verify the Display item name - *DocType*
- Sequence on form * - **23**

Click **OK** on the *Edit Relationship* dialog.

- Verify the Section name - *IDocument_C, Q, UI*
- Verify the Display item name - *DocTitle*
- Sequence on form * - **31**

Click **OK** on the *Edit Relationship* dialog.

- Verify the Section name - *IDocument_C, Q, UI*
- Verify the Display item name - *DocSubtype*
- Sequence on form * - **27**

Click **OK** on the *Edit Relationship* dialog.

16. Edit the display item order on the data form. Change the order of the *ADCDetail*, *ADCMeas*, *ADCImpact*, *ADCProposalSignature* and *ADCApproverSignature* display items in the **IADCDesignChange** sections. Highlight the display items and then select the **Edit Relationship...** button.

- Verify the Section name - *IADCDesignChange_C, Q, UI*
- Verify the Display item name - *ADCDetail*
- Sequence on form * - **186**

Click **OK** on the *Edit Relationship* dialog.

- Verify the Section name - *IADCDesignChange_C, Q, UI*
- Verify the Display item name - *ADCMeas*
- Sequence on form * - **194**

Click **OK** on the *Edit Relationship* dialog.

- Verify the Section name - *IADCDesignChange_C, Q, UI*
- Verify the Display item name - *ADCImpact*
- Sequence on form * - **190**

Click **OK** on the *Edit Relationship* dialog.

- Verify the Section name - *IADCDesignChange_C, Q, UI*
- Verify the Display item name - *ADCPProposalSignature*
- Sequence on form * - **178**

Click **OK** on the *Edit Relationship* dialog.

- Verify the Section name - *IADCDesignChange_C, Q, UI*
- Verify the Display item name - *ADCApprovalSignature*
- Sequence on form * - **182**

Click **OK** on the *Edit Relationship* dialog.

17. Edit the section order on the data form. Have the **IGenDesignedStatus** section appear after the *IADCDesignChange* section in the form. Highlight the section and then select the **Edit Relationship...** button.

- Verify the form name - *ADCDesignChange*
- Verify the Section name - *IGenDesignedStatus_C*
- Sequence on form * - **20**

Click **OK** on the *Edit Relationship* dialog.

- Verify the form name - *ADCDesignChange*
- Verify the Section name - *IGenDesignedStatus_Q*
- Sequence on form * - **20**

Click **OK** on the *Edit Relationship* dialog.

- Verify the form name - *ADCDesignChange*
- Verify the Section name - *IGenDesignedStatus_UI*
- Sequence on form * - **20**

Click **OK** on the *Edit Relationship* dialog.

18. Edit the section order on the data form. Have the **Optional_ADCDesignChange_CID** section appear as the last section in the form. Highlight the section and then select the **Edit Relationship...** button.

- Verify the form name - *ADCDesignChange*
- Verify the Section name - *Optional_ADCDesignChange_CID*
- Sequence on form * - *300*

Click **OK** on the *Edit Relationship* dialog.

19. Edit the section order on the data form. Remove the **SPFDocumentPrinting** section from the form. Highlight the section and then select the **Delete from Form** button (or Delete Relationship).

Click **OK** on the confirmation dialog.

Remove the **IWBSSItem** sections from the form. Highlight each section and then select the **Delete from Form** button (or Delete Relationship).

Click **OK** on the confirmation dialog.

(continue on the next page)

20. Verify that your modified form is like the example shown here.

The screenshot shows a Windows application window titled "Form ADCDesignChange". The interface is organized into several sections:

- Sets ClassDef for forms**: Contains a "Class definition:" field.
- Details**: Contains "Name:" and "Description:" fields.
- Document details**: Contains "Document category:", "Document type:", "Document sub-type:", and "Document title:" fields.
- Primary Interface for ADCDesignChange**: Contains "Signature Proposer:", "Signature Approver:", "Change Detail:", "Change Impact:", and "Change UOM:" fields.
- Designed Item Status**: Contains a "Status (SPF):" field.
- Owner**: Contains an "Owning group:" field.
- Configuration scope**: Contains an "Optional interfaces" section with four items:
 - Document versions :
 - Drawing representations :
 - Document items :
 - Referenced documents :

At the bottom right are "Print" and "Close" buttons.

21. Once you have modified your form, you may take a short break until the other students have finished this activity. Do not attempt the ENS example yet. You will do that in a later hands on activity.

4

CHAPTER

SmartPlant Foundation Loader and SmartPlant Loader

4. Understanding the Loader Interface

The SmartPlant Foundation Loader software allows you to use delimited text files to import data, relationships, and configuration information into the SmartPlant Foundation database.

The SmartPlant Foundation Loader connects with the data warehouse, administers the load, applies lifecycle information, performs error handling, and creates reload files.

The SPF Loader allows a system administrator to:

- load object data into the data database using an ASCII text file as input
- load administrative objects into the admin database using a specially formatted file containing system configuration definitions



SmartPlant Foundation Loader

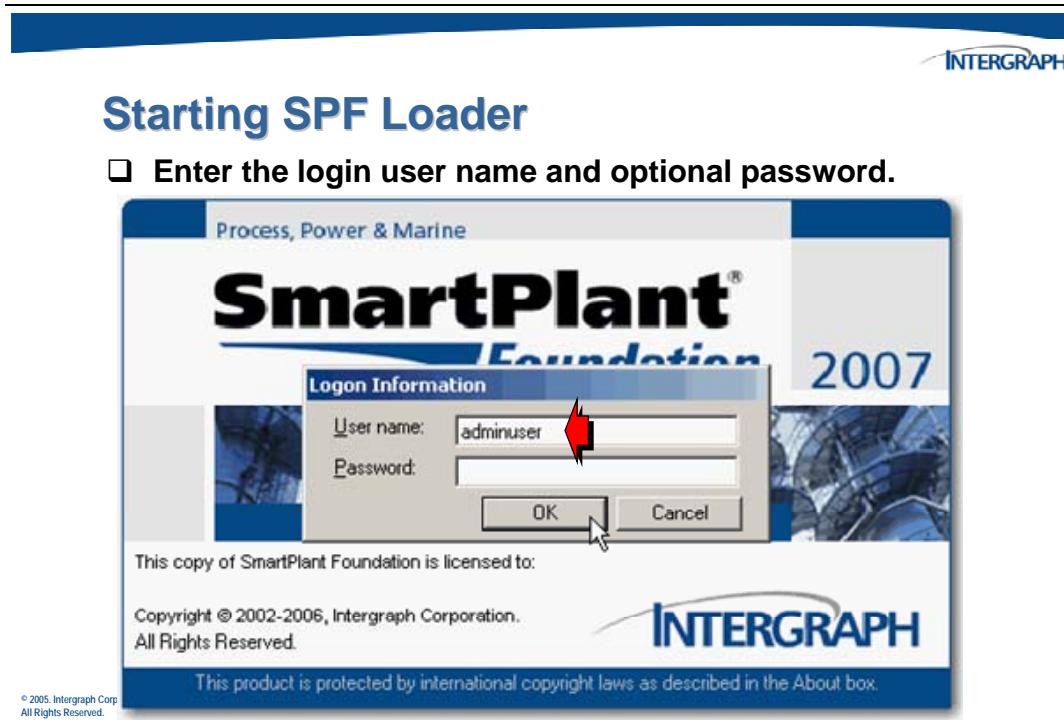
SPF load files can be created by using Notepad, WordPad or by a program that parses information from another source.

- The SPF Loader will load data from ASCII text files to the SPF databases in batch mode**
- The system administration (configuration) database can be loaded by using a load file containing *System Admin* or *Schema Object Admin* object definitions**
- The data database can be loaded by using a load file containing data object values**
- Reload and log files are generated**

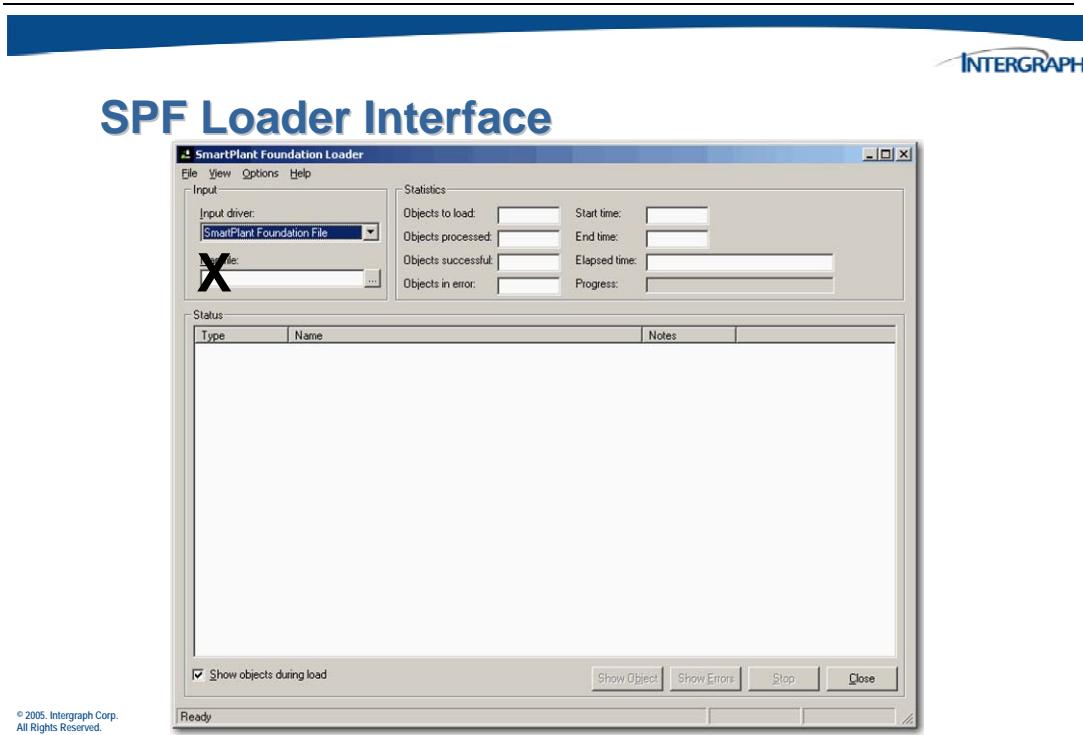
The SPF Loader is started from the desktop Program menu.



The *Logon Information* dialog will display. Use a valid SPF username to log in to the SPF Loader.



The SPF Loader application window will display.



The menu bar contains access to the available menus. The SmartPlant Foundation Loader interface contains the following areas:

The **Input Driver** field allows the user to choose the type of data to be loaded. Possible values include: SmartPlant Foundation File (XML Document is no longer supported).

The **Map File** field is no longer used.

The **Statistics** fields contain the following information:

Objects to load - the number of objects being loaded

Objects processed - the number of objects processed

Objects successful - the number of objects loaded successfully

Objects in error - the number of objects in error

Start time - the start time for the load

End time - the end time for the load

Elapsed time - the duration of the load

Progress - a graphical representation of the load as it progresses

4.1 Load File Formats

A load file uses delimited text to define object data and their relationships for load into the SmartPlant Foundation system. The syntax contained in a load file is used to specify information about each type of object. The examples shown use a *Quality Control System* (QCS) style format also known as a SmartPlant Foundation File.

SmartPlant Foundation Loader can operate in three modes: **Create**, **Update** and **Merge**. The mode can be set once per load file, and should be the first active statement in the load file.



Load File Mode

- Create** mode is used only to create new objects. If any data collisions are detected and an error occurs, you are informed that the object is not loaded. The following syntax is used to set the Create mode for a load file:

Method|Create

- Update** mode is used to update existing objects. If any objects to be updated are not found and an error occurs, you are informed that the object is not loaded. The following syntax is used to set the Update mode for a loadfile:

Method|Update



Load File Mode

Load File Modes (con't):

- Merge** mode updates an existing object or creates the object if it does not already exist. Use the following syntax to set the Merge mode for a load file:

Method|Merge

NOTE: Merge mode is the default mode for SmartPlant Foundation Loader. If you do not specify a **Method|** directive, the load file will be imported in Merge mode.

© 2005, Intergraph Corp.
All Rights Reserved.

Each line of a QCS file contains information about the objects to be loaded into SmartPlant Foundation.



Load File Format

- File uses **QCS** format (Quality Control System)
- A **dot prefix** (.) defines a new object
- Two dots** (..) indicate the information belongs to (or is associated with) the key object
- A **pipe** (|) separates the data fields

© 2005, Intergraph Corp.
All Rights Reserved.

Sample Format

.Object_Type
..Keyword|Value
..Keyword|Value|UOM|Type

4.1.1 Object Types

The following sections describe the valid object types and their associated object definition keywords.

In order to load object data into SmartPlant Foundation, the Object_Type keyword is used when defining the object internal name such as a Tag or Drawing.



Object Type Format

FORMAT --- .Object Type Keyword (Class Def)

DESCRIPTION --- Defines the type of object. Valid types are Class Data and Relationship Data.

© 2005, Intergraph Corp.
All Rights Reserved.

Object Data Keywords

Format	Description	Example
.Object_Type	Defines the internal name of the object.	.Tag
..Name/value	Defines the name of the object instance in the database.	..Name A-100
..Description/value	Defines the description of the object instance in the database.	..Description Analyzer
..Classification/value	Defines the classification of the object instance. This is necessary for classified objects. The value is the class name defined in the class library.	..Classification AA

<i>..Attribute_Name/value/UOM/Type</i>	Defines the value of the given attribute name. If the attribute is of type Quantity, the third field includes the name of UOM. If the attribute is of any other type than Text or Quantity, the fourth field should define the Type. The valid values are Count or Boolean.	Color Grey
<i>..Object Definition/value</i>	Defines the internal name of the related object such as Area, Discipline, or System. The value is the name of related object instance.	Area 6



Object Definition Format

FORMAT --- ..Object Definition Keyword|Value

DESCRIPTION --- Defines the keyword associated with the object and the value assigned to it. Valid keywords are *Name*, *Description*, *Classification*, etc. Some keywords do not apply to certain Object Types.

© 2005, Intergraph Corp.
All Rights Reserved.

Object Definition Keywords

Format	Description
Name	Stores the designated value as a name to the main object
Description	Stores the designated value as a description to the main object
LCState	Stores the designated value as Life Cycle Qualifier to the main object.
Classification	Relates the object with a classification. The class library business object that is associated with the object is the one named in the RDL field referenced on the business object definition within system administration.
TerminationDate	If set, then the object or relationship that it is referencing is terminated. The format of this date is either <i>now</i> which uses the current date time to terminate the object or yyyy/mm/dd-HH:MM:SS:xxx where: yyyy is year mm is month dd is day HH is hours MM is minutes SS is seconds xxx is milliseconds
	Example date format 2000/12/31-23:59:59:000

The following is an example of a load file.



SPF Load File Example

```
# =====Create Document Master =====
.SetConfiguration
..Configuration|EFPLANT-SC-2,PRJ-2001

.DocMaster
..ClassDefUID|ADCDesignChange
..Name|Doc1
..Description|Example of loading a document master
..RevSchemeName|RevA1
..MajorRevision|A
..MinorRevision|1
..OwningGroup|UPDATE
```

© 2005, Intergraph Corp.
All Rights Reserved.



SPF Load File Example

```
..GenDesignedItemStatus|CR
..DocCategory|P&ID Documents
..DocSubtype|Utility P&ID
..DocTitle|Load Object 1
..DocType|P&ID
```

Note: Picklist properties must match the values from the corresponding enumerated list.

© 2005, Intergraph Corp.
All Rights Reserved.

4.1.2 Field Types

For a plant independent object, the minimum parameters required are **Name** and **Description**. For a plant dependent object such as Tag, the minimum parameters required are Name, Description, Classification, and Plant.

If the first field of the data is neither a keyword (Name, Description, or Classification) nor is it found in the list of related objects (System, Area, Discipline), then the loader assumes the field is a property.

If a third and fourth value field are detected for an object, then the Loader will create the data accordingly. The third field indicates a UOM value and will treat the data as a UOM definition. The fourth field can be defined as follows:

- Text - text data
- Quantity - used in conjunction with a unit of measure and a value
- Count - used to store integer data
- Boolean - used as a condition (true/false) by setting a + or -
- Date/Time - used to store date/time information in the format yyyy:mm:dd-HH:MM:SS:xxx
where yyyy is the year, mm is the month, dd is the day, HH is the hour, MM is the minutes, SS is the seconds, and xxx is the milliseconds.

Notes: Properties must not have the same name as an object. The loader will fail to differentiate the two and will assume a relationship is being created.

Only one reference to a related object can be made in any one data set; otherwise, the second reference is terminated. If multiple relationships need to be maintained, then the relationships must be separately defined as objects in their own right.

4.1.3 Relationship Data

In order to load relationship data into SmartPlant Foundation, the internal relationship name is used as the **Relation_Type** when defining the object type. Relationship Data uses specific keywords to define each object.



SPF Load File Example

Relationship example

```
# New plant object for PBS structure      # PlantlistPlant Relationship
# -----
.EFObject                               # -----
..NAME|Madison                          .PlantListPlant
..DESCRIPTION|New Plant for PBS Structure ..LNAME|AllPlants
..CLASSDEFUID|Plant                      ..RNAME|Madison
                                         ..LOBJECT|PlantList
                                         ..ROBJECT|Plant
```

© 2005. Intergraph Corp.
All Rights Reserved.

Format	Description	Example
<i>.Relation_Type</i>	Defines the type of internal relationship name. Note: In this example, the Left object is the Tag and the Right object is the Area.	.TagArea
<i>.LName/value</i>	Defines the name of the Left (first) object.	..LName 6-00K80-102
<i>.RName/value</i>	Defines the name of the Right (second) object.	..RName AN-00019

Relationship Definition Keywords

Format	Description
LName	Defines the main object name of the Left (or first) object.
RName	Defines the main object name of the Right (or second) object.

TerminationDate	If set, then the object or relationship that it is referencing is terminated. The format of this date is either <i>now</i> which uses the current date time to terminate the object or yyyy/mm/dd-HH:MM:SS:xxx where: yyyy is year mm is month dd is day HH is hours MM is minutes SS is seconds xxx is milliseconds Example date format 2000/12/31-23:59:59:000
LKey1 - LKey3	Used to define the unique key information in order to uniquely identify the left object. If set, then the values inserted are substituted into the left object in order to make it unique. For example, if the Left object is an equipment item and the first two key fields are set (within system administration) to Manufacturer ID and Name then the LKey1 should hold the Manufacturer ID value.
RKey1 - RKey3	Used to define the unique key information in order to uniquely identify the right object.
Plant	If set, then both the LKey1 and RKey1 value is set to the Plant name set.

Note: The alternative method to load relationship data is to reference the relationship when loading an object. In this case only, one relationship type can exist to the related object.

4.1.4 Attach a File

To attach a file to an object (such as a document) within SmartPlant Foundation and transfer it to the SmartPlant Foundation vault, the object type **AttachFile** is used. The object definition keywords to the **AttachFile** object type are as follows:

- ❑ DocumentClass
- ❑ DocumentName
- ❑ FileName
- ❑ FileDescription
- ❑ DestRelativePath



SPF Load File Example

Create Document / Attach File example

```
.SetConfiguration  
..Configuration|EFPLANT-SC-2  
  
.DocMaster  
..Name|DemoPID1  
..ClassDefUID|Document  
..Description|DemoPID1  
..OwningGroup|UPDATE  
..IssueState|RESERVED  
..DocCategory|Non TEF Documents  
  
DocType|Sample  
  
.DocTitle|HotSpot Lab  
..RevName|DemoPID1  
..RevDescription|PID1 document  
#  
# Revision details:  
# -----  
..RevSchemeName|RevA1  
..MajorRevision|A  
..MinorRevision|1
```



SPF Load File Example

Create Document / Attach File example (con't):

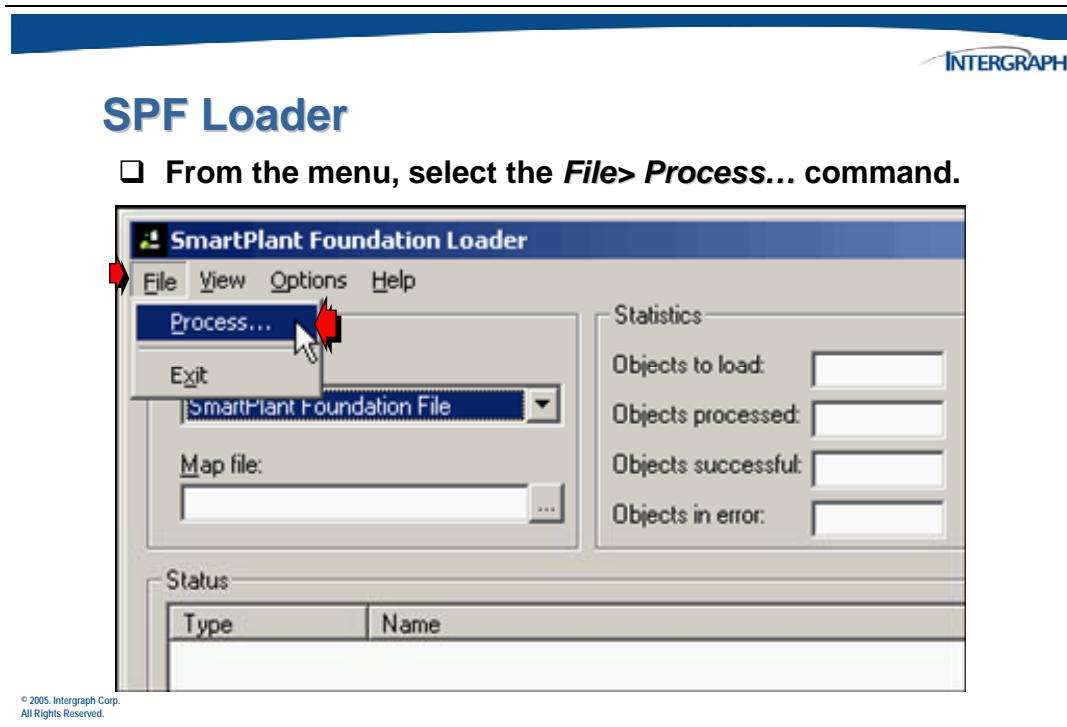
.AttachFile	..FileDescription DGN file for hotspotting
..ClassDefUID DocumentVersion	..FileType dgn
..DocumentClass DocRevision	..EditInd 1
..DocumentName DemoPID1	..ViewInd 1
..DocRevision A1	
..MajorRevision A	
..MinorRevision 1	
..DocVersion 1	
..FileClass DesignFile	
..FileName d:\temp\pid1.dgn	

4.2 Processing Load Files

In order to load files into SmartPlant Foundation, you must first select your **Input Driver**. The Input Drivers consists of an SmartPlant Foundation File. Once you have selected this file, you can then select various items to process your file. These items are available from the **Options** menu. You may also choose to view your configuration file, error file, or reload file. These options are available from the **View** menu.

The following section will provide the information necessary for loading objects into SmartPlant Foundation.

Once the Loader utility has been started, select the load file to process.



The *Process* dialog window will display.

SPF Loader

- ❑ In the *Process* dialog, choose the ellipses (...) button in order to select a load file.

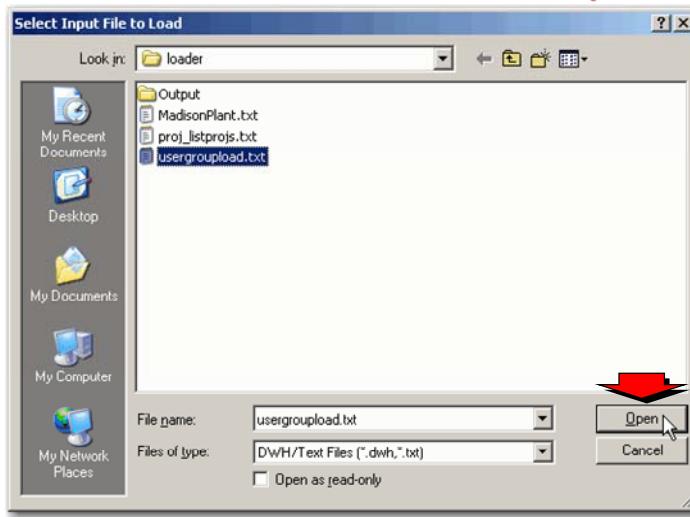


© 2005, Intergraph Corp.
All Rights Reserved.

Open the file that is to be processed by the loader. The *Select Input file to Load* dialog will display where you can select the file that contains the objects to be loaded.

SPF Loader

- ❑ Choose the file to be loaded and select **Open**.

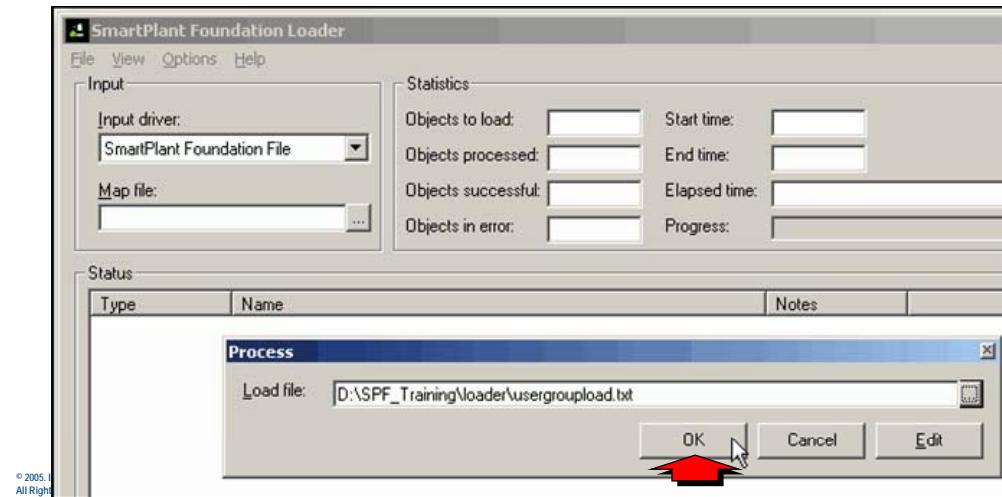


Load files will have an extension of **.dwh** (data warehouse) or **.txt**.



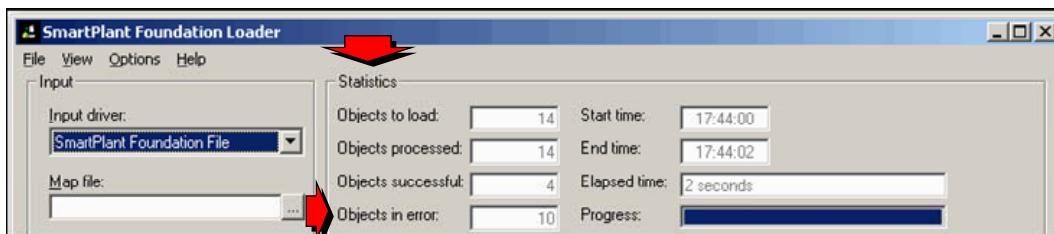
SPF Loader

- Click **OK** to begin loading the selected file into the SPF admin database.

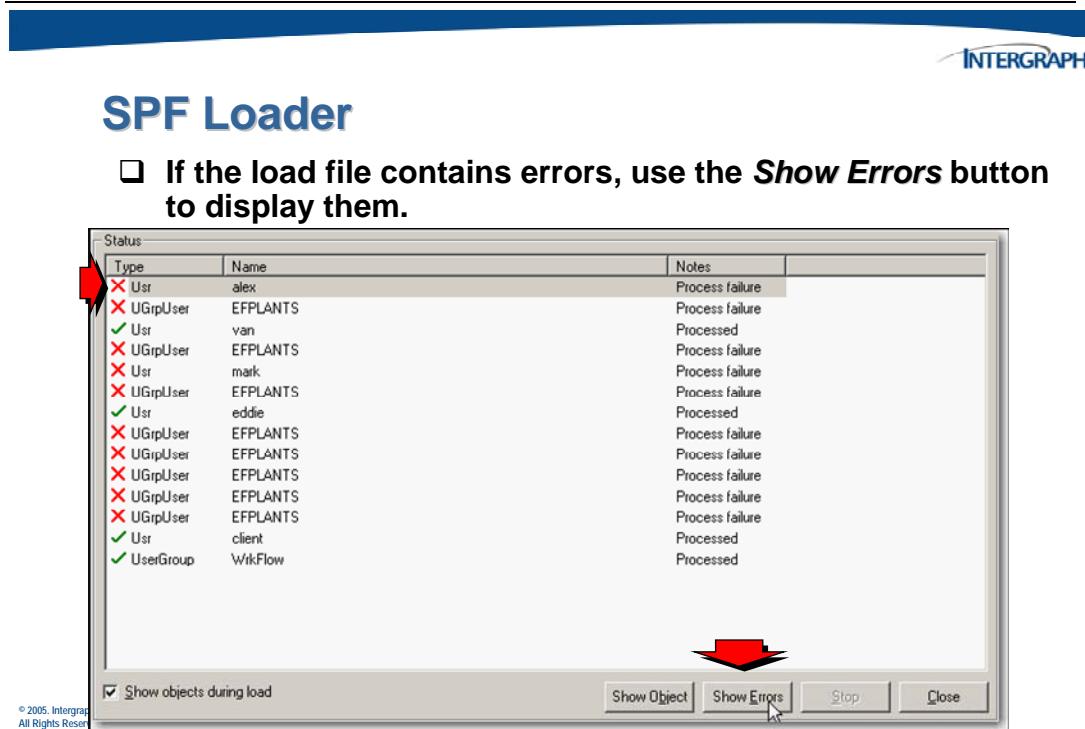


SPF Loader

The **Statistics** area will display the number of *Objects in error*.



The **Status Window** in the lower half of the *Loader* window will show the results of the load operation.



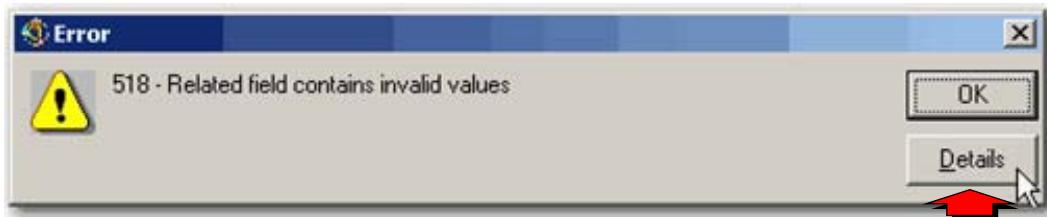
The **Status Window** contains the following information:

- the load as it is processed (error messages will be displayed here)
- a check box to **Show Objects during the load**, which indicates that load processing messages will be displayed in the window
- a command button to **Show Object**
- a command button to **Show Errors**
- a command button to **Stop** the load
- a command button to **Close** the window



SPF Loader

- Click the **Details** button to display the error details of why the load command failed.



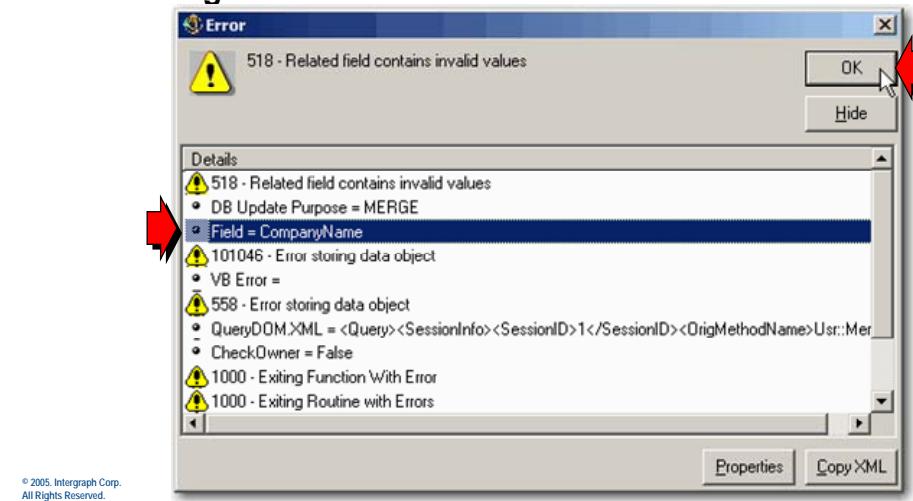
© 2005, Intergraph Corp.
All Rights Reserved.

The *Error* dialog will expand to show the error **Details**.



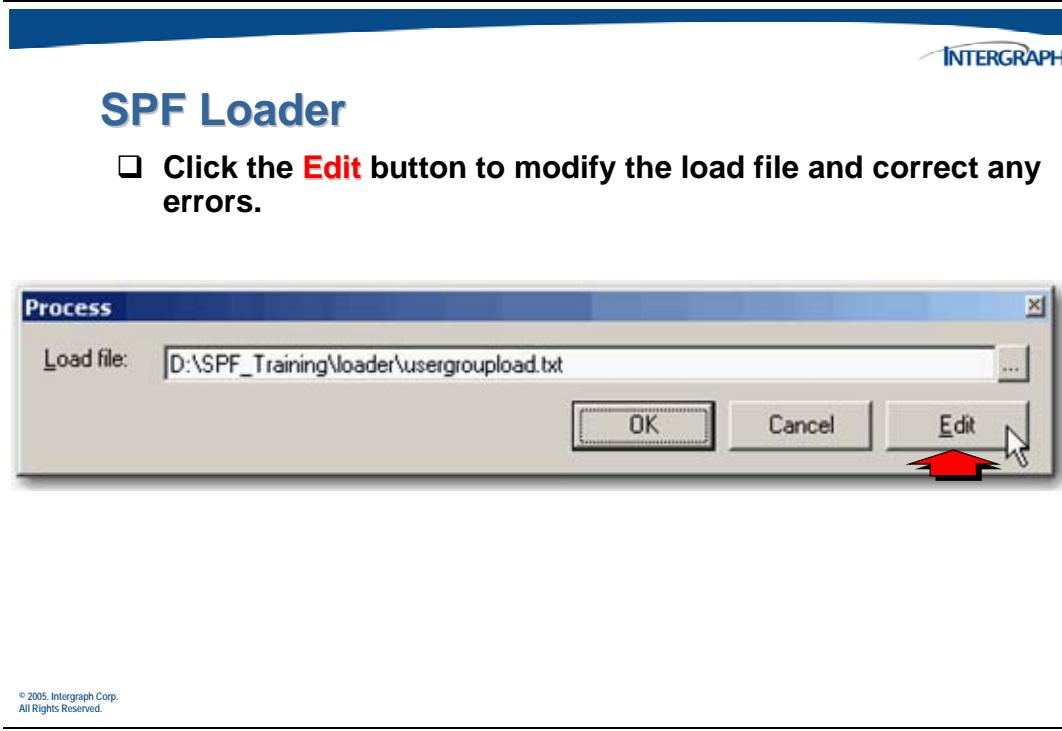
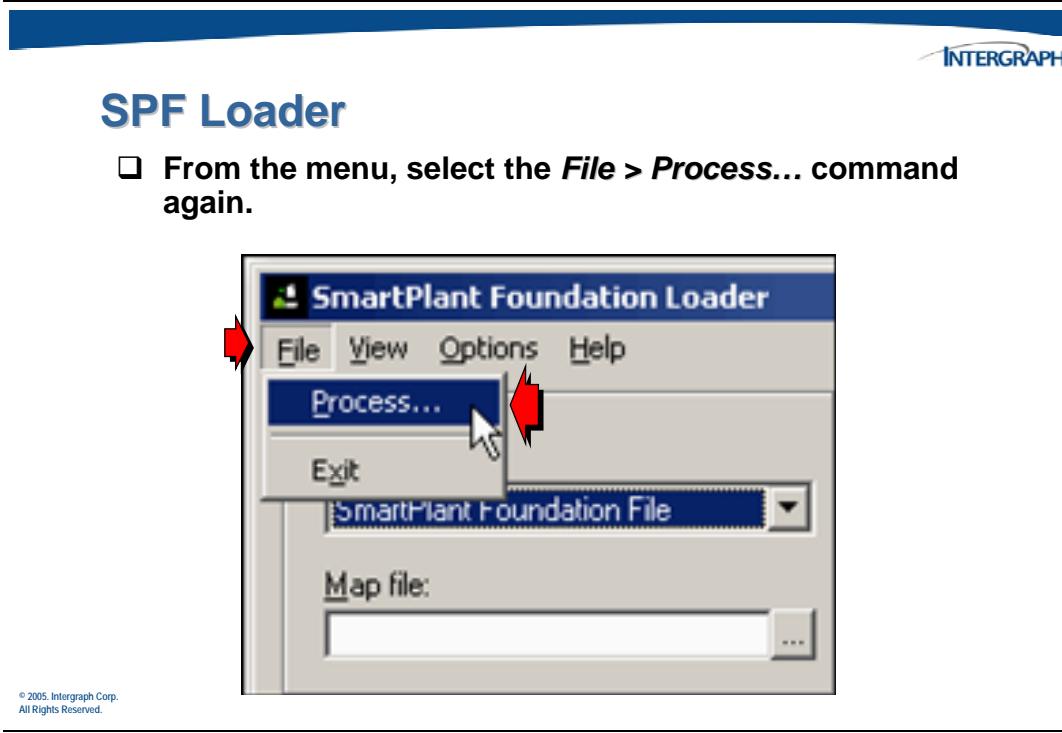
SPF Loader

- Note the name of the field and the incorrect value that might have caused the load command to fail.



© 2005, Intergraph Corp.
All Rights Reserved.

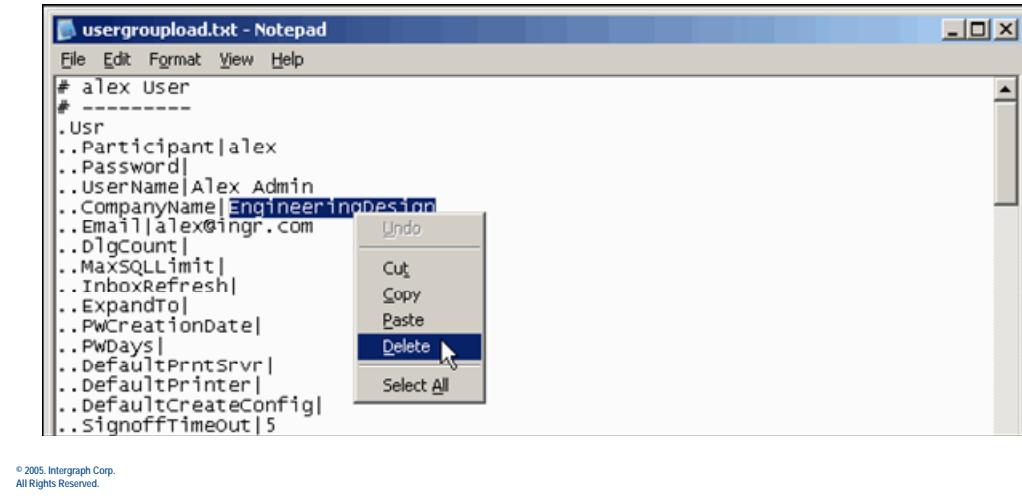
If there are errors reported by the loader, the load file can be opened and edited from the *Process* dialog.





SPF Loader

- ❑ Make any necessary changes in the load file, exit notepad and save the modifications.



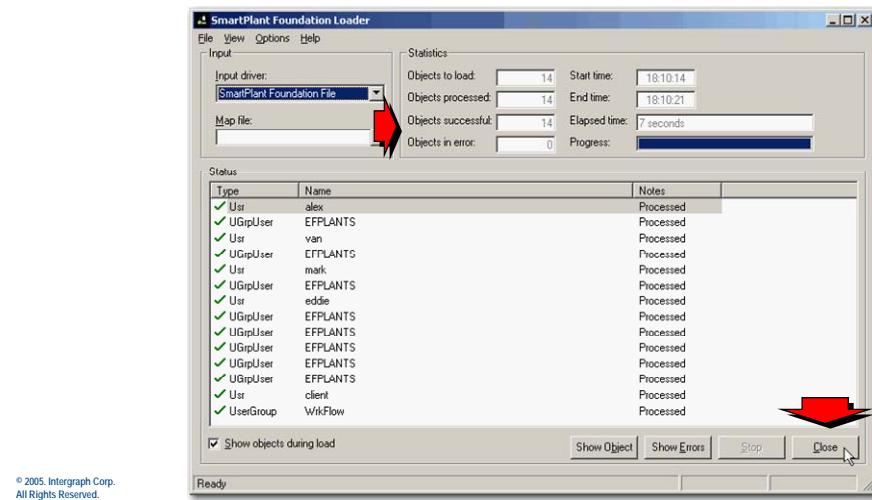
© 2005, Intergraph Corp.
All Rights Reserved.

After the errors have been corrected, use the *Process* dialog to re-run the load operation.



SPF Loader

- ❑ Once the objects have been loaded successfully, **Close** the SPF Loader.



© 2005, Intergraph Corp.
All Rights Reserved.

4.3 Introduction to SmartPlant Loader

The SmartPlant Loader software, with the aid of the SmartPlant Loader service, loads all published documents and their associated data into the SmartPlant Foundation database. The SmartPlant Loader continuously monitors SmartPlant activity for files to load. When a user publishes a document, the software registers the document and its view file with SmartPlant Foundation. However, if the user assigns the document to a workflow, the data is loaded into SmartPlant Foundation only after the document reaches a load step in the workflow. The load step typically occurs after a user approves the document. If the document is not associated with a workflow, the document immediately reaches the SmartPlant Loader queue for loading. Files are loaded in the order they reach the SmartPlant Loader queue.



SmartPlant Loader

The **SmartPlant Loader** function is to load the data content (equipments, streams etc. described by a published PFD etc.) published by a publish request in the Engineering SmartPlant.

The **SmartPlant Loader** interface displays a list of SmartPlant Foundation administration and data database connections (SmartPlant Databases) for which polling and loading are to be done. You can modify the information displayed and save it to the SmartPlant Loader database.

Once the information is saved, SmartPlant Loader signals the SmartPlant Loader service to update its SmartPlant Database objects.

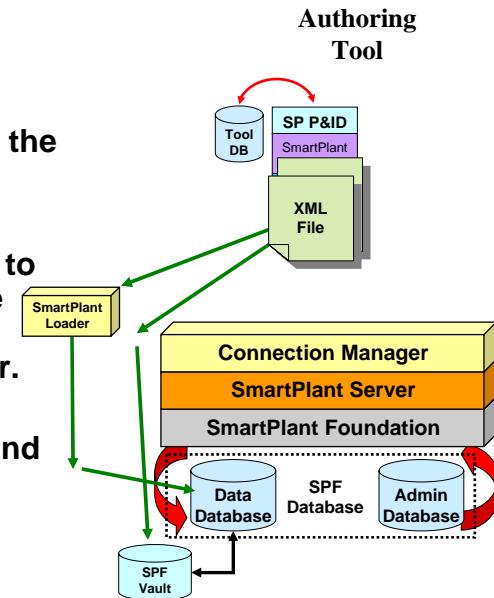
The SmartPlant Loader is installed and uninstalled by the SmartPlant Loader installation. It runs in the background automatically. When the SmartPlant Loader service starts, it reads the information it needs about SmartPlant databases from a SmartPlant Loader database. If the SmartPlant Loader service does not start after installation, you must start it manually.



SmartPlant Loader

The SmartPlant Loader component contains the SmartPlant Loader service and the SmartPlant Loader manager.

- Continuously monitors SmartPlant activity for files to load based on the database connections defined in the *SmartPlant Loader* manager.
- Data published by the authoring tools is queued and loaded into the SmartPlant Foundation database.



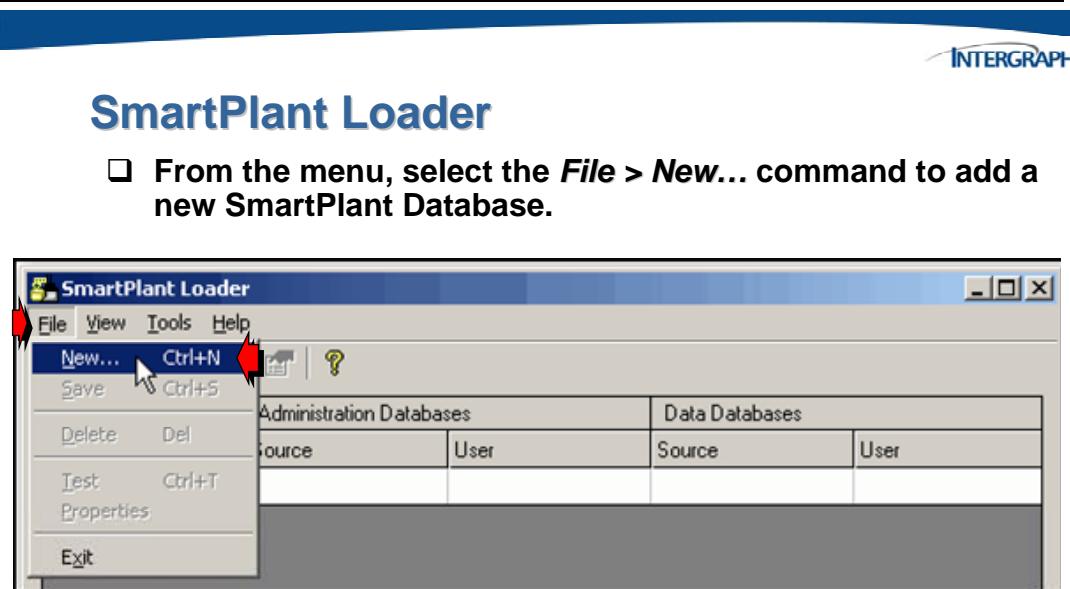
4.3.1 Starting the SmartPlant Loader

The SmartPlant Loader user interface displays a list of administration and data SmartPlant Foundation database connections (SmartPlant Databases) for which polling and loading are to be done.

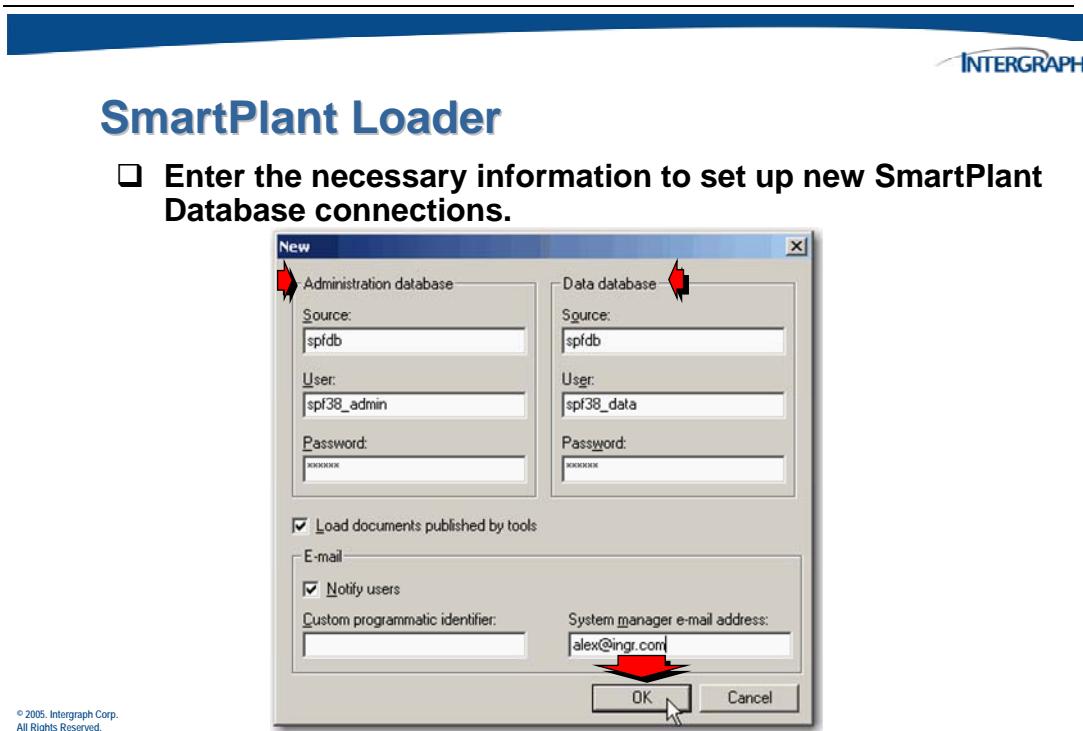


4.3.2 Creating SmartPlant Database Connections

To load files into the SmartPlant Database, you must first set up your database connections. You can also specify the schema file to be used by the SmartPlant Database and the type of e-mail notification you want to use.



The *New* dialog will display.



In the *New* dialog box, enter the **Source**, **User** name, and **Password** information for the *administration* and *data* databases.

Select the **Notify users** check box if you want e-mail notification to be sent to the subscriber list for a document when a document is loaded.

Notes: In the **Custom programmatic identifier** box, type the programmatic identifier that has been created to handle e-mail notifications.

The **Notify users** check box must be selected to use this option.

If no custom target is specified, then a default e-mail is sent to the user e-mail addresses found in the SmartPlant Foundation Data database.

In the **System manager e-mail address** box, type the e-mail address(es) of the system manager to whom problems are reported.

Note: You can list multiple system manager e-mail addresses by using semicolons to separate each e-mail address.

A *custom programmatic identifier* (ProgID) is an application's unique name that is mapped to the system registry by a class identifier (CLSID). In this case, the custom ProgID specifies a class that supports a method that the SmartPlant Loader Service calls to send e-mail notifications. For example, if the user created a DLL named MyEmailer with a class named MyMailHandler, the ProgID would be MyEmailer.MyMailHandler. MyMailHandler would implement the following method:

```
Function EmailRequest( _  
    sConnectionString As String, _  
    sAdministrationConnectionString As String, _  
    bFailNotification As Boolean, _  
    sDocumentVersionOBID As String, _  
    sRecipients As String, _  
    sEmailSubject As String, _  
    sEmailBody As String, _  
    sPlantOBID As String, _  
    sProjectOBID As String _  
) As Boolean
```

The implementation of the method should return as quickly as possible, so that the SmartPlant Loader Service does not have to wait before loading the next request in the queue. The method returns a "False" if it cannot fulfill the request or a "True" if it can. Both results are reported in the log file. If a "False" is returned, the SmartPlant Loader Service sends an e-mail notification as if the custom ProgID were not set.

If *bFailNotification* is "True", the SmartPlant Loader Service has encountered a problem and an e-mail notification is sent to the specified system manager e-mail address(es). If *bFailNotification* is "False", a normal e-mail notification is sent to all recipients.

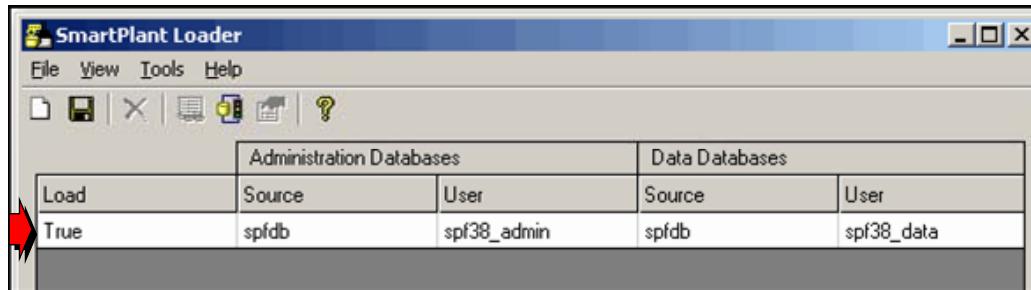
The *sRecipients* string may contain one or more e-mail addresses with each separated by semicolons.

The method may use the *sEmailSubject* and *sEmailBody* strings as formatted and supplied by the SmartPlant Loader Service, perhaps to buffer requests. The method may also use the *sDocumentVersionOBID* string and the database connection strings to do custom formatting.



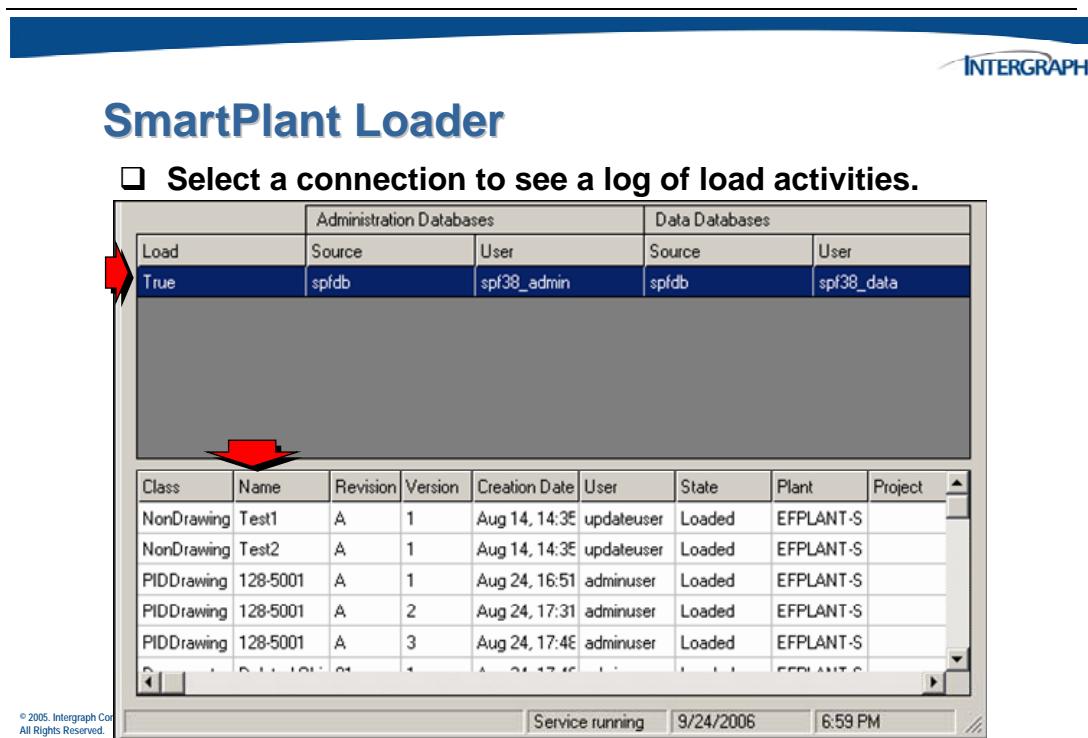
SmartPlant Loader

The new connections are displayed in the SmartPlant Loader window.



4.3.3 The SmartPlant Loader Queue

The **Loader Queue** table displays the document queue for the selected SmartPlant Database.



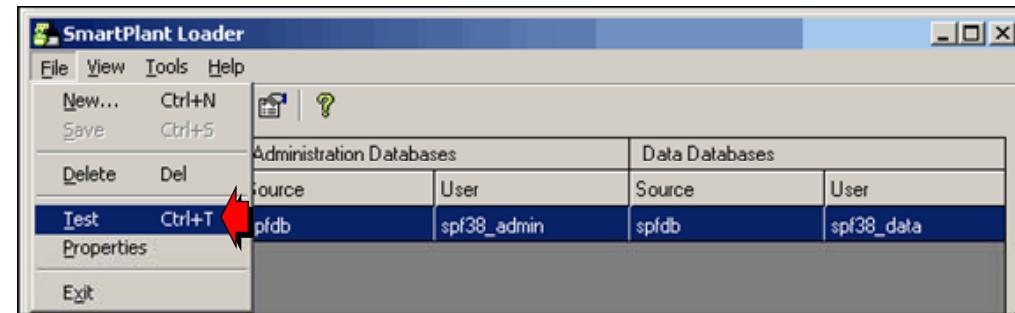
4.3.4 Testing Database Connections and Options

Once you have selected your database connections, you can choose from several options to monitor the processing of the load files. You can test your database connections that you set up for the SmartPlant Loader.

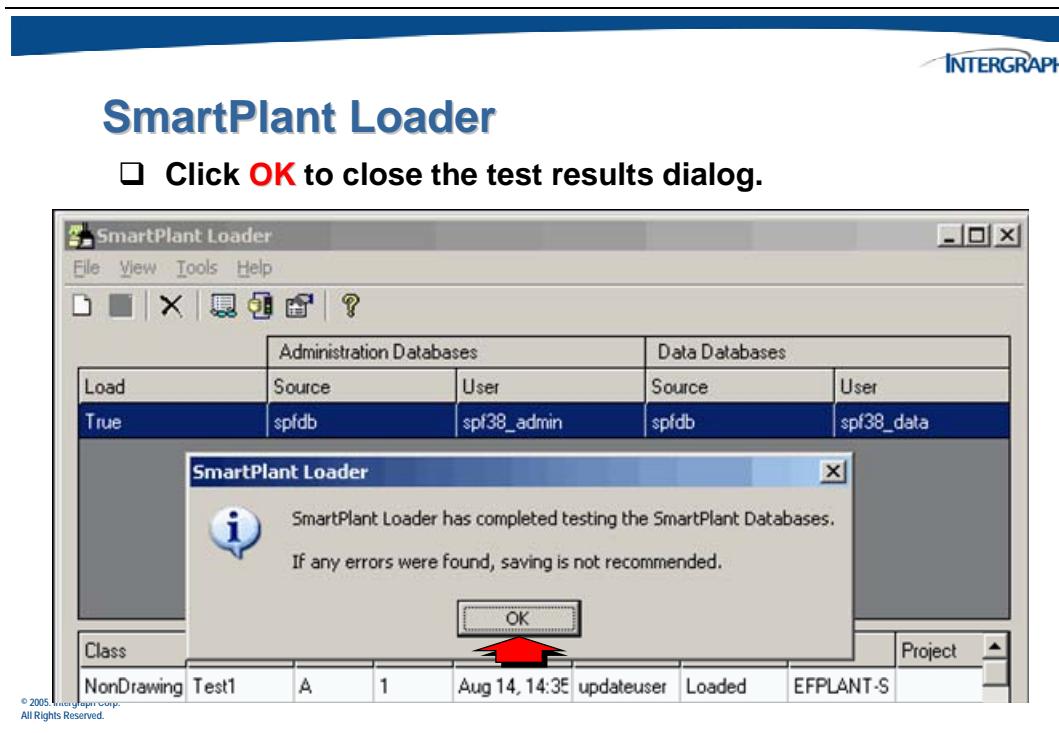


SmartPlant Loader

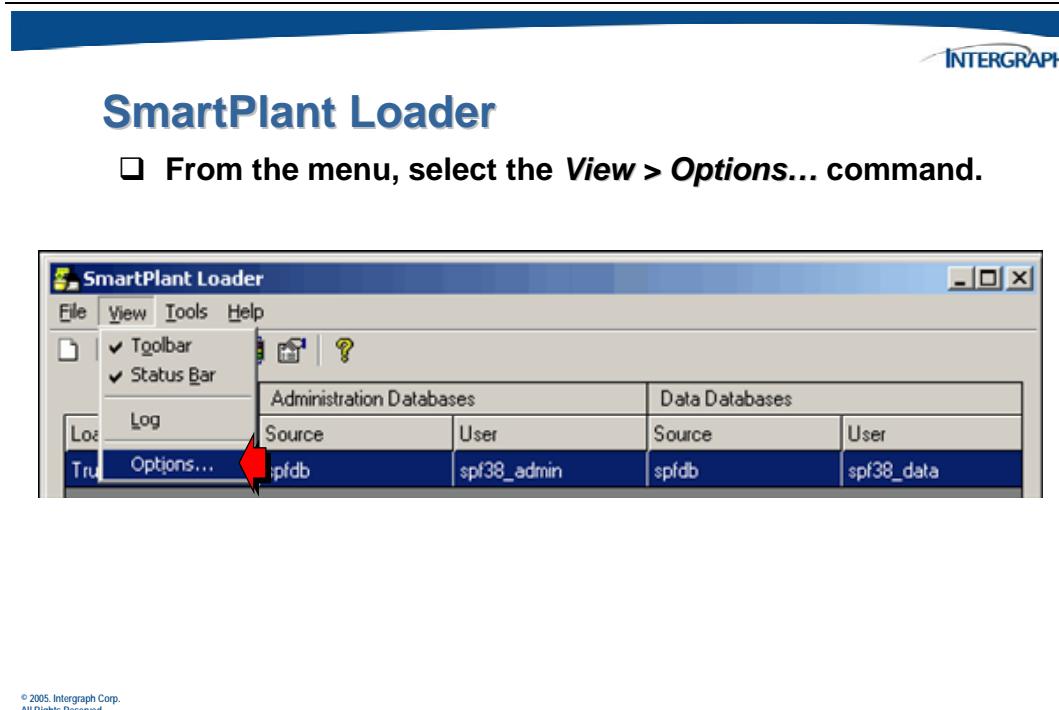
- Select the **File > Test...** command from the menu, to test schema object creation and database connections.



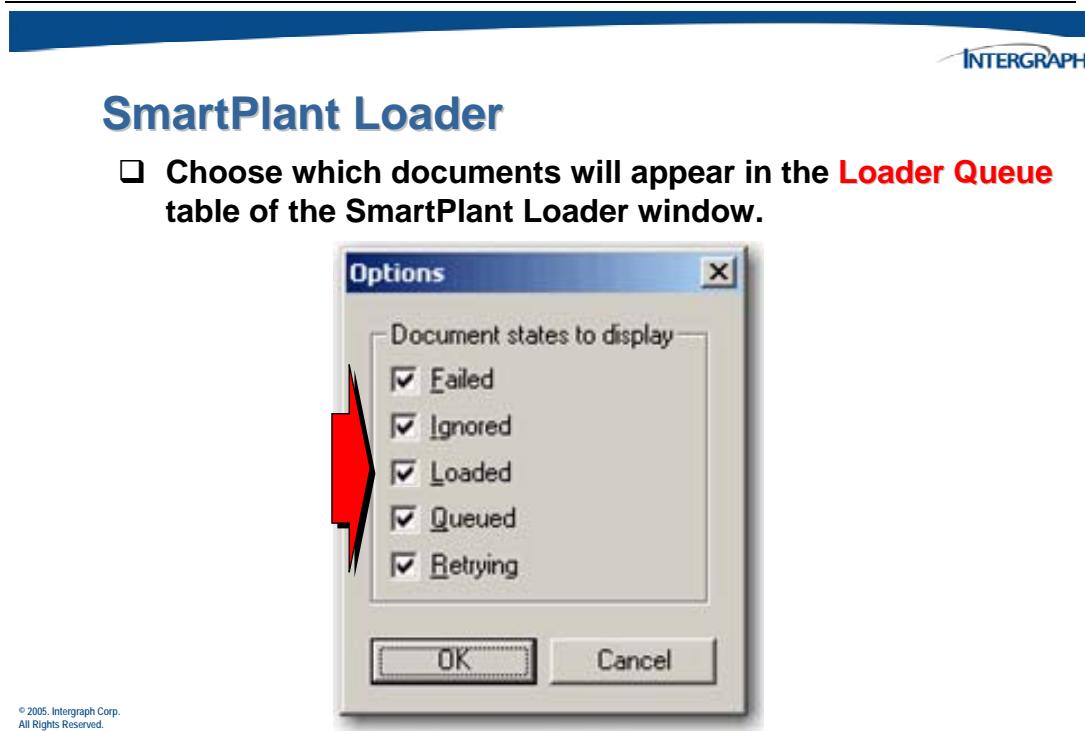
This command allows you to test schema object creation and database connections for non-idle SmartPlant Databases.



The **Options** command launches the *Options* dialog box, which allows you to select the documents to display in the Loader Queue table



The options dialog allows you to choose which documents appear in the Loader Queue table of the SmartPlant Loader window.



Document states to display:

- Failed** - Indicates that documents with a state of "failed" appear in the Loader Queue table.
- Ignored** - Indicates that documents with a state of "ignored" appear in the Loader Queue table.
- Loaded** - Indicates that documents with a state of "loaded" appear in the Loader Queue table.
- Queued** - Indicates that documents with a state of "queued" appear in the Loader Queue table.
- Retrying** - Indicates that documents with a state of "retrying" appear in the Loader Queue table.

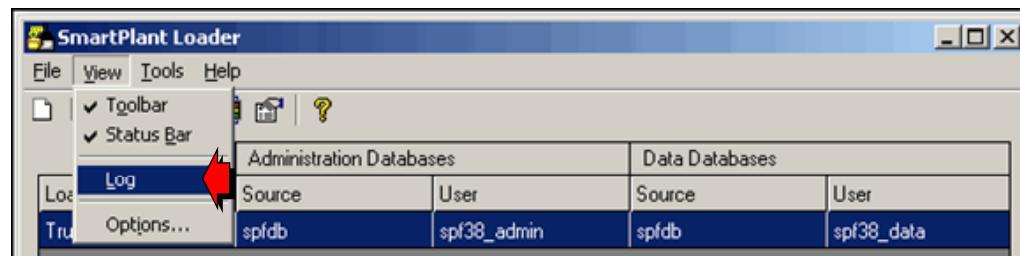
4.3.5 SmartPlant Loader Log

The **Log** command allows you to display the log file containing information about the selected SmartPlant Foundation connection. If SmartPlant Loader is producing a debug log file, and there is no SmartPlant Database selected, then the Log command will display the SmartPlant Loader's debug log file.



SmartPlant Loader

- From the menu, select the *File > Log...* command.





SmartPlant Loader

A loader results log file is displayed in a Notepad window.

```
SPFDB.SPF38_ADMIN.SPFDB.SPF38_DATA.Log - Notepad
File Edit Format View Help
08-14-2006 14:35:47:Loading: Test1 rev.A v.1.
08-14-2006 14:35:47:Object quantity: Metadata 4. Data 11. Tombstones 0.
08-14-2006 14:35:49:Loaded: Test1 rev.A v.1.
08-14-2006 14:35:50:Loading: Test2 rev.A v.1.
08-14-2006 14:35:50:Object quantity: Metadata 4. Data 11. Tombstones 0.
08-14-2006 14:35:51:Loaded: Test2 rev.A v.1.
08-24-2006 16:51:42:Loading: 128-5001 rev.A v.1.
08-24-2006 16:51:46:Object quantity: Metadata 6. Data 41. Tombstones 0.
08-24-2006 16:51:50:Loaded: 128-5001 rev.A v.1.
08-24-2006 17:31:16:Loading: 128-5001 rev.A v.2.
08-24-2006 17:31:19:Object quantity: Metadata 6. Data 68. Tombstones 0.
08-24-2006 17:31:23:Loaded: 128-5001 rev.A v.2.
08-24-2006 17:48:31:Loading: 128-5001 rev.A v.3.
08-24-2006 17:48:35:Object quantity: Metadata 6. Data 84. Tombstones 1.
08-24-2006 17:48:41:Loaded: 128-5001 rev.A v.3.
08-24-2006 17:48:42:Loading: Deleted Objects in SMARTPLANTPID rev.01 v.1.
08-24-2006 17:48:43:Object quantity: Metadata 2. Data 1. Tombstones 1.
08-24-2006 17:48:43:Loaded: Deleted objects in SMARTPLANTPID rev.01 v.1.
08-24-2006 18:00:08:Loading: 128-5001 rev.A v.4.
08-24-2006 18:00:12:Object quantity: Metadata 6. Data 119. Tombstones 7.
08-24-2006 18:00:18:Loaded: 128-5001 rev.A v.4.
08-24-2006 18:00:29:Loading: Deleted Objects in SMARTPLANTPID rev.01 v.2.
08-24-2006 18:00:31:Object quantity: Metadata 2. Data 1. Tombstones 7.
08-24-2006 18:00:33:Loaded: Deleted objects in SMARTPLANTPID rev.01 v.2.
```

© 2005, Intergraph Corp.
All Rights Reserved.

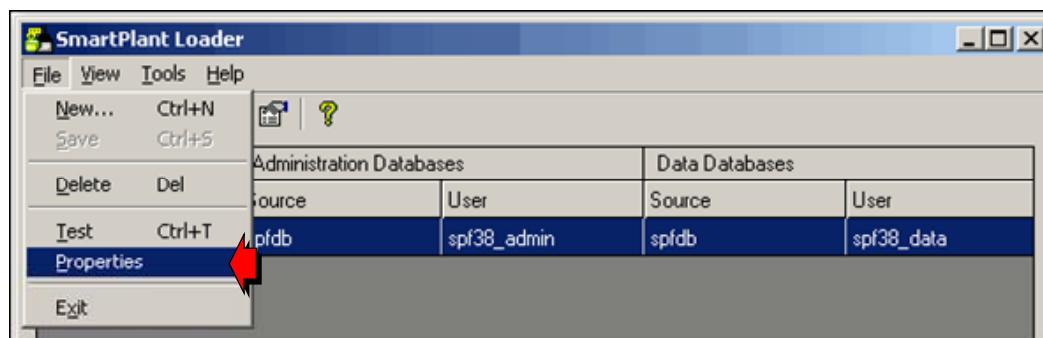
4.3.6 SmartPlant Loader Properties

The properties of any of the configured SmartPlant Databases can be displayed in the *SmartPlant Loader* window.



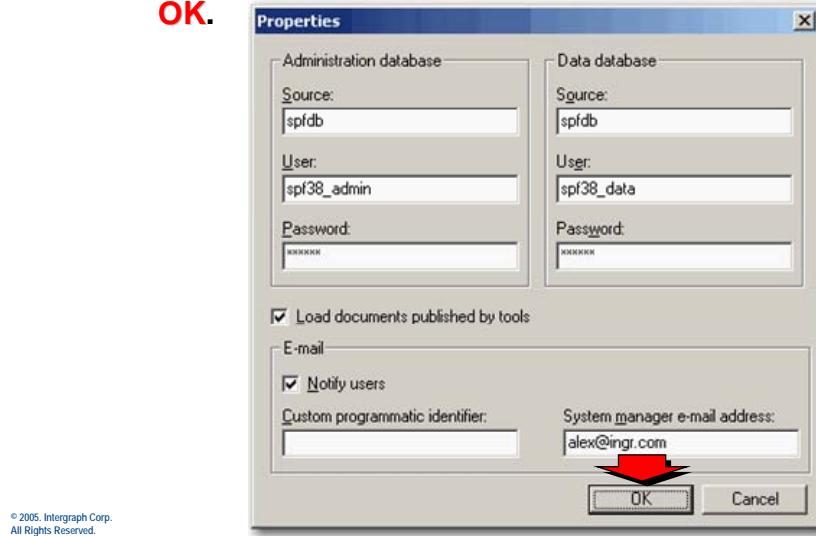
SmartPlant Loader

- Select the **File > Properties...** command from the menu, to display the properties of the currently selected SmartPlant Database connection.



SmartPlant Loader

- Use the *Properties* dialog to make any changes and click **OK**.



© 2005, Intergraph Corp.
All Rights Reserved.

INTERGRAPH

4.4 Activity 1 – Loading SPF Users and Groups

The objective of this activity is to have you load the same SPF *Users*, *User Groups* and create *User to Group* relationships that were created in the Introduction to SmartPlant Foundation and Administration I training class. This is an example of using the SPF Loader to load administrative objects.

1. Start the **SPF Loader** from the Programs start menu and log in as user *adminuser*.
2. Use the **File > Process** menu command to open the load file **d:\SPF_Training\loader\usergroupload.txt** load file.
3. Execute this load file to load users, groups and create relationships in the admin database.
4. Use the **System Administration** utility to check and verify that the objects were loaded successfully.
5. When you finish this activity, continue on to activity 2.

4.5 Activity 2 – Create Data Load File

1. Create a text file which creates three data entries for the ADCDesignChange Document object. The names and entries are your choice.
2. Use the information you have learned in the Loader and Forms chapters. Required fields in the form are required in the loader file. Also make sure you include the *Detail, Measurement, Impact, Proposer, and Approver* properties in the load file. Use the form on the next page as a guide.
3. Entries which have picklists must use the picklist entries only. You can't create your own entry for this field. Use *Schema Object Administration* to review the values for the **GenDesignedItemStatus**, **DocCategory**, **DocType** and **DocSubtype** picklists (Enumerated Sets). Add values to your load file for the **GenDesignedItemStatus**, **DocCategory**, **DocType** and **DocSubtype** properties.
4. After you create the load file, use the SPF Loader to test your file then use the *Desktop Client* to verify your loaded data.
5. When you finish this activity, you may take a short break until everyone has finished.

New Utility P&ID

Configuration
Create/update items in: EFPLANT-SC-2,PRJ-2001,

Details
Name: *
Description:

Document details
Document category: * P&ID Documents
Document type: * P&ID
Document sub-type: Utility P&ID
Document title: *

Primary Interface for ADCDesignChange
Signature Proposer:
Signature Approver:
Change Detail:
Change Impact:
Change UOM: m^3/s

Designed Item Status
Status (SPF): CR

Owner
Owning group: UPDATE

Configuration scope
IProject: * PRJ-2001

Include revision details? Collapse sections automatically?

5

CHAPTER

Creating New Methods, Menus and Toolbars

5. Introduction to Methods

In the SmartPlant Foundation system, information is retrieved by performing certain operations on Interface Definitions. The availability of these operations is controlled by assigning methods to the Interface Definitions. Users have access to these operations by selecting them from the main menus or the shortcut menu.

When a method is configured, the new method can be attached to either a component or a Client API.



Functionality Configuration

- End user functionality is configured by using methods that are based on **Client APIs**.**
- Methods can be attached to Interface Definition where they are available only when the Interface Definition instance is in focus or to menu items/toolbars when they are always available.**
- Access to functionality can be controlled by user access control, conditions and object status.**
- User groups are associated to methods by using drag and drop.**

Client APIs detail the methods allowed in SmartPlant Foundation to support the server components installed. The Client APIs reside in the components. A set of Client APIs is delivered during the software installation.



Client APIs

Client APIs are the actual functions that can be used in methods to expose system functionality for the end users

Client API can be used in several methods

- Client API name
- Client API description
- Necessary parameters

© 2005, Intergraph Corp.
All Rights Reserved.



Client APIs

These are some of the existing client APIs delivered with SPF.

Client API Name	Client API Description	Parameter 1	Parameter 2	Parameter 3
CCOEFWizard	Create EF Object Wizard	ClassDef,EnumLis...	DBObject,OptAttac...	RevDBObject
ChangeOwnerOrGroup	Change the Owner or Owning Group of ...			
ChangeOwningGroup	Function to Change the Owning Group o...			
ChangeUser	Change to another user			
ChangeUserPassword	Change User Password			
CheckIn	Function to Check in files to the vault	Web Location	Replace	Reference
CheckOut	Function to Check Out files from the vault	Show hidden files	Copy all rels (over...	
CMClientEditObj	Change Management Client Edit Obj	Method	Param1	Param2
CMClientShowObj	CM Client Show Objects	Method	Param1	Param2
CommitConfiguration	Commit Configuration			
CompareObj	Function used to compare a current revi...			
CompareObjs2	Function used to compare a current revi...			
CopyObj	Copy object and relations	(Opt EF) CustomH...	(Opt EF) RelDef t...	(Opt EF) R...
CopyToLocalFile	Used to copy a file to the local host	Show hidden files		
CopyWizard	Copying documents	MasterDBObject	RevDBObject,OptAt...	CopyFiles
CreateAdhocQuery	Create Adhoc Query		Allow group sharin...	
CreateClassObj	Used to Create an object from a Class O...	ClassName	PurPose	
CreateClassObj2	Function used in New Main Menu to dis...	BObjClassName	RDLClassName	BObjName

© 2005, Intergraph Corp.
All Rights Reserved.

Conditions in SPF are set up to restrict the applicability of actions. These objects can be attached to methods, workflows, and relationships to apply enhanced filtering. If the condition is not met on the selected object, then the user interface does not make this option available to the user.



Conditions

Conditions can be created against any property.

- Condition name**
- Condition description**
- Condition default (property=value)**

© 2005, Intergraph Corp.
All Rights Reserved.

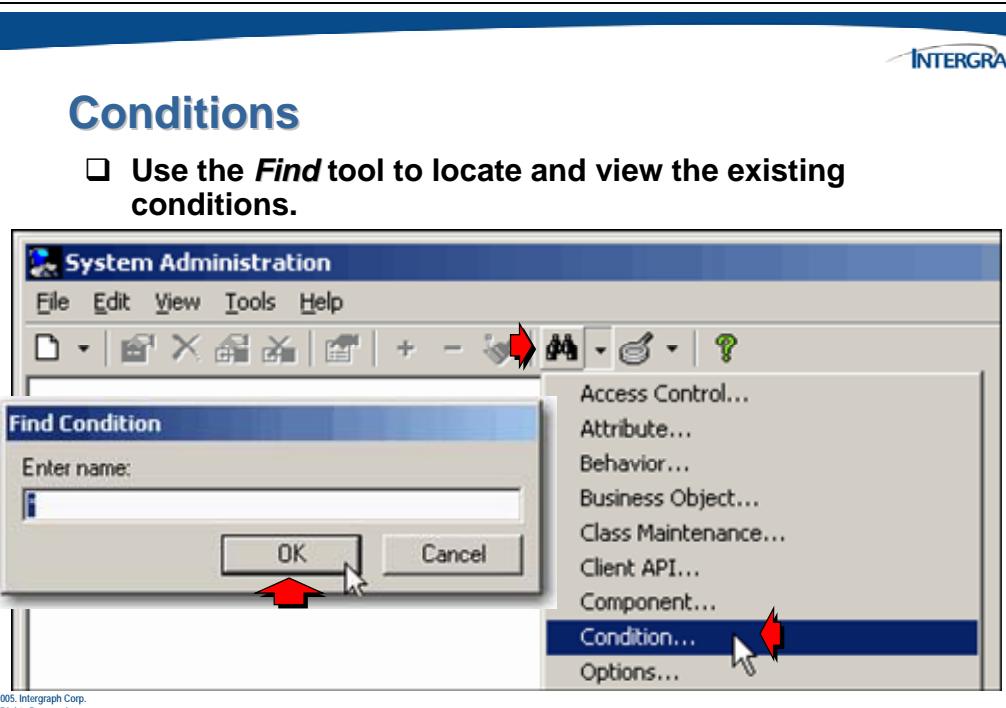
The following functions can be used when creating condition expressions:

Function Name	Description	Example
ABS	Specifies the absolute value of a number.	ABS(Obj.Value1) > 3
AVG	Specifies the average value for a returned set of values.	AVG(Obj.Value1, obj.Value2) < 5
COUNT	Counts the number of objects in a set.	COUNT(Obj->TagArea) > 0
INSTR	Searches for a particular string in the values returned for the attribute.	INSTR(Obj.Name, "Admin User") > 0
LEFT	Specifies characters on the left side of a string.	LEFT(Obj.Name) = "ABC"
LEN	Counts the number of characters in a string.	LEN(Obj.Name) > 5
LOWER	Converts the contents of the returned string to	LOWER(Obj.Name) =

	lowercase characters and compares the results to the specified string.	"graph"
MAX	Specifies the highest of the returned values.	MAX(Obj.Value1, Obj.Value2) > 5
MID	Returns all characters from a given position in a string to the end of the string.	MID("Intergraph", 6) would return a value of "graph".
MIN	Specifies a minimum value for returned values.	MIN(Obj.Value1, Obj.Value2) < 3
RIGHT	Specifies characters on the right side of a string.	RIGHT(Obj.Name) = "XYZ"
ROUND	Rounds the returned number to the nearest integer.	ROUND(Obj.Value1) >= 3
SUM	Evaluates the sum of all returned values.	SUM(Obj.Value1, Obj.Value2) < 5
TRUNC	Specifies the integer portion of a number.	TRUNC(Obj.Value1)=3
UPPER	Converts the contents of the returned string to uppercase characters and compares the results to the specified string.	UPPER(Obj.Name) = "ADMIN"

Note: For more detailed information on Conditions, refer to the SmartPlant Foundation System Administration User's Guide.

SmartPlant Foundation has existing conditions delivered that can be used in the configuration of methods.

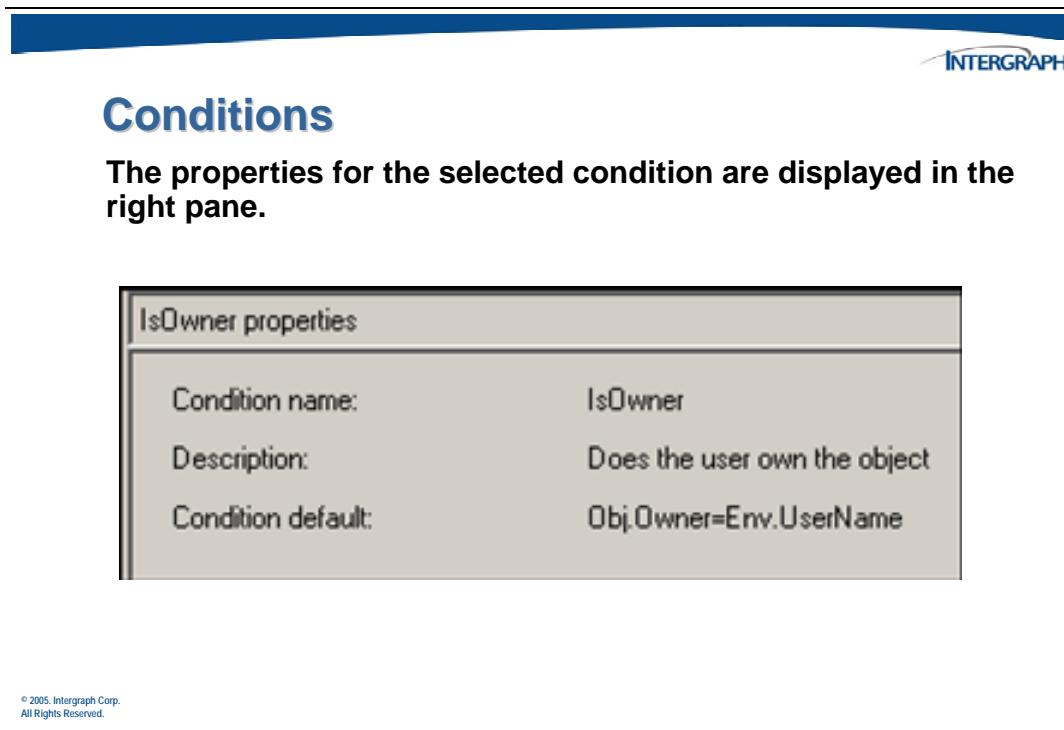
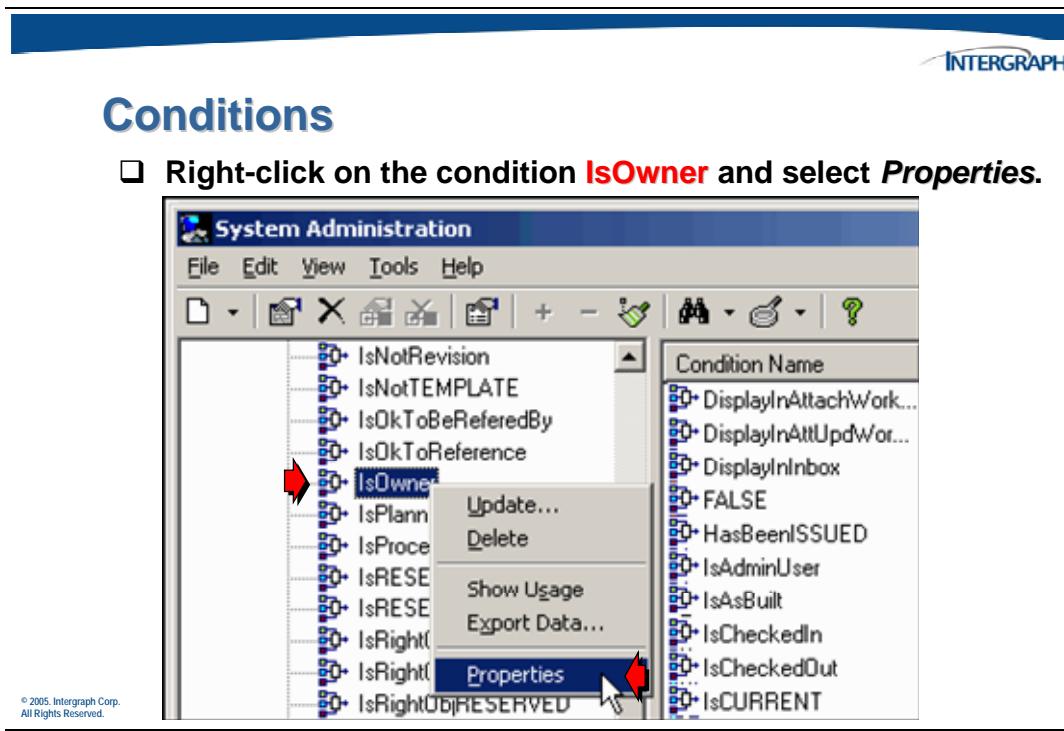


From the tree, click on the method title to populate the right pane with method properties.

The screenshot shows a list of existing conditions in the 'System Administration' application. The list is titled 'These are some of the existing conditions delivered with SPF.' and is presented in a table format. The columns are 'Condition Name', 'Description', and 'Condition Default'. The table lists various conditions such as 'DisplayInAttachWorkflow', 'DisplayInAttUpdWorkflow', 'DisplayInInbox', 'FALSE', 'HasBeenISSUED', 'IsAdminUser', 'IsAsBuilt', 'IsCheckedIn', 'IsCheckedOut', 'IsCURRENT', 'IsCURRENTOrSUPERSEDED', 'IsFileConversionNeeded', 'IsIntergraphEmployee', 'IsISSUED', 'IsLeftOwner', 'IsLeftOwnerAndRightObjNot...', and 'IsNotCURRENT'. Each row provides a brief description and the default condition value or expression. A red arrow points from the text 'From the tree, click on the method title to populate the right pane with method properties.' to the 'IsLeftOwner' condition entry in the table.

Condition Name	Description	Condition Default
DisplayInAttachWorkflow		Env.CALLEDFROMDIALOG IN ('AltWorkflow')
DisplayInAttUpdWorkflow		Env.CALLEDFROMDIALOG IN ('AltWorkflow')
DisplayInInbox		Env.CALLEDFROMDIALOG='Inbox'
FALSE	Reserved Condition for FALSE	2<>2
HasBeenISSUED	Xmtl has been issued	Obj.IssueState='ISSUED' or Obj.IssueStat...
IsAdminUser	Test if user is an admin user	INSTR(Env.UserUserGroups,'ADMIN')
IsAsBuilt	Is LCState = As Built	Obj.LCState='As Built'
IsCheckedIn	Is the revision object checked in?	Obj.AccessFlag=2 and Obj.IssueState='W...
IsCheckedOut	has an owner (access flag = 1) so is ch...	Obj.AccessFlag=1
IsCURRENT	Current ISSUED Document	Obj.IssueState='CURRENT'
IsCURRENTOrSUPERSEDED	CURRENT Or SUPERSEDED Document	Obj.IssueState='CURRENT' or Obj.IssueSt...
IsFileConversionNeeded	Check if it is needed to convert a file	Len(Obj.FileType)< 0
IsIntergraphEmployee	Test User Company Name	Env.UserCompanyName='Intergraph'
IsISSUED	An Issued Transmittal	Obj.IssueState='ISSUED'
IsLeftOwner	Must own left object (eg folder)	Obj1.Owner=Env.UserName
IsLeftOwnerAndRightObjNot...	Must own left obj & right obj not rev (fold...	Obj1.Owner=Env.UserName and Obj2.Cla...
IsNotCURRENT	Not a Current ISSUED Document	Obj.IssueState <> 'CURRENT'

Below is an example of the **IsOwner** delivered condition.



5.1 Configuring Methods

When configuring a new method, it can be designated a standalone method. Standalone methods are attached to menu items rather than Interface Definitions. These methods execute an action when no object is selected. For example, searching for a tag is achieved with a standalone method.



Methods

- Method name**
- Display name**
- Client API (related to Client API)**
- Arguments based on selected Client API**
- Component (related to Component)**
- Object state (relates to workflow state)**
- Flags for stand-alone method and UI availability**
- Cascade name**
- Condition name (relates to Condition)**
- Method access (values 0 or 1)**
- Email message**

Before configuring a new method, it might be helpful to perform a search and review some of the samples that can be delivered with an SPF system.



Configuring New Methods

- Perform a *Find* to locate and view the existing create new Document method.

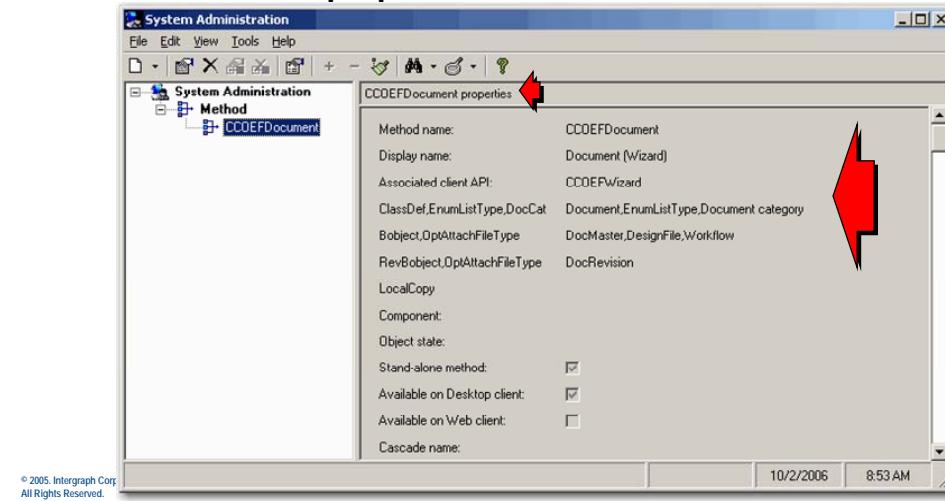


In this example, the **CCOEFDocument** method is shown.



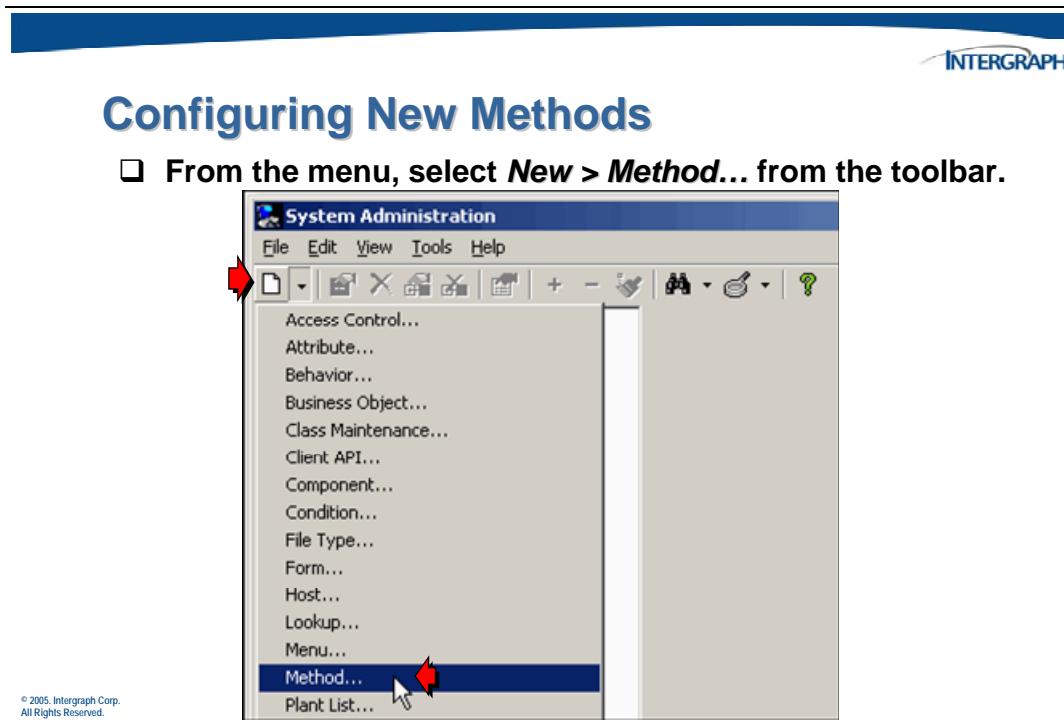
Configuring New Methods

- Use the *Properties* command from the shortcut menu to review the properties of the **CCOEFDocument** method.

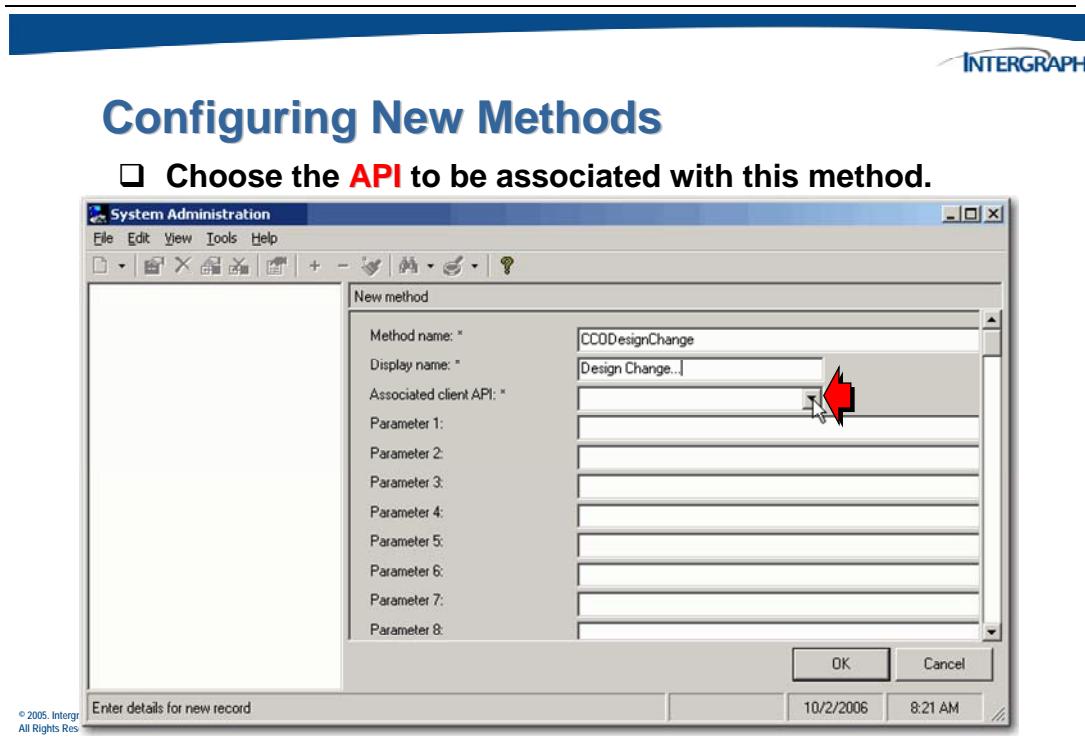


The **CCOEFDocument** method can be used as an sample to be followed in configuring a new method to be used for creating *ADCDesignChange* document instances. The Client API used in this method is the **CCOEFWizard** api as shown in the above example.

The following will demonstrate configuring a new method that will be associated with the *IADCDesignChange* Interface Definition and allow users to create new *ADC Design Change* objects. Methods are configured using the System Administration utility and a user must be logged in as an administrative user.



The *New method* form will display in the right pane.



The *New method* form contains the following fields:

- Method Name** - specifies the name/purpose of the method and is a required field.
- Display Name** - specifies the name of the method as you want it to display. This is a required field.



Configuring New Methods

Zoomed in view of the *New Method* form.

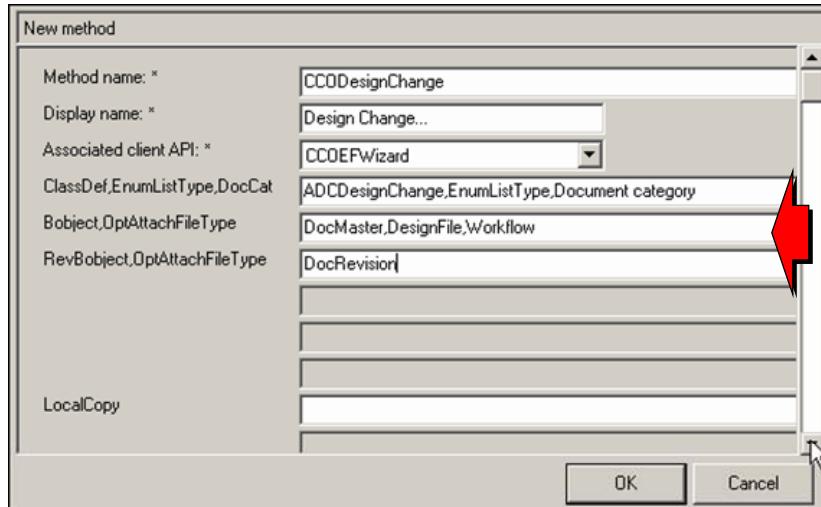
The screenshot shows the 'New method' dialog box. On the left, there are fields for 'Method name:' (CCODesignChange), 'Display name:' (Design Change...), and 'Associated client API:' (a dropdown menu). The dropdown menu is open, displaying a list of APIs: CCOEFWizard, CMClientEditObj, CMClientShowObj, ChangeOwnerOrGroup, ChangeOwningGroup, ChangeUser, and ChangeUserPassword. A red arrow points to the top of the dropdown menu. Below the dropdown are fields for 'Parameter 1' through 'Parameter 8', each with an empty input field. At the bottom left of the dialog box, there is a copyright notice: © 2005. Intergraph All Rights Reserved.

The *New method* form fields (con't):

- **Associated Client API** - associates a *Client API* with the method. Select the API from the list. Depending on which API is selected will control which fields are available in the rest of the *New method* form. This is a required field.

Configuring New Methods

- Enter the appropriate parameters for the selected **client API**.



The CCOEFWizard Client API

The CCOEFWizard API supports creating objects in the SmartPlant model. The CCOEFWizard API arguments describe the classification to create, the master object, with its related attachments, and the revision object with its related attachments to be created.

The CCOEFWizard API, uses commas as a delimiter to allow multiple attachments to be configured on the master, or revision object. The CCOEFWizard API relates all attachments to the revision object if a revision is created, and relates the attachments to the master object, if a revision is not created, such as in Offline files.

The CCOEFWizard API invokes the CCOEFWizard

CLIENTAPINAME ARG1

ARG2

ARG3

CCOEFWizard ClassDef,EnumListType,DocCat Bobject,OptAttachFileType RevBobject,OptAttachFileType

.ClientAPI

..ClientAPIName|CCOEFWizard

..ClientAPIDesc|Create EF Object Wizard

..Arg1|ClassDef,EnumListType,DocCat

..Arg2|Bobject,OptAttachFileType

..Arg3|RevBobject,OptAttachFileType

..Arg4| Not Used (This should be used to invoke another method)

CCOEFWizard API Argument Definitions

- Arg1** - Specifies the ClassDef, the classification object and the classification of the object at which to start the drilldown.
-

Note: ClassDef is required for Arg1.

Method values for this argument are derived from the Framework Administrator “Class Definitions” and “Enumerated Set”. For Example:

To display a classification tree displaying all document type classifications use:

“Document,EnumListType,Document category”

To display a single “Contract” form use:

“Contract,EnumDet,ContractDocument”

- Arg2** - Specifies the business object being created and any related objects such as *DesignFile*, *OfflineFile*, and *Workflow* that are to be attached to the master.

Method values for this argument are derived from the SPF SysAdmin “Business Object” table. For Example:

To create a Document master use “DocMaster

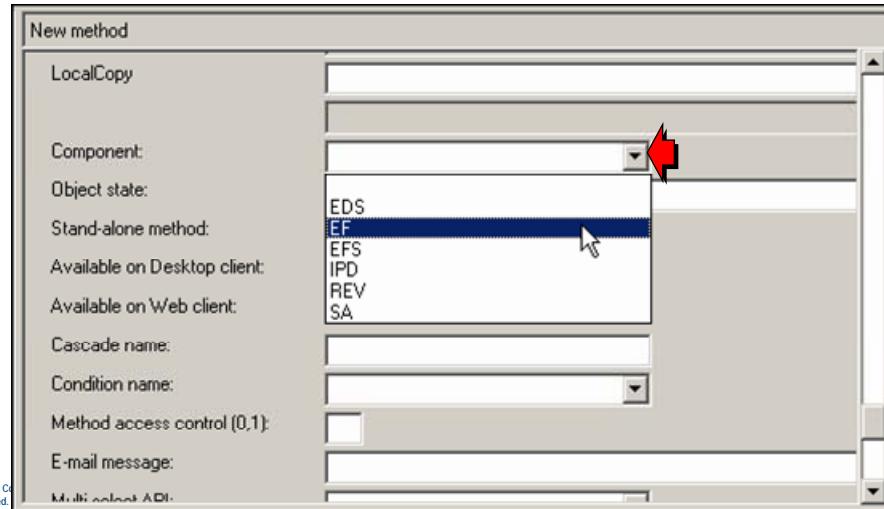
- Arg3** - Specifies the revision object and any related objects such as *DesignFile*, *OfflineFile*, and *Workflow*.

Method values for argument3 are derived from the SPF SysAdmin Business Object table. The current SmartPlant model always creates a revision with the Master, and always makes all attachments on the Revision.



Configuring New Methods

- Select the **EF component** which will be associated with this method.

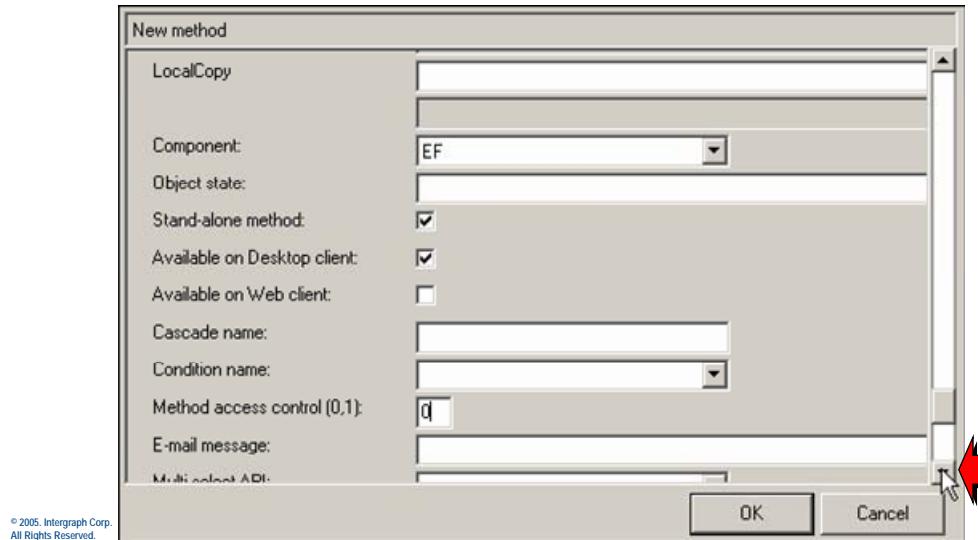


The *New method* form fields (con't):

- **Component** - specifies a component object. EF for most SmartPlant objects
- **Object state** - used to filter the appearance of the method based on the state of the business object to which it is attached.

Configuring New Methods

- Use the scroll bar to see the remaining property fields.



The *New method* form fields (con't):

- Stand-alone method** - specifies that this method to be a standalone method.
- Available on Desktop client** - specifies that this method to be available in the Desktop Client interface.
- Available on Web client** - specifies that this method to be available in the Web Client interface.
- Cascade name** - defines a submenu structure on the right click menu. Enter the name of the top-level menu in the field.
- Condition name** - restricts the validity of this method to certain conditions. Select a condition from the available list.
- Method Access Control** - controls the standard ownership states of a method. Enter the appropriate numeric value in the field.

The following are valid numeric states:

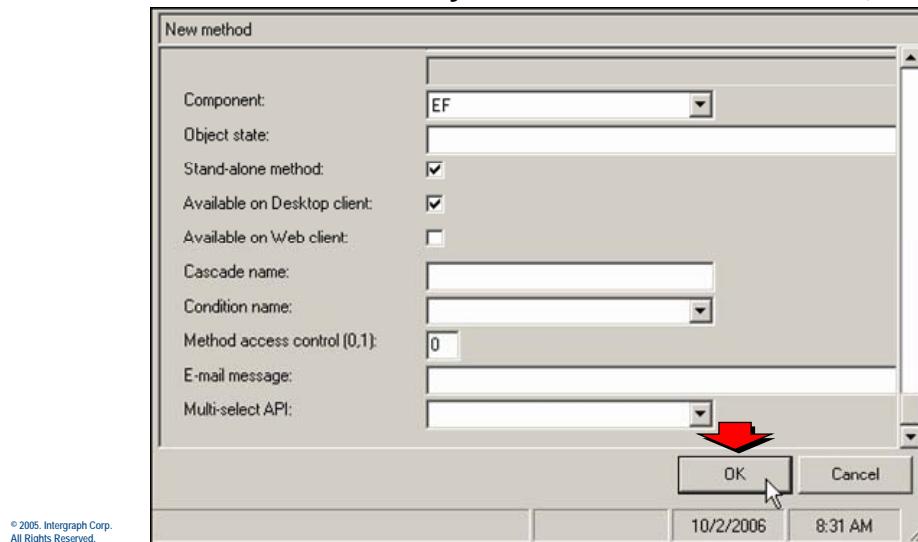
- | | |
|-----------------|---|
| 0 = open | An ownership state of 0 indicates that the object is open to all users who have access privileges. |
| 1 = user access | An ownership state of 1 indicates that only the user who can successfully checked out an object has access privileges to the object. For example, the Update access control method should be set to 1 to restrict access when the object is displayed. This allows only valid users in active user groups to access the |

object. However, certain methods, such as a Detail method, normally have a state equal to 0 because all users should have viewing privileges.



Configuring New Methods

- When the necessary values have been entered, click **OK**.



The *New method* form fields (con't):

- **Email message** - configures an email message to be sent to users that have subscribed to this event. Enter the email text in the field. If a user subscribes to an object, they can register for email notification when a particular command takes place on the object.

The text of the e-mail message must be 100 characters or less. You can include the following strings as placeholders in the e-mail message where you want SmartPlant Foundation to substitute specific information for the event:

\$CLASS - Use this string in the e-mail message as a placeholder for the class of the object. For example, the class of a P&ID might be PIDDrawing.

\$NAME - Use this string in the e-mail message as a placeholder for the name of the object.

\$METHOD - Use this string in the e-mail message as a placeholder for the name of the method.

\$USERNAME - Use this string in the e-mail message as a placeholder for the name of the user.

- **Multi-select API** – specifies to the Client API that you want to use when this method is run on multiple selected objects. Selecting a multi-select client API allows users to select multiple objects in the client list view and then invoke this method. A blank value indicates that this method is not supported on multiple objects.

A general multi-select API called MultiObject is delivered with SmartPlant Foundation. This multi-select API processes each command in sequence on the selected objects. The only client commands that specific multi-object select processing are *Check Out*, *Check In*, *Undo Checkout*, and *Sign Off*.

For example, the Revision CheckIn method uses the Checkin API if one object selected in the SmartPlant Foundation client, but it uses MultiCheckIn API if more than one object is selected.

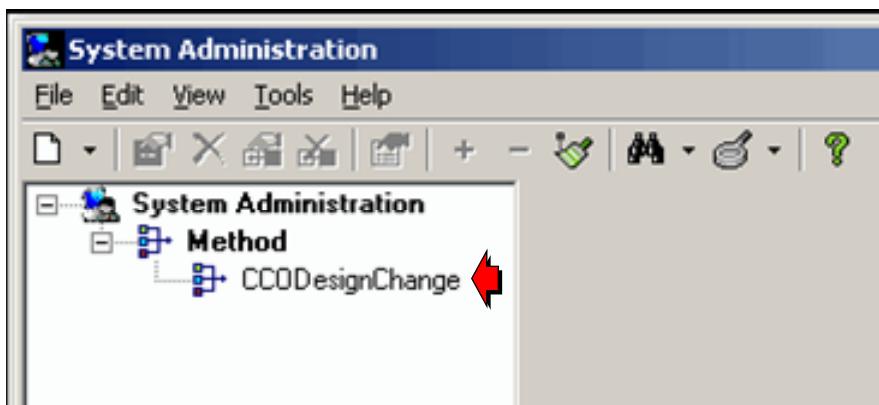
The *Tree View* will display the method that has been added.



INTERGRAPH

Configuring New Methods

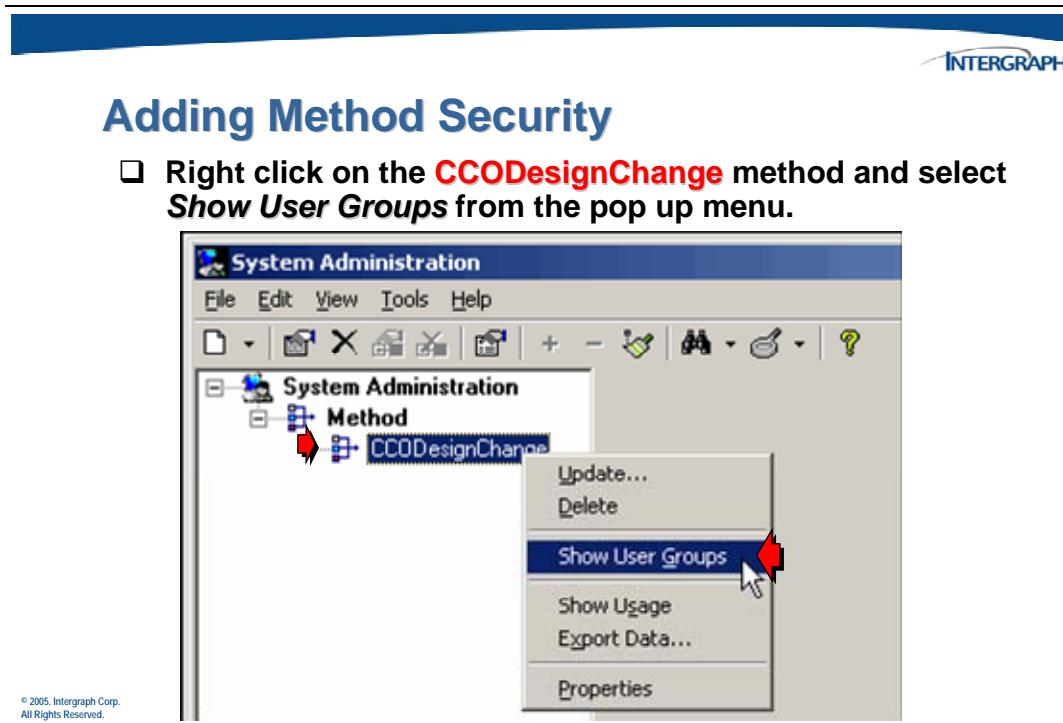
The new method is displayed in the **Tree View**.



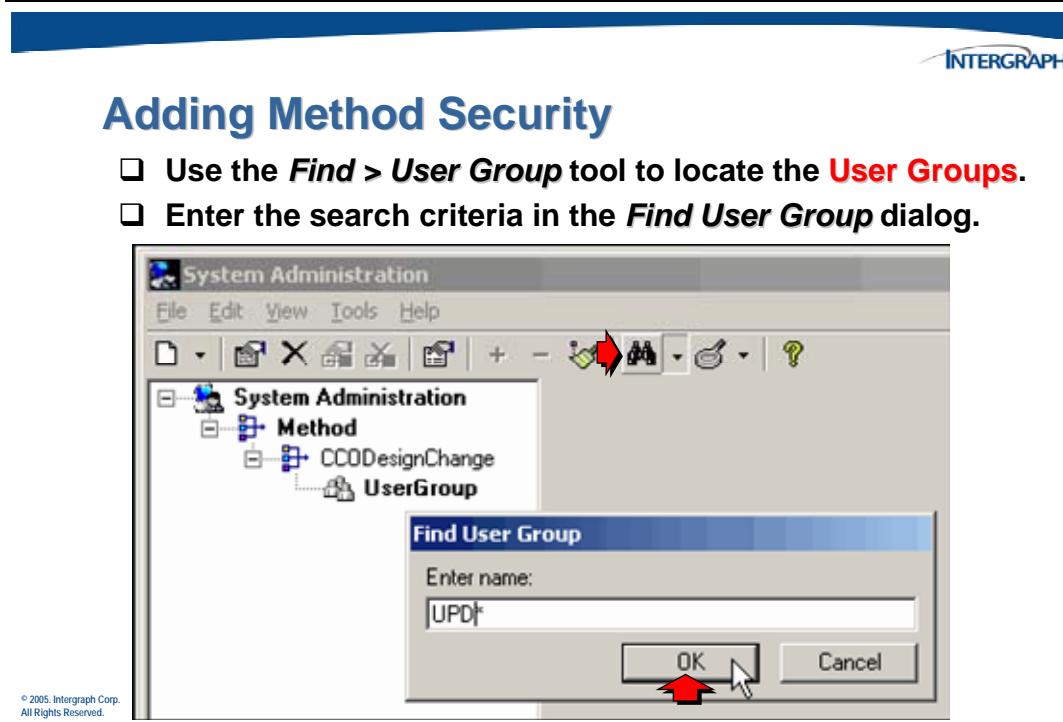
5.1.1 Setting Method Security

To access the new method, access security must be set. This will be accomplished in the same manner as other access security, that is a User Group will be dragged and dropped onto the UserGroup relation for the method.

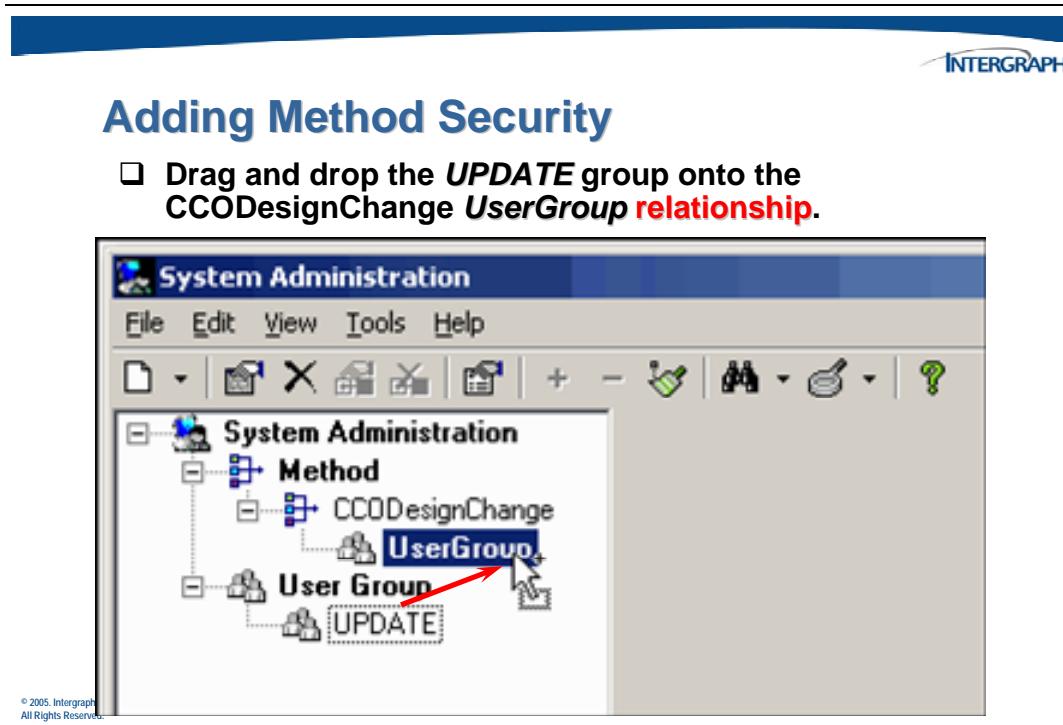
To set method access security:



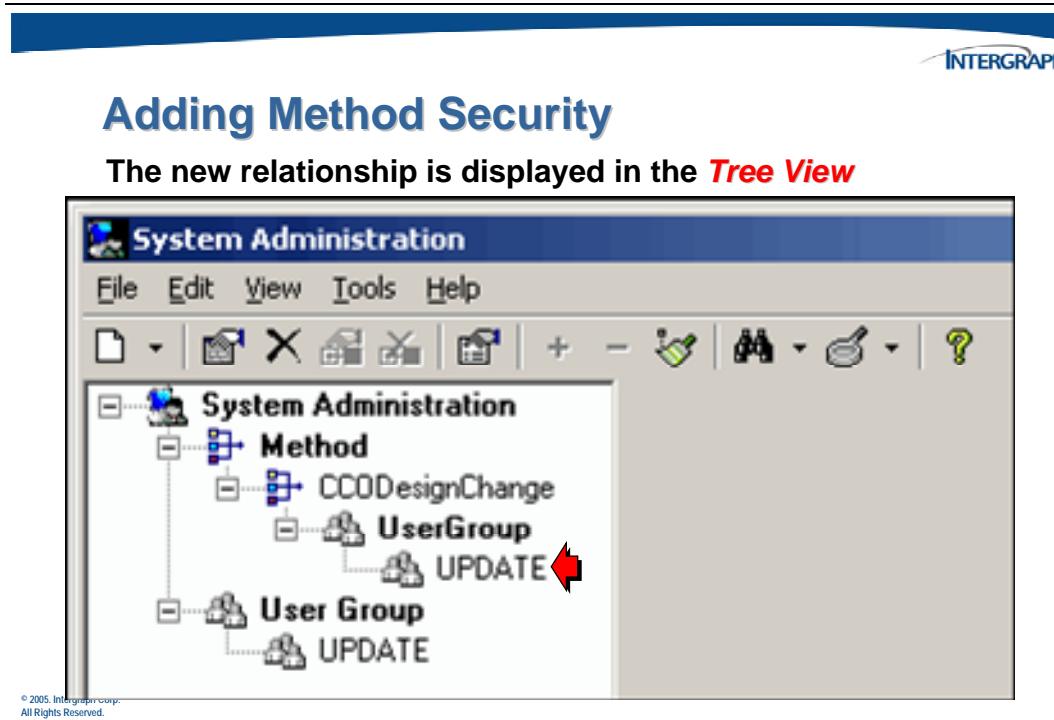
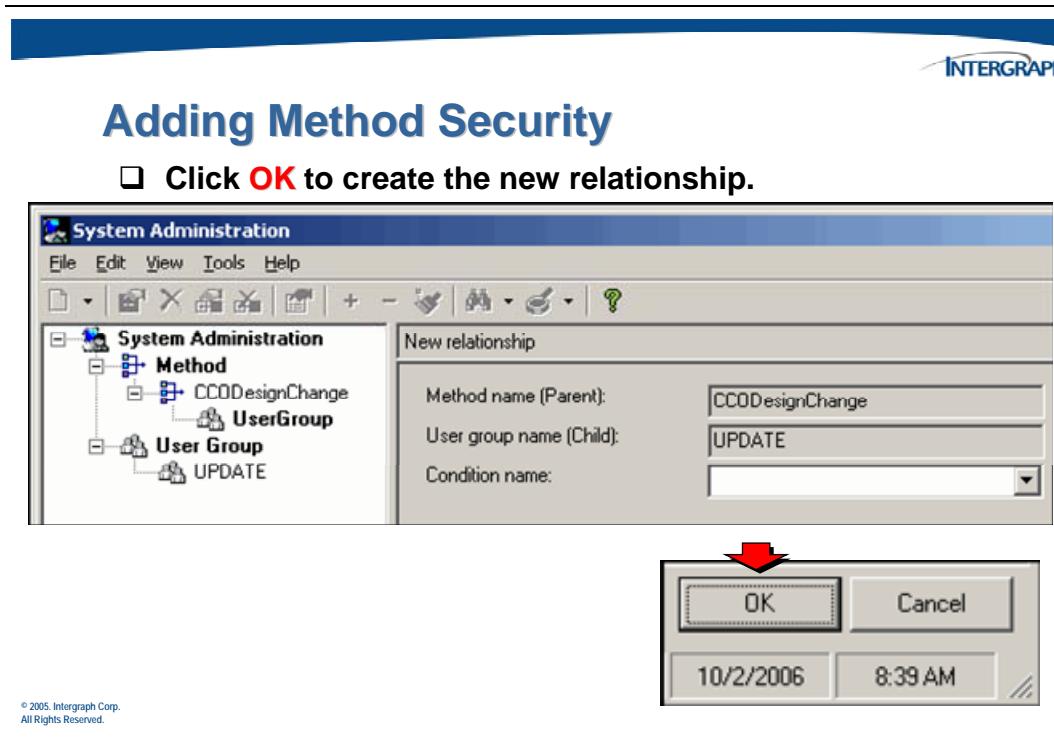
Before a *User Group* can be dropped onto the method relationship, a search has to be performed. The *Find User Group* dialog will display.



Now, relate a user group to the new method.

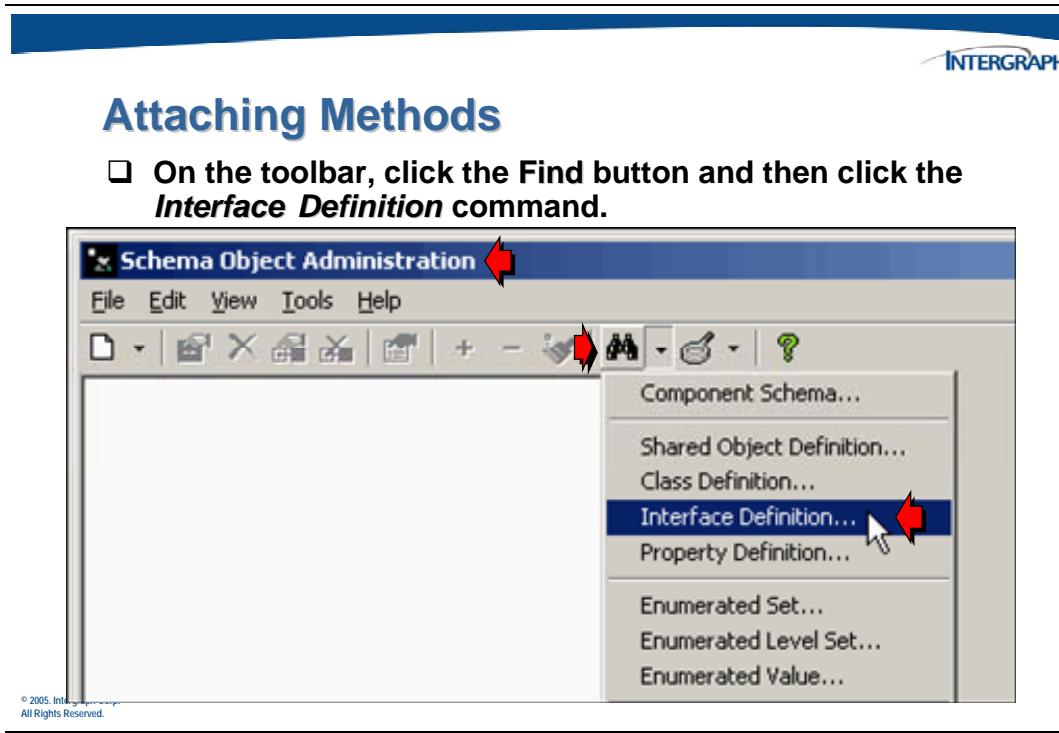


A condition can be added to the relationship security.



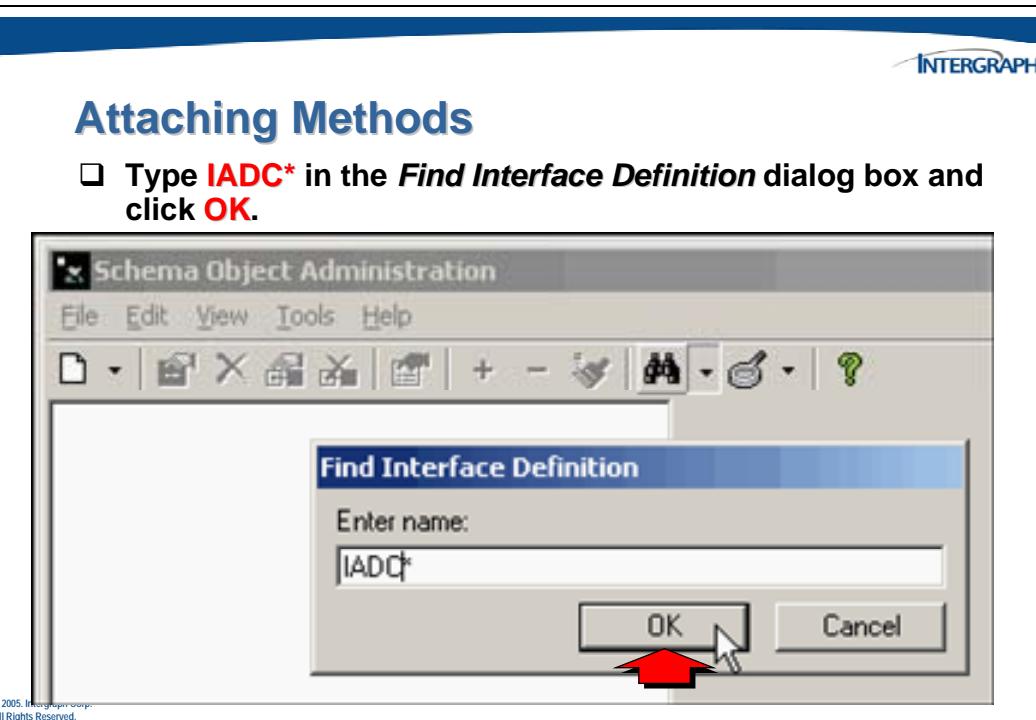
5.2 Attaching Methods

Attaching a method to an interface definition means that the method is available for all objects associated with the interface definition. For example, if you attach a method to the **IADCDesignChange** interface definition, right-clicking all class definitions that realize the **IADCDesignChange** interface in the SPF client displays the selected method on the shortcut menu.

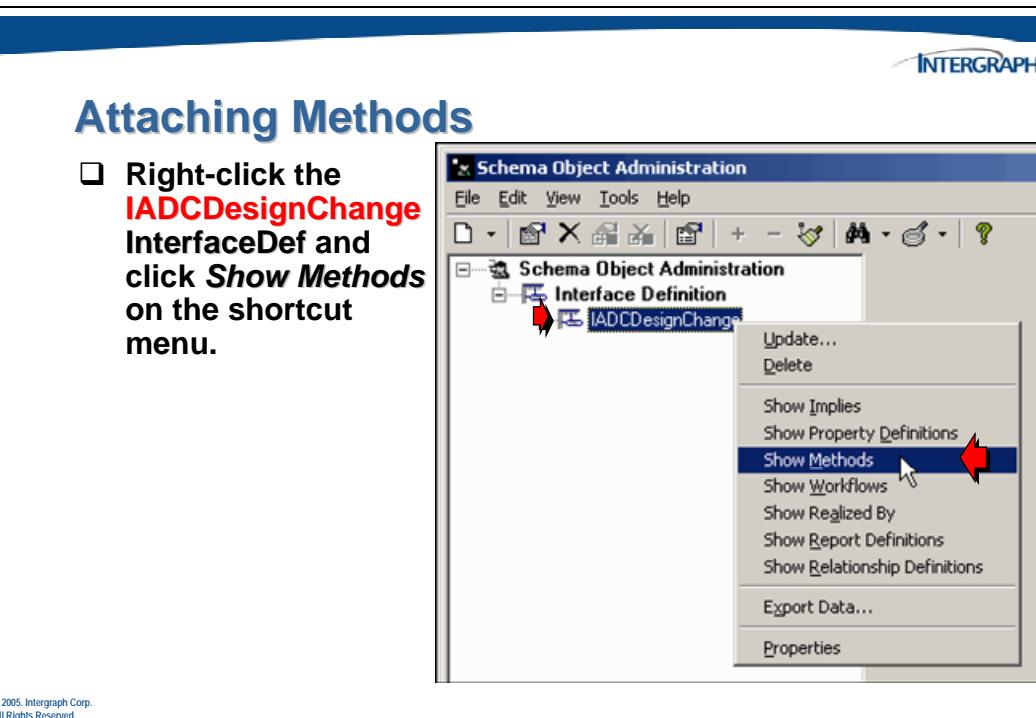


For example, there may be a custom SPF method that will calculate the insulation density needed for a given item (equipment, pipeline, and so on). However, the method should not be attached to a ClassDef because there is an InterfaceDef that is supported by objects that are insulated.

The *Find Interface Definition* dialog box appears.



The **IADC*** *interface definition* will display in the Tree view.

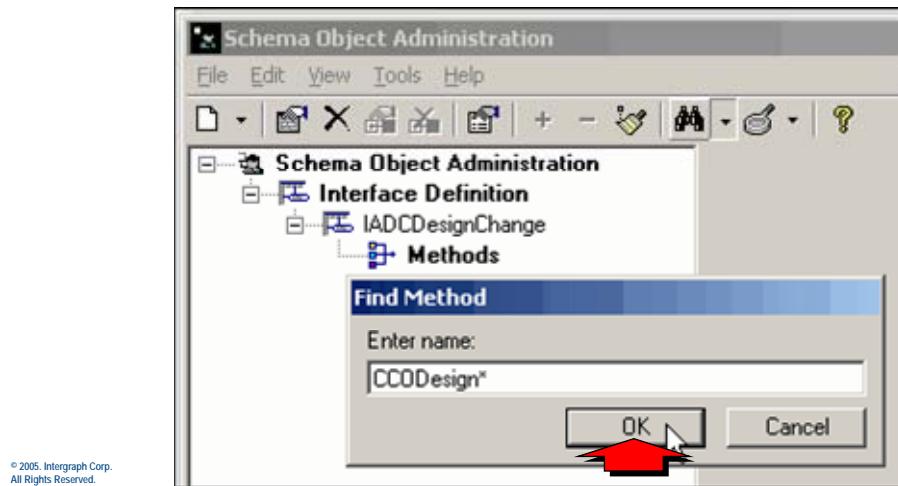


Next, perform a search to locate the custom method that was configured in the System Administration utility. In the *Find Method* dialog box, enter the search criteria to locate the custom method.



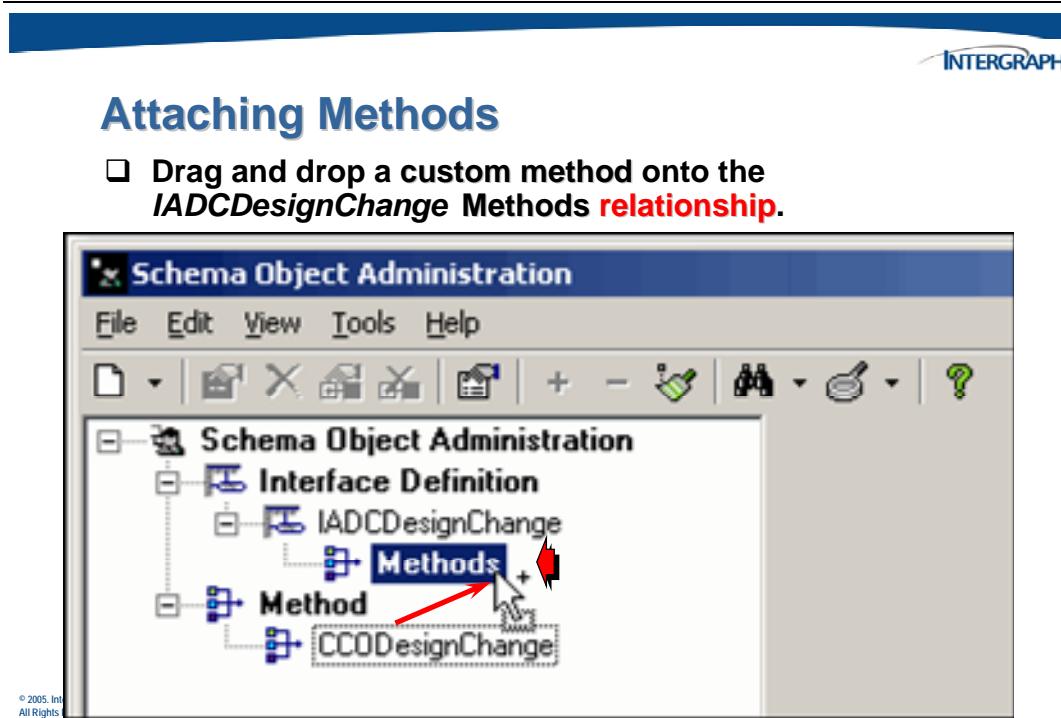
Attaching Methods

- Use the **Find > Method...** tool to locate the **Method**.
- Type **CCODesign*** in the **Find Method** dialog box.

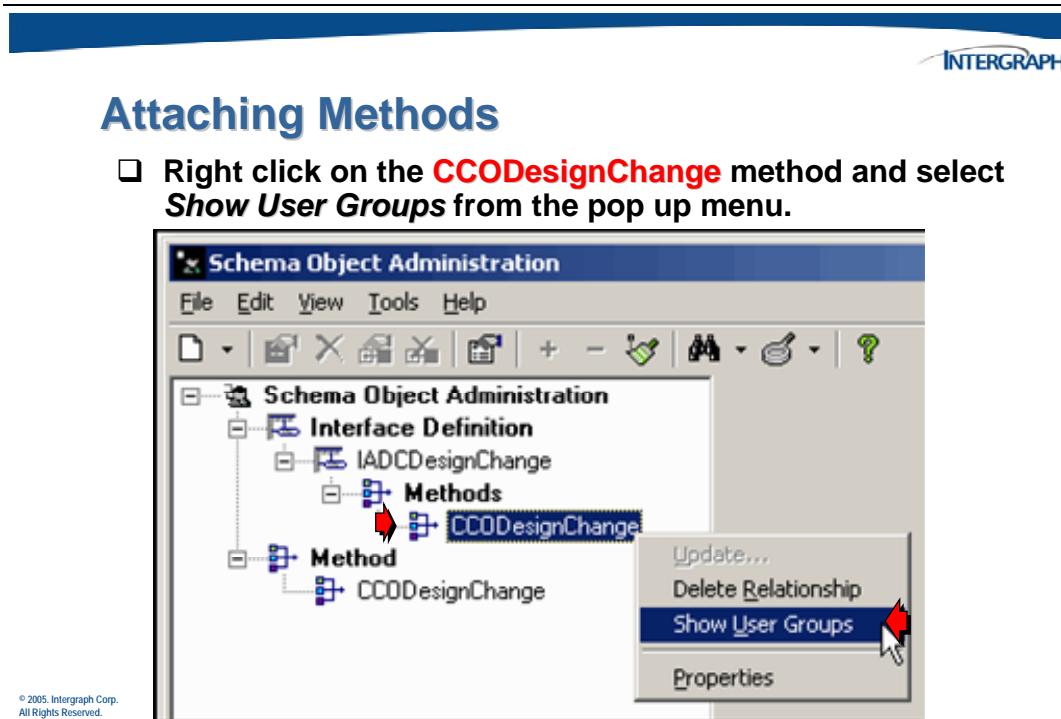


© 2005, Intergraph Corp.
All Rights Reserved.

Select the custom method in the Tree view and drag it onto the InterfaceDef Methods relationship.



The custom method should already have a user access relationship.

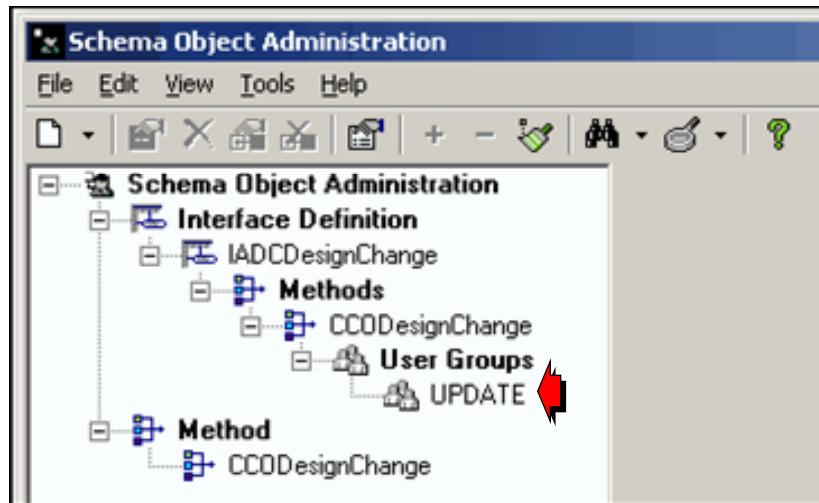


The tree view will expand to show the *User Groups* relationship.



Attaching Methods

The method already has user access security from an earlier drag and drop operation.



© 2005, Intergraph Corp.
All Rights Reserved.

5.3 Configuring New Menu Items

Menus are used to group various actions performed in SPF Client.



Menu Nodes

Menu nodes are static and always displayed.

- Menu name**
- Display name**
- Parent node (for nested menus)**
- Sequence number (0-999)**

© 2005. Intergraph Corp.
All Rights Reserved.

Menu items (actions) can be created for each menu specifying the action to be performed, the method to be used, and other related properties. A user must be a member of a user group associated with the menu item in order to gain access to that item.



Menu Items

Menu items are created under menu nodes

- Menu name (parent)
- Menu item name (child)
- Display as name
- Icon (not used)
- Method name (related to Method with stand-alone flag)
- Separator required indicator flag
- Sequence number

User access is controlled by associating user group to method that is attached to menu item.

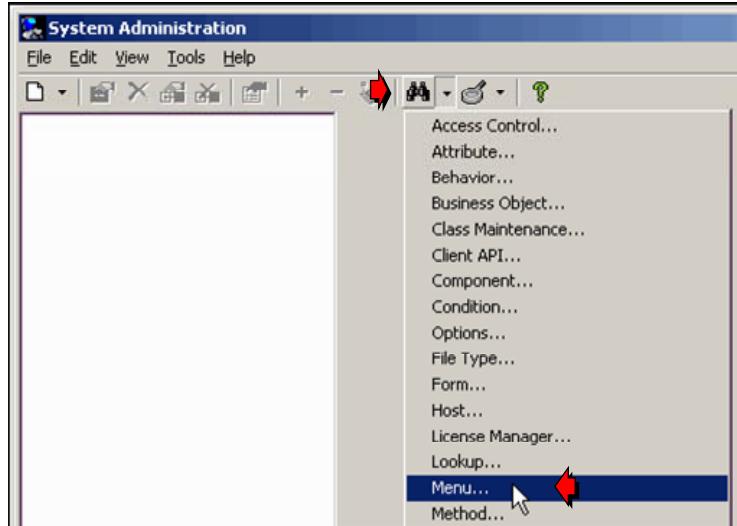
© 2005, Intergraph Corp.
All Rights Reserved.

In the following example, an existing menu will be located and a new menu item added to it.



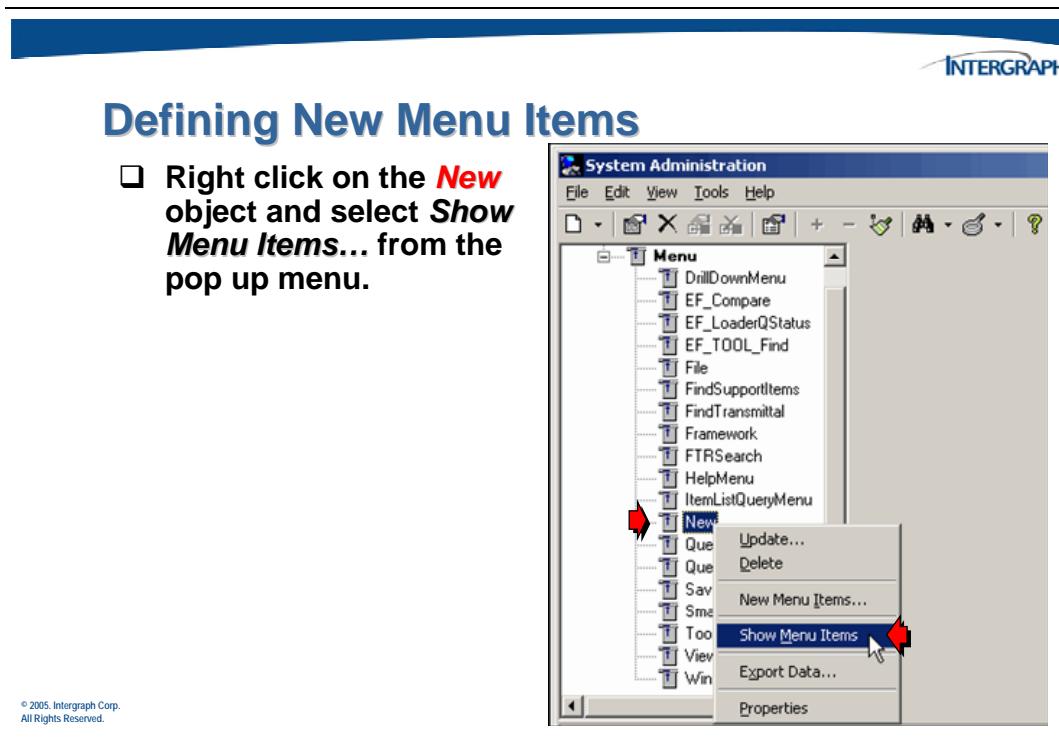
Defining New Menu Items

- Use the **Find > Menu...** tool to locate the existing **Menus**.



© 2005, Intergraph Corp.
All Rights Reserved.

Locate the menu node, which is to contain the new item.



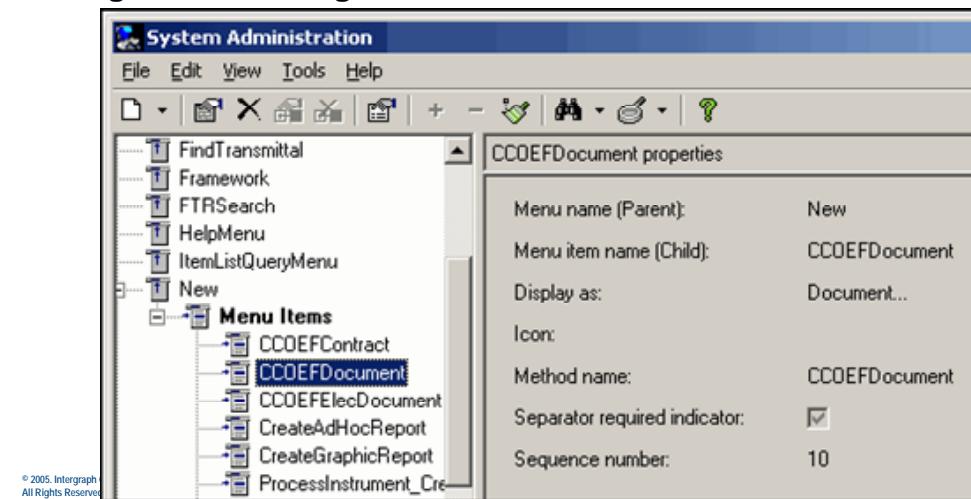
Like the method earlier, use the document menu item (**CCOEFDocument**) as a sample for creating a new menu item to use with new *Design Change* documents.





Defining New Menu Items

Review the properties of an existing menu item to use as a guide for adding a new menu item.

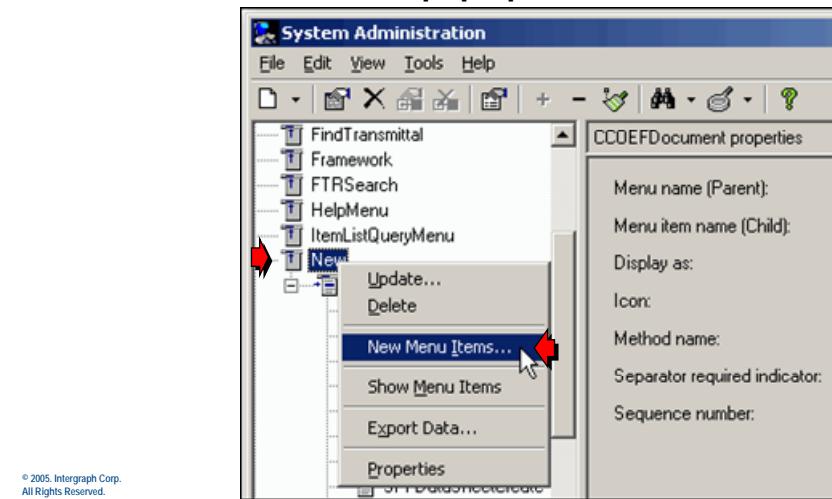


Now, a new menu item (command) is ready to be configured.

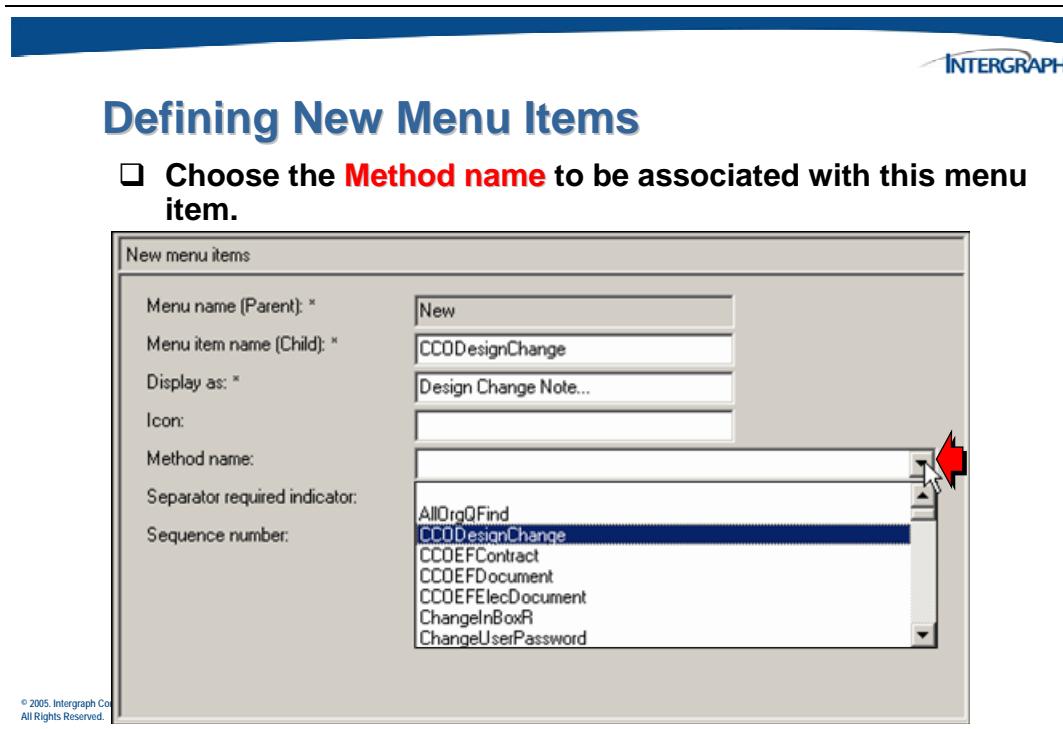


Defining New Menu Items

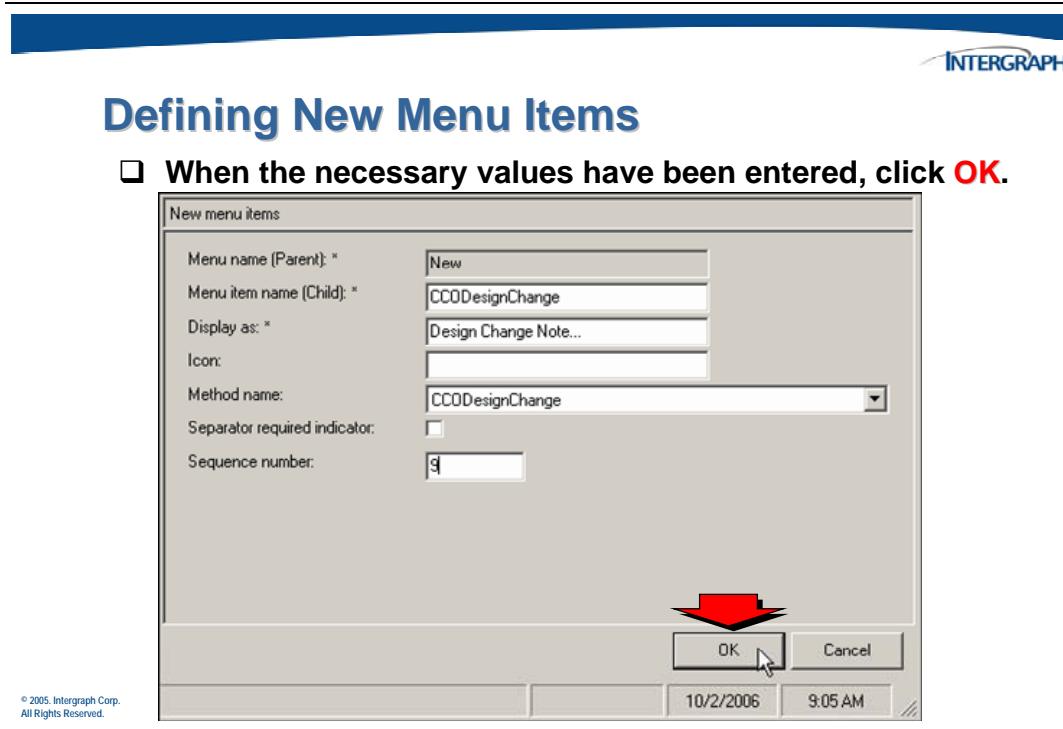
- Right click on the **New** object and select **New Menu Items...** from the pop up menu.



The *New menu items* form will display.



Continue to define the remaining values for the new menu item.



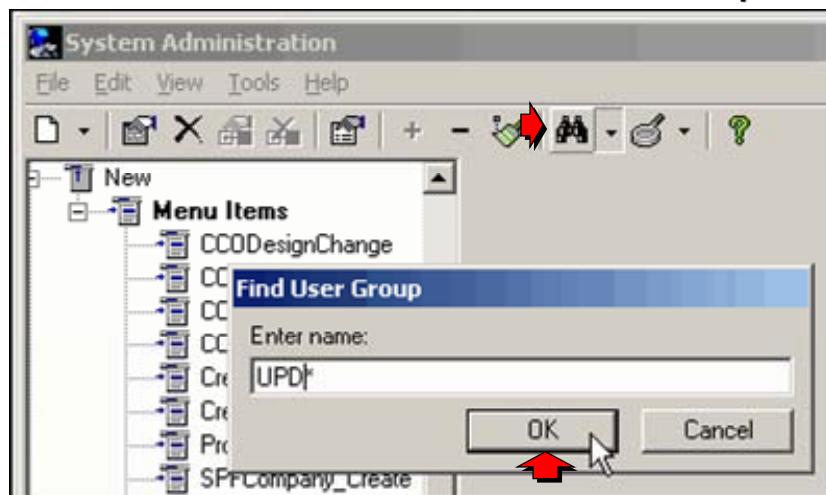
5.3.1 Adding Menu Item Security

Just like toolbars, once the menu item has been created, the necessary access security must be added. Perform a search to locate the user groups if they are not already displayed in a window.



Adding Method Security

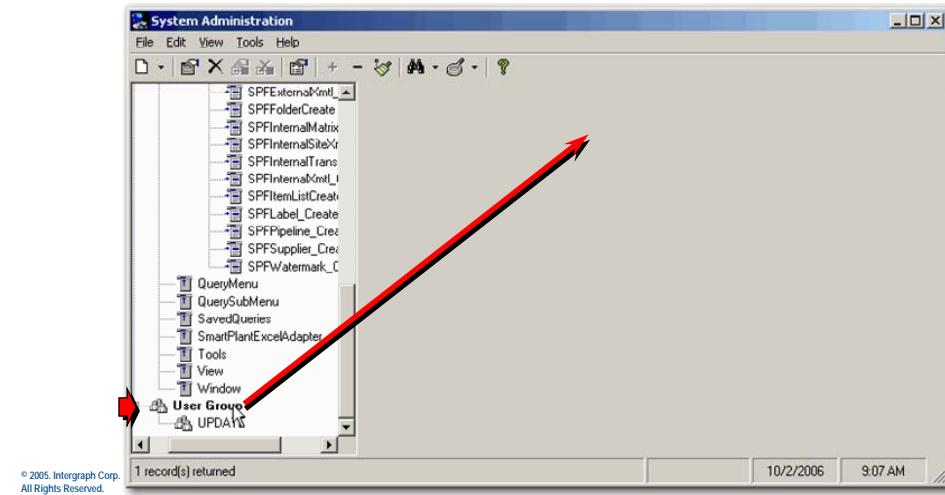
- Use the **Find > User Group** tool to locate the **User Groups**.
- Enter the search criteria in the **Find User Group** dialog.





Defining Menu Item Security

- Click on the **User Group** heading to display the User Group details in the right pane.

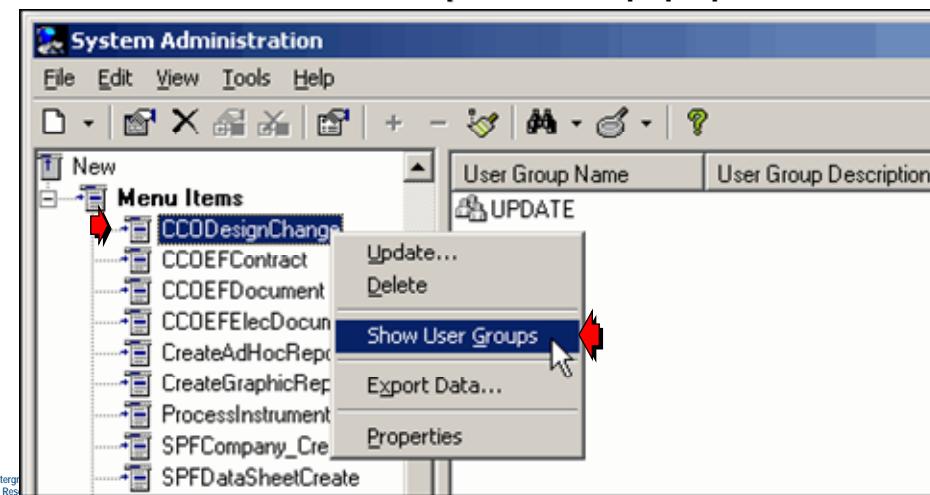


To create the security access, create a relationship between a *user group* and the *menu item*.



Defining Menu Item Security

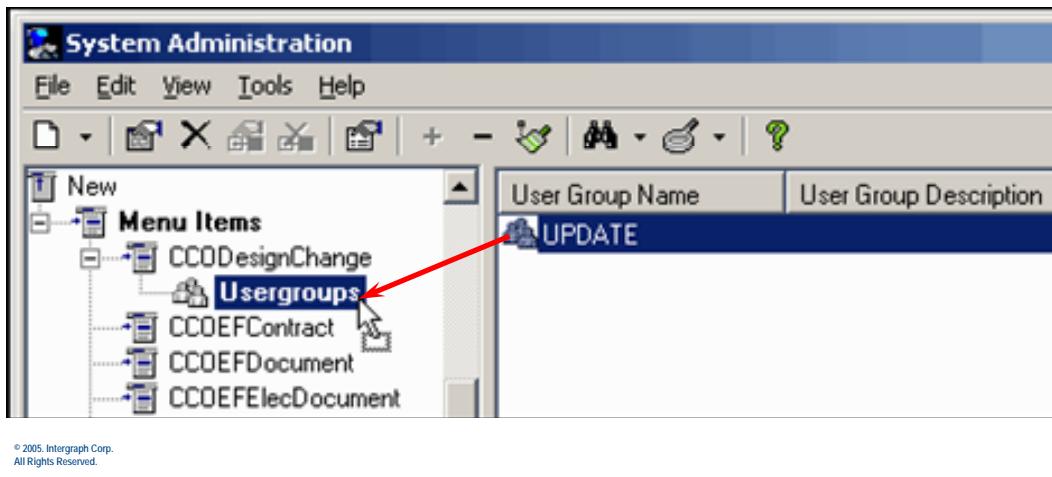
- Right click on the **CCODesignChange** menu item and select **Show User Groups** from the pop up menu





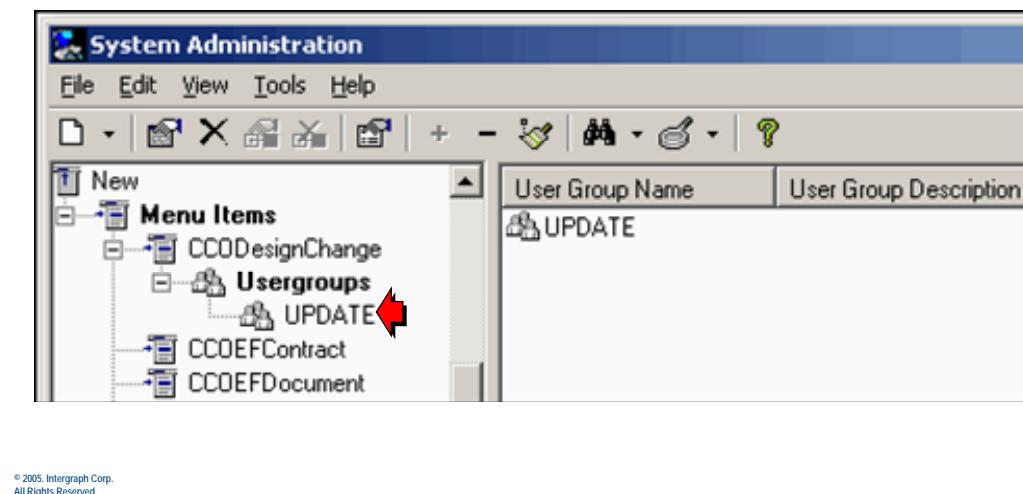
Defining Menu Item Security

- Drag and drop the **UPDATE** group onto the **CCODesignChange UserGroup relationship**



Defining Menu Item Security

The new menu item relationship is displayed in the **Tree View**



5.4 Activity 1 – Configuring Methods

The objective of this activity is to create a method for *IADCDesignChange*. This method will be used to create instances of these Interface Definitions.

If you are not currently logged into your machine:

1. Log on to your operating system (if not already logged in):
spfuser with no password

2. Start the *SysAdmin Utility* by selecting *Start > All Programs > Intergraph SmartPlant Foundation > System Administration*.

3. When the *Login* dialog window appears, use the *User name* **adminuser** with no password and click on **OK**.

4. Create a method to use with your Interface Definitions. From the System Administration menu, select the **New > Method** tool. In the *New method* form, enter the following:

<input type="checkbox"/> Method Name -	<i>CCODesignChange</i>
<input type="checkbox"/> Display Name -	<i>Design Change...</i>
<input type="checkbox"/> Associated client API -	<i>CCOEFWizard</i>
<input type="checkbox"/> ClassDef, EnumListType, DocCat -	<i>ADCDesignChange, EnumListType, Document category</i>
<input type="checkbox"/> Bobject,OptAttachFile Type -	<i>DocMaster,DesignFile,Workflow</i>
<input type="checkbox"/> RevBobject,OptAttachFileType-	<i>DocRevision</i>
<input type="checkbox"/> LocalCopy-	leave blankup
<input type="checkbox"/> Component -	<i>EF</i>
<input type="checkbox"/> Object state -	leave blank
<input type="checkbox"/> Stand-alone method -	<i>toggle enabled</i>
<input type="checkbox"/> Available on Desktop client -	<i>toggle enabled</i>
<input type="checkbox"/> Available on Web client -	leave blank
<input type="checkbox"/> Cascade name -	leave blank
<input type="checkbox"/> Condition name -	leave blank

- Method Access Control - *0*
- Email message - leave blank
- Multi-select API - leave blank

Click **OK** to configure the new method.

5. Add a user group to the method relationship to allow a group create functionality.

- Right click on the method name and select **Show User Groups** to view the relationship.
- Drag and drop the user group objects to the method relationship object.
 - *UPDATE to CCODesignChange*

Verify in the *Tree View* pane that relations between the user groups and the method has been created.

6. Add the method to the Interface Definitions method relationship to allow create functionality.

- Perform a search for the Primary Interface Definition (Hint: IADC*)
- Right click on the Interface Definition name and select **Show Methods** to view the relationship.
- Drag and drop the method object to the Interface Definition relationship object.
 - *CCODesignChange to IADCDDesignChange*

Verify in the *Tree View* pane that relationship between the method and the Interface Definition relationship has been created.

7. Verify that the user group associated with the method and with the Interface Definition relationship for navigate/show functionality.

- Right click on the Interface Definition method relationship name and select **Show User Groups** to view the relationship.

Verify in the *Tree View* pane that relations between the user groups and the method Interface Definition relationships exist.

8. You are now going to add additional methods to the Interface Definitions method relationship to allow users *Detail*, *History* and *Update* functionality.

- Perform a search for all the methods

- View the list of methods in the right pane
- Drag and drop the method objects to the *IADCDesignChange* Interface Definition relationship objects.
 - *GenDetail*
 - *GenHistory*
 - *GenUpdate*

Verify in the *Tree View* pane that relations between the methods and the Interface Definition relationships have been created.

9. Continue this hands on session and complete activity 2.

5.5 Activity 2 – Configuring Menu Items

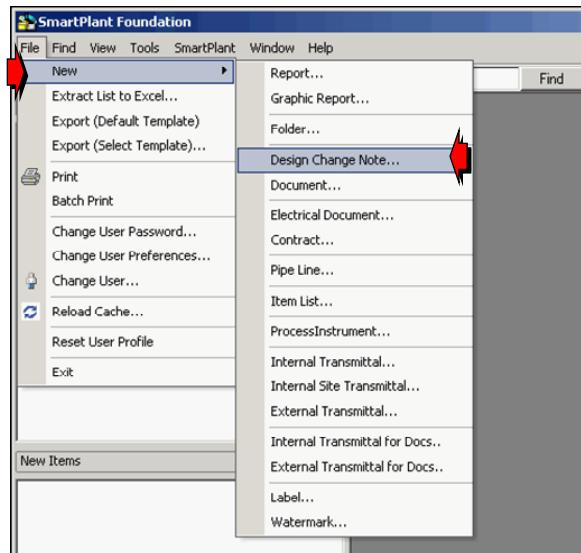
The objective of this activity is to create a menu item for the **Design Change** class and attach the *CCODesignChange* method created in activity 1 of this chapter.

If you are not currently logged into your machine:

1. Log on to your operating system (if not already logged in):
spfuser with no password
2. Start the *SysAdmin Utility* by selecting *Start > All Programs > Intergraph SmartPlant Foundation > System Administration*.
3. When the *Login* dialog window appears, use the *User name* **alex** with no password and click on **OK**.
4. Perform a search to find existing menu nodes.
5. Add a new menu item to be able create a new instance of the Design Change class.
6. Set an appropriate sequence number for the new menu item..
7. After you create the menu option, test your method and add a new *Design Change* document interactively. You can use the folder path **D:\SPF_Training** to attach the *Ortho Piping.igr* file to test the new method. **Do not** submit the new document to a workflow.

Activity 2

- This is an example of how the menu should look in order to test your new create method.



© 2005. Intergraph Corp.
All Rights Reserved.

You may take a short break until the other students have finished activity 2.

5.6 Additional Model Components

If you wish to create a module like SPF's EF or REV when your code is completed, you can create your own component.



Components

Components are dll's including executable code

- Component name**
- Component address**

Example: Component Name: EF

Component Address: SPF38EFBSL.grpDispatch

© 2005. Intergraph Corp.
All Rights Reserved.

-
- Component Name** - the name of your module
 - Component Address** - how to find the dll you create

SPF38EFBSL is the dll that was created (the dll name will be entered here).

.grpDispatch is the SPF code which calls the dll you create.

SPF will search in the main install directory for all dll's (i.e. \\Smartplant\\Foundation\\2007).

5.7 Context Sensitive Menus

The *Right Mouse Button* operations are made up of the *Relationships* you create and the methods, which are related to the Interface Definitions. The security is setup with the User Group linked to the relationship or method.

User Interface Configuration

- Context sensitive right hand mouse button menu includes all methods and relationship expansions that the user is authorized to execute
- Static user interface contains menus and toolbars
- Appearance of menu items in menus and toolbar buttons is also controlled by user access

© 2005, Intergraph Corp.
All Rights Reserved.

User Interface Configuration

Right Mouse Button
Menu

Code –
All Relations

Code

Methods

Relationships

- Change Owning Group...
- Copy...
- Details
- Edit Relationships...
- Files
- Generate PDF Copies
- History
- Show Nested References
- Subscription
- Transmittal History
- Update...
- Workflow
- Show All Relationships
- Show All Versions
- Show Document Master
- Show External Transmittals
- Show Files
- Show Folder
- Show Internal Transmittals
- Show Label
- Show Latest Revision
- Show Plant Items
- Show Reference File Revisions
- Show Reference Revisions
- Show Revisions Referenced By
- Show Top Files
- Show Watermark
- Clear Selection
- Select All
- Refresh
- Open in New Window

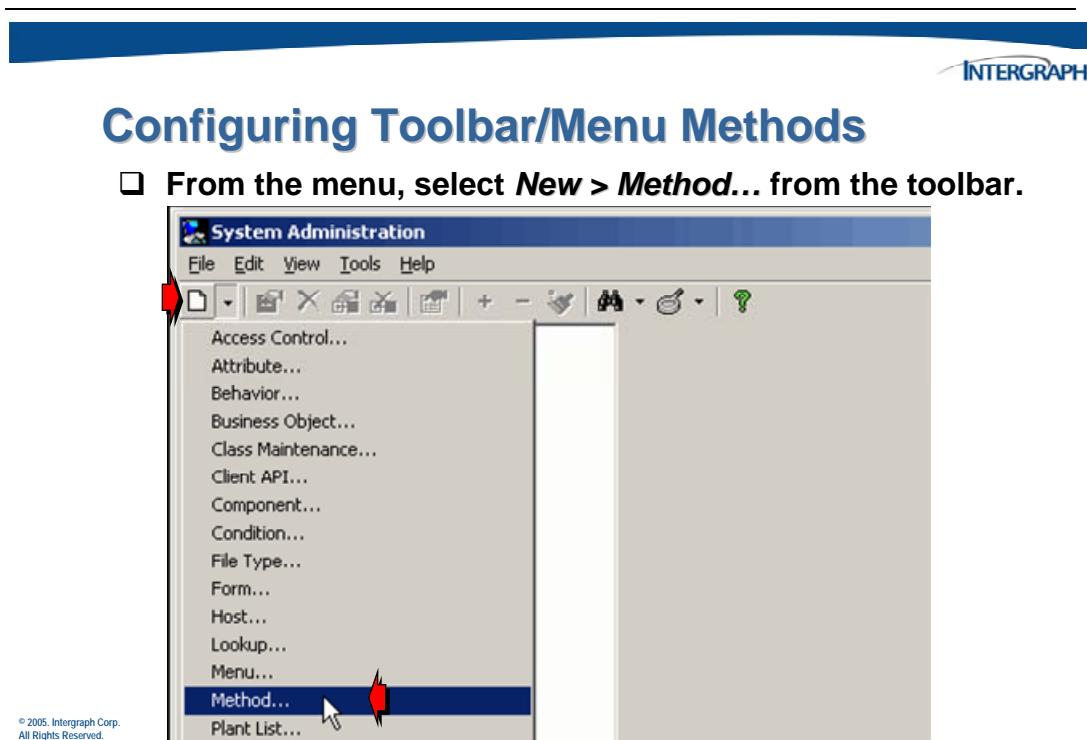
© 2005, Intergraph Corp.
All Rights Reserved.

5.8 Configuring Toolbar and Menu Methods

Toolbar items provide a shortcut method for performing various actions in an SPF Client. Before the toolbar or menu item is created, a method must be defined for use in the toolbar item or the menu item.

In the following example, a new method, **QuickFindDesignChange**, will be created in order to be associated with a toolbar item and a menu item.

From the System Administration menu, select **File > New > Method**. The *New method* dialog will display.



Some new methods can be designated a **standalone** method. Standalone methods are attached to menu items in addition to Interface Definitions. These methods execute an action when no object is selected.

The *New method* form will be displayed.



Configuring Toolbar/Menu Methods

Choose the **API name** to be associated with this method.

New method

Method name: *	QuickFindDesignChange
Display name: *	Design Change...
Associated client API: *	<input type="button" value="▼"/>
BObject Name	ProjectFilter
Interface Name	Publish
Display As	PutWorkflowOnHold
Rev Status	<input type="button" value="▼"/>
Query Filter (Condition Name)	QFindEFObj
Column Display (BObject Name)	QFindObject
	QueryEFObj
	QueryObj
	RegisterAndCheckIn

© 2005. All Rights Reserved.

Continue to define the new method attributes.



Configuring Toolbar/Menu Methods

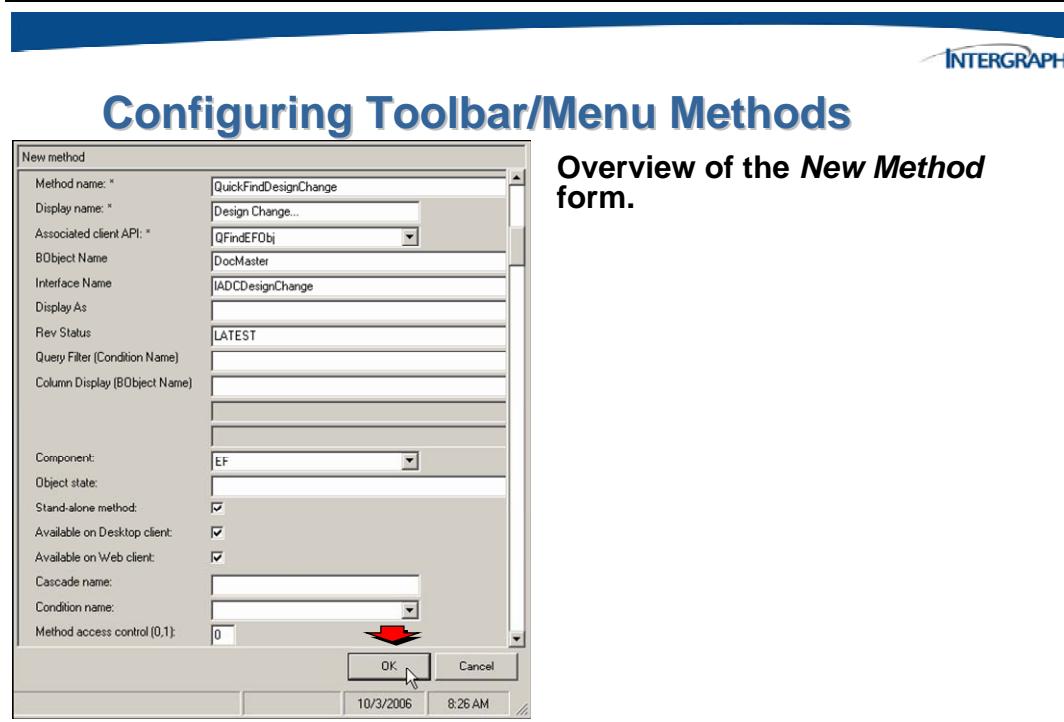
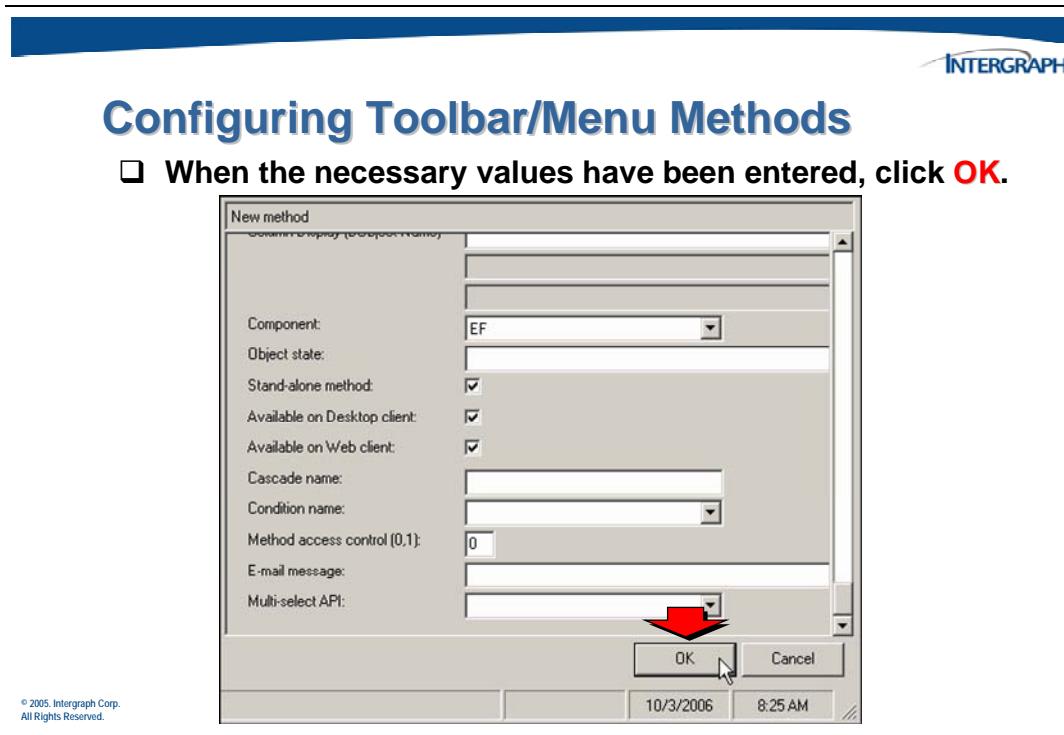
Use the scroll bar to enter values in the remaining fields.

New method

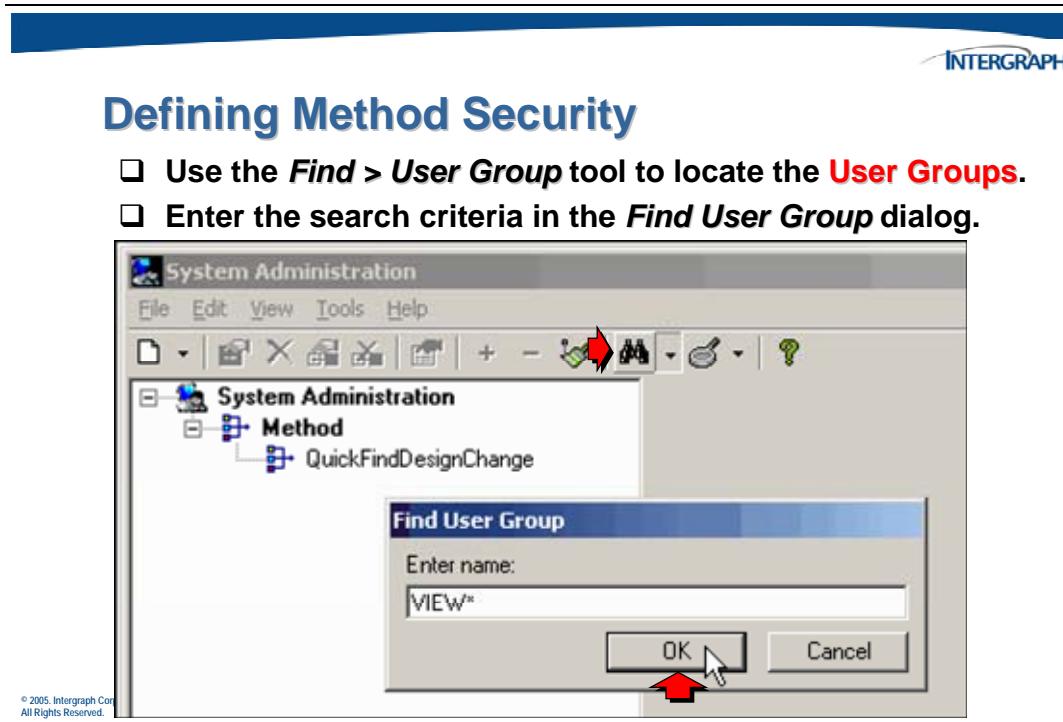
Method name: *	QuickFindDesignChange	<input type="button" value="▼"/>
Display name: *	Design Change...	<input type="button" value="▼"/>
Associated client API: *	QFindEFObj	<input type="button" value="▼"/>
BObject Name	DocMaster	
Interface Name	IADCDesignChange	
Display As		
Rev Status	LATEST	
Query Filter (Condition Name)		
Column Display (BObject Name)		

© 2005. All Rights Reserved.

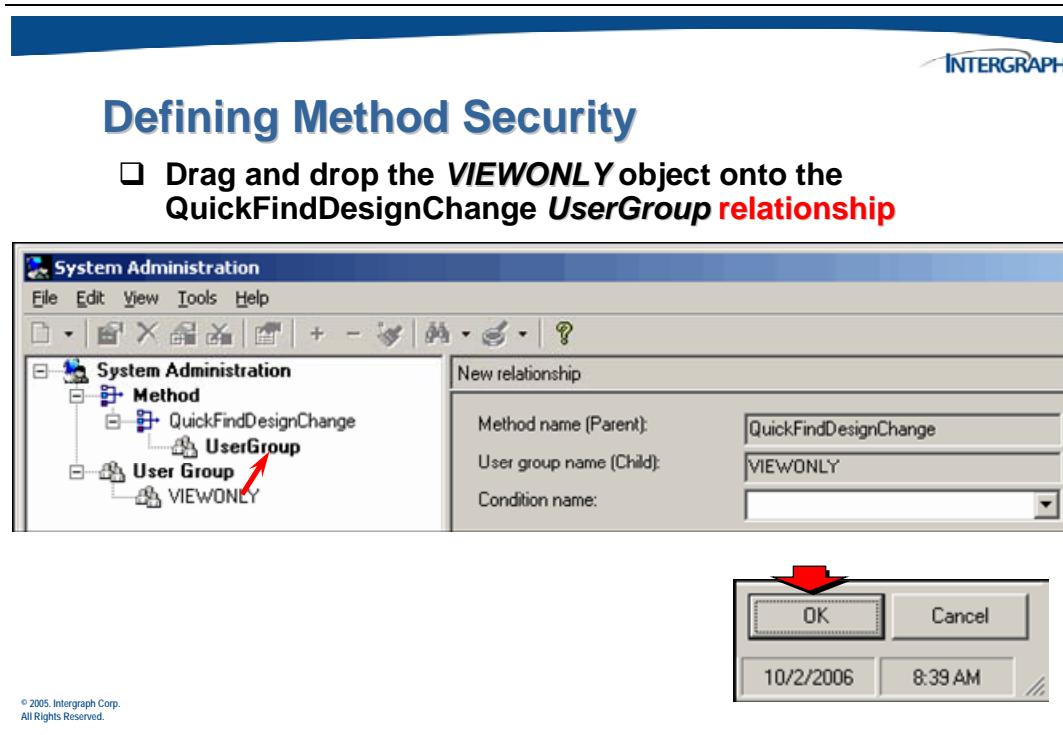
Note that in this example, the *Standalone method* toggle has been enabled.



Remember to set the access security on the new method.



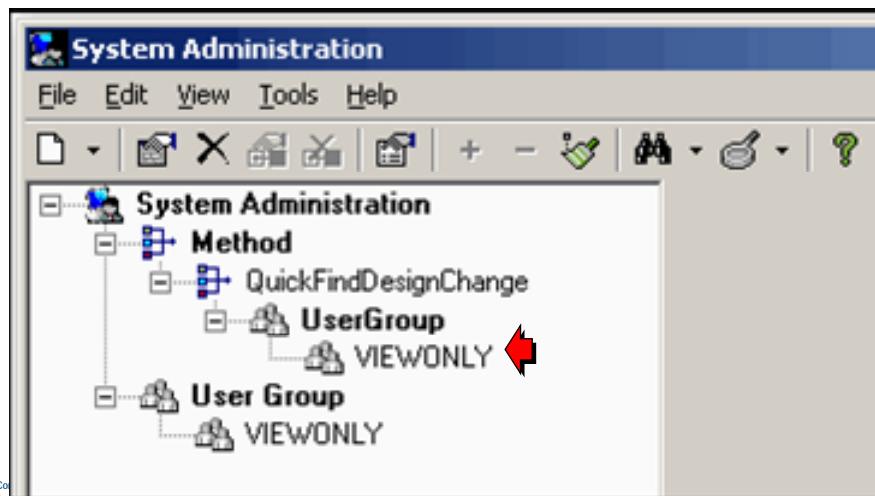
Since this is a search method, VIEWONLY group access can be used for access security.





Defining Method Security

The new method security relationship is displayed in the **Tree View**



5.8.1 Creating New Toolbar Item

Toolbar items (icons) are created and added to the toolbar. Each item contains the action to be performed, the method to be used, and other related properties. A user must be a member of a user group associated with the toolbar in order to gain access to that toolbar item.



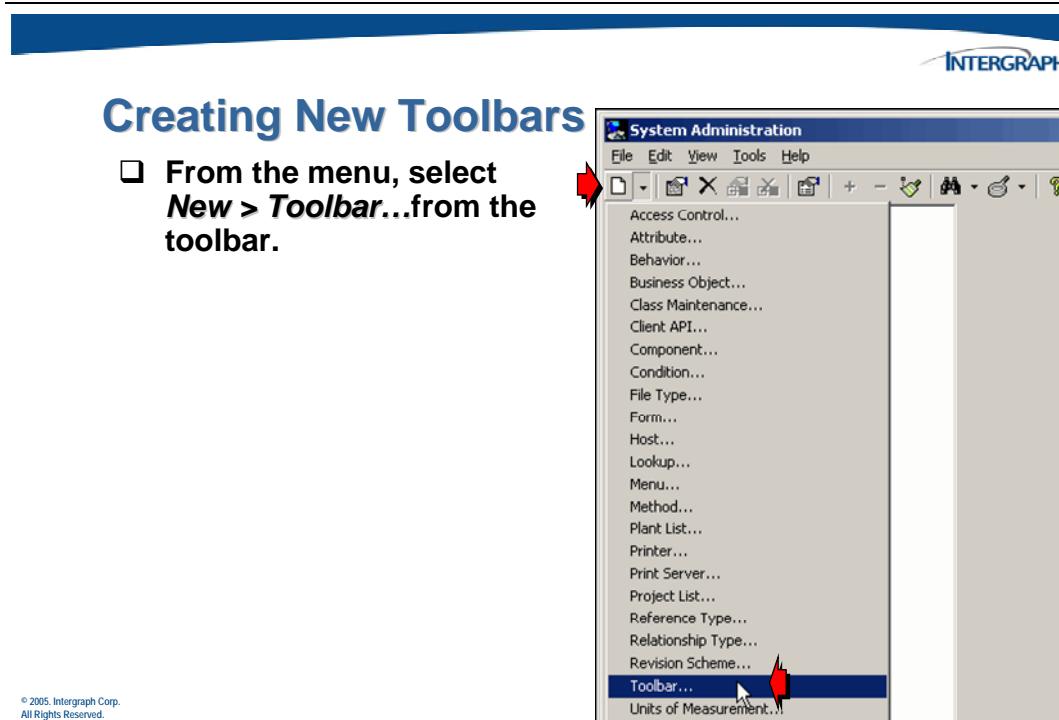
Toolbars

Toolbars are available in the **Web and **Desktop** clients**

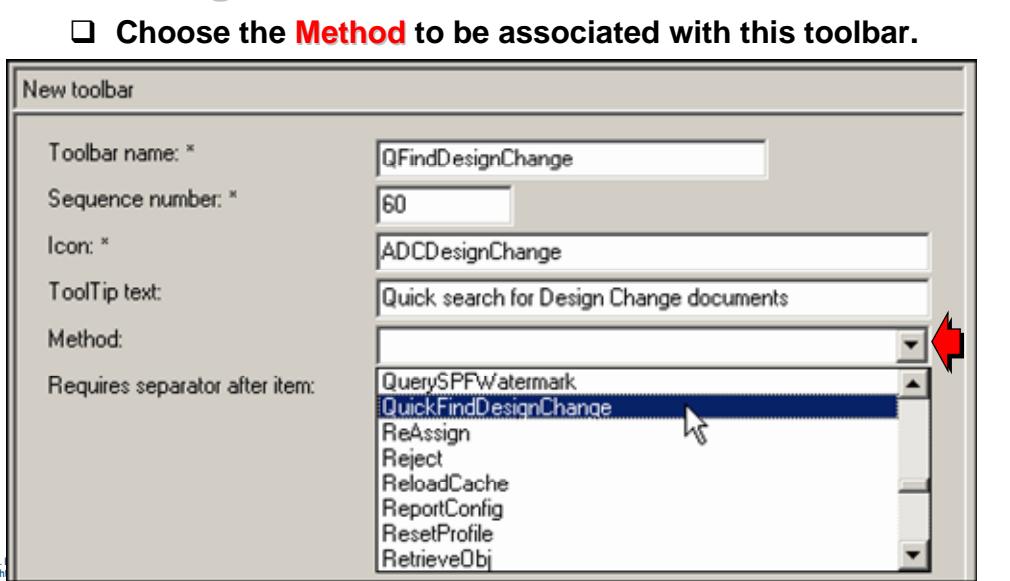
- Toolbar name**
- Sequence number**
- Icon**
- Tool tip text**
- Method (related to Method with stand-alone flag)**
- Requires separator after item indicator**

User access is controlled by associating user group to method that is attached to tool bar item

Create a new toolbar that can be used from the client interfaces. This new toolbar will use the new standalone method.



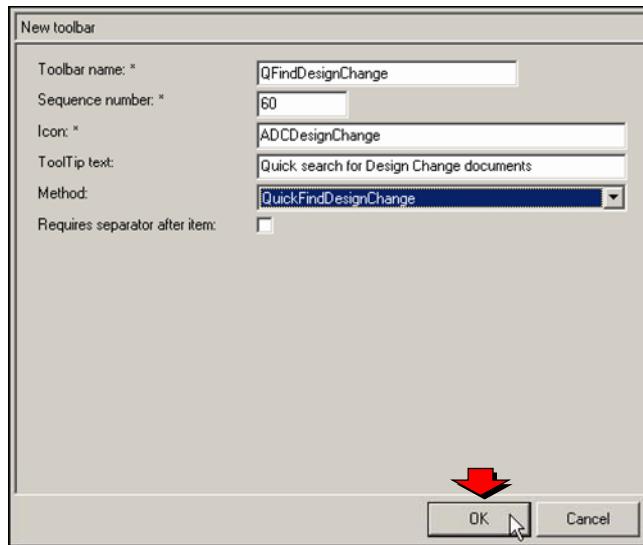
The *New toolbar* form will display in the right pane.





Creating New Toolbars

- When the necessary values have been entered, click **OK**.

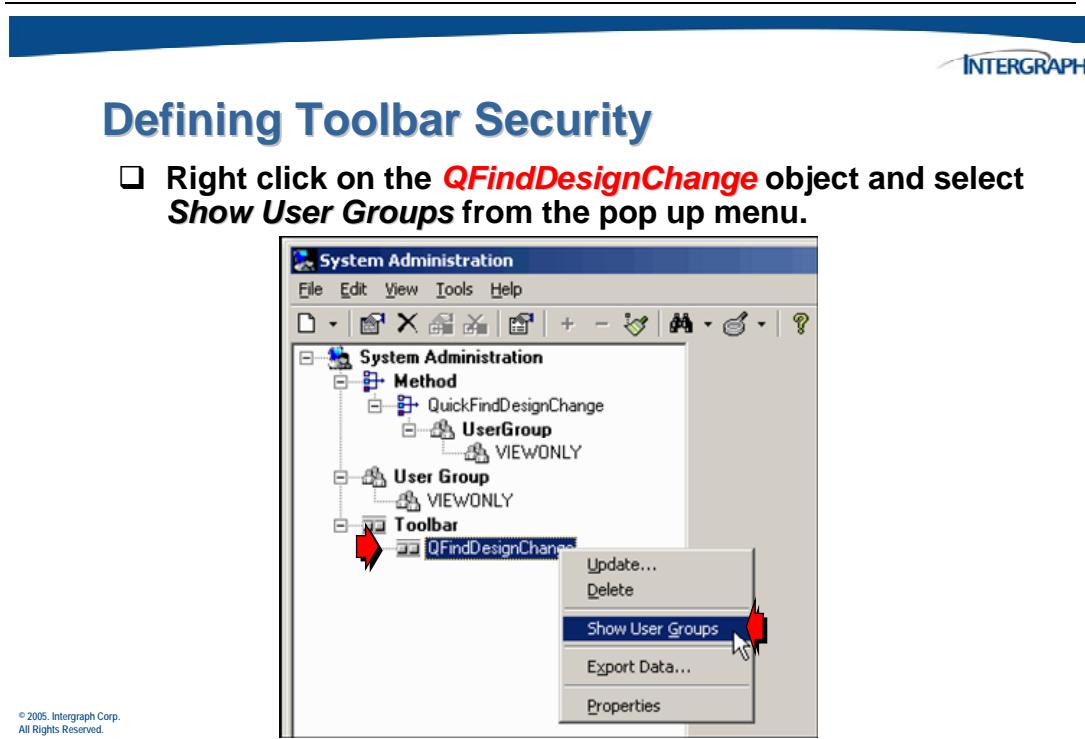


© 2005, Intergraph Corp.
All Rights Reserved.

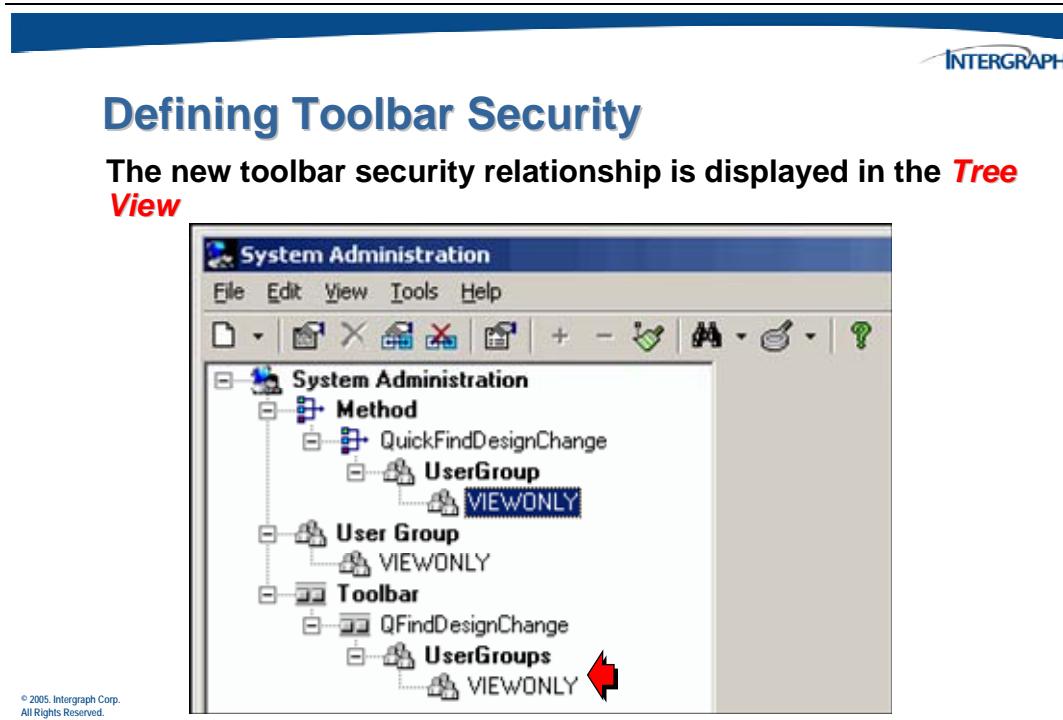
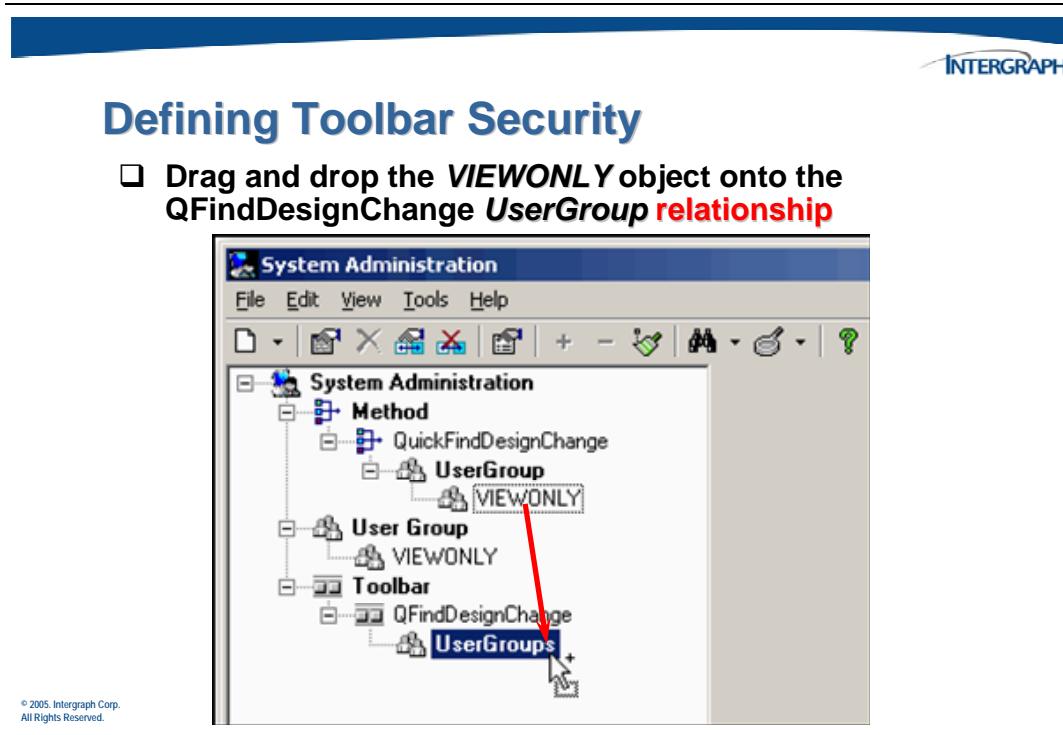
5.8.2 Adding Toolbar Security

Once the toolbar has been created, the necessary access security must be added. Perform a search to verify that the new toolbar has been defined successfully. This is not necessary but it will allow you to see how your new toolbar fits in with the existing toolbars.

Display the *UserGroup* relation for the toolbar.



Next, add the user group relationship to the new toolbar item.

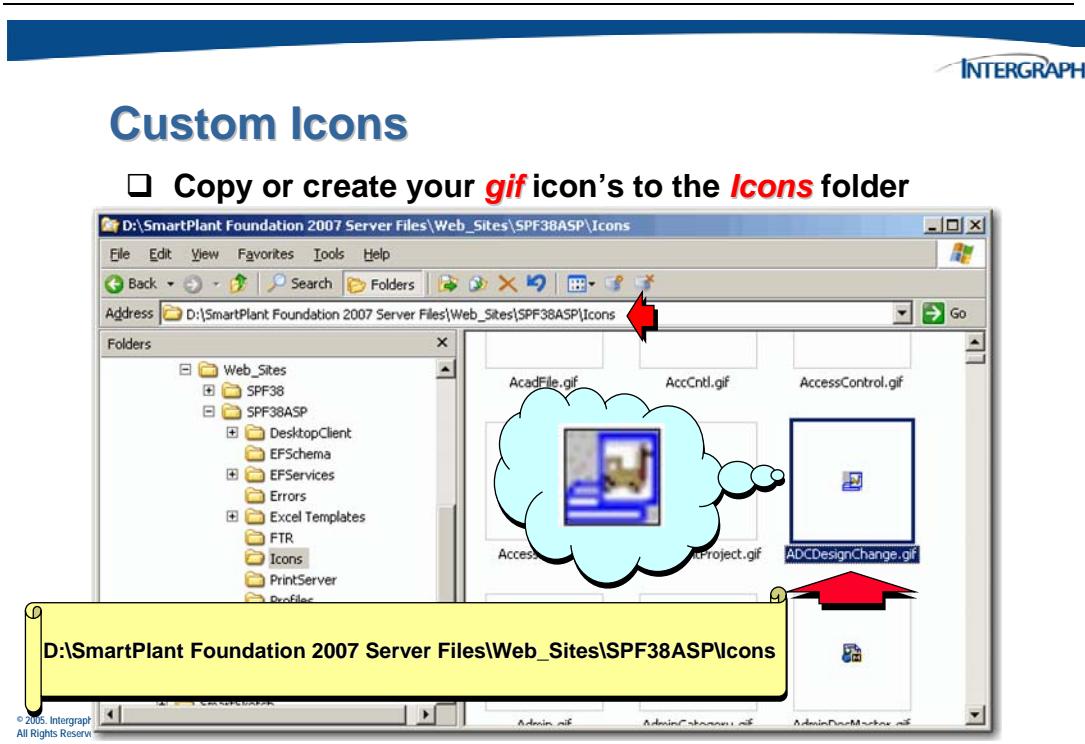


5.8.3 Defining Custom Icons

Icons are used on the toolbars and in the view windows to represent objects. Anytime a custom object or toolbar item is created, a corresponding icon must be created for that item.

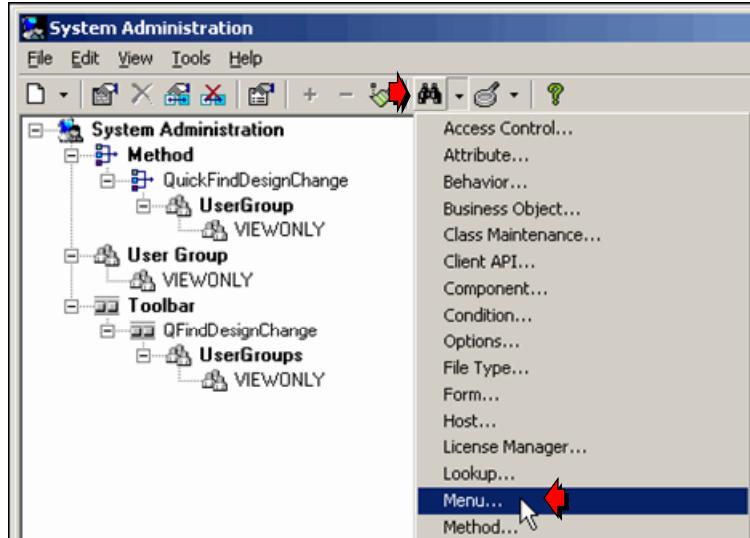
Until a custom icon is created, the default icon used is a ? as noted when the methods were tested earlier. A program, which creates GIF files, can be used to create custom icons.

Icons that are **gif's** are used for the Desktop client. Once these gif icons are created, they need to be placed in the **Icons** folder, within the SPF product location (SmartPlant Foundation 2007 Server Files\Web_Sites\SPFASP).



Defining Query Menu Items

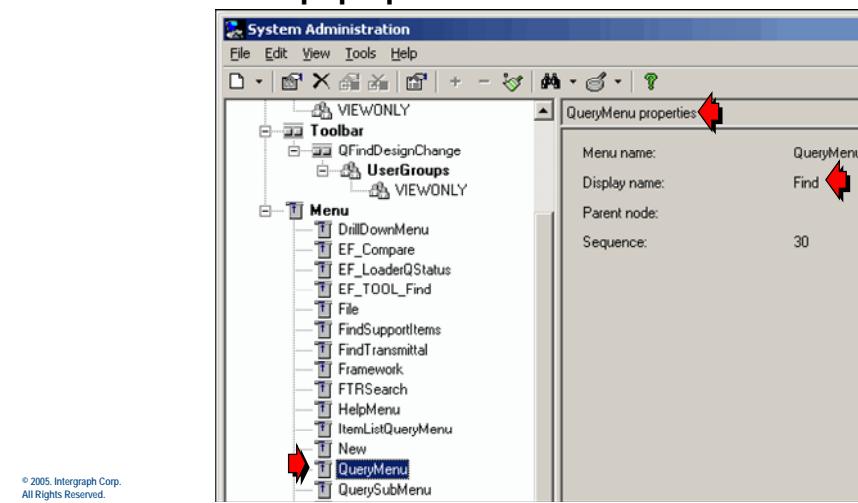
- Use the **Find > Menu...** tool to locate the existing **Menus**.



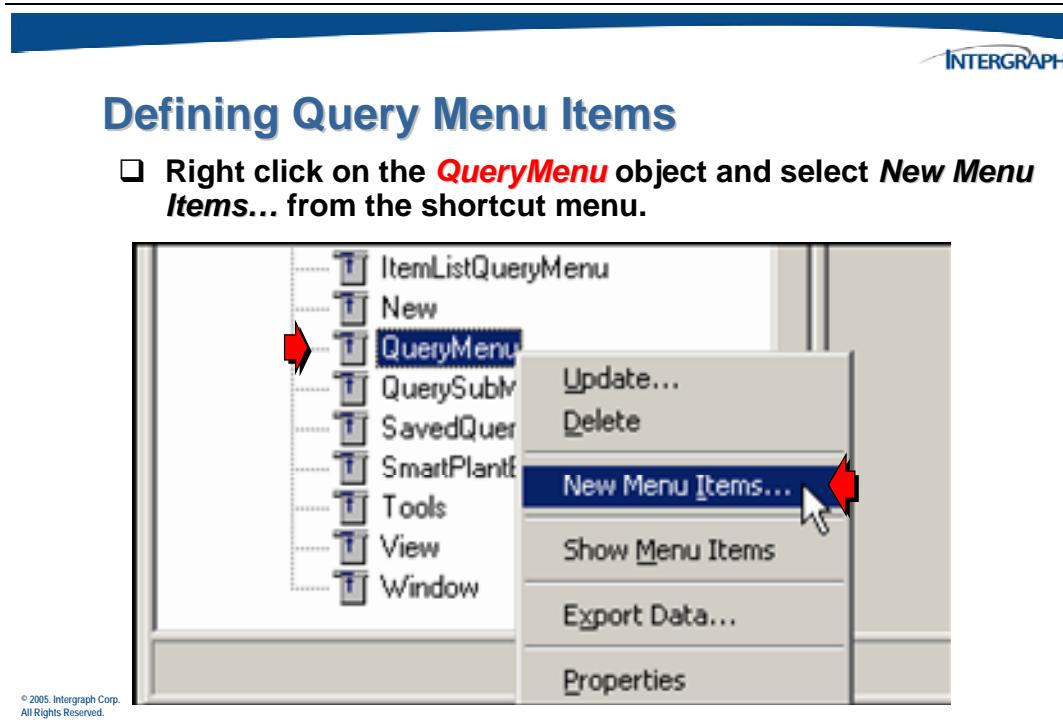
The client Find command is actually called **QueryMenu**.

Defining Query Menu Items

- Right click on the **QueryMenu** object and select **Properties** from the pop up menu.



Next, add a new **QueryMenu** menu item.



The *New menu items* form will display.

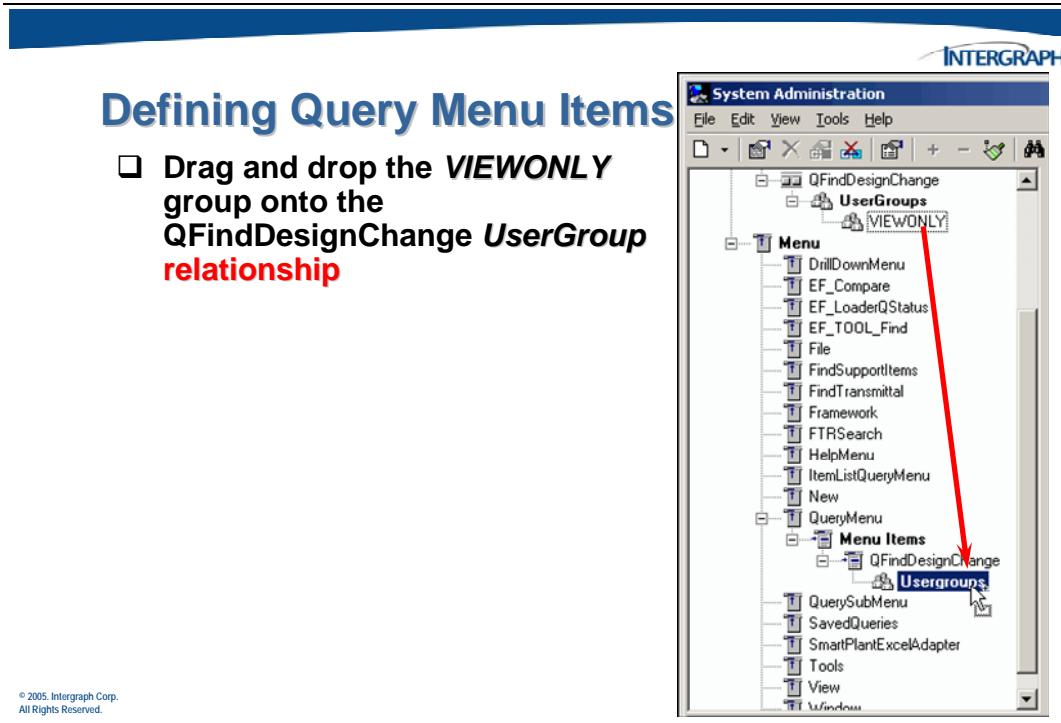
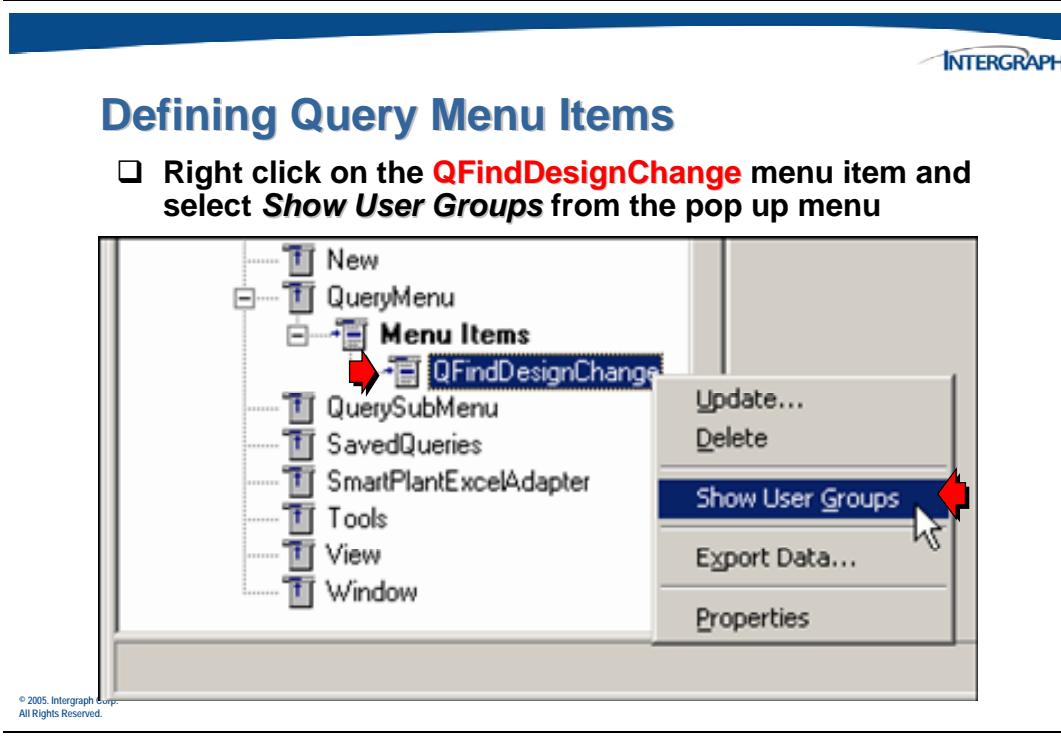
Defining Query Menu Items

- When the necessary values have been entered, click **OK**.

The screenshot shows the 'New menu items' dialog box. It has a title bar 'New menu items'. Inside, there are several input fields:

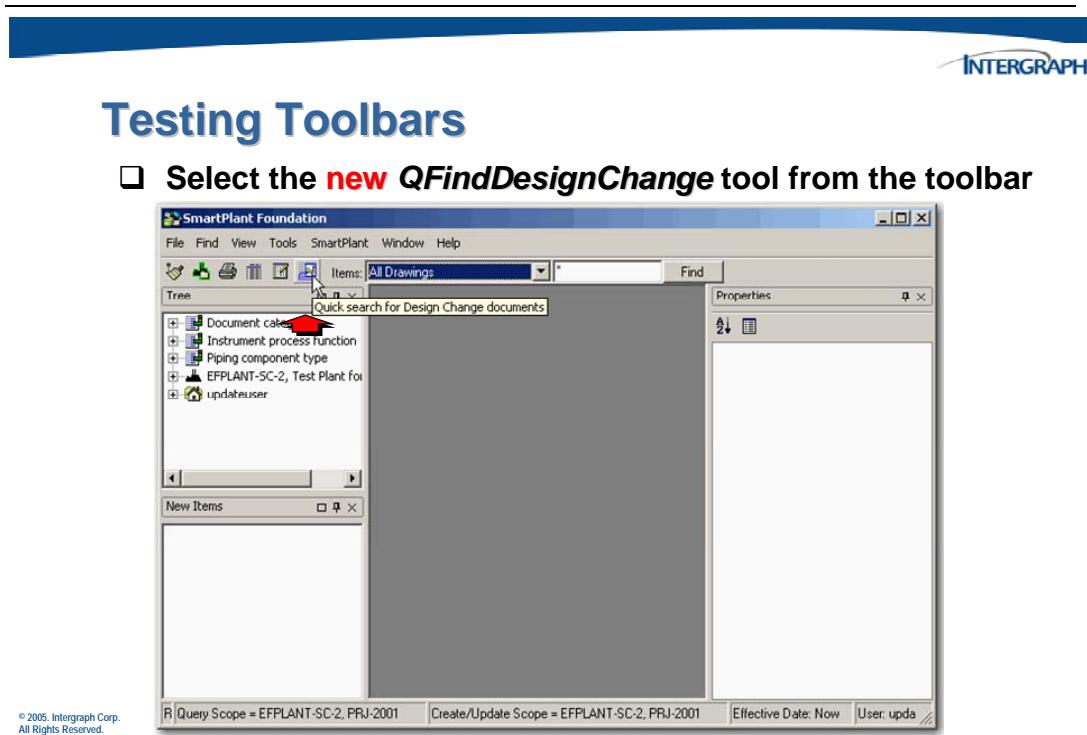
- Menu name (Parent):
- Menu item name (Child):
- Display as:
- Icon:
- Method name:
- Separator required indicator:
- Sequence number:

To add the security access, create a relationship between a *user group* and the *menu item*.

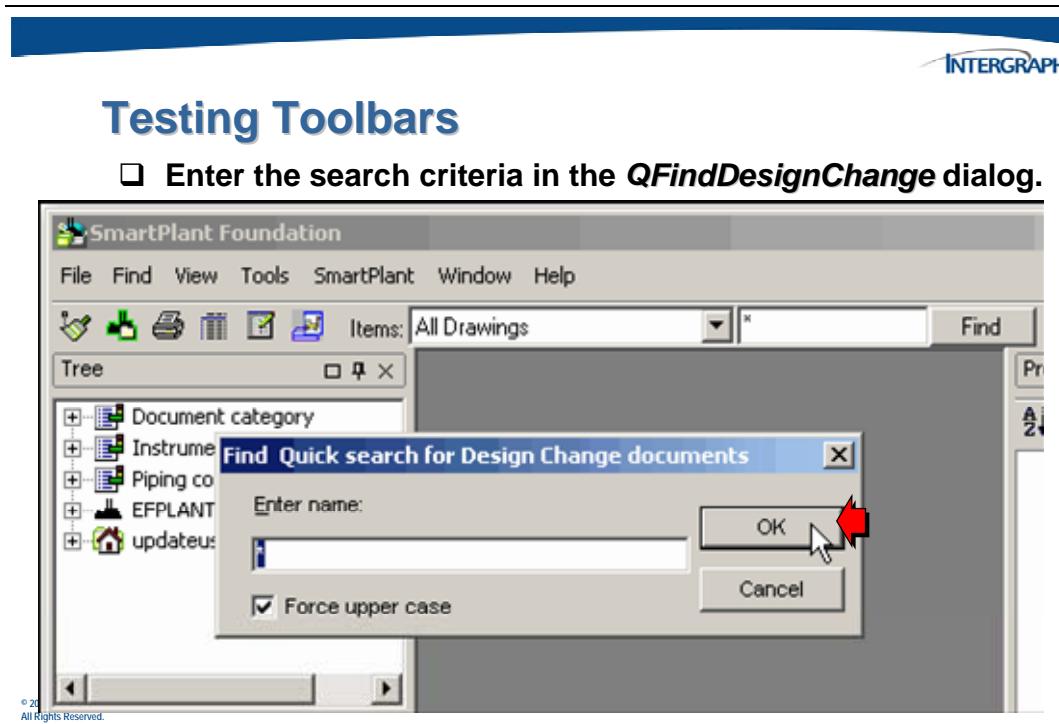


5.9 Testing Toolbars

Now that toolbars have been created, access security defined and custom icons created, they can be tested for functionality. To do this, again exit from the System Administration utility. Start the client interface as an UPDATE user so that the toolbar can be tested.



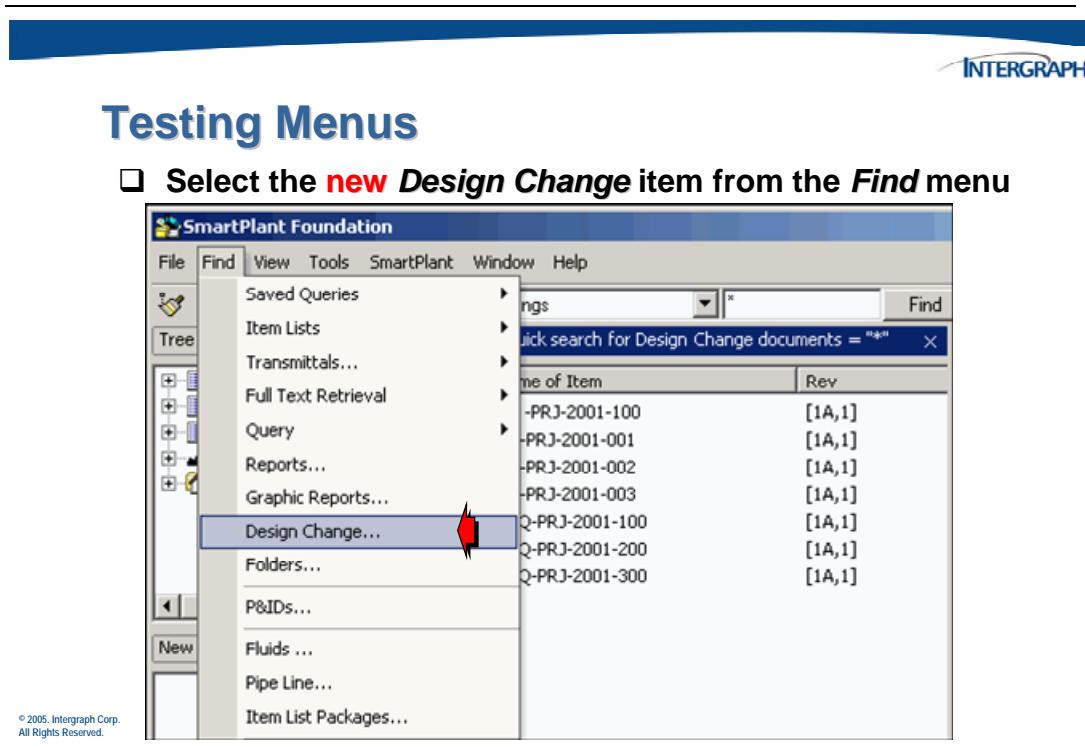
The *Find* dialog will display.



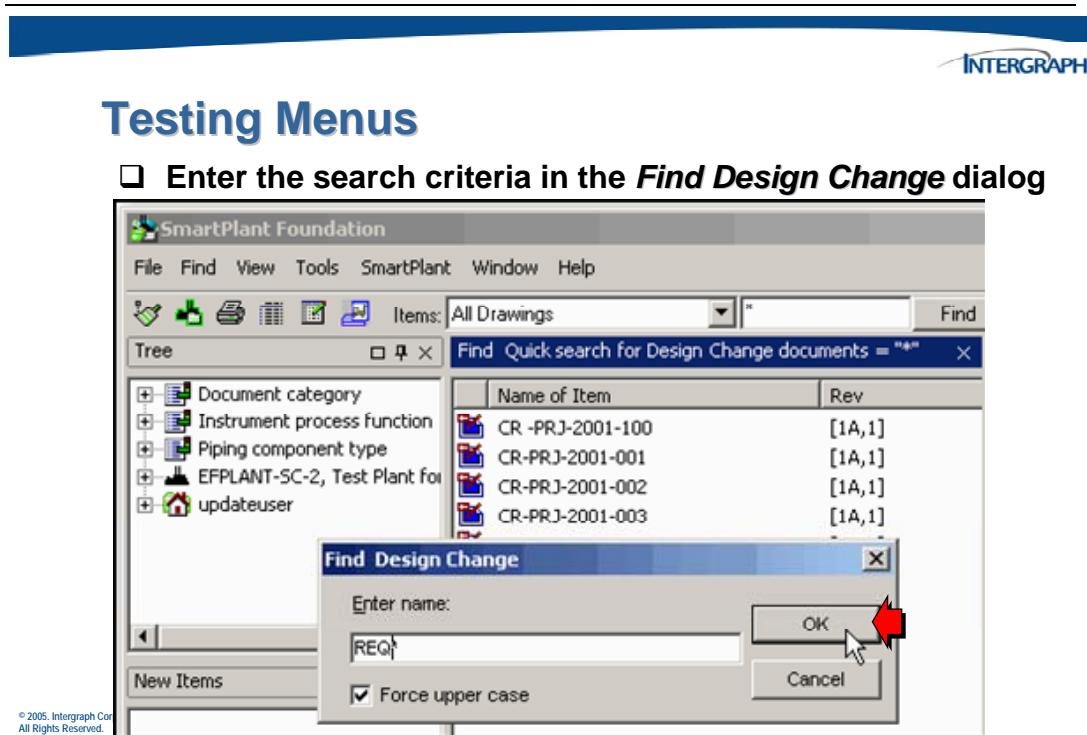
Once the search criteria is entered, click **OK** to test the *Find*.

5.10 Testing Menus

Once the new menu items have been created and access security defined they can be tested. To do this, again exit from the System Administration utility. Start the client interface as an UPDATE user so that the menu item can be tested.



The *Find Design Change* dialog will display.



Once the search criteria is entered, click **OK** to test the *Find*.

5.11 Units of Measurement

Units of Measurement (UOMs) in SPF are set up to identify the units in which the particular property is recorded. Administrators can create a custom UOM (Unit of Measurement) if needed.



Units of Measurement

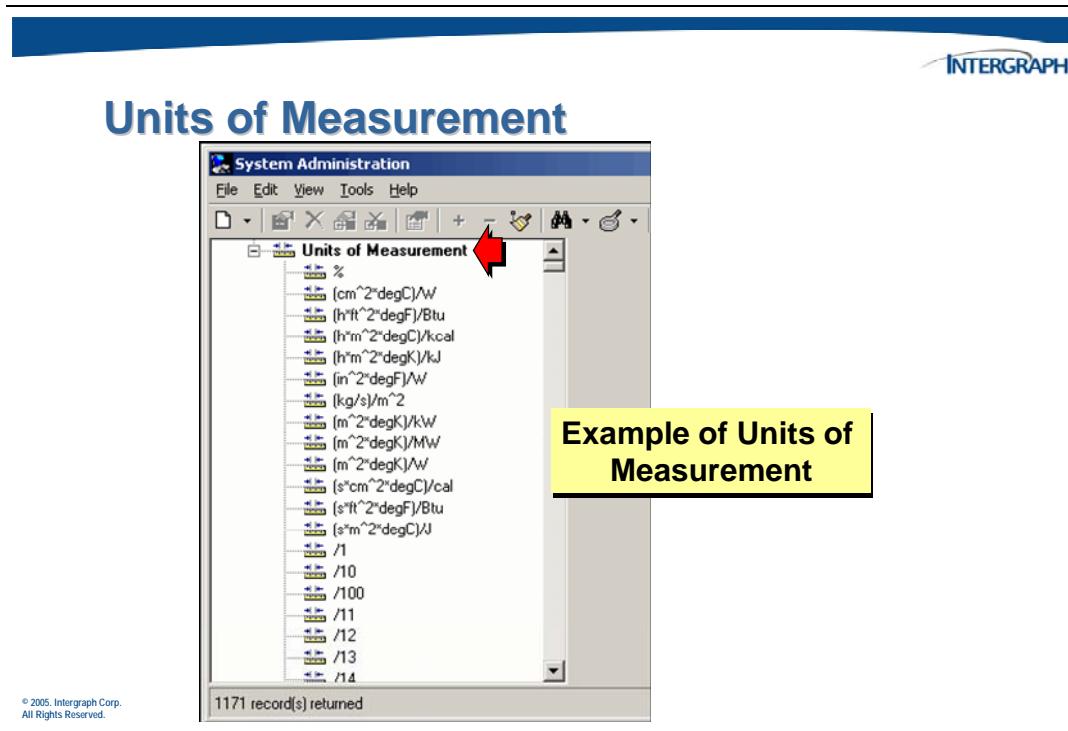
***Units of Measurement* is used with the property type *Quantity*.**

Value is always stored in standard Units of Measurement and converted in searches and updates.

- Units of measurement name**
- Units of measurement description**

Note: To ensure the proper execution of the conversion factors for Units of Measurement, new UOM's will need to be created in the Schema Editor and then loaded into the SPF Admin database.

Perform a search to find and display the available **Units of Measurement** in the *System Administration* interface.



5.11.1 UOM Sets

UOM sets are groups of UOMs that measure the same property, but in different units. A UOM set can be attached to a data form display item to provide a picklist of valid UOMs.



Units of Measurement Sets

Units of Measurement values can be grouped to logical sets

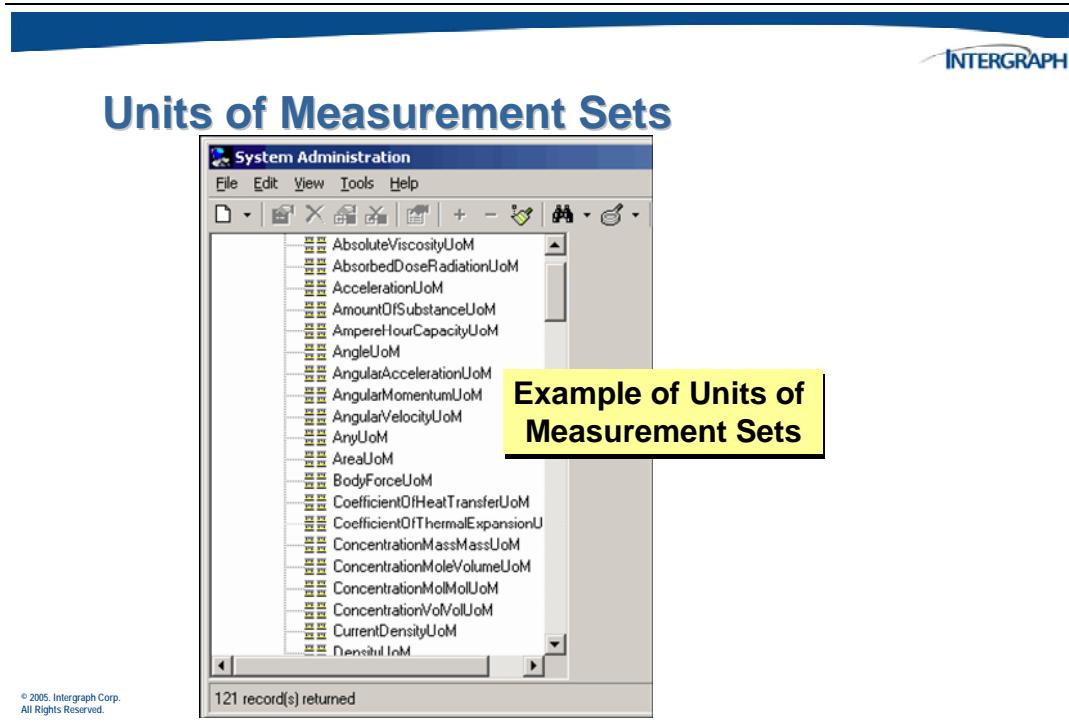
The set name is used for data items to limit number of available values for selection

Units of Measure values are dragged and dropped onto a set

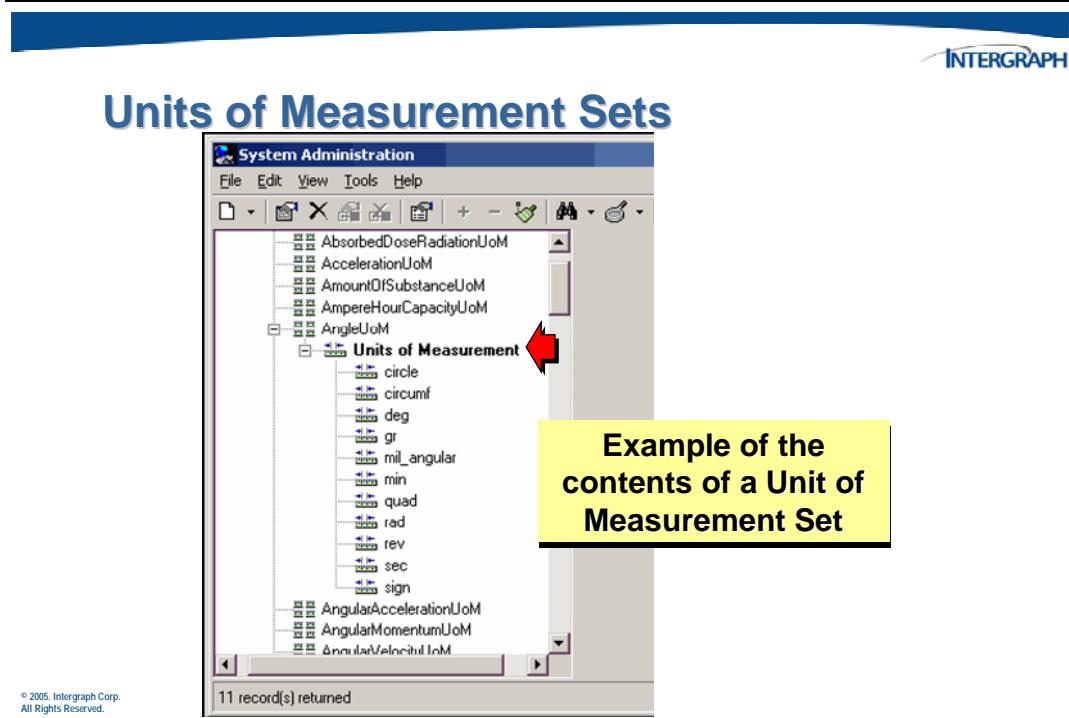
- Units of measurement set name

UOM Sets define a list of UOM's for a short list of UOM's to be provided in a Form instead of the full list of UOM's.

Perform a search to find and display the available **Units of Measurement Sets** in the *System Administration* interface.



Expand the tree to see the contents of one of the UOM Sets.



5.12 Revision Schemes

Revision Schemes in SPF are set up to control the revision attributes on an Interface Definition of the Revision behavior. The revision group object is typically a document class and the revision object is its revision.



Revision Schemes

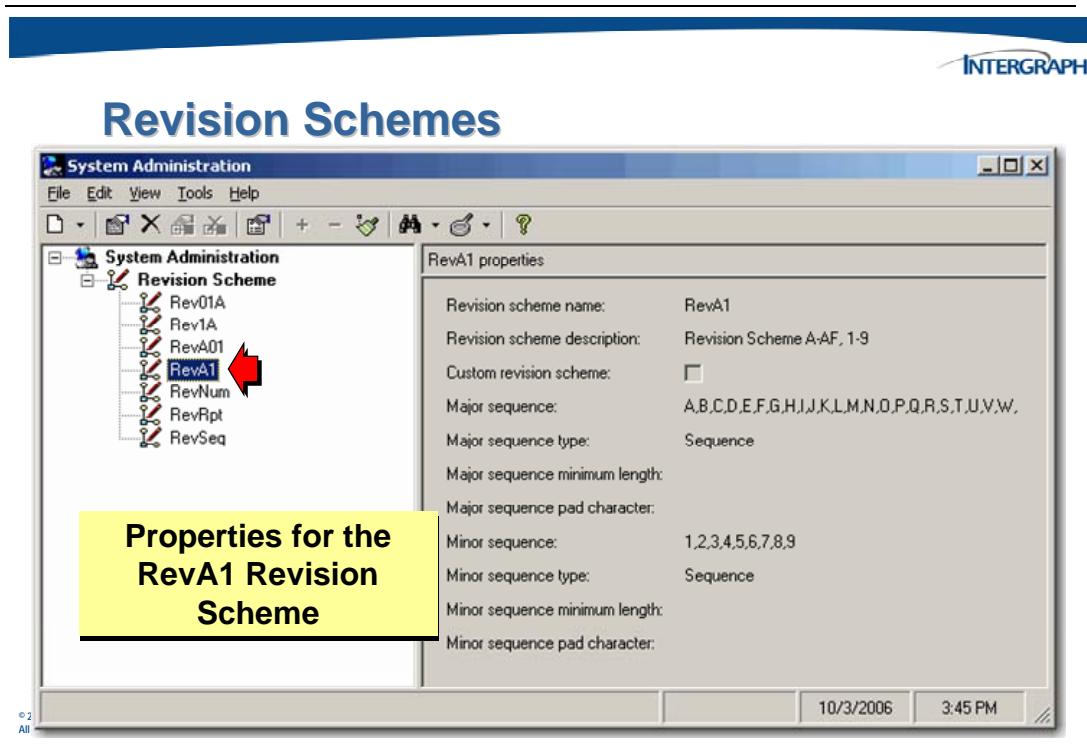
Revision scheme defines how the next revision number will be formed

It is valid only for business objects with behaviour Revision

- Revision scheme name**
- Revision scheme description**
- Custom revision scheme indicator (future)**
- Major sequence**
- Minor sequence**

Perform a search to find and display the available **Revision Schemes** in the *System Administration* interface.

The revision attributes include the *Major* and *Minor* revisions, which are combined to create a sequence. The *Revision Scheme* defines a valid sequence for these attributes.



This will specify a *Revision Scheme* list for Document Revisions to follow, i.e. A, B, C, 1A, 2A, 3A, A1, B2, C3,

5.13 Activity 3 – Configuring Toolbars and Menus

The objective of this activity is to create a toolbar and menu item to allow a user to find design changes. You will also use the drag and drop method to user security to the new menu items and toolbar.

If you are not currently logged into your machine:

1. Log on to your operating system (if not already logged in):
spfuser with no password
2. Start the *SysAdmin Utility* by selecting *Start: All Programs > Intergraph SmartPlant Foundation > System Administration*.
3. When the *Login* dialog window appears, use the *User name* **adminuser** with no password and click on **OK**.
4. Create a new method to use with your toolbar and menu item. From the System Administration menu, select the *New > Method...* tool. In the *New method* form, enter the following:

<input type="checkbox"/> Method Name -	<i>QuickFindDesignChange</i>
<input type="checkbox"/> Display Name -	<i>Design Change...</i>
<input type="checkbox"/> Associated client API -	<i>QFindEFObj</i>
<input type="checkbox"/> BObject Name -	<i>DocMaster</i>
<input type="checkbox"/> Interface Name -	<i>IADCDesignChange</i>
<input type="checkbox"/> Rev Status	<i>LATEST</i>
<input type="checkbox"/> Component -	<i>EF</i>
<input type="checkbox"/> Object state -	leave blank
<input type="checkbox"/> Stand-alone method -	<i>toggle enabled</i>
<input type="checkbox"/> Available on Desktop client -	<i>toggle enabled</i>
<input type="checkbox"/> Available on Web client -	<i>toggle enabled</i>
<input type="checkbox"/> Cascade name -	leave blank
<input type="checkbox"/> Condition name -	leave blank

- | | |
|--|-------------|
| <input type="checkbox"/> Method Access Control - | <i>0</i> |
| <input type="checkbox"/> Email message - | leave blank |

Click **OK** to create the new method.

5. Create a new toolbar to use with the method created in step 4 above. From the System Administration menu, select the *New > Toolbar...* tool. In the *New toolbar* form, enter the following:

- | | |
|--|------------------------------|
| <input type="checkbox"/> Toolbar name - | <i>QFindDesignChange</i> |
| <input type="checkbox"/> Sequence number - | <i>60</i> |
| <input type="checkbox"/> Icon - | <i>ADCDesignChange</i> |
| <input type="checkbox"/> ToolTip Text - | <i>Design Change</i> |
| <input type="checkbox"/> Method - | <i>QuickFindDesignChange</i> |
| <input type="checkbox"/> Requires separator after item - | leave blank |

Click **OK** to create the new toolbar.

6. You are now going to add a user group to the toolbar relationships to allow access security.
- Right click on the new toolbar name and select *Show User Groups* to view the relationship.
 - Drag and drop the user group object to the toolbar relationship object.
 - *UPDATE* to *QFindDesignChange*

Verify in the *Tree View* pane that relations between the user groups and the toolbar have been created.

7. Define a new custom icon to use with your toolbar. You can copy an existing icon from the SPF product directory.

Location to copy icons from:_____

Remember, **gif's** get copied to the \SmartPlant Foundation 2007 Server Files\Web_Sites\<your site>\Icons folder.

8. Test the new toolbar that has been created.

9. You are now going to add a menu item to the menu node to allow users search functionality.
 - Perform a search for the existing *Menu* node objects
 - Right click on the **QueryMenu** object name and select **New Menu Items...**
 - In the *New menu items* form, enter the following:
 - Menu name (Parent) - **QueryMenu**
 - Menu item name (Child) - **QFindDesignChange**
 - Display as - **Design Change...**
 - Icon - leave blank
 - Method name - **QuickFindDesignChange**
 - Separator required indicator - leave blank
 - Sequence number - **90**

Click **OK** to create the new menu item.

10. You are now going to add a user group to the menu item relationships to allow them user access security.
 - Right click on the **QFindDesignChange** menu item name and select **Show User Groups** to view the relationship.
 - Drag and drop the user group objects to the menu item relationship object.
 - *VIEWONLY* to **QFindDesignChange**
 - Verify in the *Tree View* pane that relations between the user groups and the menu item have been created.
11. Test the new menu item that has been created.
12. Add a new method and menu item to perform a *Search* for Design Change objects. This will be different from the quick find that was shown in this chapter. This command will be found under the *QuerySubmenu* item in the menu structure. Use the document object as an example.

QueryDocument properties	
Method name:	QueryDocument
Display name:	Document Revisions
Associated client API:	QueryEFObj
EF BObject Name	DocMaster
EF ClassDef Name	Document
Display As	Documents
Rev Status	LATEST
Query Filter (Condition Name)	
Column Display (BObject Name)	
Component:	EF
Object state:	
Stand-alone method:	<input checked="" type="checkbox"/>
Available on Desktop client:	<input checked="" type="checkbox"/>
Available on Web client:	<input checked="" type="checkbox"/>
Cascade name:	
Condition name:	
Method access control (0,1):	0
E-mail message:	
Multi-select API:	

QueryDocument properties	
Menu name (Parent):	QuerySubMenu
Menu item name (Child):	QueryDocument
Display as:	Documents...
Icon:	
Method name:	QueryDocument
Separator required indicator:	<input type="checkbox"/>
Sequence number:	50

When you finish this activity, you may take a short break until everyone has finished.

Possible solutions for activity step 12.

New method

Method name: *	QueryDesignChange
Display name: *	Design Changes
Associated client API: *	QueryEFObj
EF BObject Name	DocMaster
EF ClassDef Name	ADCDesignChange
Display As	Design Changes
Rev Status	LATEST
Query Filter (Condition Name)	
Column Display (BObject Name)	
Component:	EF
Object state:	
Stand-alone method:	<input checked="" type="checkbox"/>

New menu items

Menu name (Parent): *	QuerySubMenu
Menu item name (Child): *	QueryDesignChange
Display as: *	Design Changes...
Icon:	
Method name:	QueryDesignChange
Separator required indicator:	<input type="checkbox"/>
Sequence number:	49

6

CHAPTER

SmartPlant Change Management

6. Introduction to SPF Change Management Administration

SPF Change Management Administration consists of two key objects, which are workflows and user groups. SPF workflows perform a series of steps, which are used to achieve a process. A workflow template defines steps, user groups, statuses, dependencies, and check lists. Each step of the workflow performs a specific task and these steps must be created before defining a workflow template. Relationships are created between the steps and the workflow to ensure that each task within the workflow is performed correctly.



Introduction to SPF Change Management

The Change Management Administration module is used for workflow configuration

Workflows are built by using:

- Steps**
- Statuses**
- Checklists**
- Users**
- User groups**

© 2005, Intergraph Corp.
All Rights Reserved.

Workflows contain the steps necessary to achieve a change management process. One of the main benefits of a workflow is that after a change object is attached to a workflow, each step completed updates the status on the change object. Thus, you can view how the change is progressing in its associated workflow.

6.1 User Groups and Change Management

User groups contain a list of users based upon their particular working discipline or department. Workflows can be related to specific user groups thus filtering the list of workflows available to particular user groups.



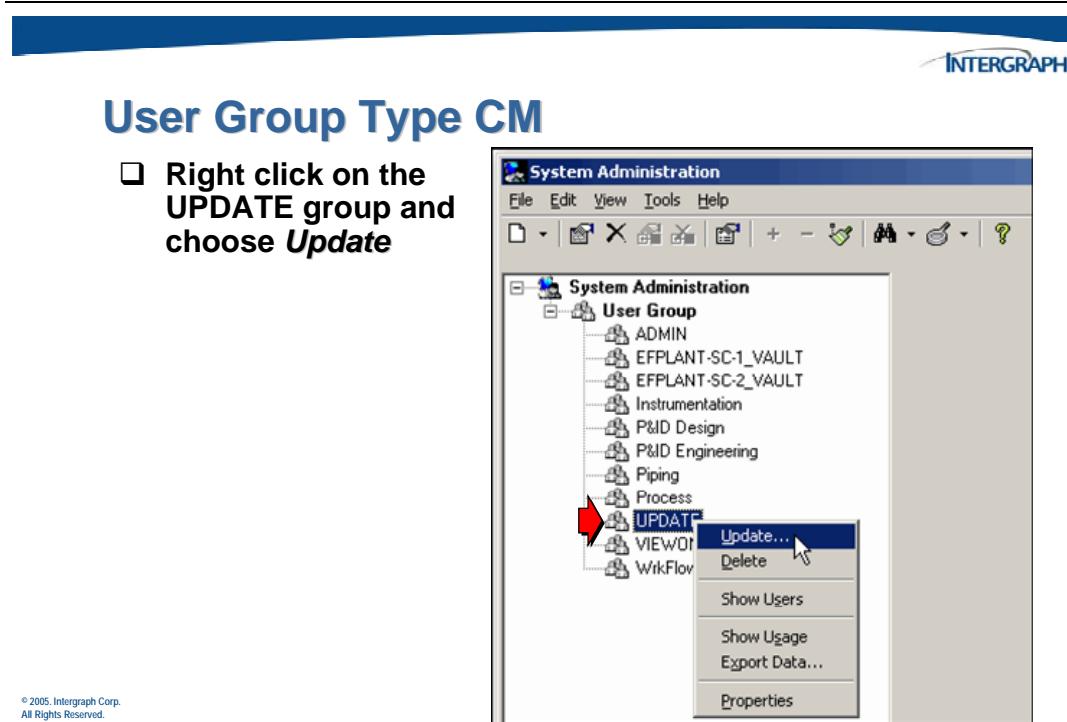
Users and User Groups

Users and user groups are configured by using System Administrator module

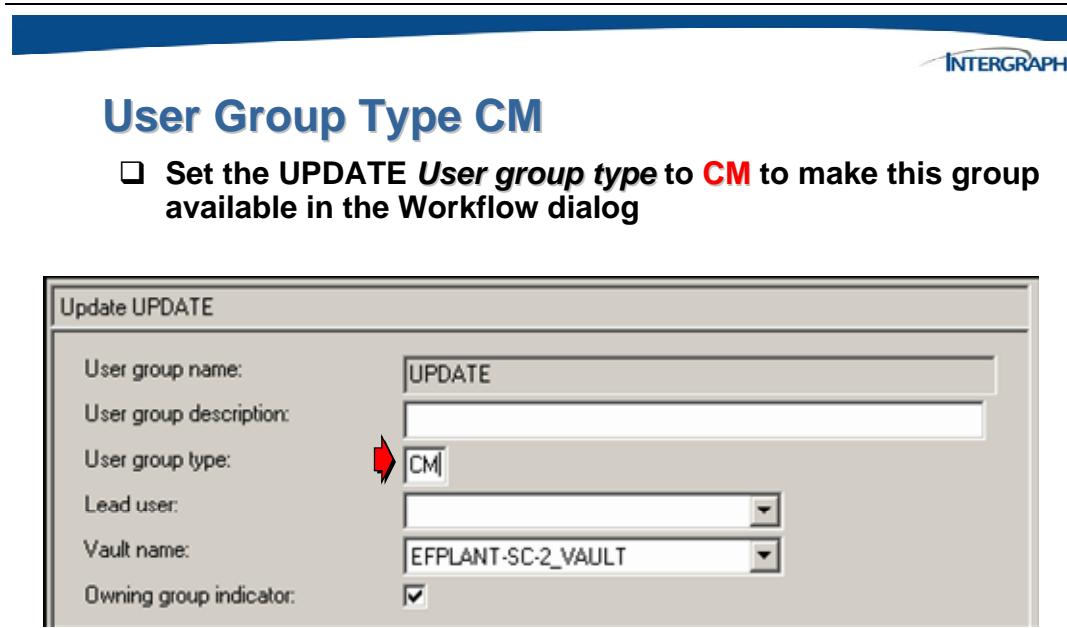
Only user groups with type CM are used in workflows

Users and user groups are created in the **SmartPlant Foundation System Administration** module but assigning users and user groups to workflows can be performed using lists in the **SmartPlant Foundation Change Management Administration** module.

Use the **SmartPlant Foundation System Administration** module to update the *User group type* field.



For a user group to be available from the list when you create a workflow, you must have **CM** in the **User group type** field.





User Group Type CM

- Set the **VIEWONLY User group type** to **CM** to make this group available in the Workflow dialog

The dialog box has the following fields:

- User group name: VIEWONLY
- User group description: (empty)
- User group type: CM (highlighted with a red arrow)
- Lead user: (dropdown menu)
- Vault name: EFPLANT-SC-2_VAULT (dropdown menu)
- Owning group indicator: checked

© 2005, Intergraph Corp.
All Rights Reserved.



Interactive Activity

You can follow along interactively as we build a sample workflow.

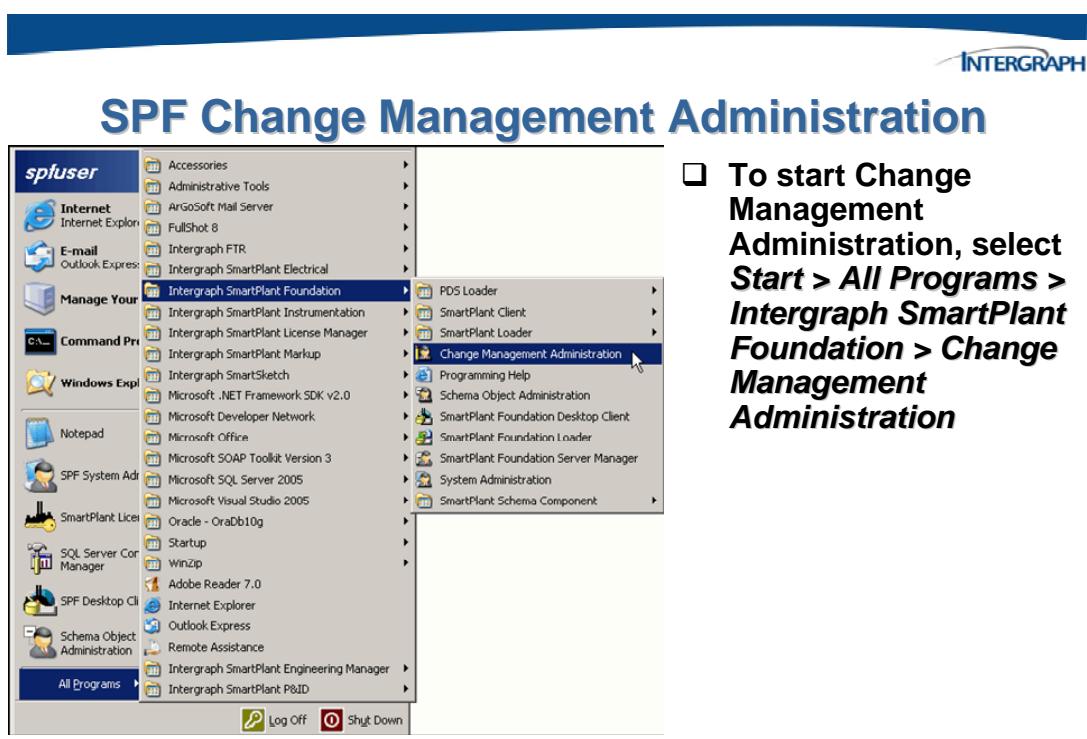


Use the sample in this chapter as an example of building and using a workflow.

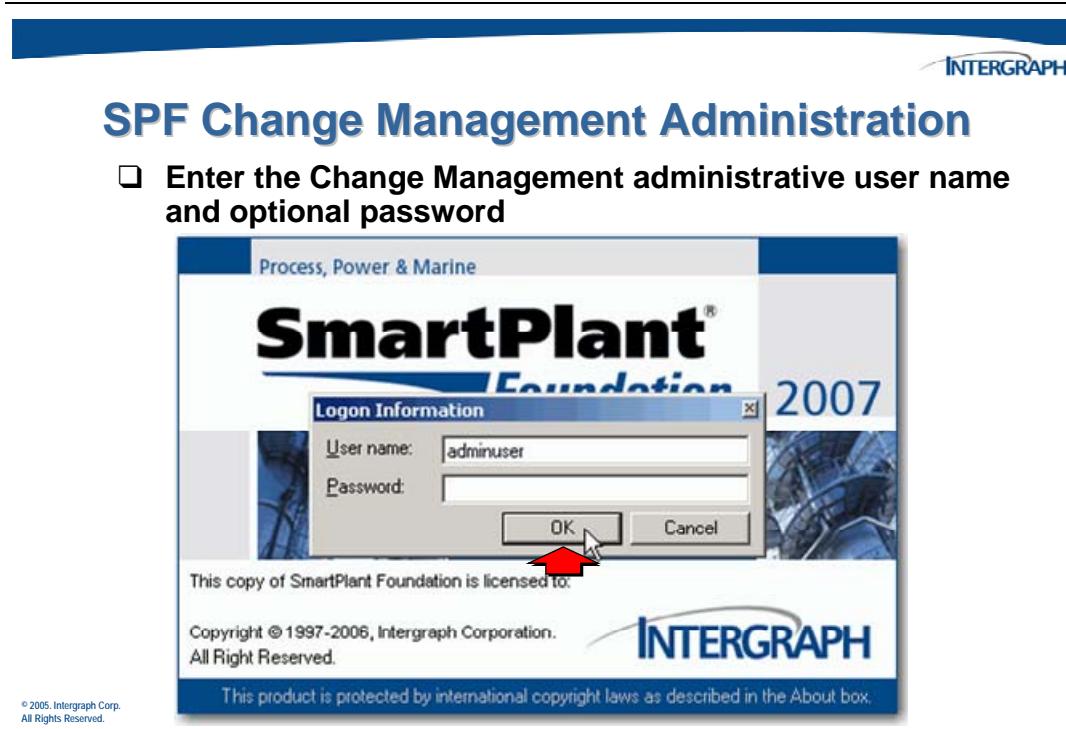
© 2005, Intergraph Corp.
All Rights Reserved.

6.2 Creating Workflows

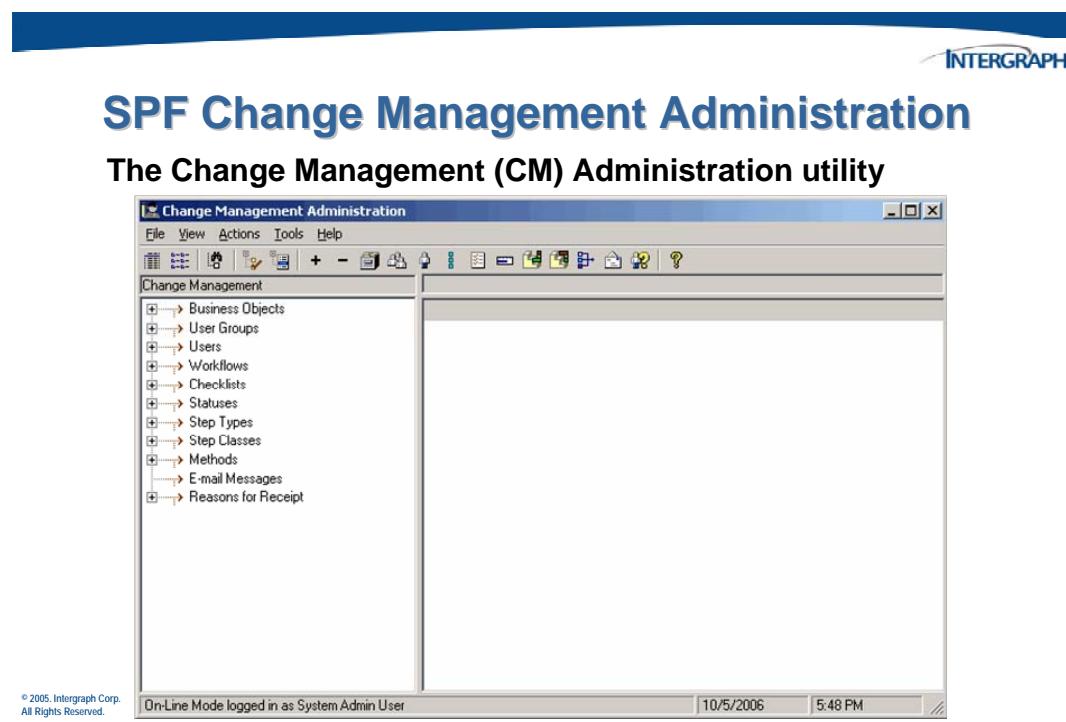
Using the **SmartPlant Foundation Change Management Administration** module, you can create, view, update, copy, and delete a workflow.



The *Login Information* dialog will display.



The *Change Management Administration* utility window will be displayed.



6.2.1 Step Types

Step Types define the behavior of steps and step classifications. Each step is assigned to a step classification, which contains a step type definition. The default step types delivered with the Change Management Administration software are:



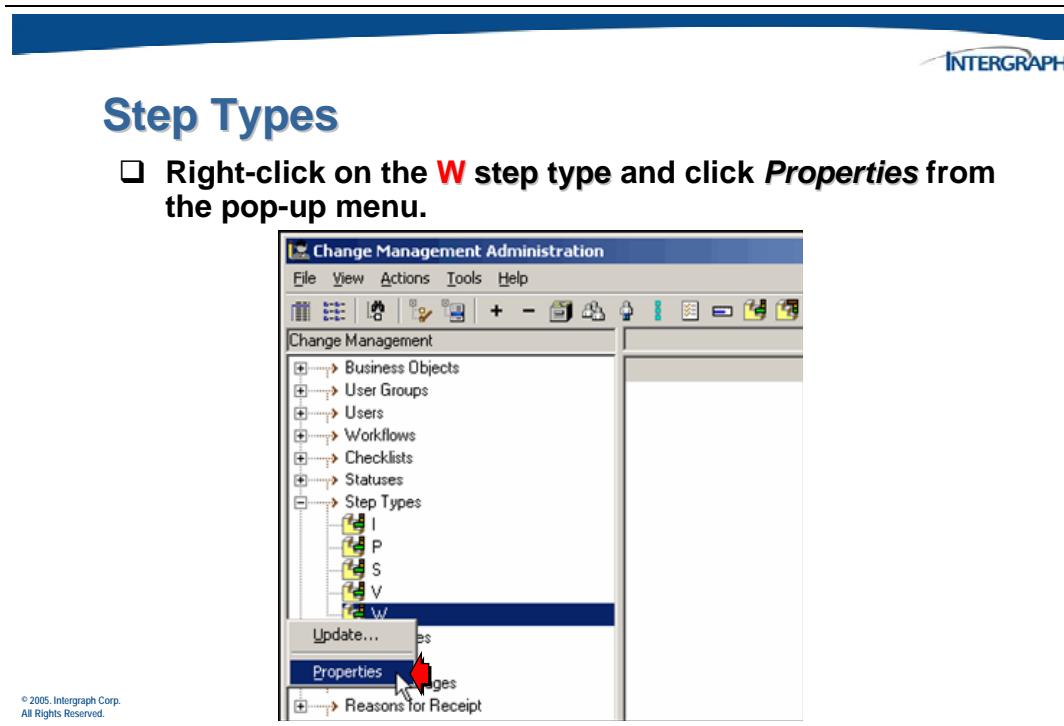
Step Types

Step Types define the behavior of steps and step classifications

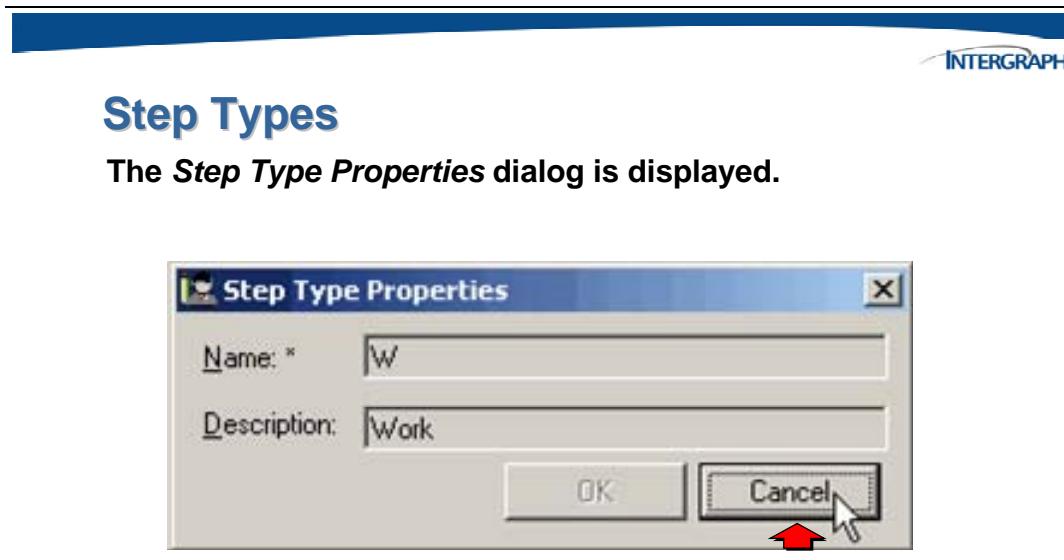
The default step types are:

<u>Step</u>	<u>Type</u>	<u>Description Purpose</u>
I	Information	This step contains no subsequent steps.
W	Working	This step assigns work to the steps recipients.
P	Process	This step is a non-interactive step.
V	Voting	This step is for processing the results of parallel steps.
S	Sign Off	This step is for document sign off.

You can also view information about the step type. To view step type information, right-click the step type, and choose **Properties**.

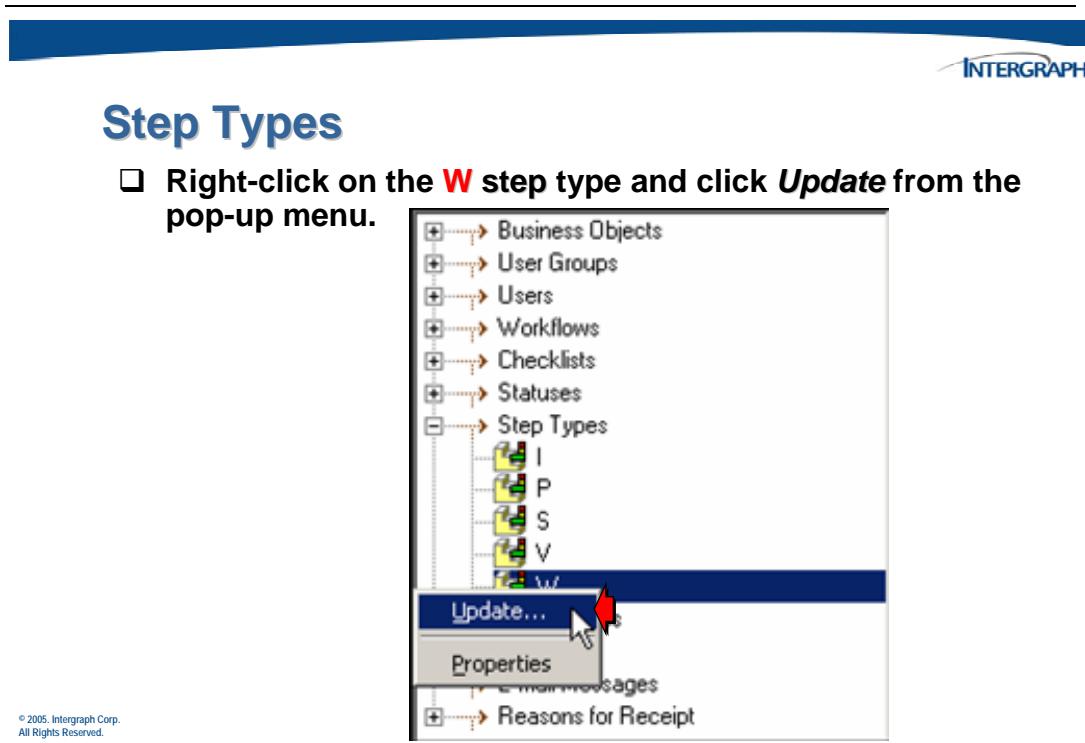


Selecting the **Properties** command displays the *Step Type Properties* dialog box for the selected step type as shown below.



6.2.2 Update a Step Type

A **Step Type** object can be updated and the description field changed. To start, double-click the *Step Types* entry in the tree view to expand its associated list.

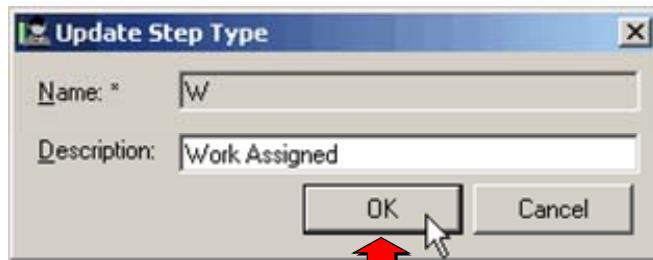


In the example above, the **W** step type is selected for update.



Step Types

The *Update Step Type* dialog is displayed.



© 2005, Intergraph Corp.
All Rights Reserved.

In the *Update Step Type* dialog box, change the **Description** field if necessary and click **OK**.

6.2.3 Step Classes

Step Classes are used in conjunction with step types to define the behavior and appearance of steps. Methods are added to each step class to define commands that can be performed when a particular step is located in the user's inbox. The default step classes delivered with the Change Management Administration software are:



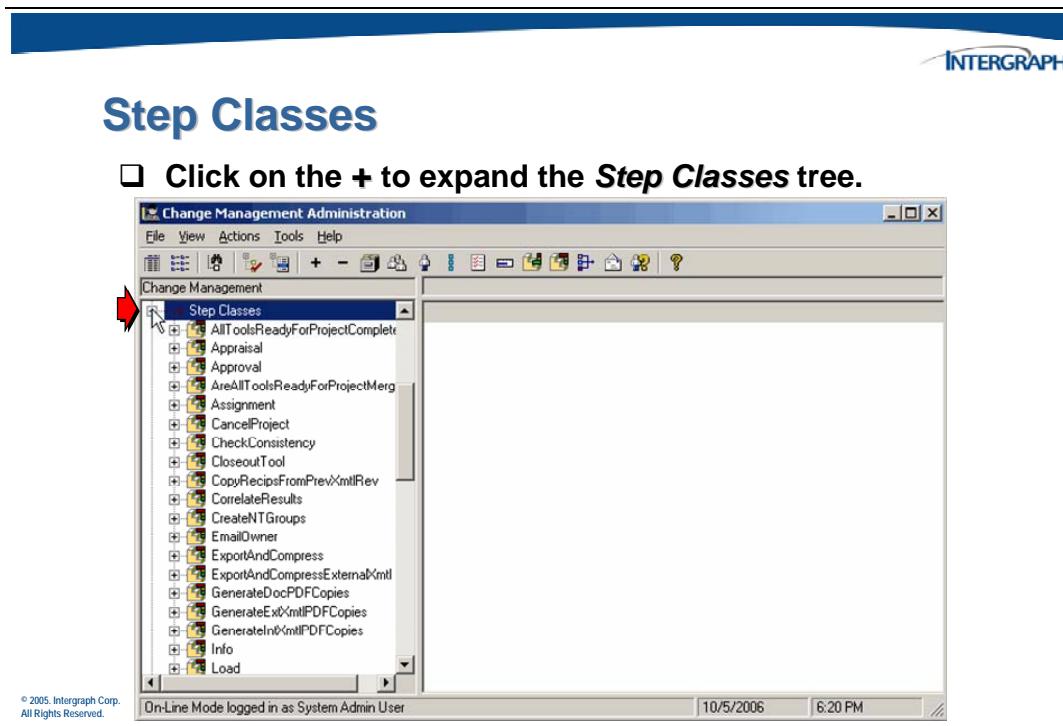
Step Classes

Step Classes are used in conjunction with step types to define the behavior and appearance of steps

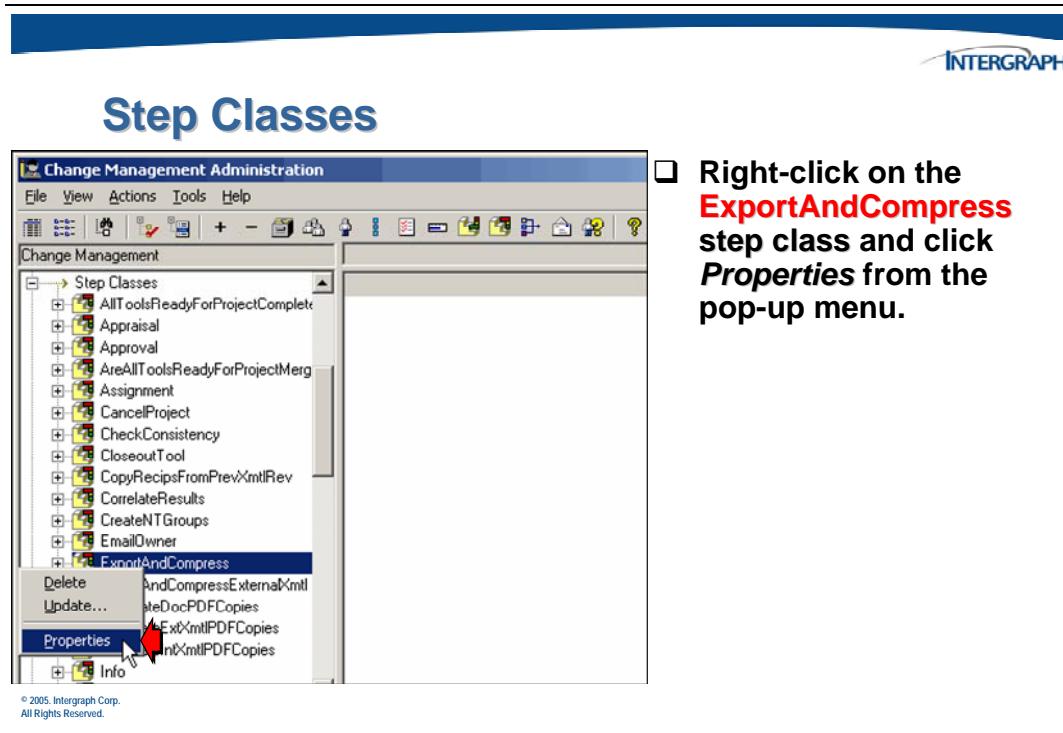
The default step classes are:

<u>Step Class</u>	<u>Description</u>	<u>Step Type</u>
Info	For Information	I
Assignment	Work Assignment	W
Approval	Approval Voting	W
Voting	Voting Result Testing	V
Sign Off	Automatic Document Revision Sign Off	S
E-mail Owner	E-mail Owner of the Change Object	P
Synchronize NT Groups By Project	Windows Authentication Synchronization of Users and User Groups.	P

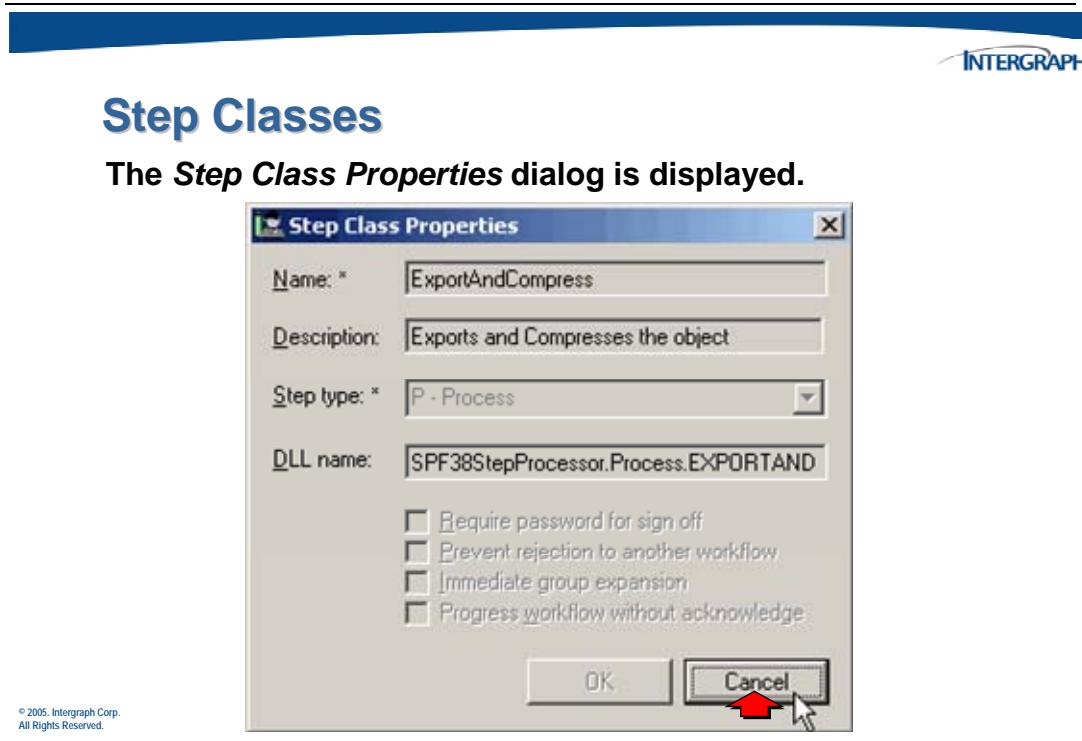
Step classes organize steps with similar functionality.



To view step class information, right-click the step class that you want and choose **Properties**.



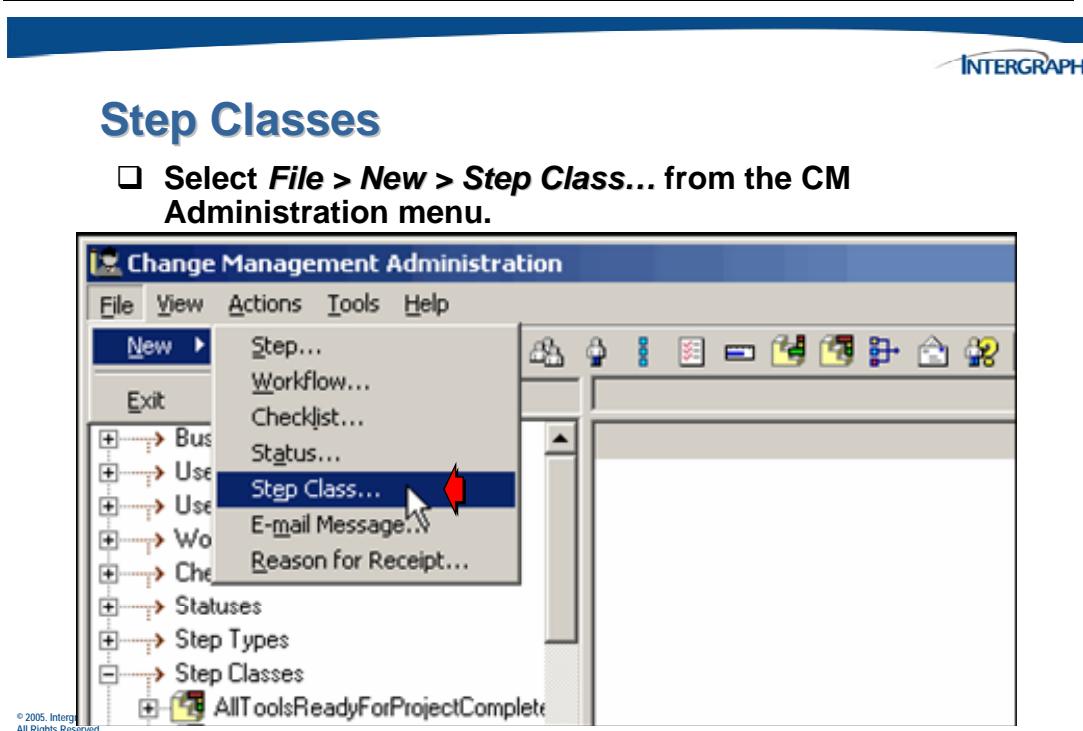
Selecting the **Properties** command displays the *Step Class Properties* dialog box for the selected step class as shown below.



Note If the "W-Work" step type is selected, the *Create and Update Step Class* dialog boxes will allow you to choose if a password is required for signoff off a step in a workflow. If the "P-Process" step type is selected, you will need to specify the process DLL name.

6.2.4 Create a Step Class

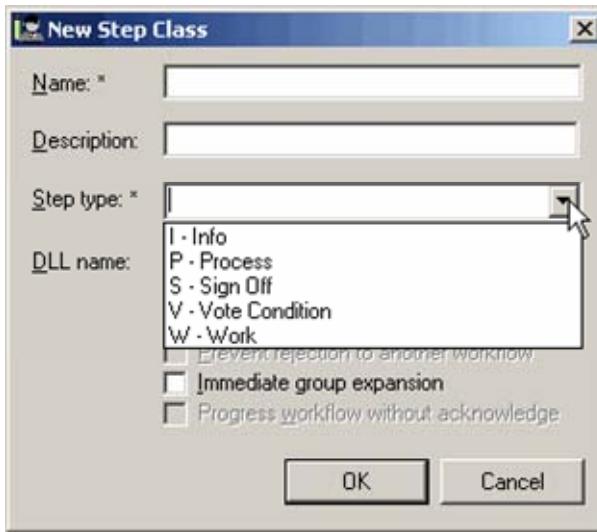
A workflow can be customized by adding new site specific **Step Classes**.





Step Classes

The **New Step Class** dialog is displayed.



1. In the **New Step Class** dialog box, type the name of the step class in the **Name** box.
2. In the **Description** box, type a description relevant to your step class.
3. In the **Step Type** box, click to select a step type for the step classification you are creating.
 4. If you select a **Step Type** of **P - Process**, you need to define the DLL to be executed with the step in the **DLL Name** box. The check boxes are enabled based upon the type of step that you select.
 - If you select a **Step Type** of **W - Work**, you can select the **Require password for sign off** check box, which will require the user to enter a password when signing off a step in a workflow.
 - If you select a **Step Type** of **I - Info**, you can select the **Progress workflow without acknowledge** check box, which allows a workflow to continue even if an information-only assignee has not acknowledged receipt of this action.
5. Click **OK**.

6.2.5 Steps

Steps are used during the definition of a workflow template. Each step in a workflow defines a process that must be performed in order to complete the workflow. Step Types and Step Classes are used to define the behavior of each step.

Using the **Change Management Administration** module, you can create, update, and delete steps for a workflow.



Steps

A Step defines a process that must be performed in order to complete the workflow

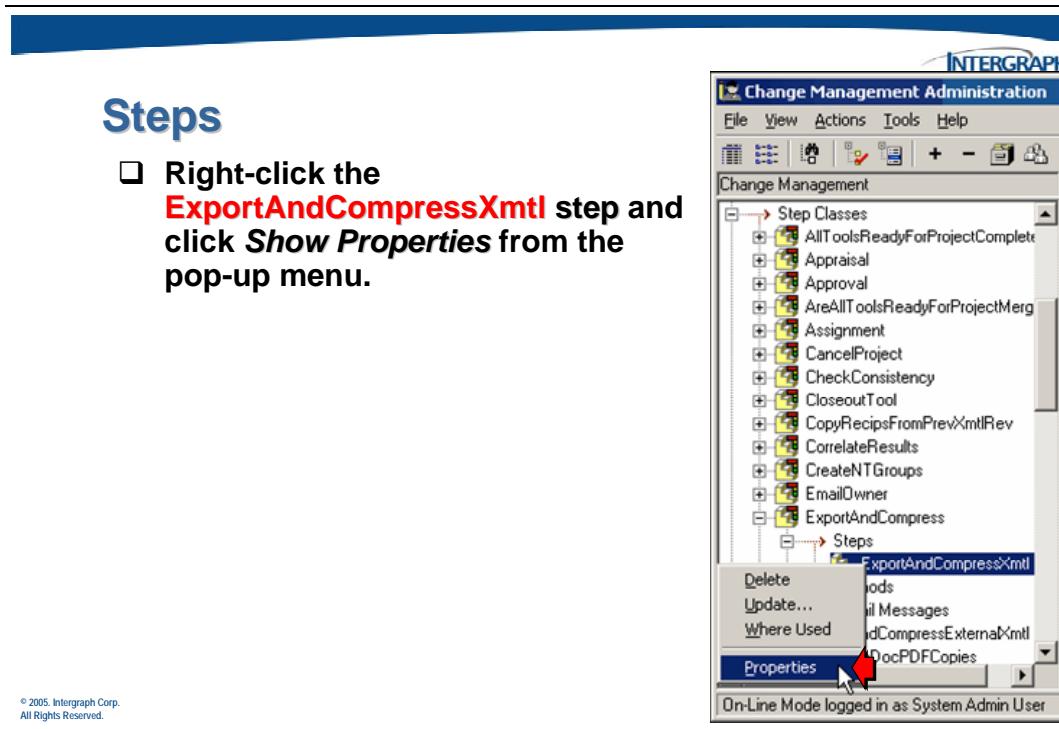
A Step is makeup of:

- Name
- Description
- Step Class
- Post DLL Name

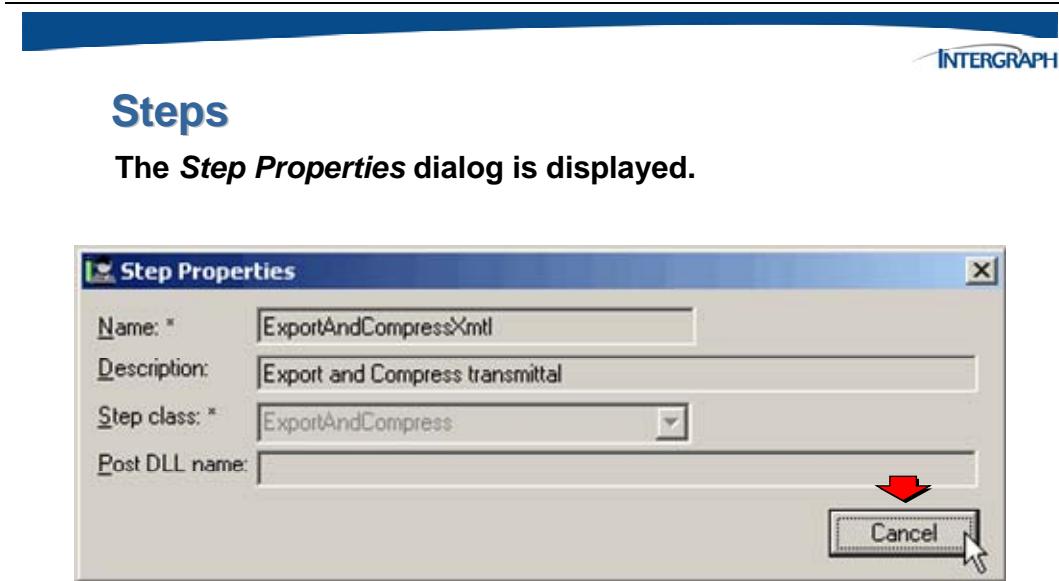
DLL Format:

<DLL>.<class type?>.<action name>,parameter1,parameter2
SPF38StepProcessor.Property.CREATECHKLISTFORSTEP,XmtIDocs,True

You can also view information about the step. To view step information, or to see which workflows a step is assigned to, right-click the step and choose **Properties** or **Where Used**.



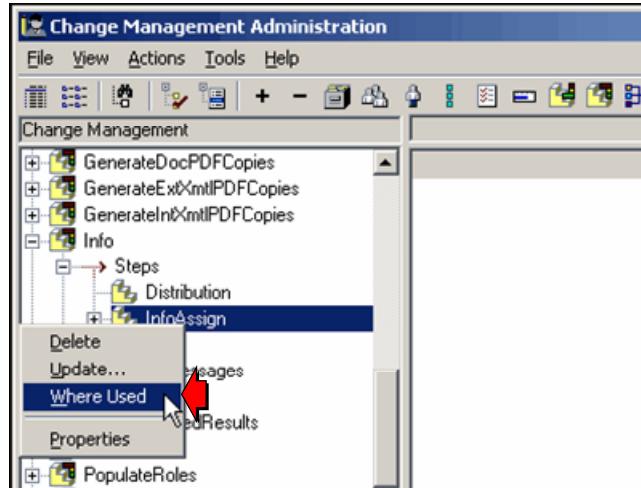
Selecting the **Properties** command displays the *Step Properties* dialog box for the selected step as shown below.



Selecting the **Where Used** command displays the Workflows the Step is used, the selected step as shown below.

Steps

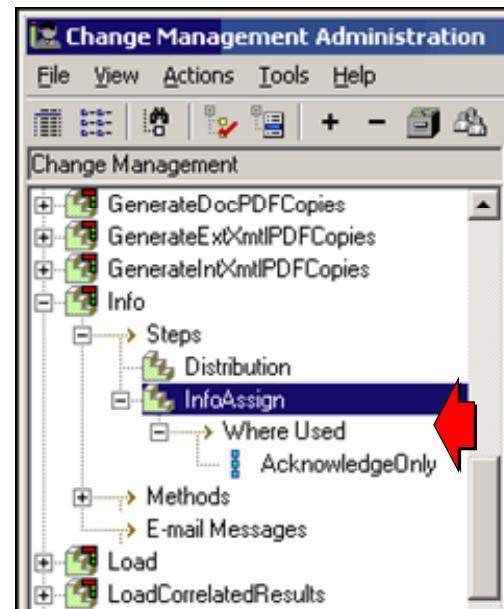
- ❑ Right-click the **InfoAssign** step and click **Where Used** from the pop-up menu.



This will display which workflows are currently using the *InfoAssign* step.

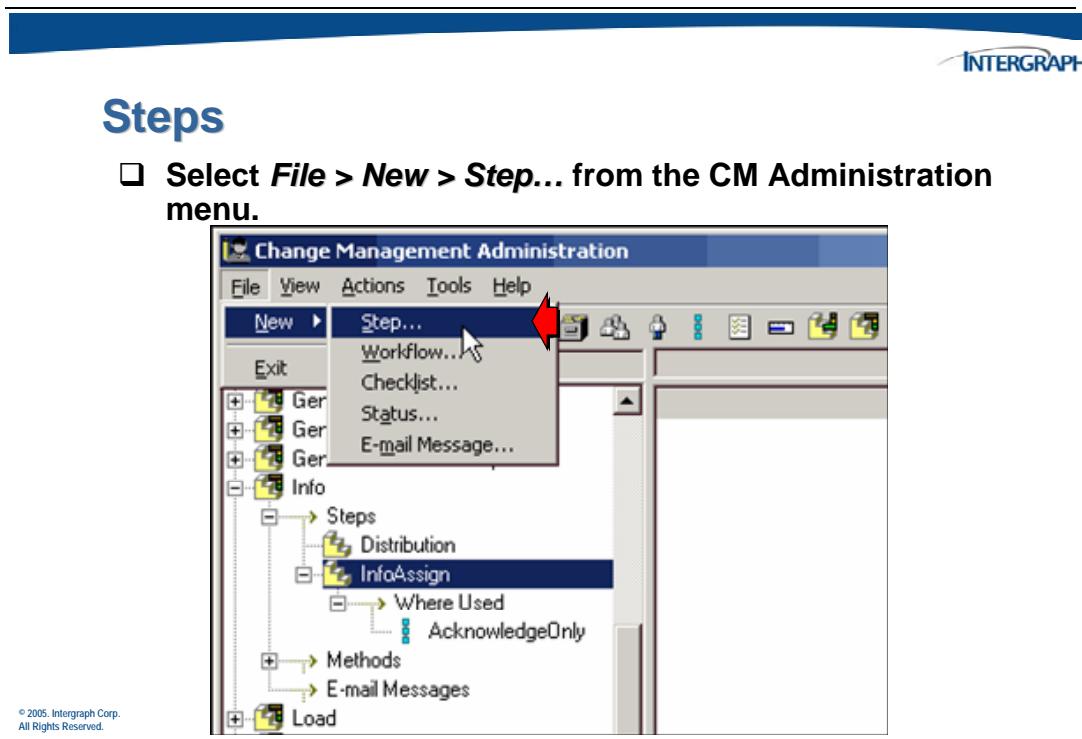
Steps

- ❑ The **InfoAssign** step is expanded to show the **Where Used** relationship.



6.2.6 Create a Step

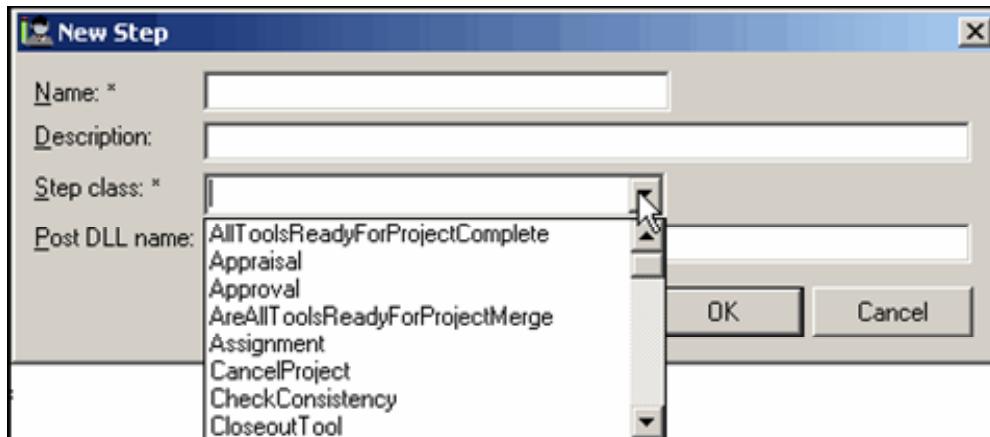
Customizing a workflow can include creating new workflow steps.





Steps

The **New Step** dialog is displayed.



© 2005, Intergraph Corp.
All Rights Reserved.

1. In the **New Step** dialog box, type the name of the step in the **Name** box.
2. In the **Description** box, type a description relevant to your step.
3. In the **Step Class** box, click to select a step classification for the step you are creating.
4. In the **Post DLL name** box, type the name of the DLL to be executed after the step is started in the workflow. For example, if you wanted to create a checklist for each of the recipients on an issue step, which would remind users that they have documents to review, then you could specify an action to occur when the issue step has started. The format would be as follows:

<DLL>.<class type?>.<action name>,parameter1,parameter2, where the number of parameters is determined by the method

Example:

SPF38StepProcessor.Property.CREATECHKLISTFORSTEP,XmtlDocs,True

5. Click **OK**.

6.2.7 Using Reasons for Receipt

Reasons for Receipt objects are a new object in Change Management Administration, which allows you to define why a user is receiving a transmittal. This new object is used on the transmittal and matrix recipient management forms. It is also a keyword in the e-mails set up in **Options**. Although the reason for receipt object is set up in Change Management Administration, it can be exported through the System Administration module. The Reason for Receipt object contains the following properties: *Name*, *Description* and *Information Only Indicator* (I – Info, W – Work).



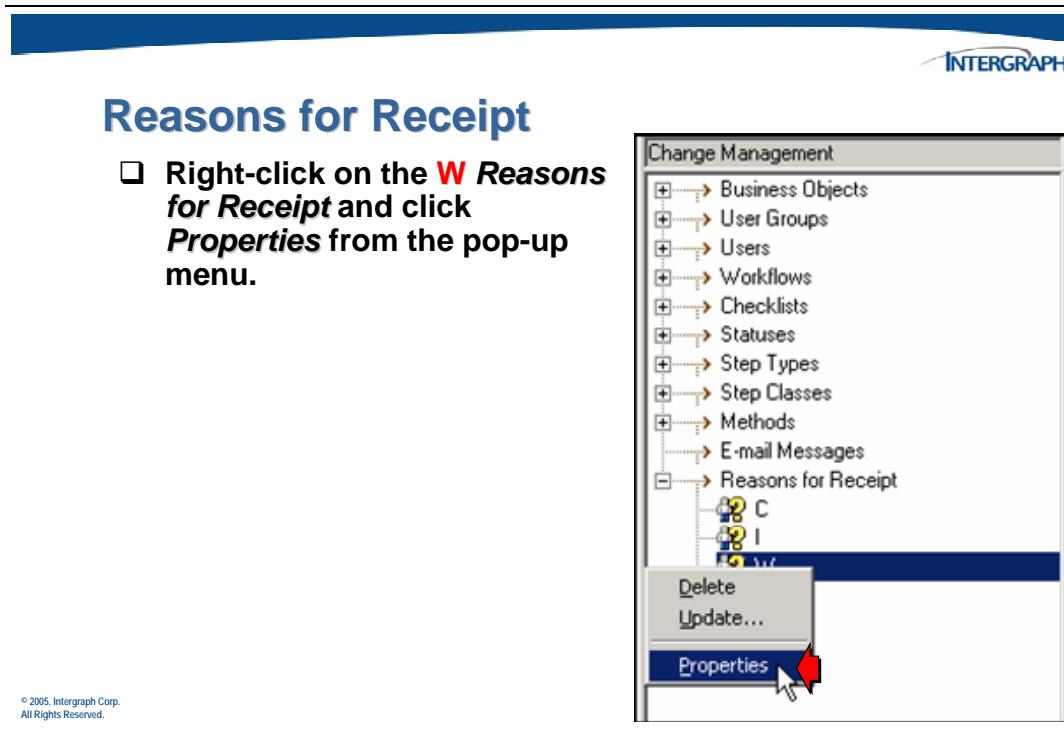
Reasons for Receipt

Reasons for Receipt objects define why a user is receiving a transmittal.

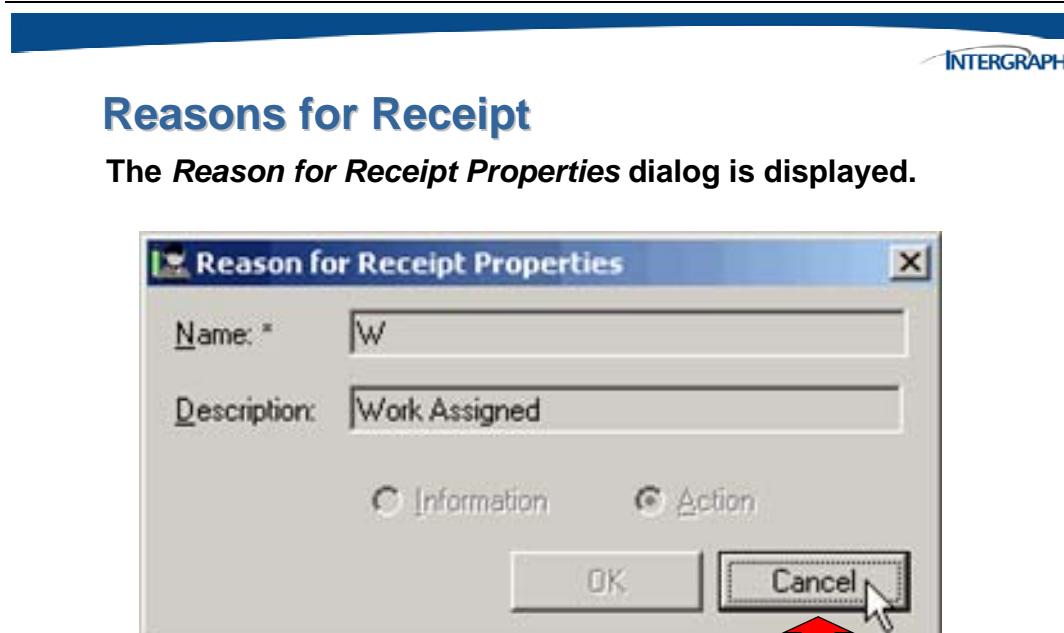
It is used on the **Transmittal** and **Matrix Recipient** management forms and also for email.

© 2005. Intergraph Corp.
All Rights Reserved.

Using the Change Management Administration module, you can create, update, and delete reasons for receipt. You can also view information about existing reasons for receipt. To view reason for receipt information, right-click the reason for receipt object you want and choose **Properties**.

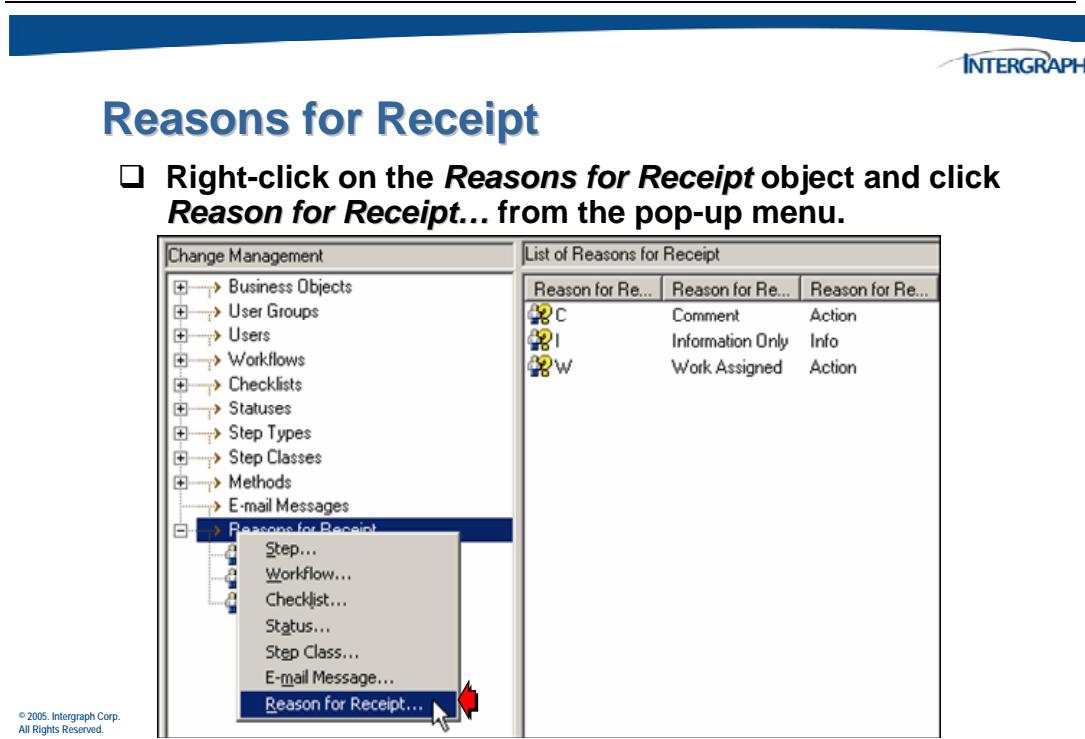


Selecting the **Properties** command displays the *Reason for Receipt Properties* dialog box for the selected reason for receipt as shown below.



6.2.8 Create a Reason For Receipt

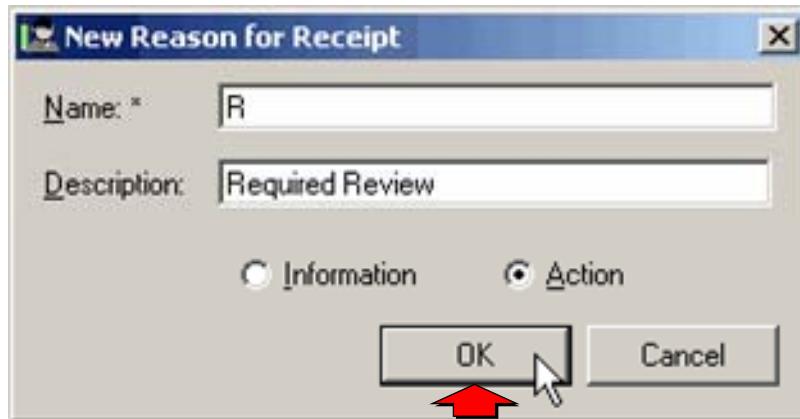
Custom **Reasons for Receipt** can be created for use in a transmittal process. The following example demonstrates adding a custom *Reason for Receipt*.





Reasons for Receipt

- Enter **New Reason for Receipt** information and click **OK**.



© 2005, Intergraph Corp.
All Rights Reserved.

1. In the *New Reason For Receipt* dialog box, type a name and description in the **Name** and **Description** boxes.
2. Select the type of receipt. The receipt types are: **Information** - transmittal is received for information only **Action** - transmittal is received and requires action
3. Click **OK**



Reasons for Receipt

Zoomed in view of the *List of Reasons for Receipt* pane.

Reason for Receipt Name	Reason for Receipt Description	Reason for Receipt Step Type
C	Comment	Action
I	Information Only	Info
R	Required Review	Action
W	Work Assigned	Action

6.2.9 Workflow Check Lists

Check Lists are a series of items, tasks, or questions that the user can be forced to complete before a step can be completed in a workflow. Each step in a workflow has the option of assigning a check list to the step. Check list objects are used when defining a workflow template but must exist before they can be referenced by the workflow.



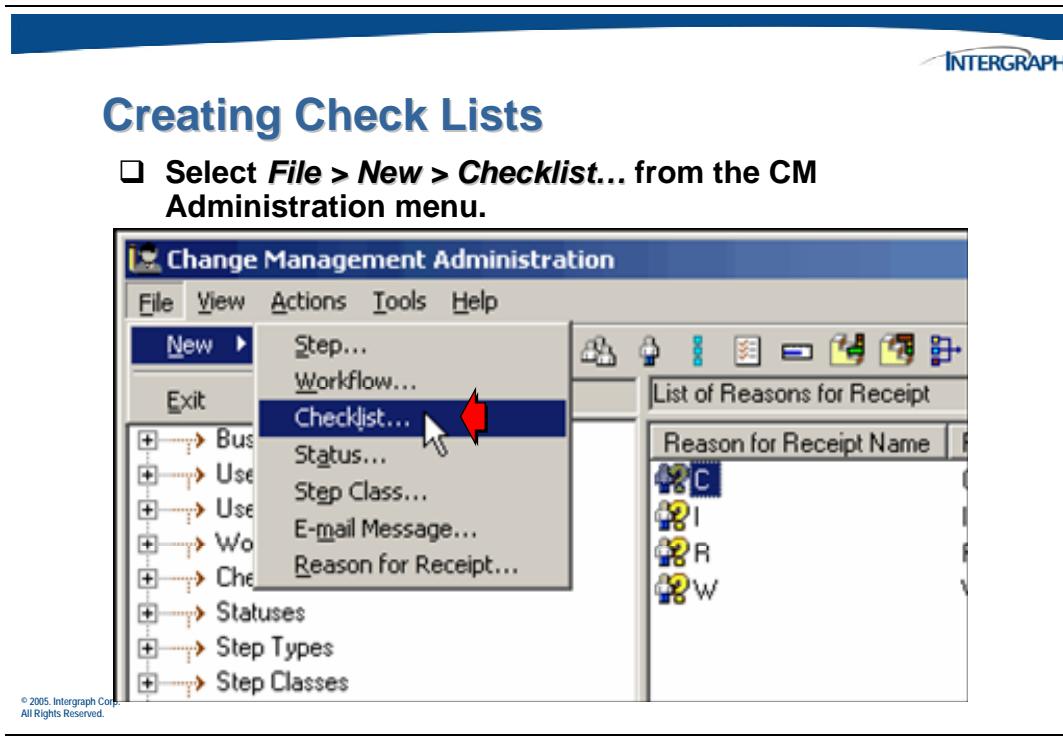
Check Lists

Check list can be attached to a workflow step and it has to be completed before step can be completed

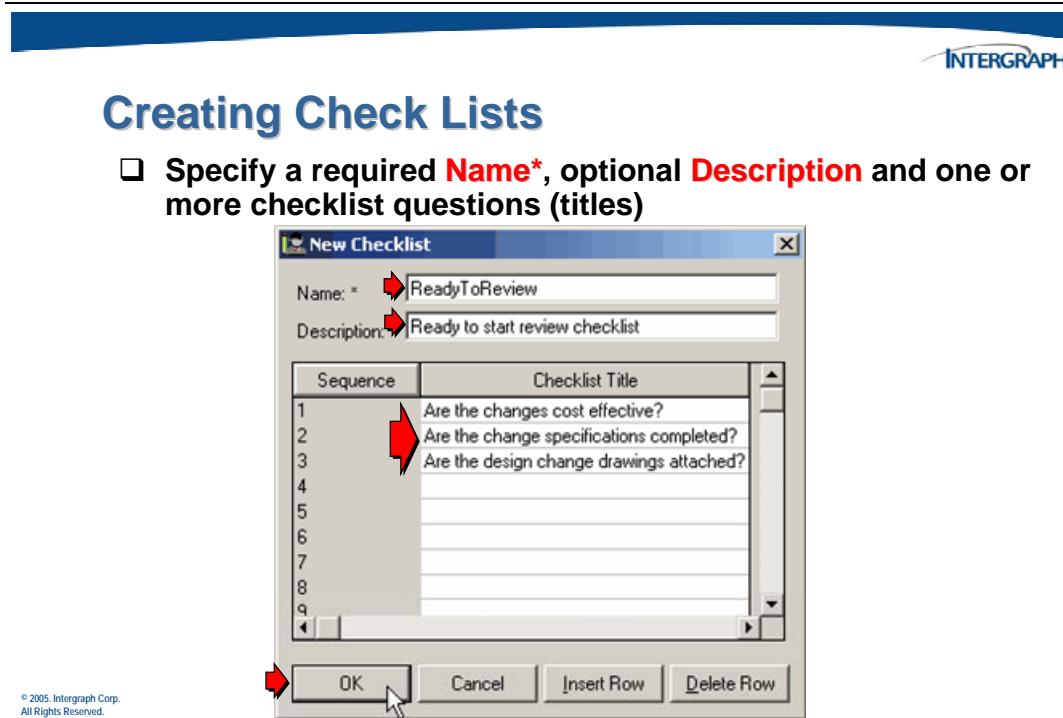
Check list can include several items

- Check list name**
- Check list description**
- Check list titles**

To create a new workflow check list:



The *New Checklist* dialog will display.



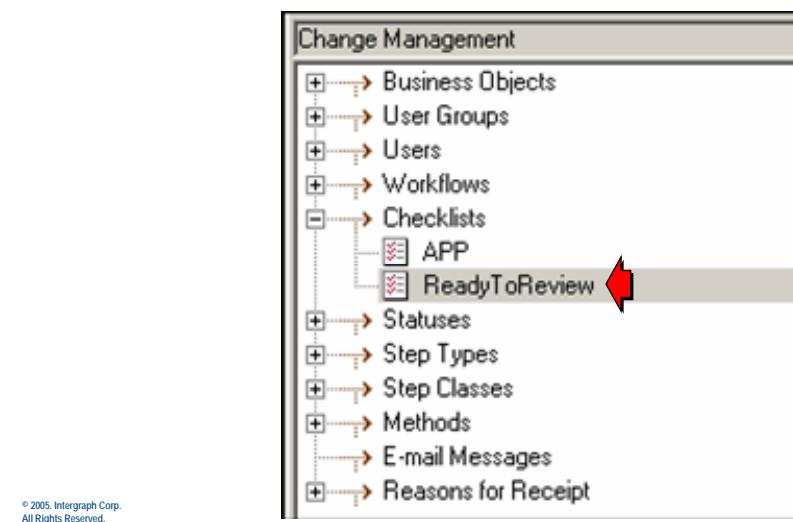
The following fields are used to define a checklist:

- Name** - a unique name to define the checklist
- Description** - a brief description for the check list, which is an optional field
- Checklist Title** – defines a question or action item for each check list step in the box next to its associated sequence number. Each sequence number is a unique number that is automatically incremented and can not be changed manually. There are twenty-five sequence steps predefined.
- Insert Row** - used to insert additional check list items (beyond the initial 25). A blank row is inserted above the selected row.
- Delete Row** – used to delete a check list item



Creating Check Lists

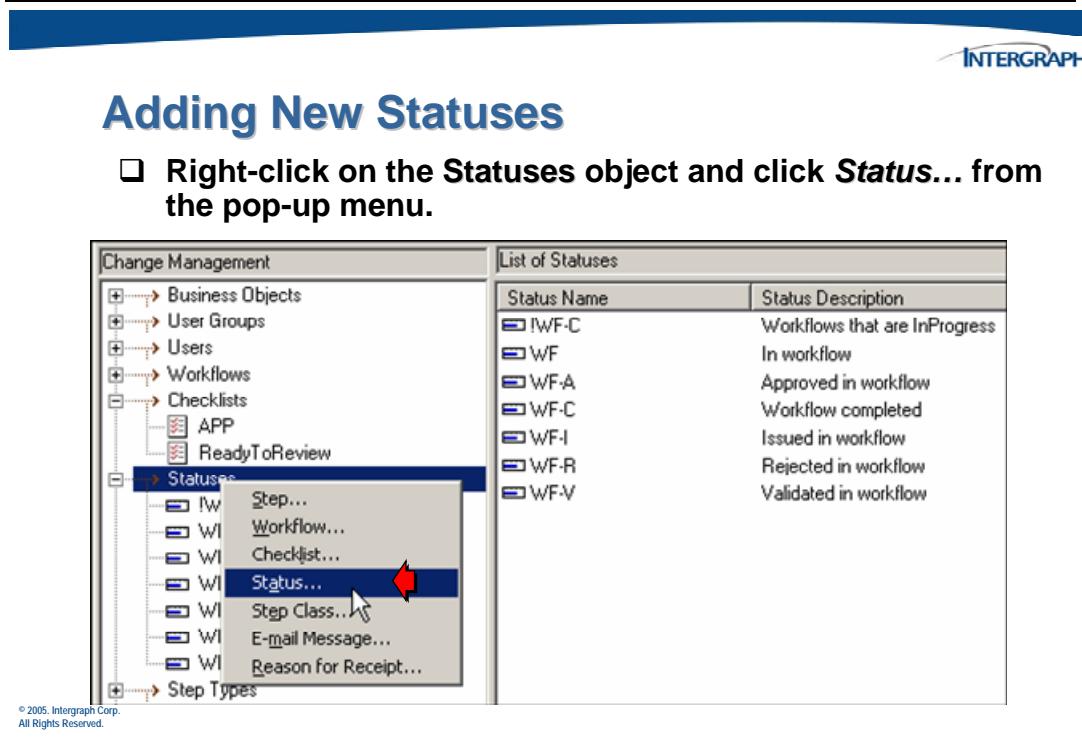
The new checklist is added to the CM tree



6.2.10 Workflow Statuses

Status objects are used during the definition of a workflow template. A status defines the state of an object in the workflow at the completion of each step in a workflow. Using the Change Management Administration module, you can create, update, and delete statuses.

The following example demonstrates how to add new statuses to Change Management.

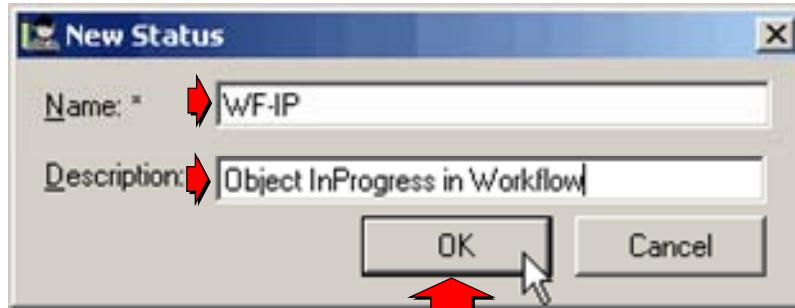


The *New Status* dialog will display.



Adding New Statuses

- Enter a required status **Name** and an optional **Description** and click **OK**.



© 2005, Intergraph Corp.
All Rights Reserved.



Adding New Statuses

The new statuses are now available to be used in workflows

List of Statuses	
Status Name	Status Description
IWF-C	Workflows that are InProgress
WF	In workflow
WFA	Approved in workflow
WF-C	Workflow completed
WF-I	Issued in workflow
WF-IP	Object InProgress in Workflow
WF-R	Rejected in workflow
WF-RR	Ready for Review Rejected
WF-S	Workflow Started
WF-V	Validated in workflow

© 2005, Intergraph Corp.
All Rights Reserved.

6.2.11 Building a Workflow

Workflows allow you to record a process using a template. Within a workflow, you can define the steps required to achieve the process and also set planned and target dates for the step and the duration of the step.



Building a Workflow

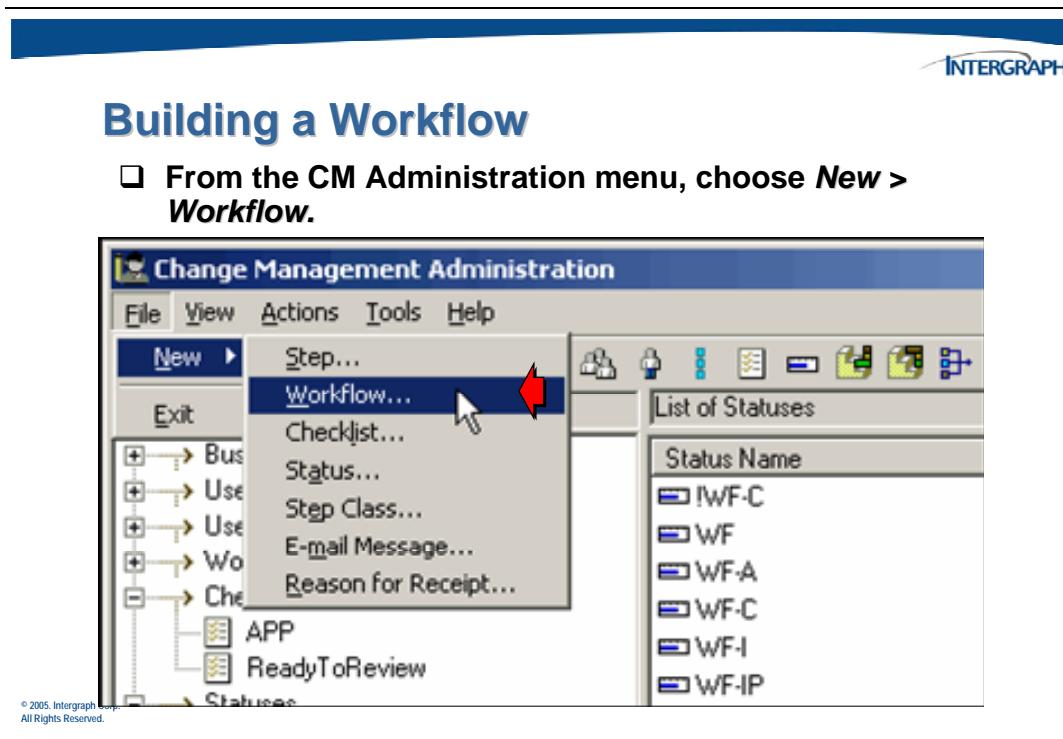
A Workflow can be attached to an Interface Definition

Workflows consists of steps that are organized sequentially:

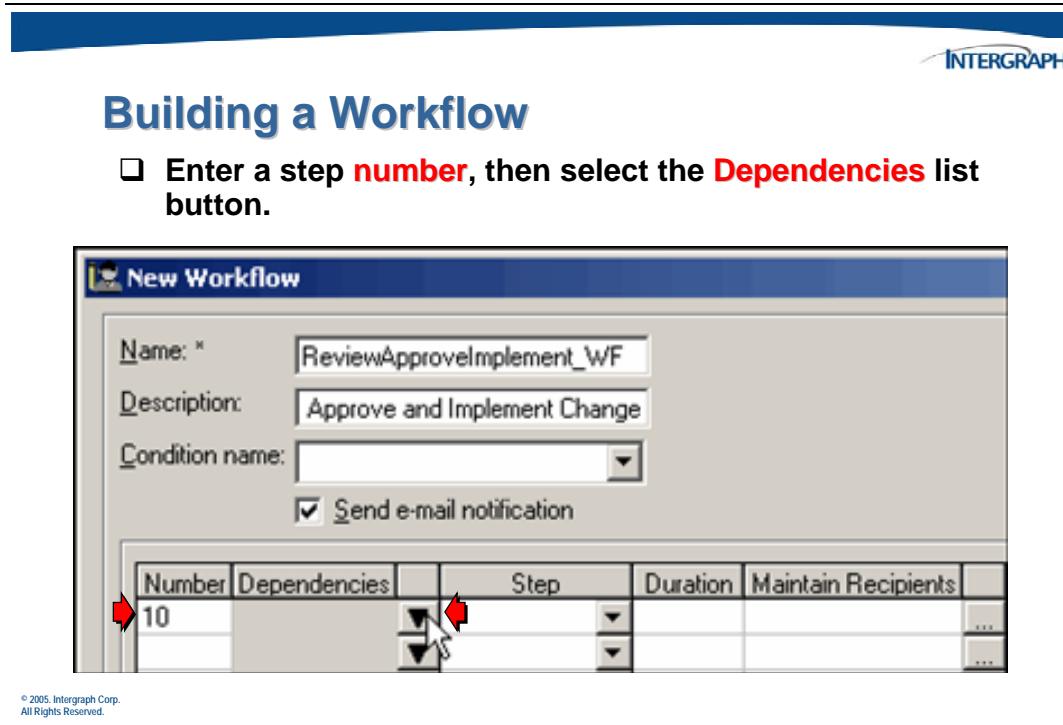
- Steps can be parallel or linear**
- Step can be assigned to individual user or user group**
- Maximum duration can be assigned to step**
- Step can have success and failure exits**

Workflows contain the steps necessary to achieve a change management process. These steps can be serial, parallel, or a combination of both. Each step is assigned a sequence number, a duration period, and a user (Assignee) or user group. Each step can optionally have a list of check list questions, which the user must perform at each step before completion of the change. The order of processing for each step is defined by dependencies. By default, any steps that have a dependency of **First** are started when the workflow template is attached to the change object. Subsequent steps start when their prerequisite sequence number (steps) is completed. Parallel steps are configured by specifying a number of steps that start based upon the completion of a particular step sequence number. These new steps then start and all must be completed before moving on to the next step.

To build a new workflow:

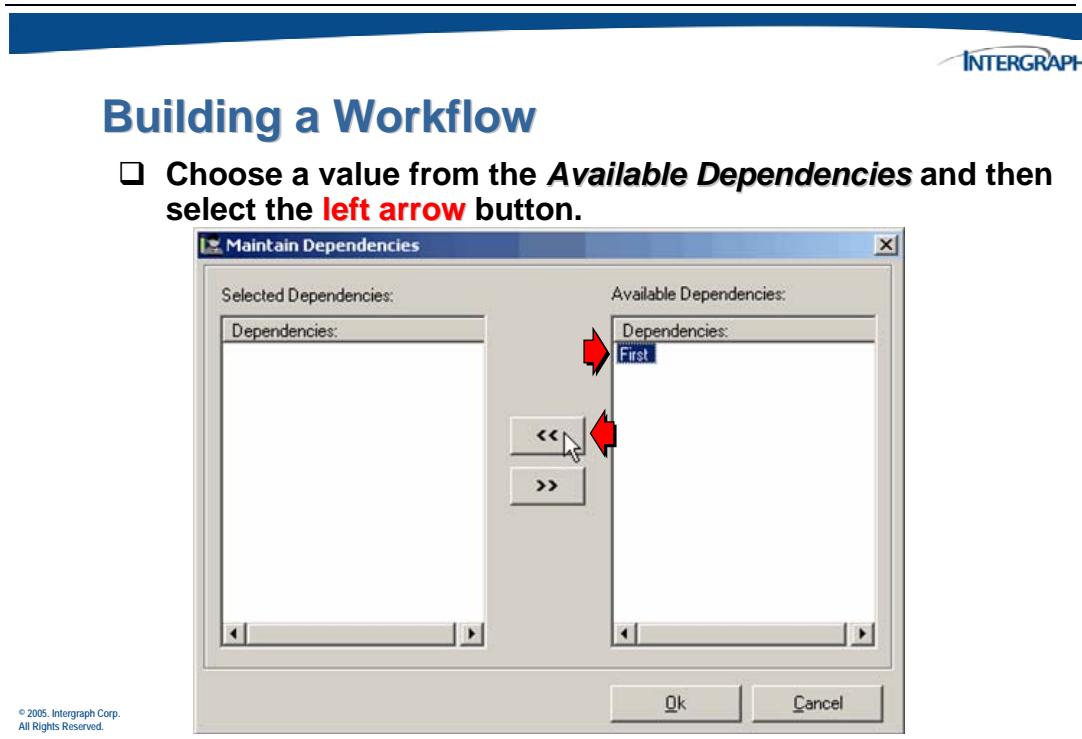


The *New Workflow* dialog will display.



The **New Workflow** dialog window contains the following fields:

- Name** - specifies the new workflow name and is required
- Description** – defines a short description for the workflow. This is an optional field.
- Condition name** - specifies a condition name for the workflow. Select an option from the list. Conditions are defined in SmartPlant Foundation System Administration and are used to determine if it is appropriate to attach a workflow to the selected object.
- Send e-mail notification** – enable the check box if the users associated with this workflow need e-mail notification.
- Number** - defines a number that specifies the order in which the assigned step will be performed.

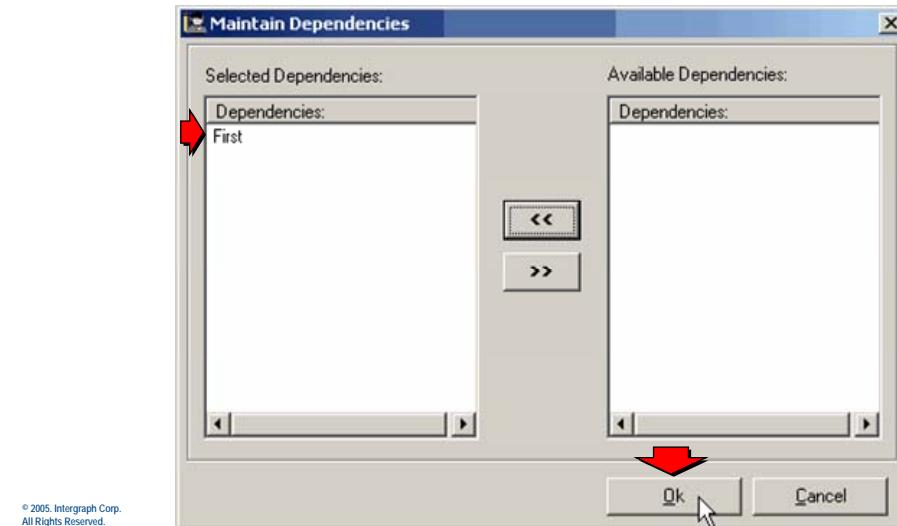


- Dependencies** - specifies any prerequisites for the step.



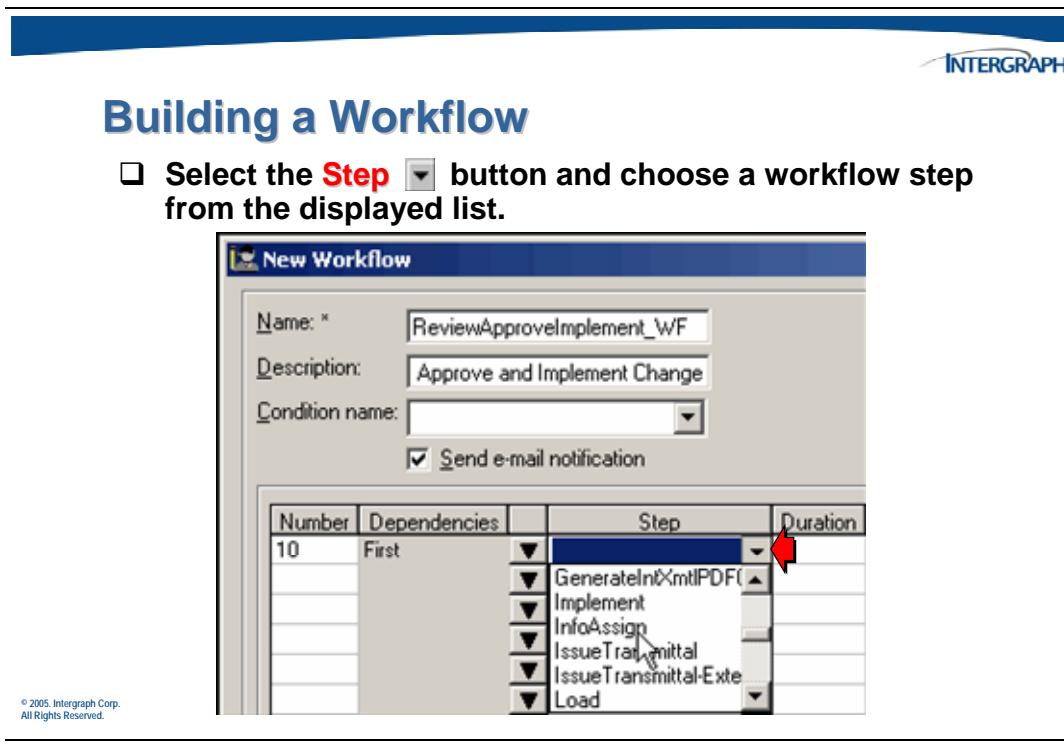
Building a Workflow

- Verify the correct **Dependency** and click **OK**.



Since this is the first step in the workflow, the single choice available from the *Maintain Dependencies* dialog is **First**, meaning this is the initial starting point for the workflow. The first step selected must have a dependency of **First**, which cannot be used as a sub-dependency for any step other than the initiating step.

Choose a workflow step name to be used with this sequence number.



The **New Workflow** dialog window fields (con't):

- ❑ **Step** - specifies the step to be performed from the list. Each step must be unique and a step already in use by this workflow may not be selected again.



Building a Workflow

- ❑ Choose the ellipses to select the **recipient** that will perform this workflow step.

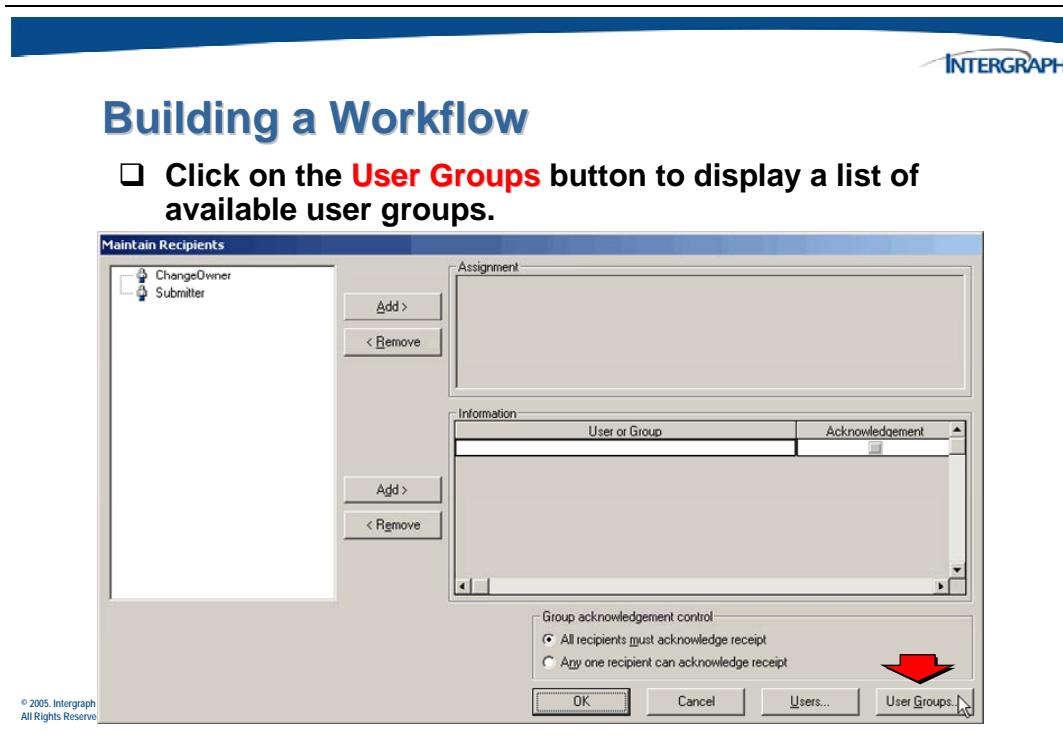
The screenshot shows the "New Workflow" dialog window. At the top, there are fields for "Name" (ReviewApproveImplement_WF), "Description" (Approve and Implement Change), and "Condition name". A checkbox for "Send e-mail notification" is checked. Below these is a table with columns: Number, Dependencies, Step, Duration, Maintain Recipients, and Status. The first row has values: 10, First, InfoAssign, 1, and an empty "Maintain Recipients" field. The "Status" column contains an ellipsis button (three dots) which is highlighted with a red arrow. The bottom left corner of the dialog window has a copyright notice: "© 2005. Intergraph Corp. All Rights Reserved."

Number	Dependencies	Step	Duration	Maintain Recipients	Status
10	First	InfoAssign	1		...

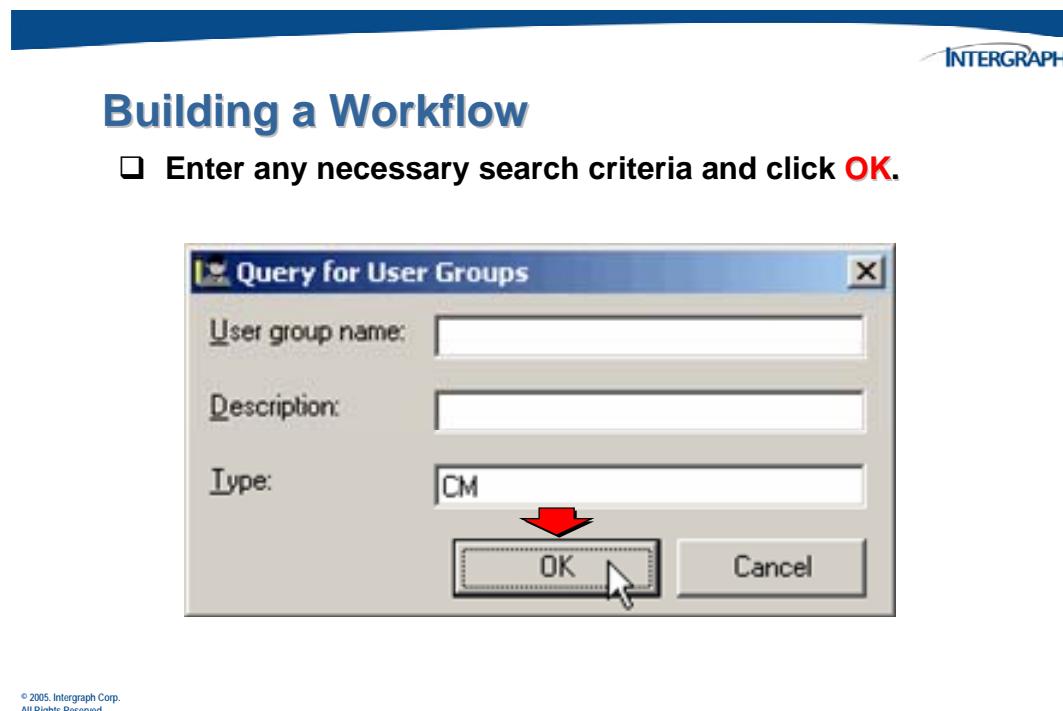
The **New Workflow** dialog window fields (con't):

- ❑ **Duration** - specifies the duration range for the step. Scroll up or down numerically to select a duration range of working days. The duration range is from 1 to 60 working days and a value of zero is not allowed.
- ❑ **Maintain Recipients** – specifies the User or User Group assigned to the step. If you assign a User Group to the step, all users in the group receive the object in their users **To Do List**.

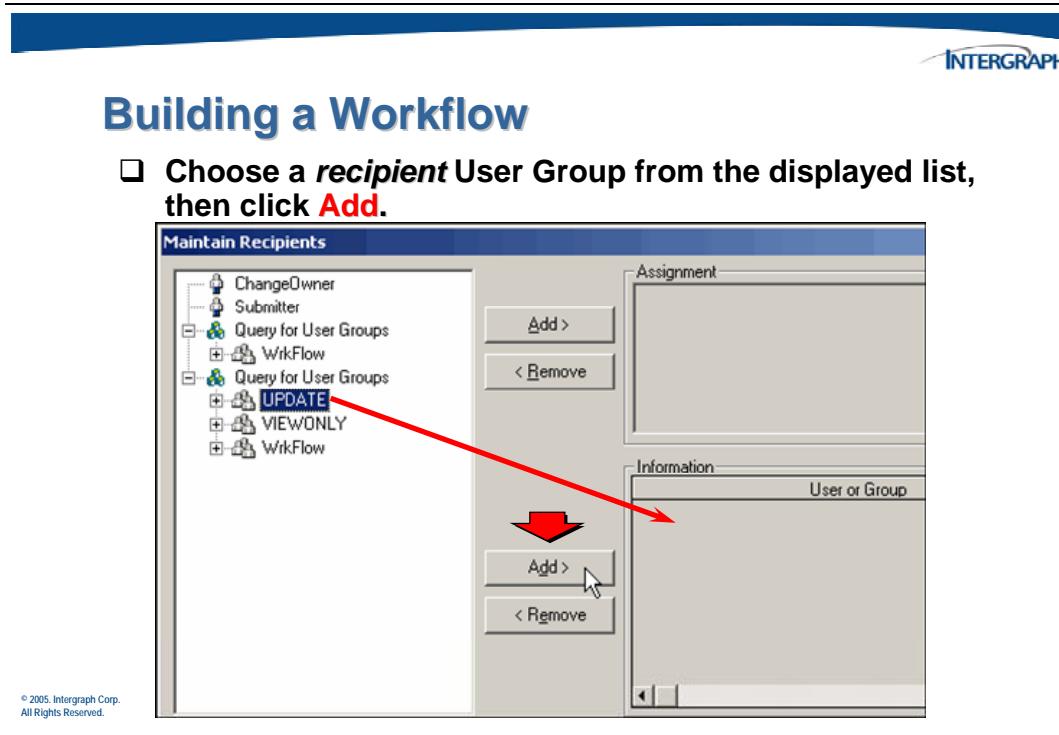
The *Maintain Recipients* dialog will display.



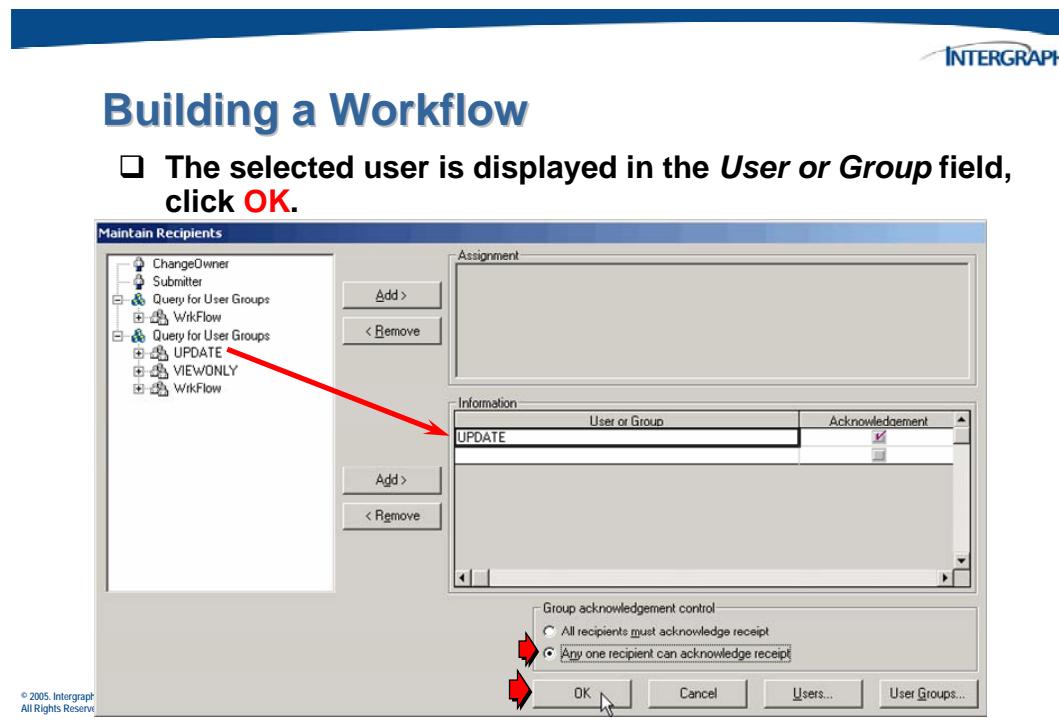
A *Query for User Groups* dialog will display allowing a search for specific User Groups.



The *Tree View* will expand to show the results of the **User Group** query.



You can set the *Group acknowledgment control* to **one** or **all** recipients.





Building a Workflow

- Select the **Status** button to choose an existing workflow status.

The screenshot shows the 'New Workflow' dialog window. At the top, there are fields for 'Name' (ReviewApproveImplement_WF), 'Description' (Approve and Implement Change), and 'Condition name'. A checked checkbox for 'Send e-mail notification' is also present. Below these is a grid table with columns: Number, Dependencies, Step, Duration, Maintain Recipients, Status, and Check Lists. The 'Step' column contains 'InfoAssign'. The 'Status' column has a dropdown arrow pointing down, which is highlighted with a red arrow in the screenshot. A mouse cursor is visible over the dropdown menu, which lists several workflow statuses: WF-I, WF-R, WF-V, WF-IP, WF-RR, and WF-S. The copyright notice '© 2005, Intergraph Corp. All Rights Reserved.' is at the bottom left of the dialog.

The **New Workflow** dialog window fields (con't):

- Status** - specifies a workflow Status from the list. The default statuses are:

- **WF-C** – Workflows that are InProgress
- **WF** – In workflow
- **WF-A** – Approved in workflow
- **WF-C** – Workflow completed
- **WF-I** – Issued in workflow
- **WF-R** – Rejected in workflow
- **WF-V** – Validated in workflow

And the new ones that were added:

- **WF-IP** – Object InProgress in Workflow
- **WF-RR** – Ready for Review Rejected
- **WF-S** – Workflow Started

- Check Lists** – assigns an existing check list to the step. Select a check list from the list in the **Check Lists** field.
- Complete** - specifies that the requirements for this step must be completed. This indicates to the user, via the To Do List, that all check list items must be completed by

the user if a check list is attached to this step. If a check list is specified for the step, but the Complete button is not checked, then the user can complete some, none, or all of the check list items.

Continue by defining the next step in the workflow.



Building a Workflow

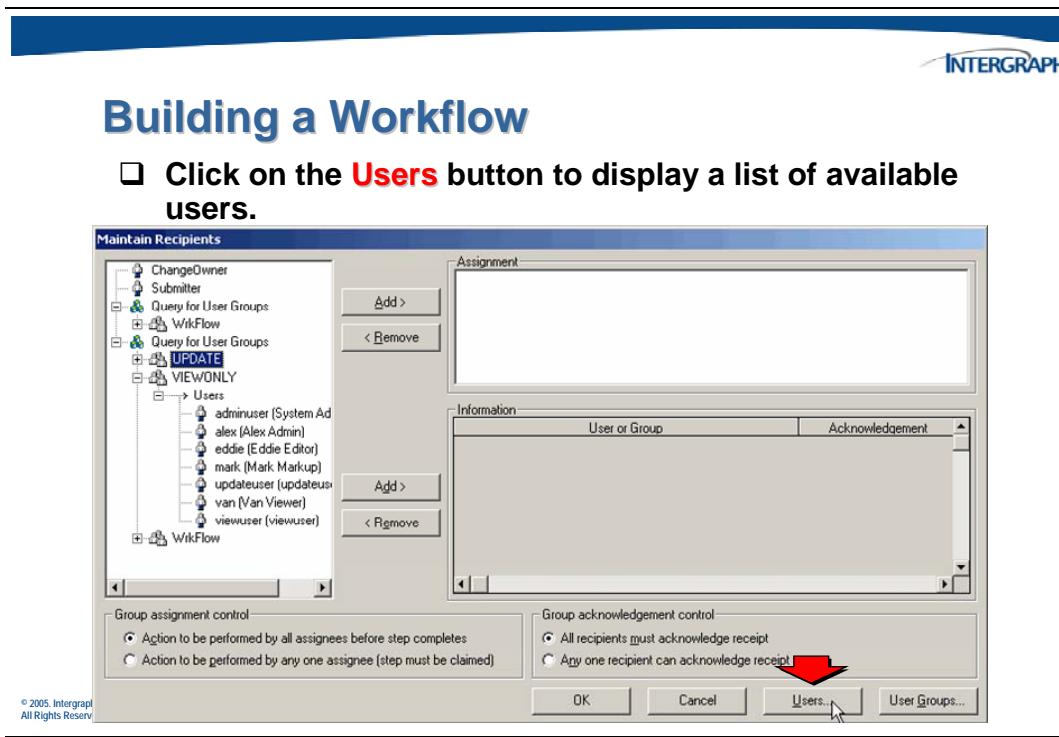
- Choose the ellipses to select the **recipient** that will perform this workflow step.

New Workflow

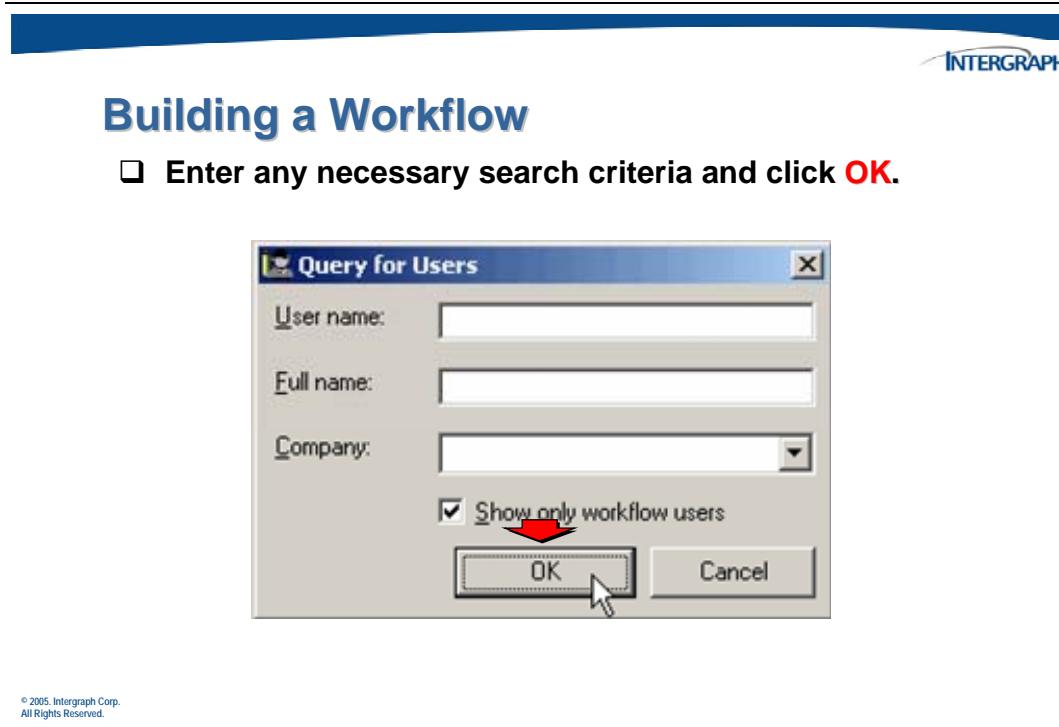
Name: *	ReviewApproveImplement_WF				
Description:	Approve and Implement Change				
Condition name:	<input type="button" value="▼"/>				
<input checked="" type="checkbox"/> Send e-mail notification					
Number	Dependencies	Step	Duration	Maintain Recipients	Status
10	First	InfoAssign	1	...UPDATE	WF-S
20	10	PrepareForReview	1
			
			
			

© 2005. Intergraph Corp.
All Rights Reserved.

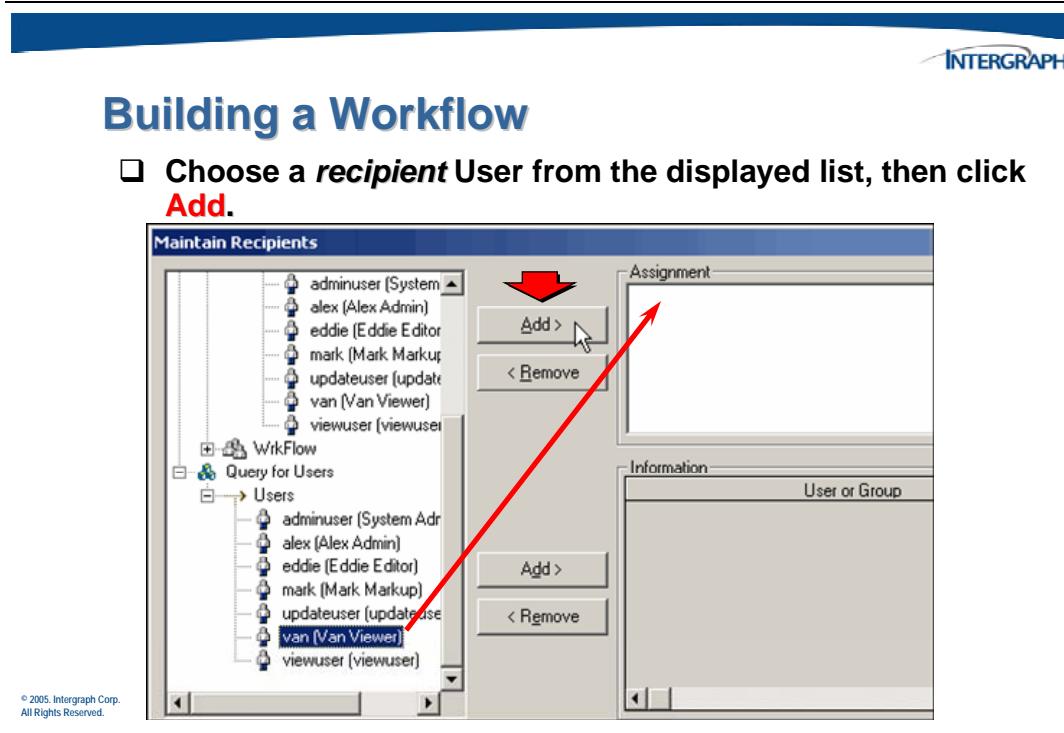
The *Maintain Recipients* dialog will display.



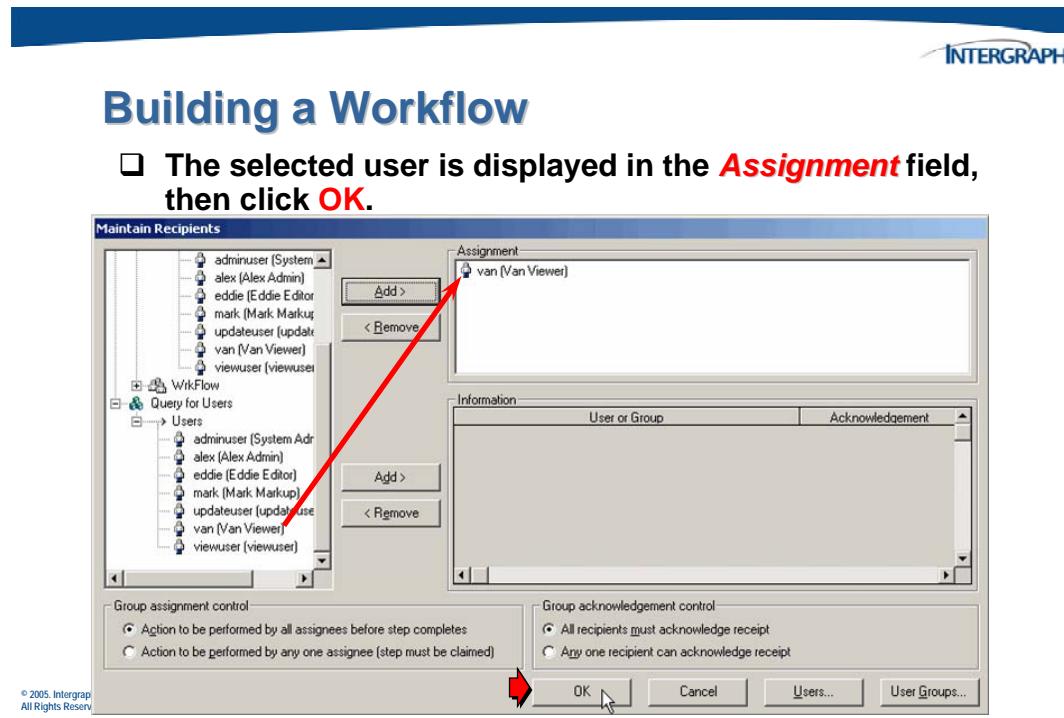
Users can also be added as recipients.



The *Query for Users* area of the tree is populated with the results of the query.



In this example, *van* is being added as a recipient to a different step.



This step will have a check list associated with it.

New Workflow

Name:	ReviewApproveImplement_WF					
Description:	Approve and Implement Change					
Condition name:						
<input checked="" type="checkbox"/> Send e-mail notification						
Number	Dependencies	Step	Duration	Maintain Recipients	Status	Check Lists
10	First	InfoAssign	1	...UPDATE	WF-S	
20	10	PrepareForReview	1	van	WF-IP	
						APP ReadyToReview

© 2005, Intergraph Corp.
All Rights Reserved.

To determine what happens if a step is not processed successfully, configure the **Reject Step** information.

Maintain Recipients	Status	Check Lists	Complete	Accept Step	Reject Step
...UPDATE	WF-S				
van	WF-IP	ReadyToReview			...
...					...
...					...
...					...
...					...
...					...

OK Cancel Insert row Delete row View

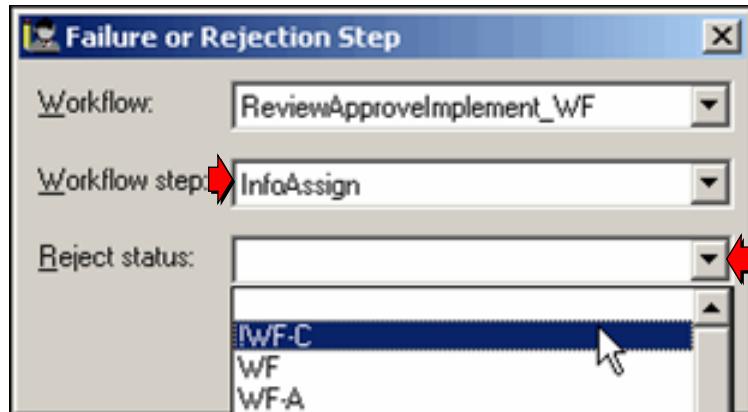
© 2005, Intergraph Corp.
All Rights Reserved.

When the **Reject Step** ellipses are selected, a *Failure or Rejection Step* dialog is displayed.



Building a Workflow

- Select the reject step (Workflow step) and a *Reject status* and click **OK**.



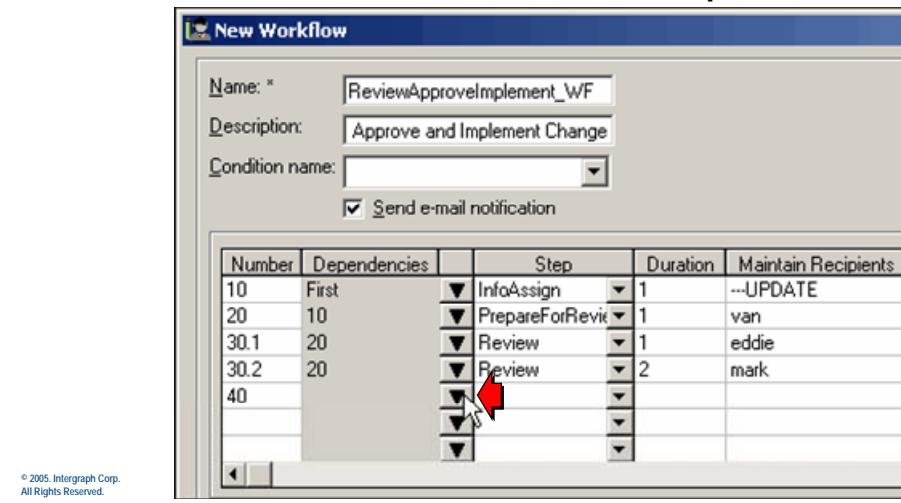
© 2005, Intergraph Corp.
All Rights Reserved.

Continue to build the workflow template by adding the next steps.



Building a Workflow

- Choose the list button to select the **Dependencies** that will be associated with this workflow step.

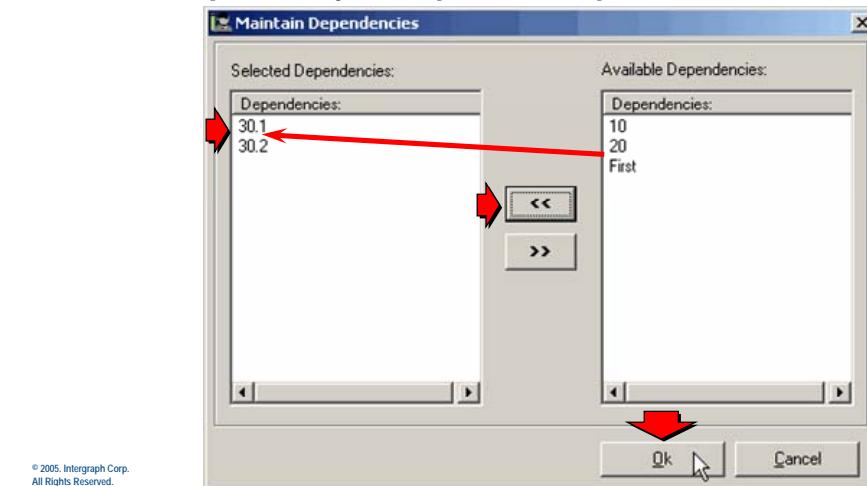


Parallel steps can be chosen as a dependency for a particular step, such as Steps **30.1** and **30.2** being dependencies for Step **40**.



Building a Workflow

- Multiple Dependencies** can be selected to define a dependency for a parallel step.

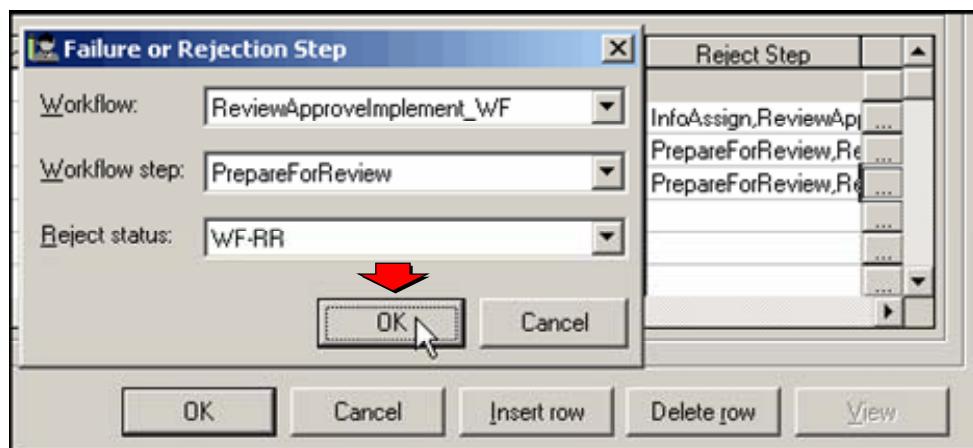


The **Dependencies** field now shows both **30.1 & 30.2**, which means that Step 40 cannot be performed until Steps 30.1 and 30.2 are complete. Continue to define the workflow template.



Building a Workflow

- Once the Rejection Step has been configured, click **OK**.





Building a Workflow

- When all of the steps have been defined for the workflow, click **OK**.

New Workflow

Name:	ReviewApproveImplement_WF									
Description:	Approve and Implement Change									
Condition name:										
<input checked="" type="checkbox"/> Send e-mail notification										
Number	Dependencies	Step	Duration	Maintain Recipients	Status	Check Lists	Complete	Accept Step	Reject Step	
10	First	InfoAssign	1	...UPDATE	WF-S					InfoAssign.Rev...
20	10	PrepareForRev	1	van	WF-IP	ReadyTo...				PrepareForRev...
30.1	20	Review	1	eddie	WF-IP					PrepareForRev...
30.2	20	Review	2	mark	WF-IP					PrepareForRev...
40	30.1&30.2	Approve	1	updateuser	WF-A	APP				Approve.Revi...
50	40	Implement	1	alex	WF-I					Implement.Rev...

OK **Cancel** **Insert row** **Delete row** **View**

© 2005, Intergraph Corp.
All Rights Reserved.



Building a Workflow

The *Tree View* shows the newly defined custom workflow

Change Management Administration

File View Actions Tools Help

Change Management

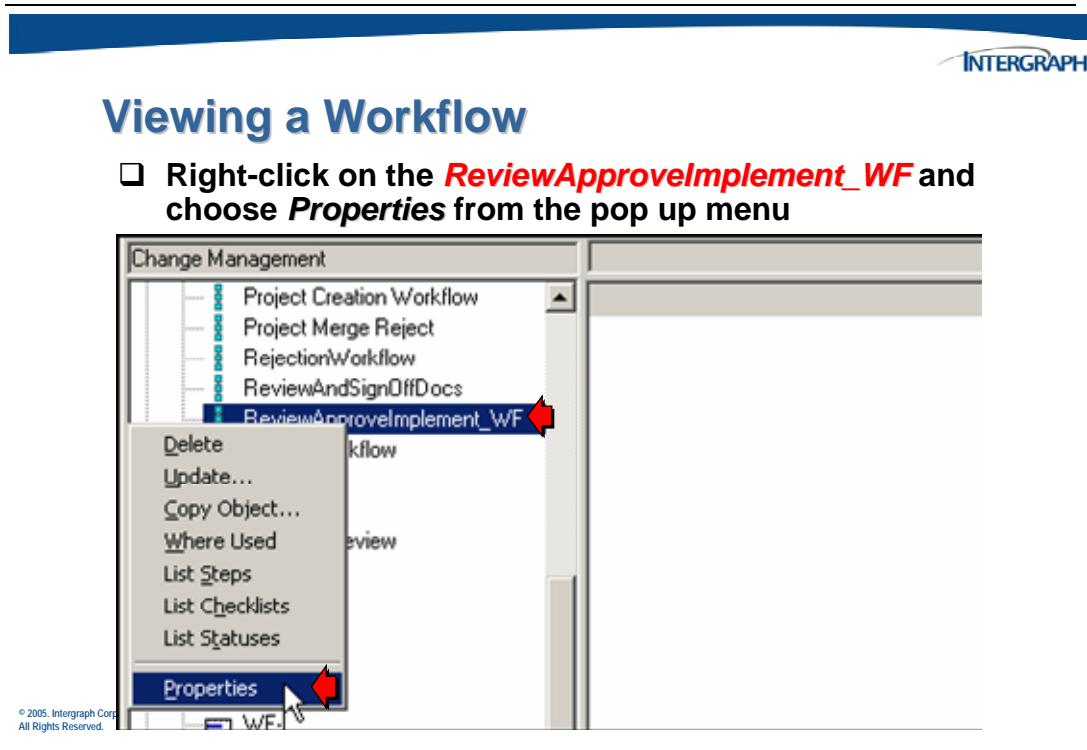
- Workflows
 - AcknowledgeOnly
 - ApproveVoteWorkflow
 - CancelExternalTransmittal
 - CancelInternalTransmittal
 - ExcelDocumentCorrelateWorkflow
 - ExternalTransmittalLifeCycle
 - InternalTransmittalLifeCycle
 - PDSDocumentCorrelate
 - Project Cancellation Workflow
 - Project Complete Reject
 - Project Completion Workflow
 - Project Creation Workflow
 - Project Merge Reject
 - RejectionWorkflow
 - ReviewAndSignOffDocs
 - ReviewApproveImplement_WF**
 - SignOffWorkflow

© 2005, Intergraph Corp.
All Rights Reserved.

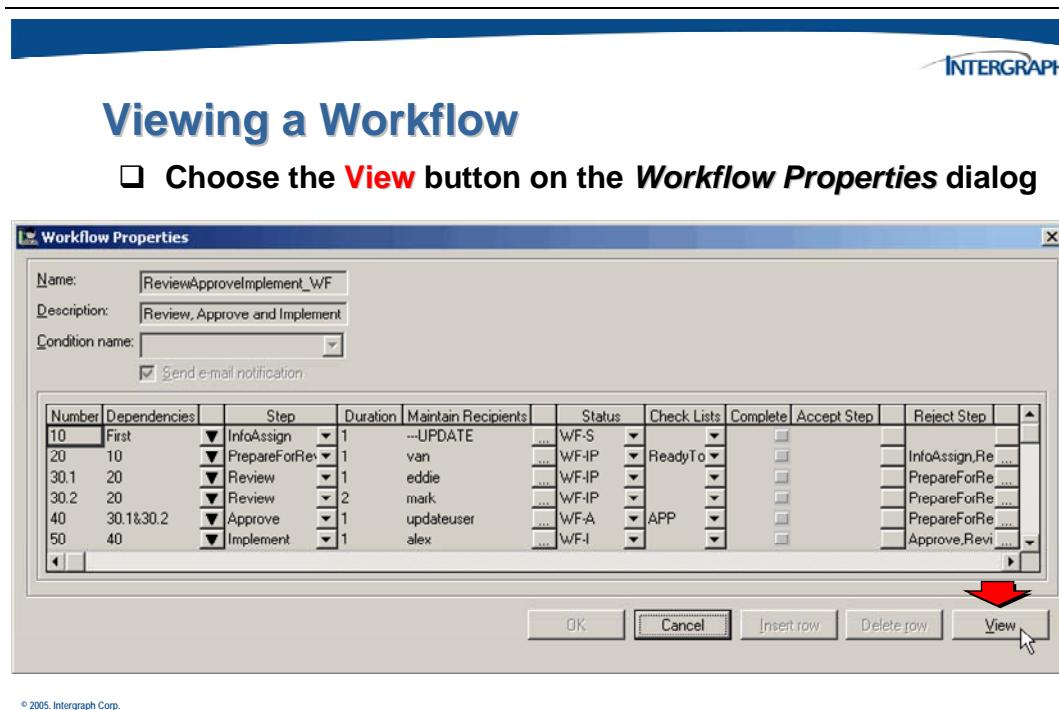
6.3 Viewing Workflows

Once the workflow has been created, a graphical display can be used to review the workflow steps.

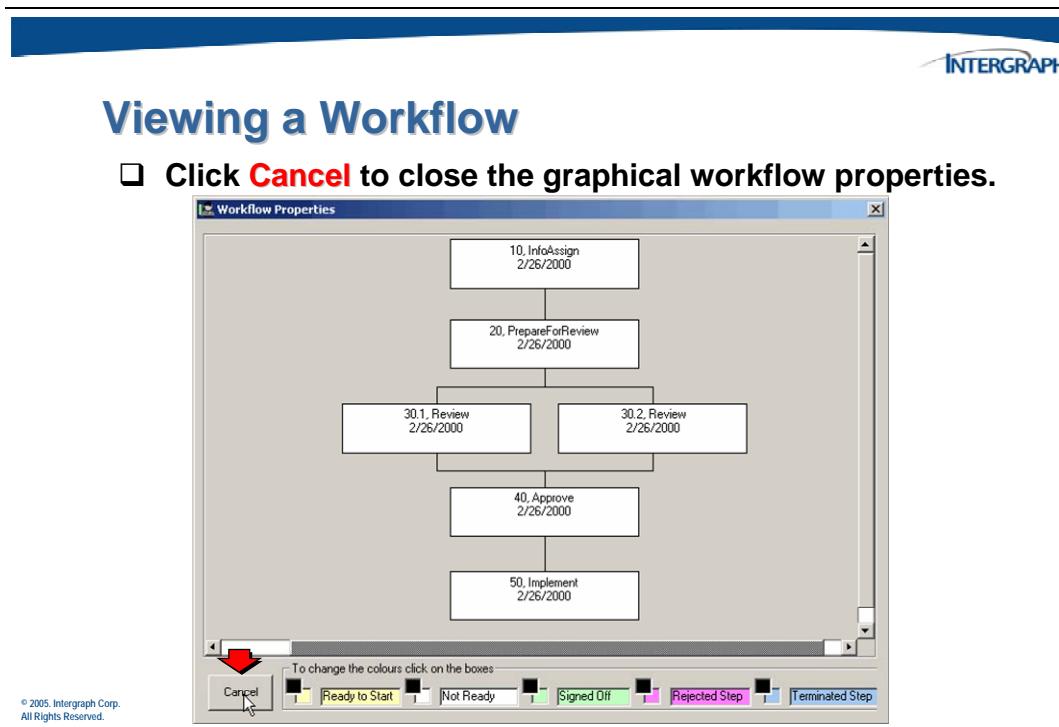
Locate the workflow in either the *Tree View* or *List View* windows.



The *Workflow Properties* dialog will display.



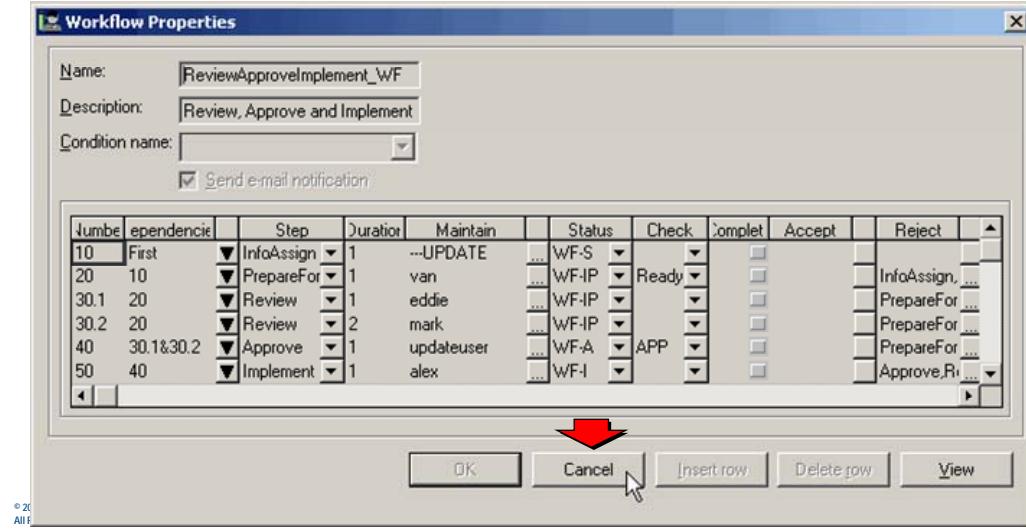
Each node in the graphical display represents a step in the workflow and the status of the step is represented by a color. In the example below, the color **white** represents a state of **Not Ready**.





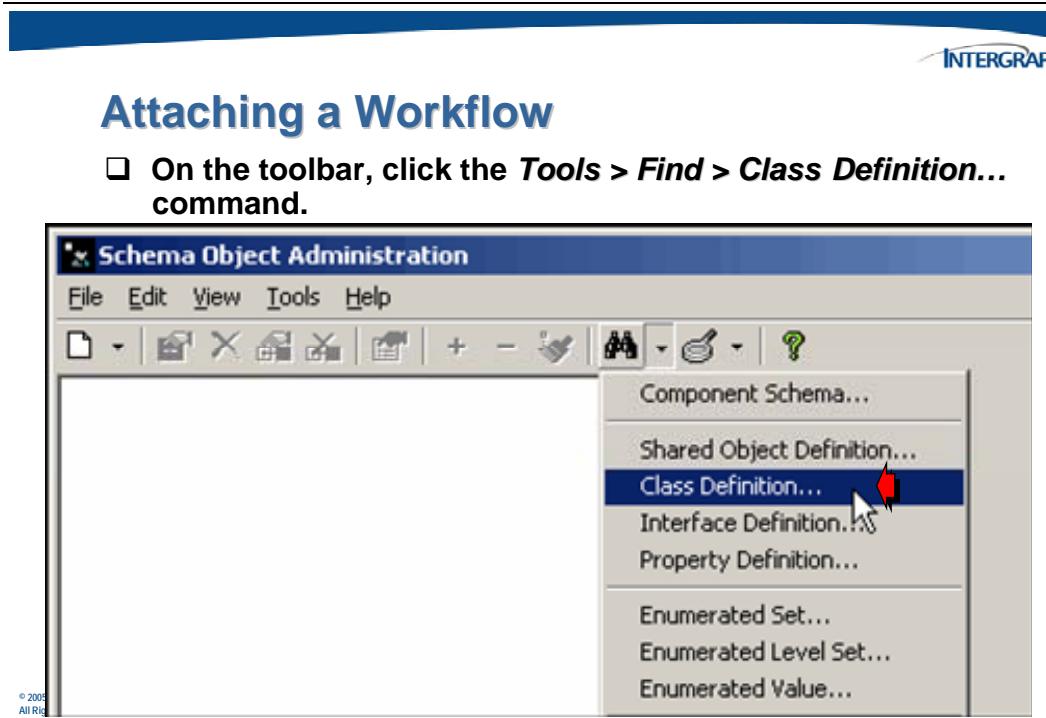
Viewing a Workflow

□ Choose **Cancel** to dismiss the *Workflow Properties* dialog.



6.4 Attaching Workflows

Attaching a workflow to an interface definition means that the workflow is available for all objects or classes associated with the interface definition. For example, if you attach a workflow to the **IDocumentVersion** interface definition, all classes that realize the **IDocumentVersion** interface definition can be submitted to the selected workflow.



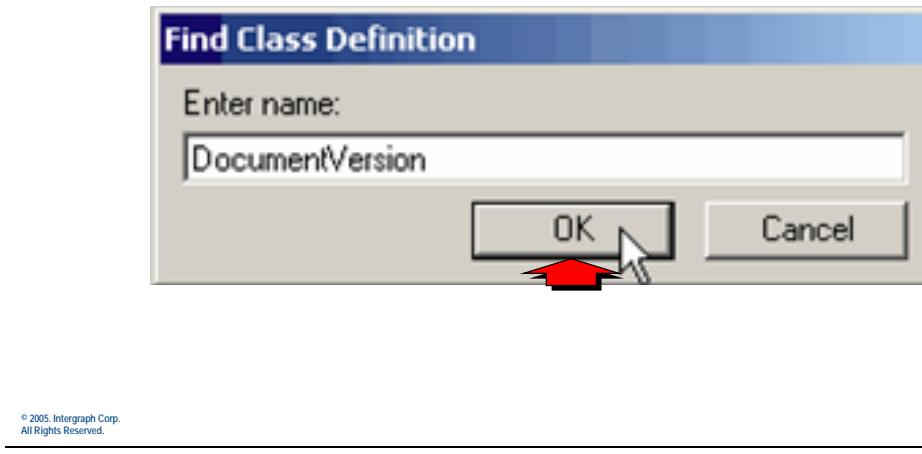
A search will be performed to locate the *DocumentVersion* class. This is because when a document object is created, in reality two object instances are created – the document master and the document revision. The *DocumentVersion* class is associated with the revision object that gets created. When an object gets submitted to a workflow, it's not the document master that is routed through the workflow steps, it's a specific **version** of the document revision.

The *Find Class Definition* dialog box displays.



Attaching a Workflow

- Type **DocumentVersion** in the *Find Class Definition* dialog box and click **OK**.

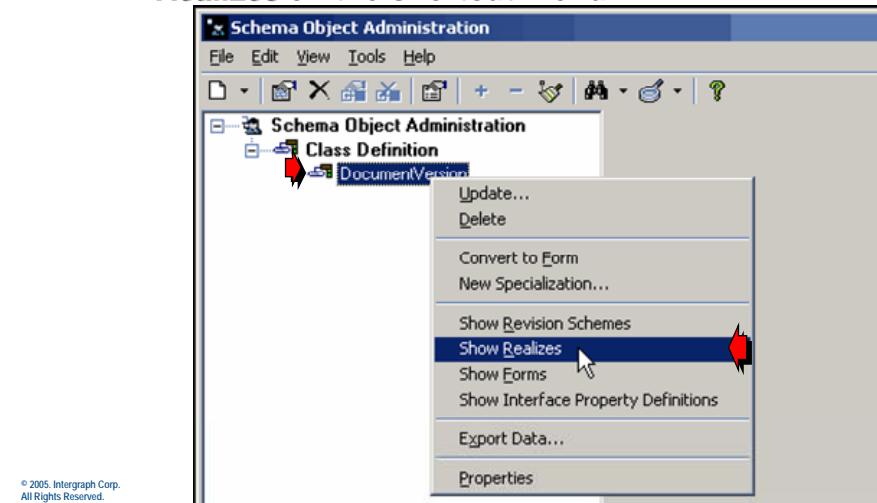


Select the *DocumentVersion* class def in the tree view.

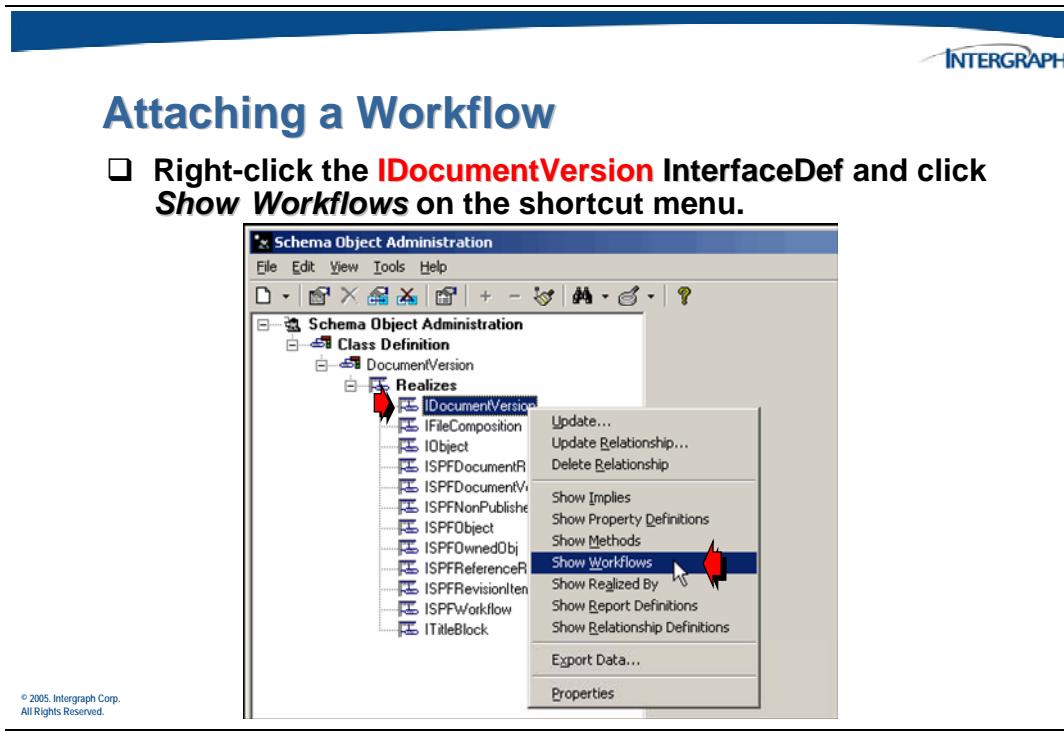


Attaching a Workflow

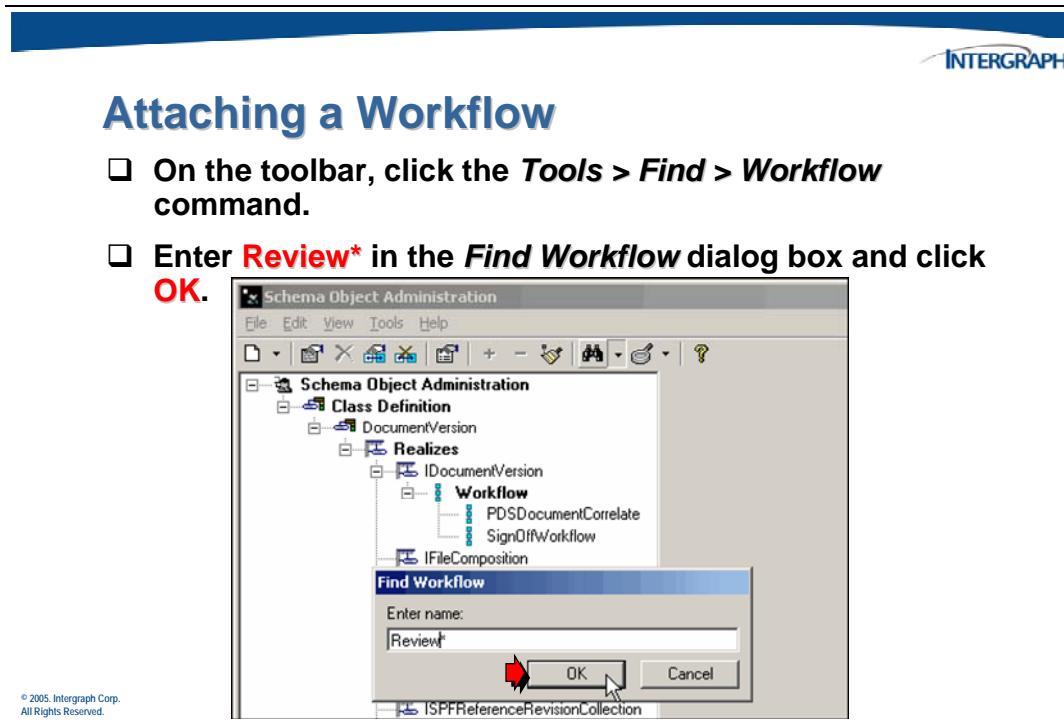
- Right-click the **DocumentVersion** ClassDef and click **Show Realizes** on the shortcut menu.



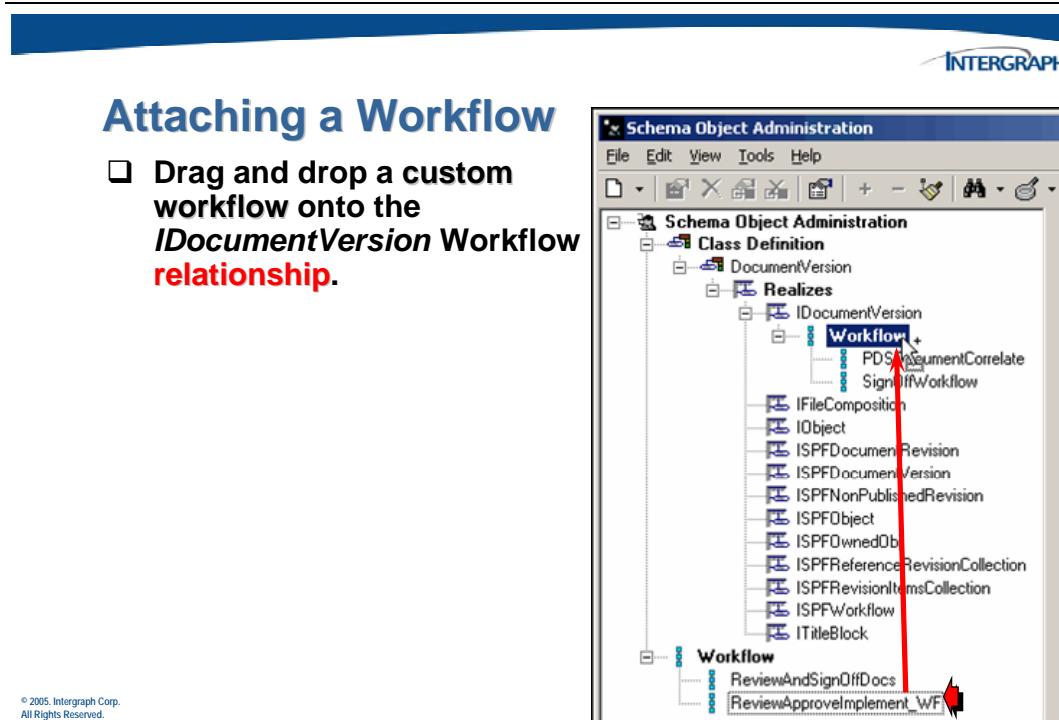
Next, view the interfaces that are realized by the class *DocumentVersion*.



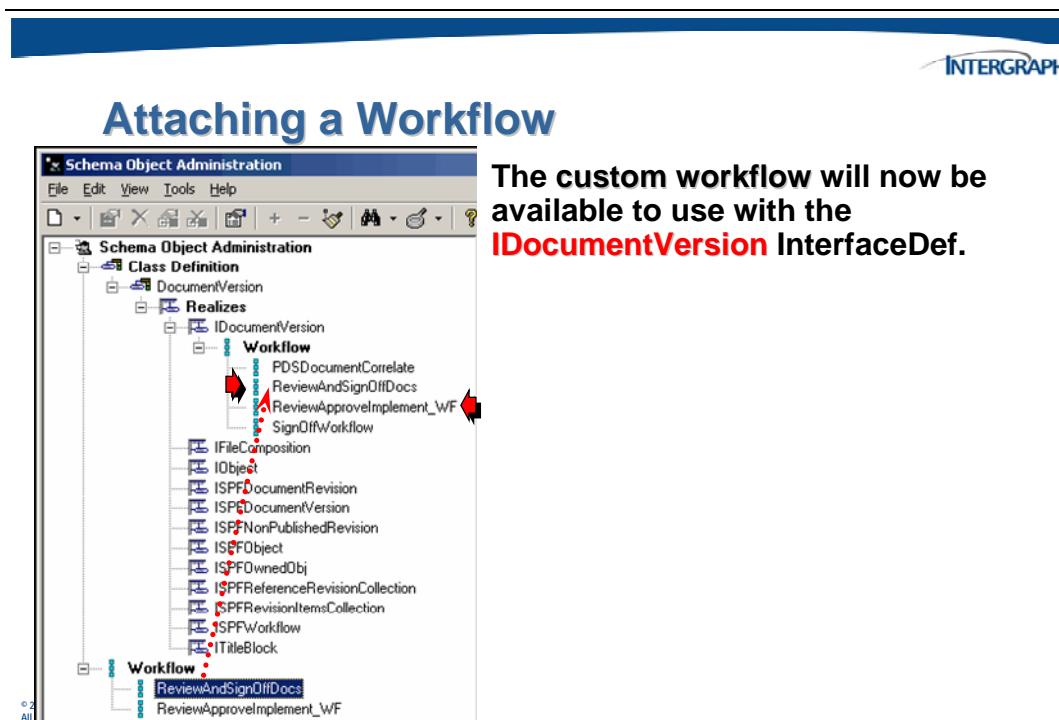
Use the **Find Workflow** dialog to enter the search criteria to locate a custom workflow.



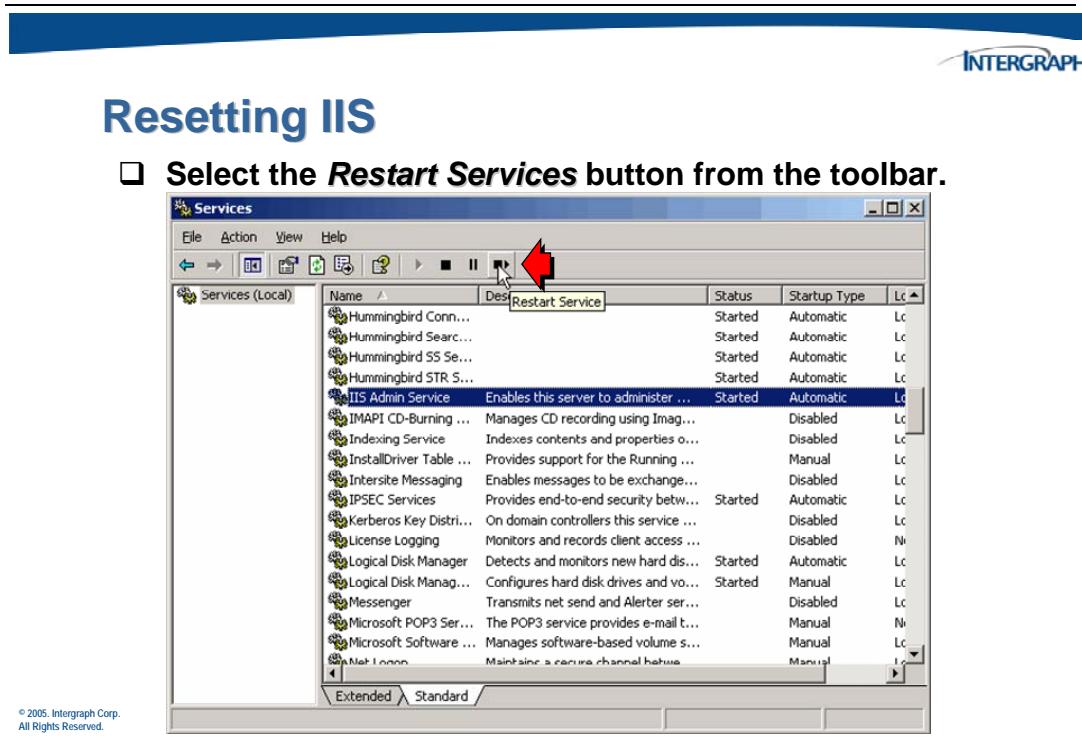
Select the custom workflow, and drag it onto the *Workflow* relationship for the InterfaceDef in the tree view.



Repeat this procedure for any additional workflows that you want to be available to select from the Desktop Client.



You will need to restart your IIS services once the new workflow is attached.



6.5 Other Workflow Functions

SPF Change Management Administration provides several change management functions, which are available and configurable from the **Tools** menu. You can configure the following options:

- E-mail Options
- Change Statuses
- Access Control
- Maintain Non-working Days

The **Options** menu provides two utilities to configure CM Admin options. The **Maintain Calendar Work Week** utility allows you to set non-working days to calculate planned and target dates for workflow steps. The **Options** utility provides a method for configuring e-mail, change statuses, and access control options.



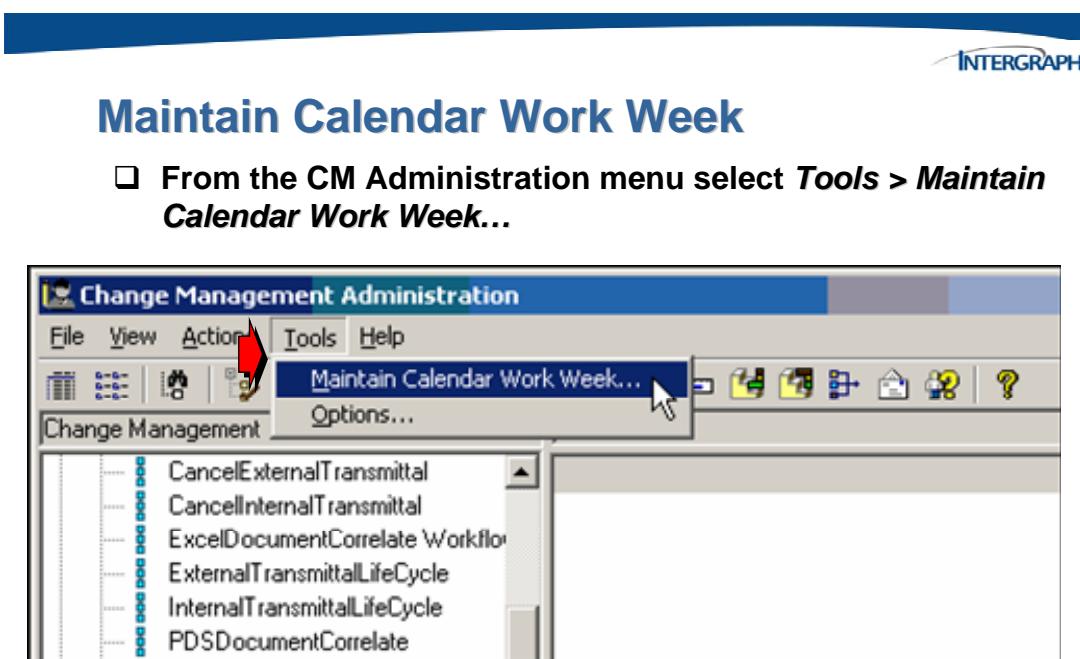
Other Workflow Functions

The “Maintain Calendar Work Week” function allows the configuration of a calendar for duration calculation.

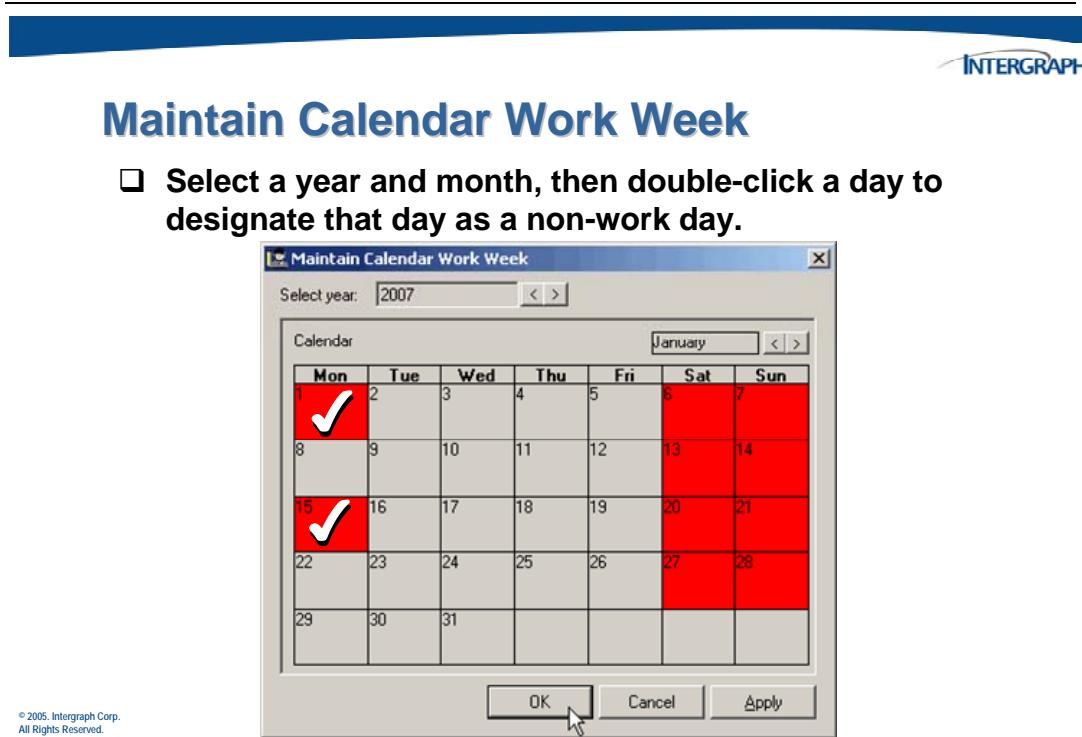
The “Options” function allows you to configure emails, statuses, access control and process control.

6.5.1 Maintain Calendar Work Week

The **Maintain Calendar Work Week** command enables the administrator to set the non-working days such as weekends and holidays. The administrator is able to maintain not only the current years' non-working days, but also past and future years (including leap years).



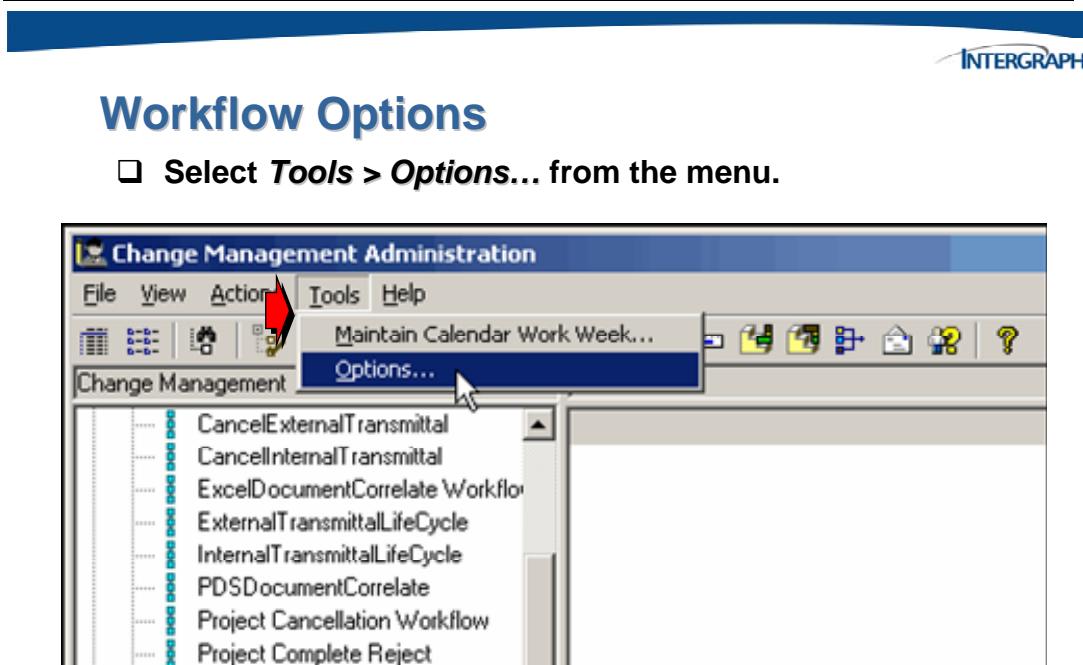
The *Maintain Calendar Work Week* dialog will display.



When a step is processed in a workflow, it has an associated duration specifying the number of days allotted for completion of the step. In order to calculate the planned (or target) completion date, the software must access the calendar settings to determine the number of working days it will take to complete the step. For example, if the step has a duration of three days starting on the 10th, the target date for completion would be the 16th (inclusive day's calculation) because of the non-working days on the weekend and a Monday holiday.

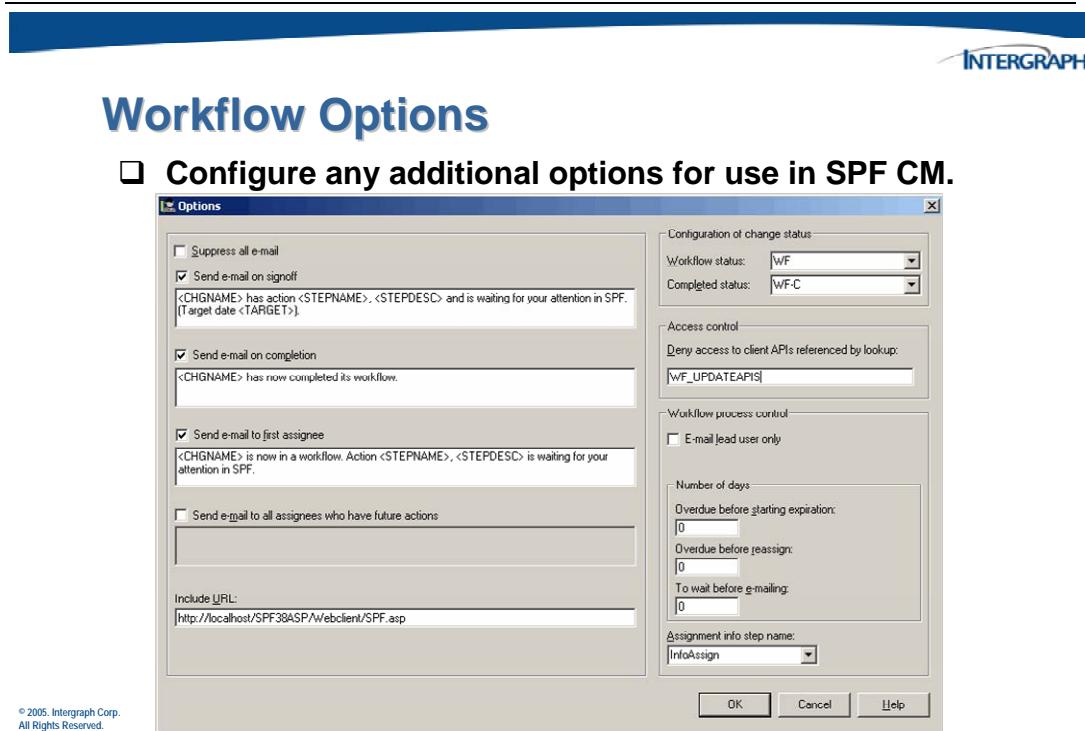
6.5.2 Options

The **Options** command displays the *Options* dialog box used to configure e-mail notification regarding workflows assigned to respective users. Using the **Options** dialog box, you can set the URL path for your **SPF Web Client** module where you can display your To Do List.



© 2005, Intergraph Corp.
All Rights Reserved.

Using reserved words, you can also configure all e-mail text for *SignOff*, *Change* or *Sub-Change Completion*, and *Attaching a Workflow* notifications.



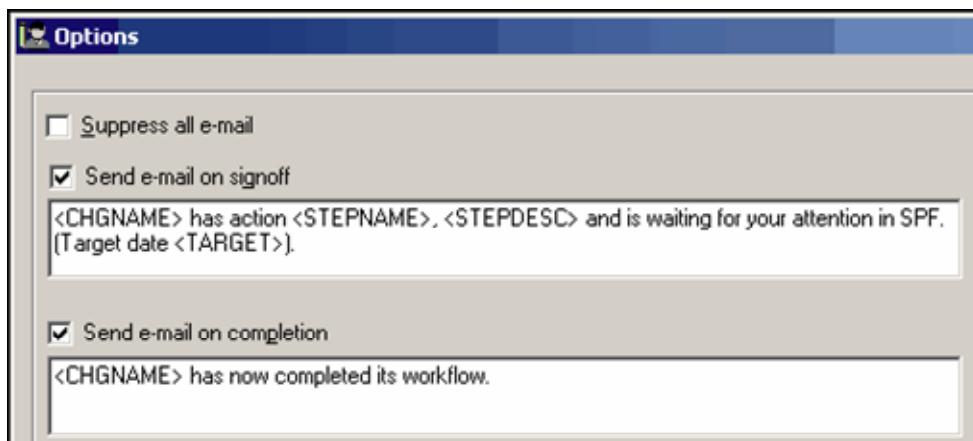
The left part of the dialog provides the email text for each operation. **Suppress** turns off email. *Signoff*, *Completion*, *first Assignee*, and *Future assignees* in a later step are enabled or disabled (checked is enabled) and the text string listed will be sent. The entry <STEPNAME> is replaced with the step name from the workflow. The URL is added for the web address to SPF.

Note: The URL value in this example is <http://localhost/SPF38ASP/Webclient/SPF.asp>.
The URL value will depend on the configuration of the web client within Server Manager.



Workflow Options

Zoomed in view of the *Options* dialog.

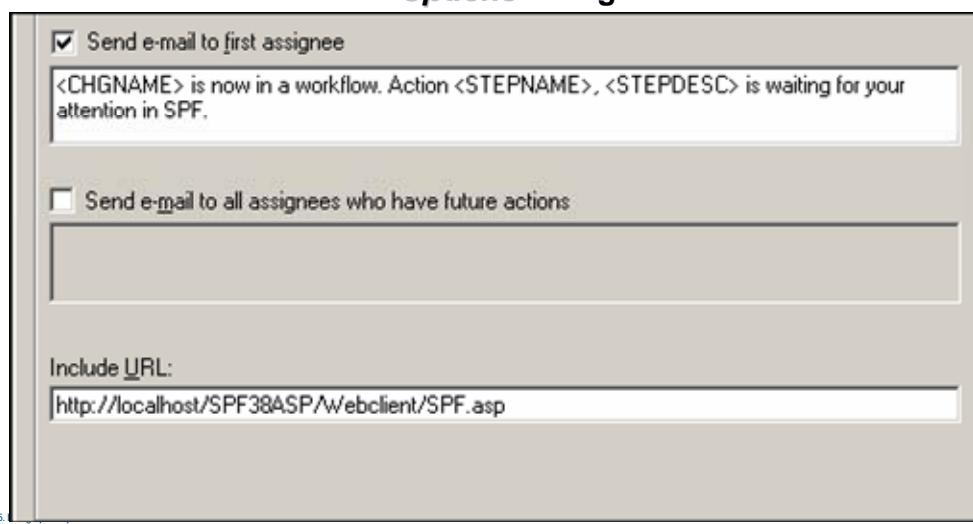


© 2005, Intergraph Corp.
All Rights Reserved.



Workflow Options

Zoomed in view of the *Options* dialog.

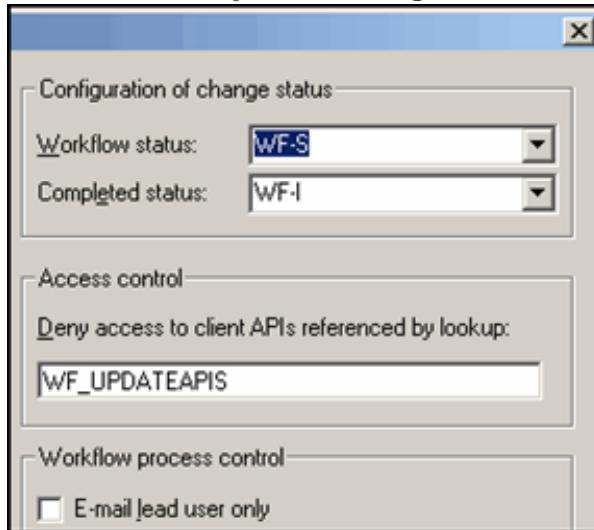


© 2005
All Rights Reserved.



Workflow Options

Zoomed in view of the *Options* dialog.



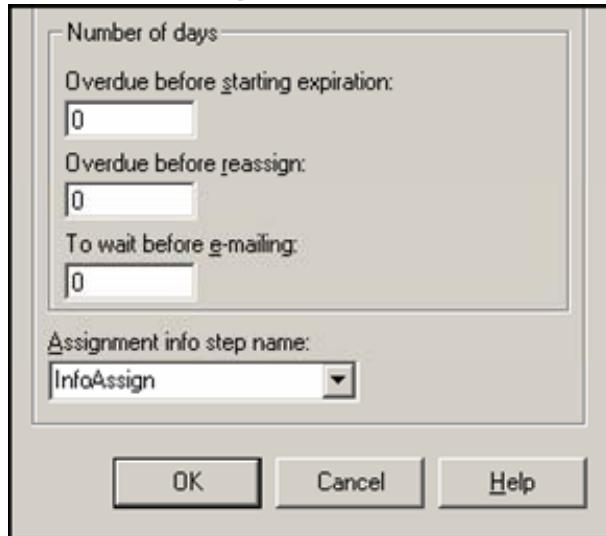
The right part of the dialog defines the following parameters:

- Configuration status** - the default status of starting the first and last step of all workflows, unless changed by the workflow.
- Access control** - lookup which lists API's, which are not available in a workflow. This option is currently not enabled.
- Workflow control** - enabled will send email to the lead user assigned in the User group.



Workflow Options

Zoomed in view of the *Options* dialog.



© 2005, Intergraph Corp.
All Rights Reserved.

- Number of days** - sets the number of days for each command to run.
- Assignment info step name** - specifies the step that is displayed in the To Do List when a user is assigned as an *Info Only* recipient on an Assignment step.

6.6 Using Workflows

Workflows allow you to record a process using a template. If a user wants to add a workflow to an interface definition for which no workflows have been previously added to the interface definition, they will be presented with a list of all workflows available in the system for the class of the selected item.

Since the workflow has been configured to send email notifications, open the email client.



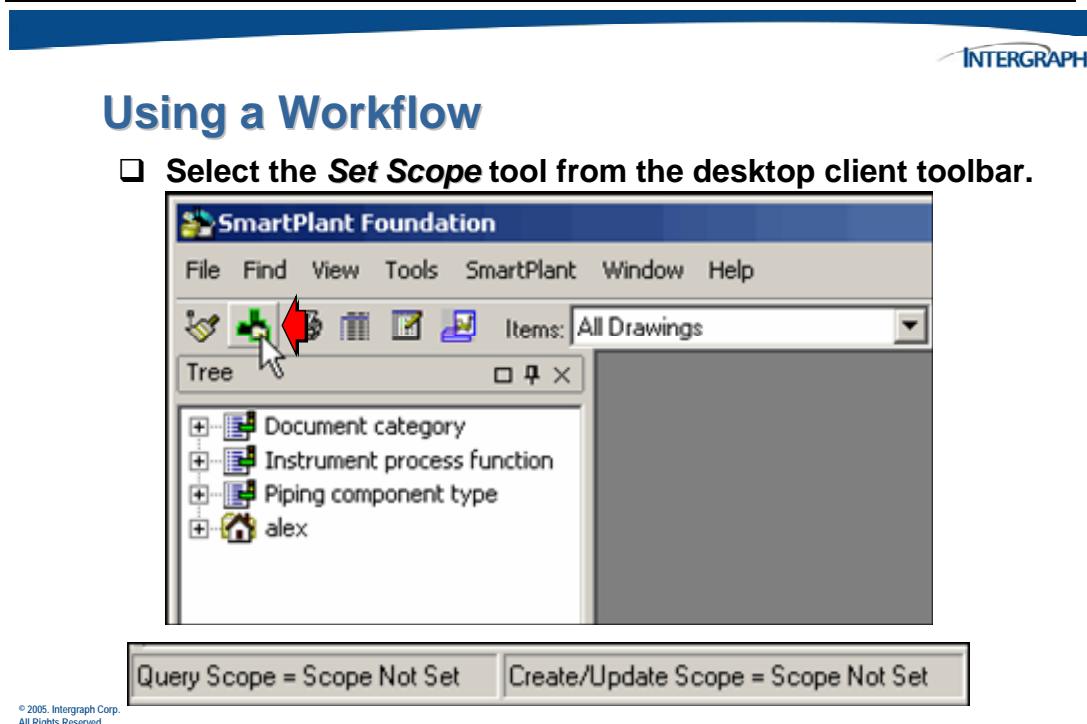
Using a Workflow

- Double-click on the Outlook Express email icon from the Windows desktop.

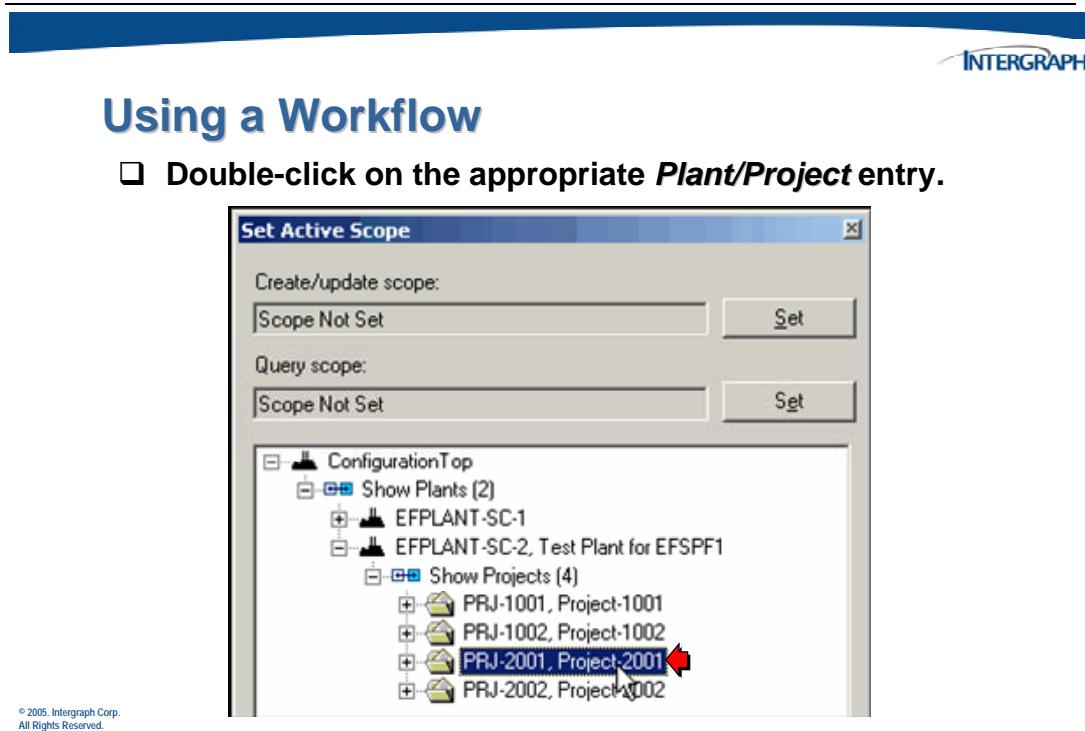


6.6.1 Setting the Active Scope

Before creating objects or searching for objects, a user must first set the active scope. This will specify a *Plant* and a *Project* with which the user will be confined for the operations.



The *Set Active Scope* dialog will display.



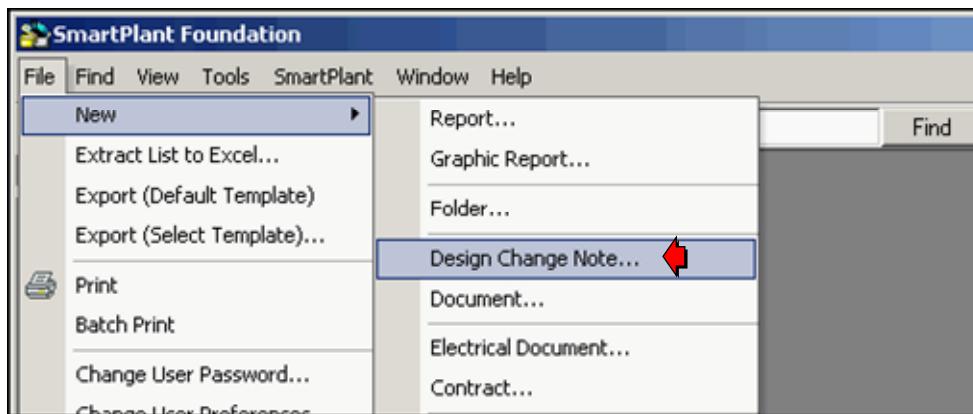
6.6.2 Submitting a Document to the Workflow

The first step is to create an object, such as a document, which will be used by the workflow in a custom review/approval process.



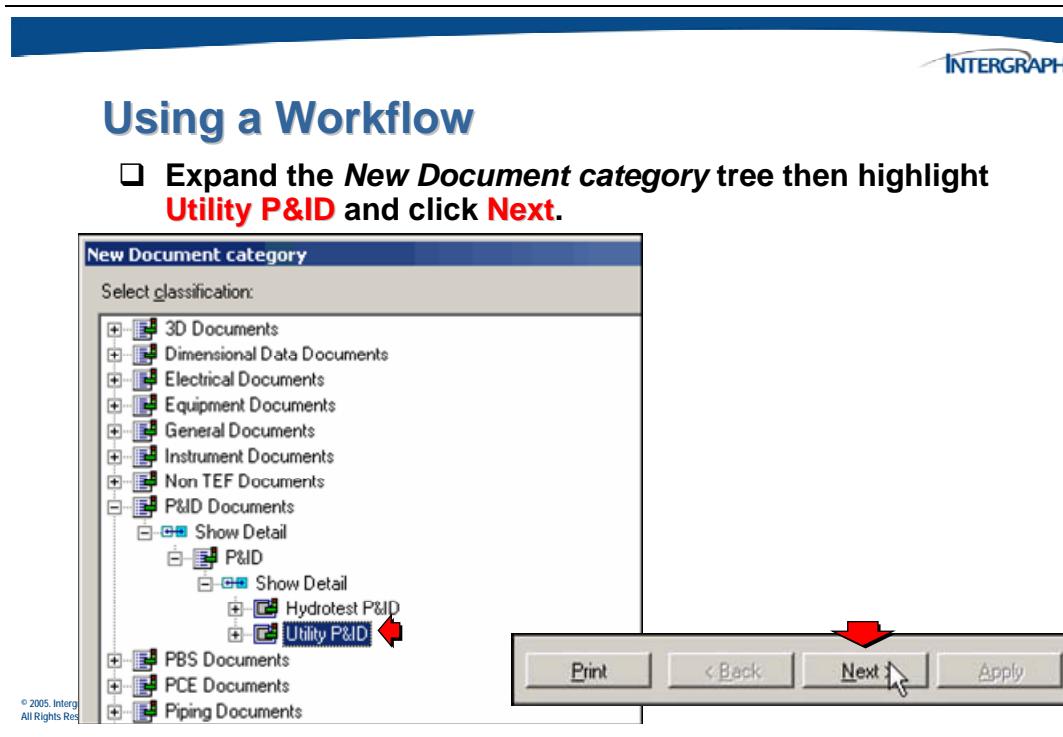
Using a Workflow

- Select **File > New> Design Change Note...** from the desktop client menu.

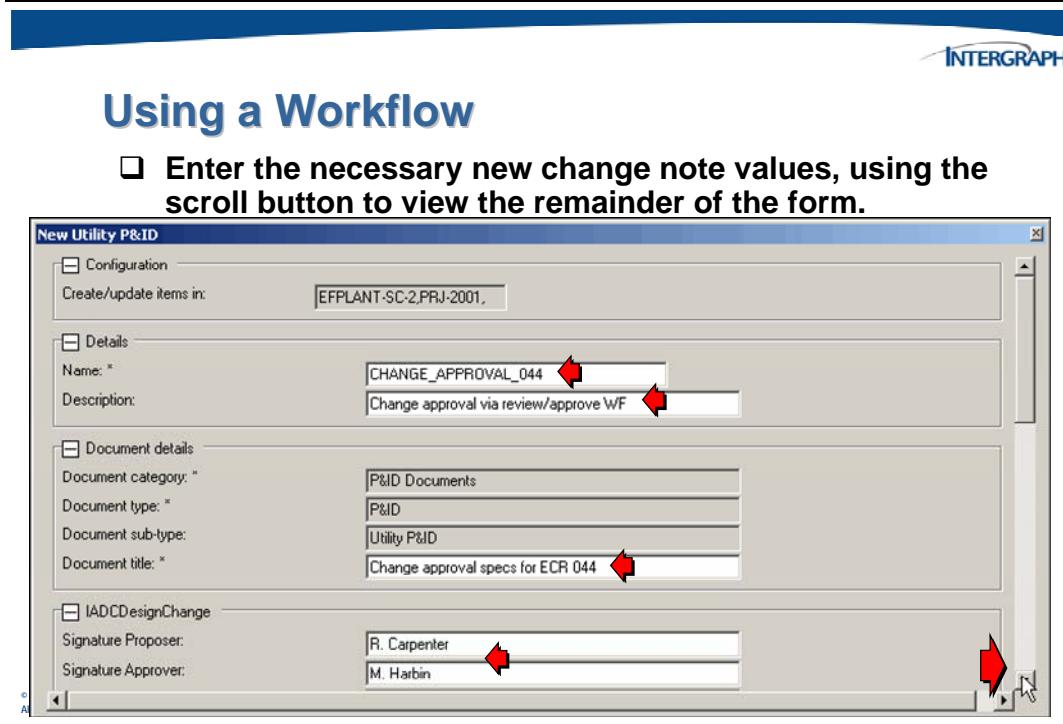


© 2005. Intergraph Corp.
All Rights Reserved.

A New DocCategories window will display.



The New Utility P&ID create document dialog will display.





Using a Workflow

- Continue entering the new document values and scrolling.

Change Detail: N/A

Change Impact: TBD

Change UOM: m^3/s

Designed Item Status

Status (SPF): CR

Owner

Owning group: UPDATE

Configuration scope

Project: PRJ-2001

Optional interfaces

Document items

Document versions

Drawing representations

© 2005, Intergraph Corp.
All Rights Reserved.



Using a Workflow

- Finish entering the desired values and click **Next**.

Referenced documents

Document version

Rev Scheme Name: * Rev1A

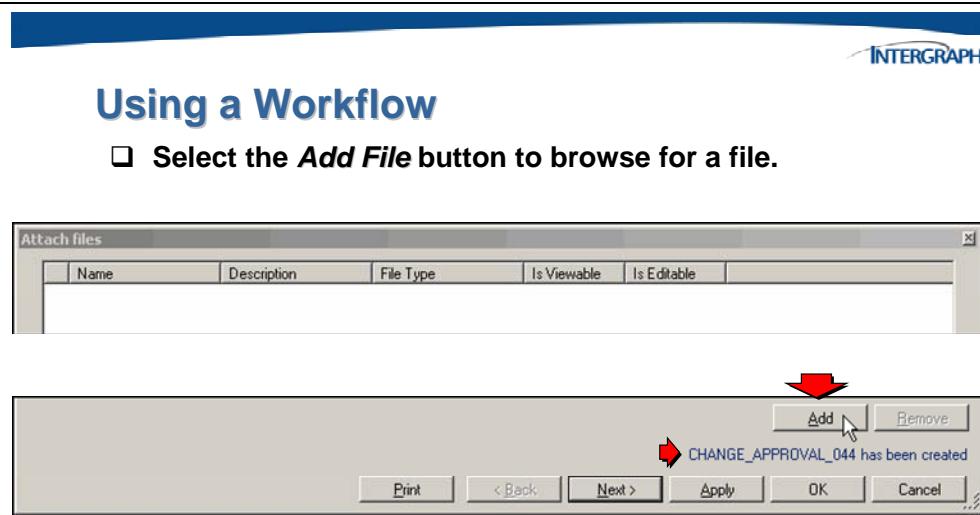
Major Revision: * 1

Minor Revision: A

Include revision details? Collapse sections automatically?

Next >

An electronic file can be attached to the document and processed by the workflow. The *Attach files* dialog will display allowing you to attach a file to this new document. This procedure can also be performed at a later time.

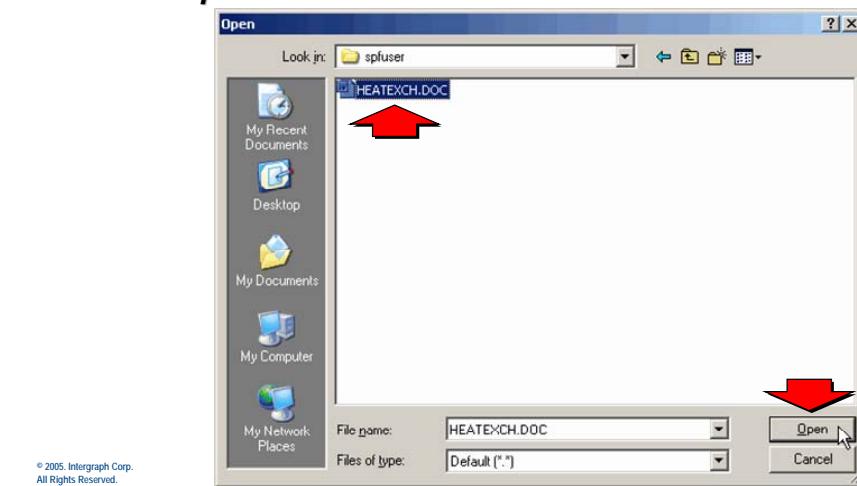


© 2005, Intergraph Corp.
All Rights Reserved.

A folder browser window will display. Use this browser to locate the file that will be attached to the new document. The file can reside in any folder on the client.

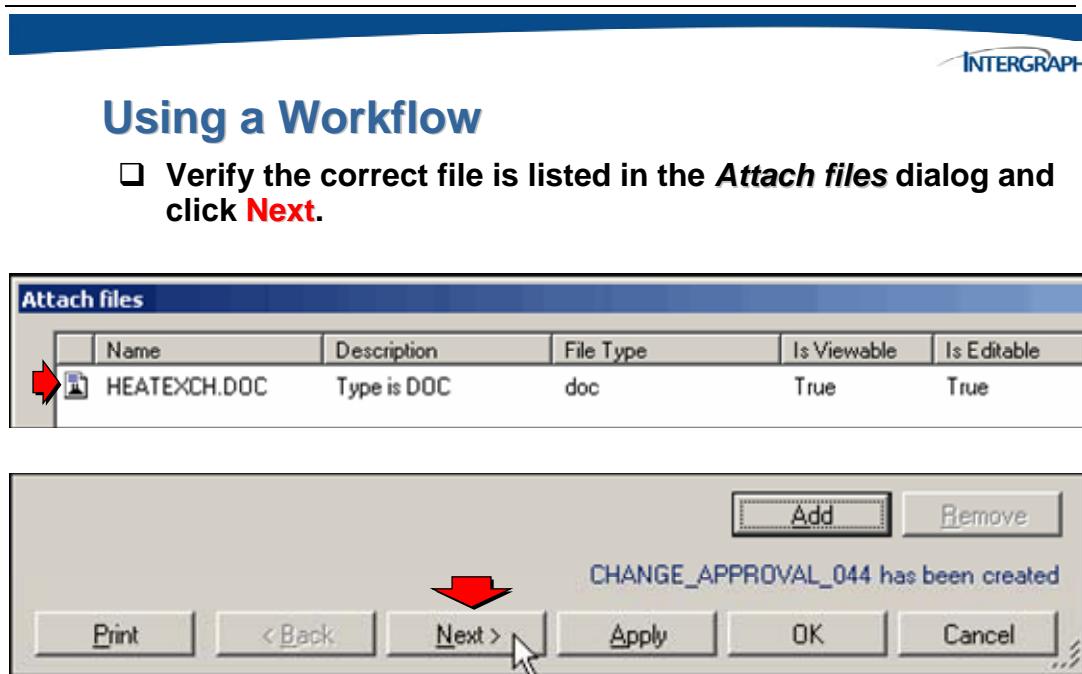
Using a Workflow

- Open a folder and choose a file to be attached then select **Open**.



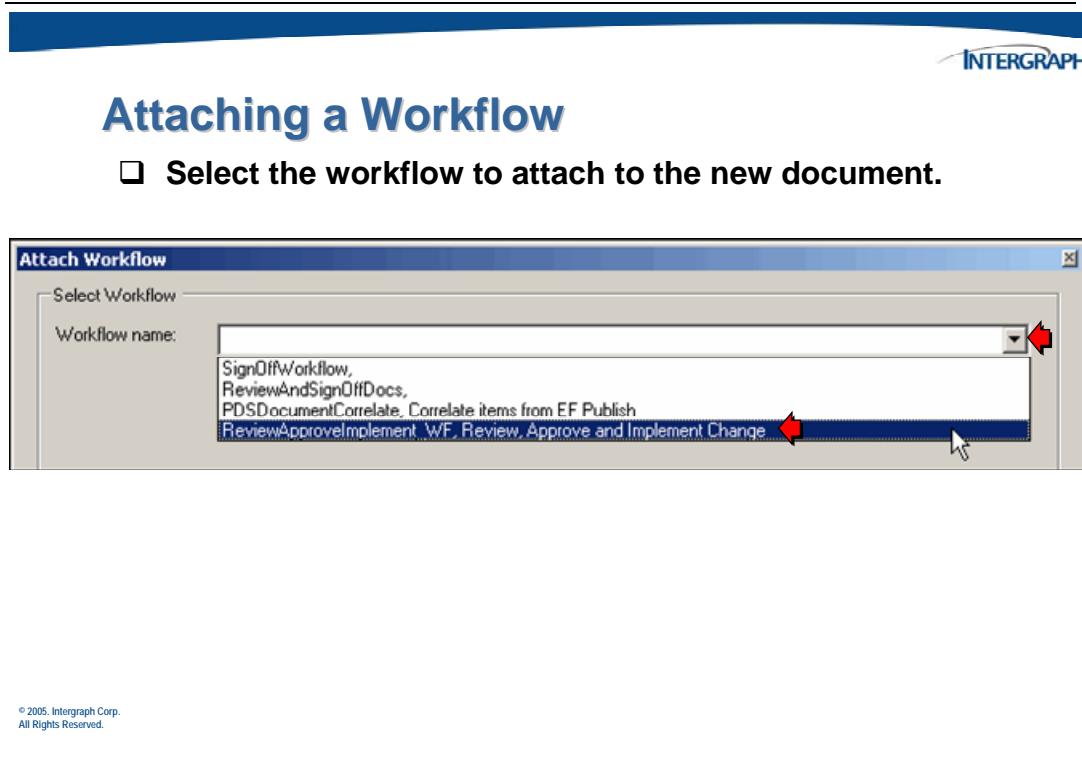
© 2005, Intergraph Corp.
All Rights Reserved.

The selected file will be displayed in the *Attach files* window.



6.6.3 Attaching a Workflow

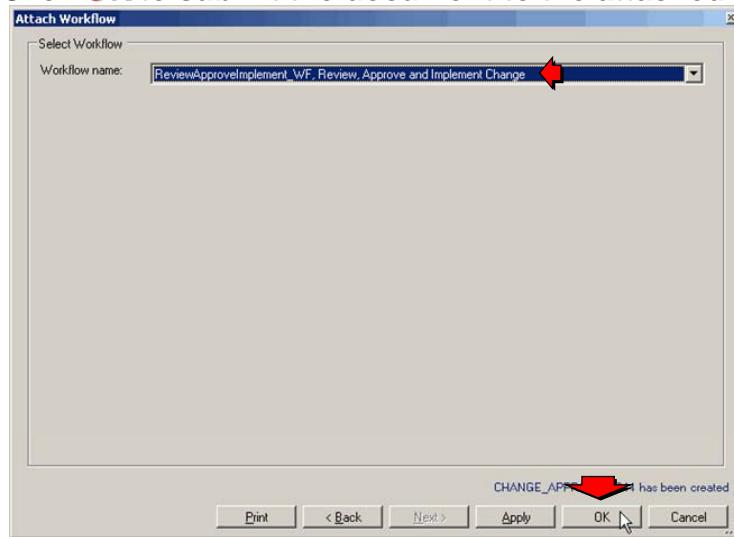
In order to submit the document and attached file to the workflow, the workflow must first be attached to the document. To start the workflow review and approval process, continue with the following steps:





Attaching a Workflow

- Click **OK** to submit the document to the attached workflow.

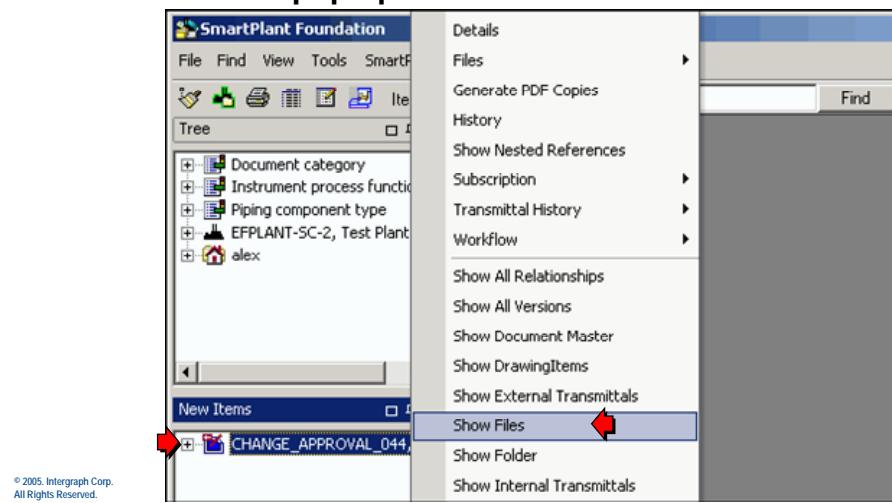


Attaching the workflow will also submit the document to the workflow. Verify that the file was successfully attached to the document.



Verifying Attached Files

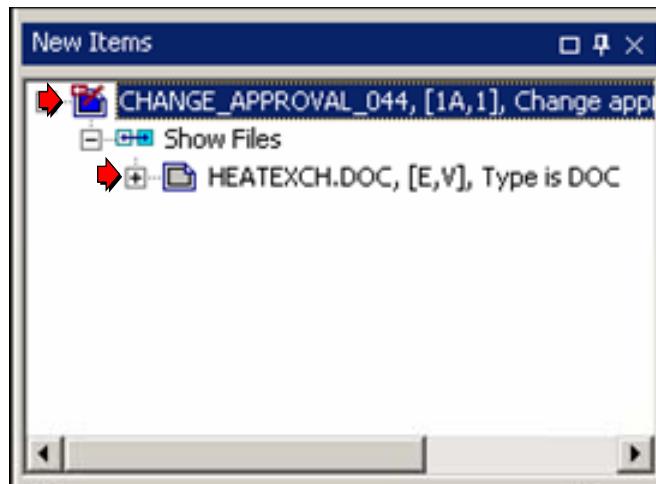
- Right-click on the new **document** and choose **Show Files** from the pop up menu.



The *New Items* window will expand to show the attached file.

Verifying Attached Files

- Verify that a file is attached to the selected document.



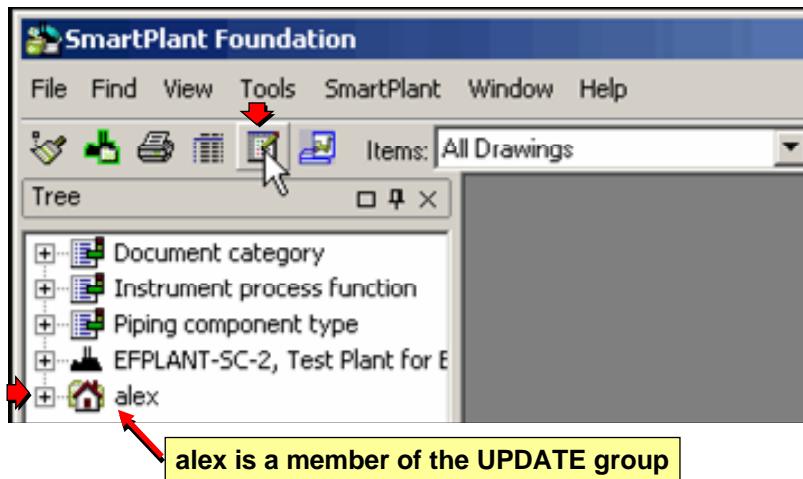
© 2005. Intergraph Corp.
All Rights Reserved.

6.6.4 Viewing a Workflow

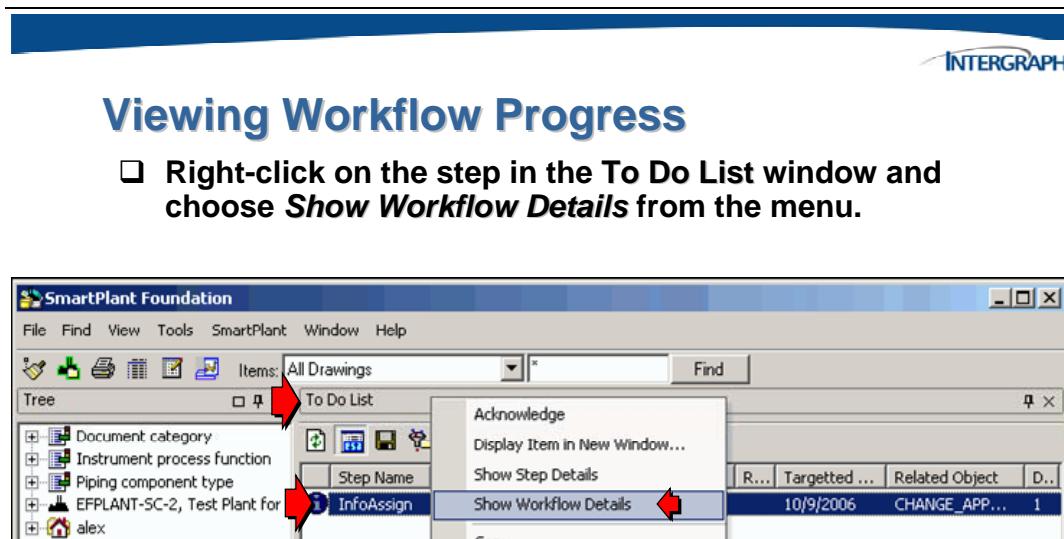
Once an object has been submitted to the workflow, the active workflow can be displayed graphically. The first recipient in the list for the selected workflow, *ReviewApproveImplement_WF*, was the **UPDATE** user group. That means that any member of that user group will receive a task in their **To Do List** as well as an email notification.

Viewing Workflow Progress

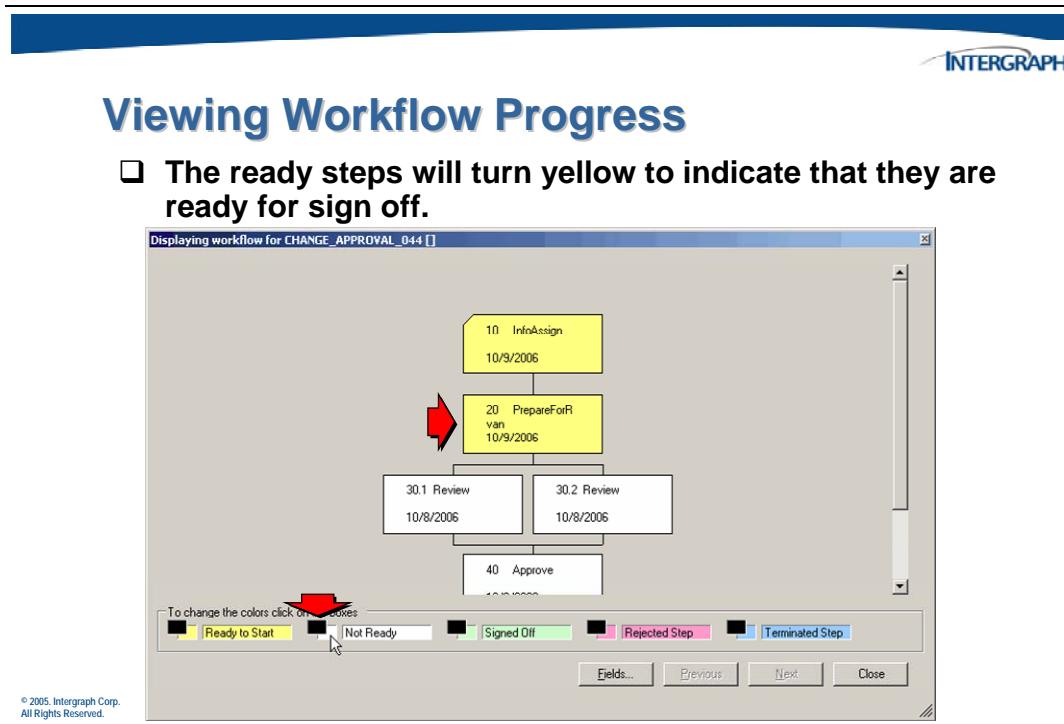
- From the toolbar, choose the **To Do List** icon.



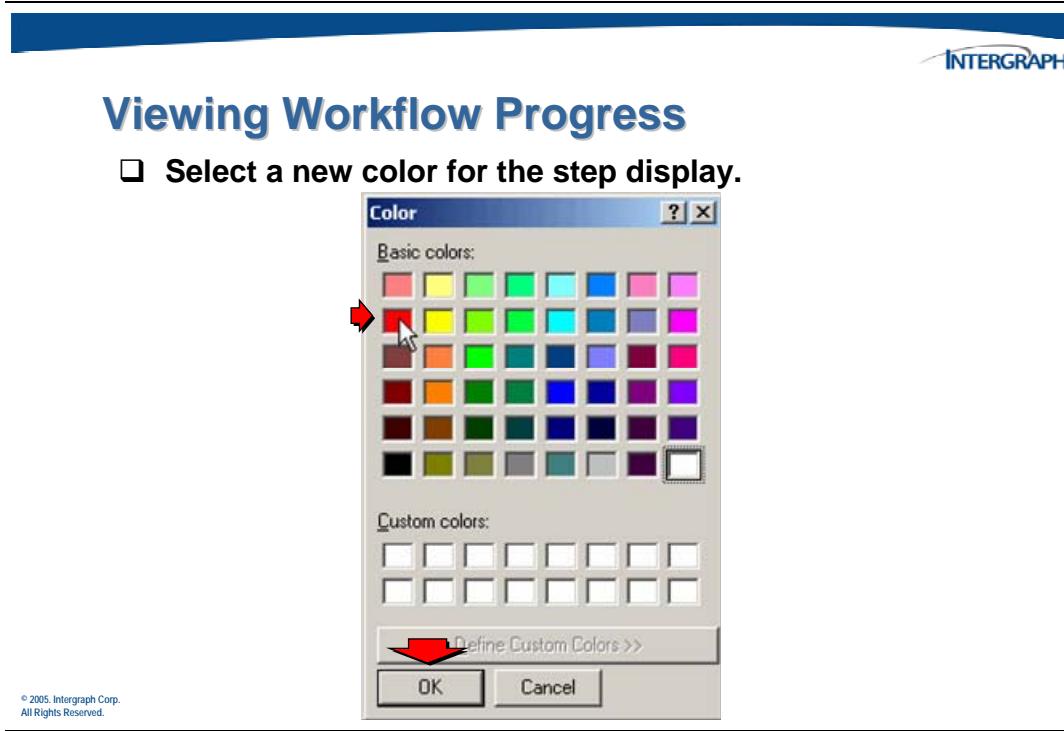
The **To Do List** will show the first step in the workflow, *InfoAssign*.



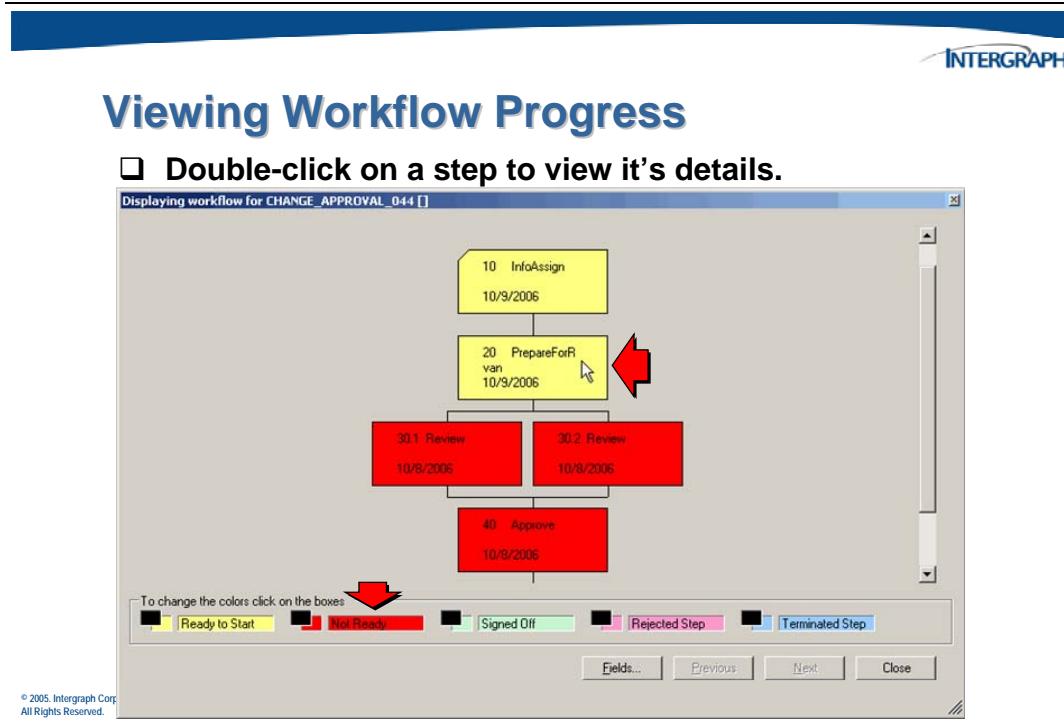
A graphical workflow dialog will be displayed.



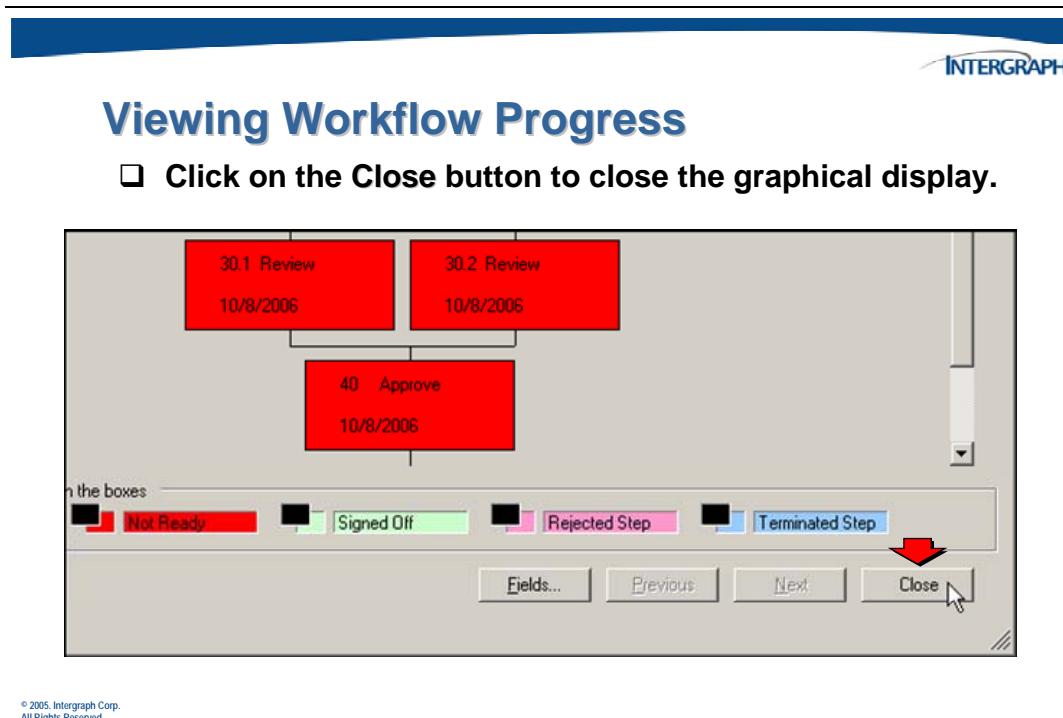
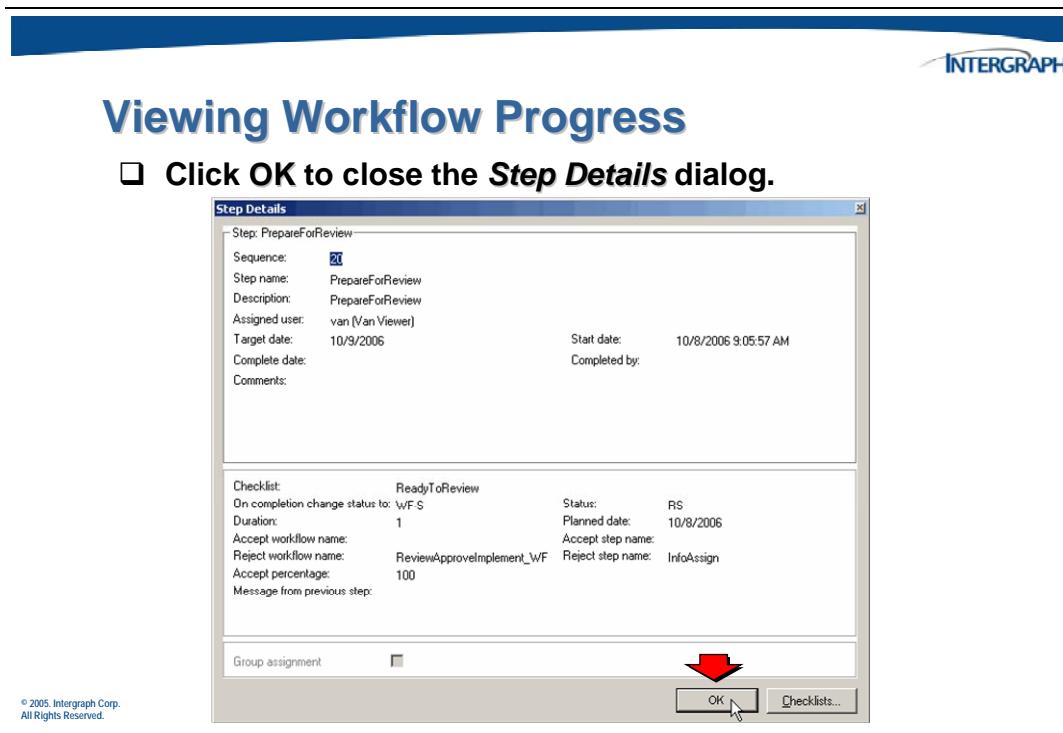
You can configure the display color of any of the workflow steps in the graphical display.



The *Not Ready* steps are now displayed in red. A step's details can also be reviewed.



A *Step Details* dialog window will display.



6.6.5 Workflow Signoffs

A workflow is made up of different steps with one or more users assigned to perform some kind of task within each step.



Using a Workflow

- Log in as different users to test the workflow. Listed below are the recipients that were defined by the workflow.

Maintain Recipients	
---UPDATE	...
van	...
eddie	...
mark	...
updateuser	...
alex	...

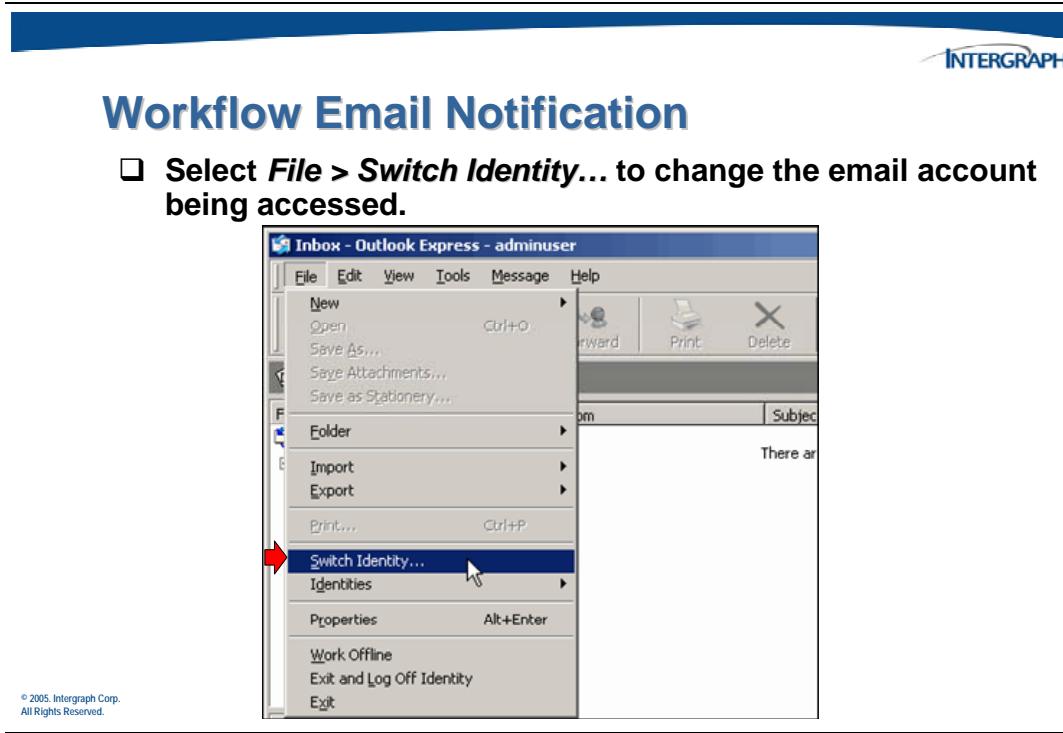
Red arrows point from the recipient names in the table to their corresponding user names listed to the right:

- updateuser → updateuser
- van → van
- eddie → eddie
- mark → mark
- updateuser → updateuser
- alex → alex

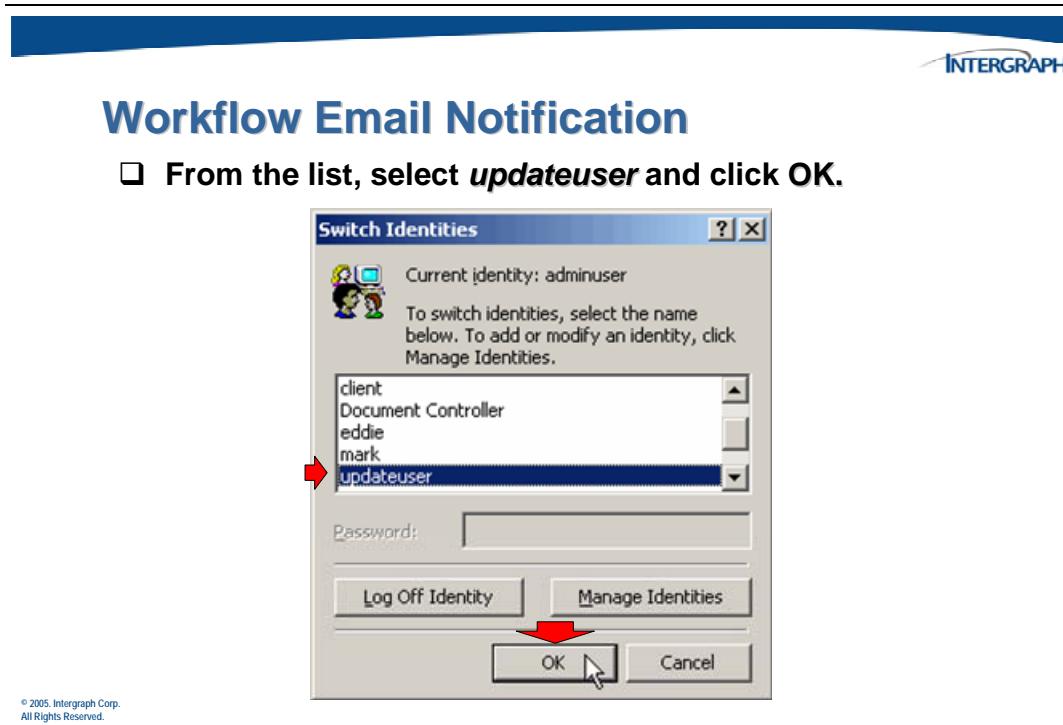
© 2005. Intergraph Corp.
All Rights Reserved.

The recipient for the first step in the workflow will log in to the SPF client and check the **To Do List**. Using the information about the workflow from the previous pages, this recipient should belong to the UPDATE user group.

Submitting a document to a workflow will also send out an email notification. You may have to switch the email client to the correct user.



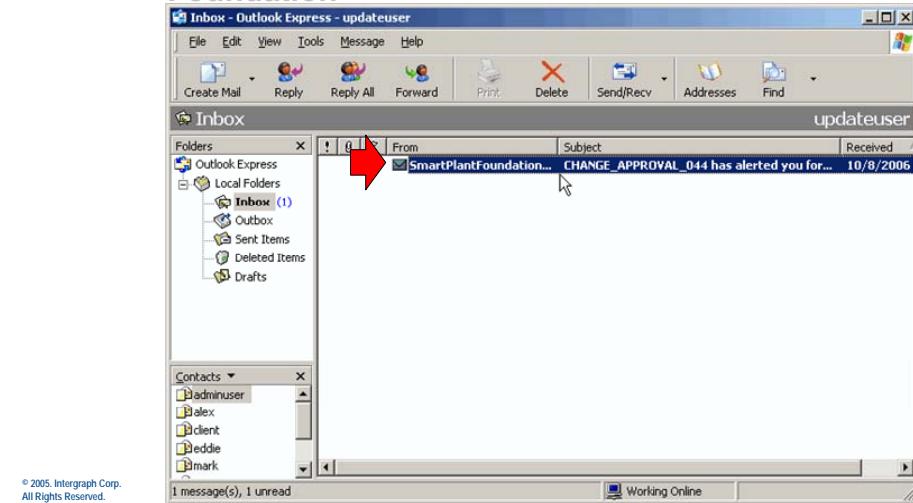
The first user for this particular workflow will be **updateuser**.





Workflow Email Notification

- Double-click to view the mail message sent by SmartPlant Foundation.

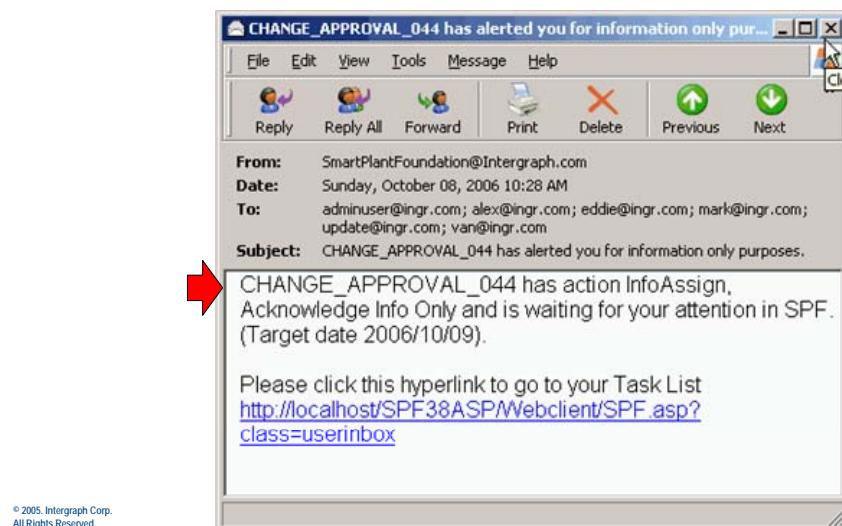


The selected mail message will be opened.



Workflow Email Notification

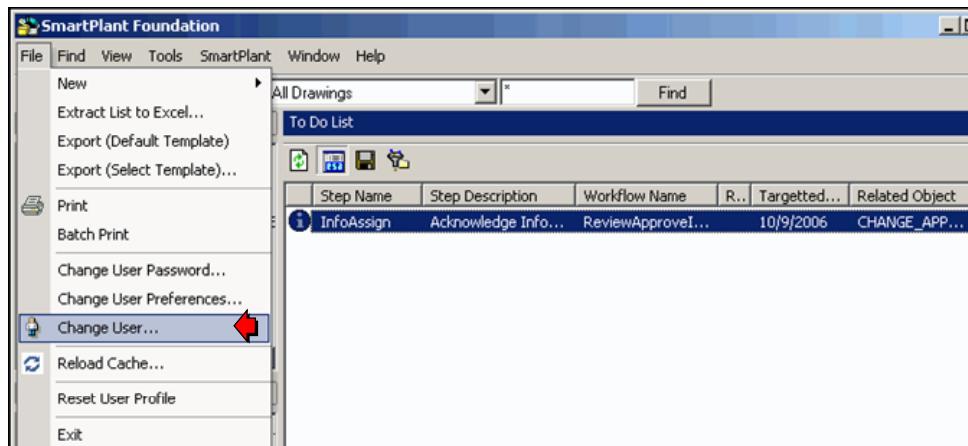
The required action is detailed in the mail message.



Since user **alex** created the new document and submitted it to the workflow, log out and log back in as the first **workflow** recipient.

Using a Workflow

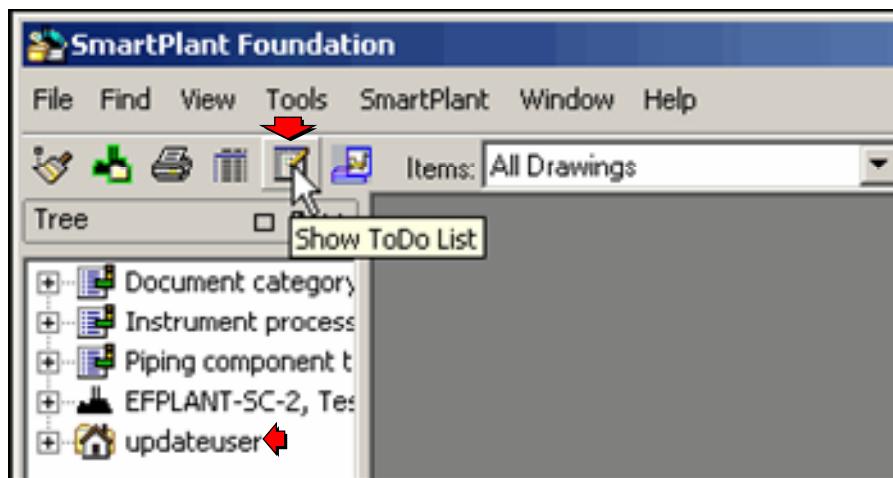
- Select the **File > Change User...** command from the menu.



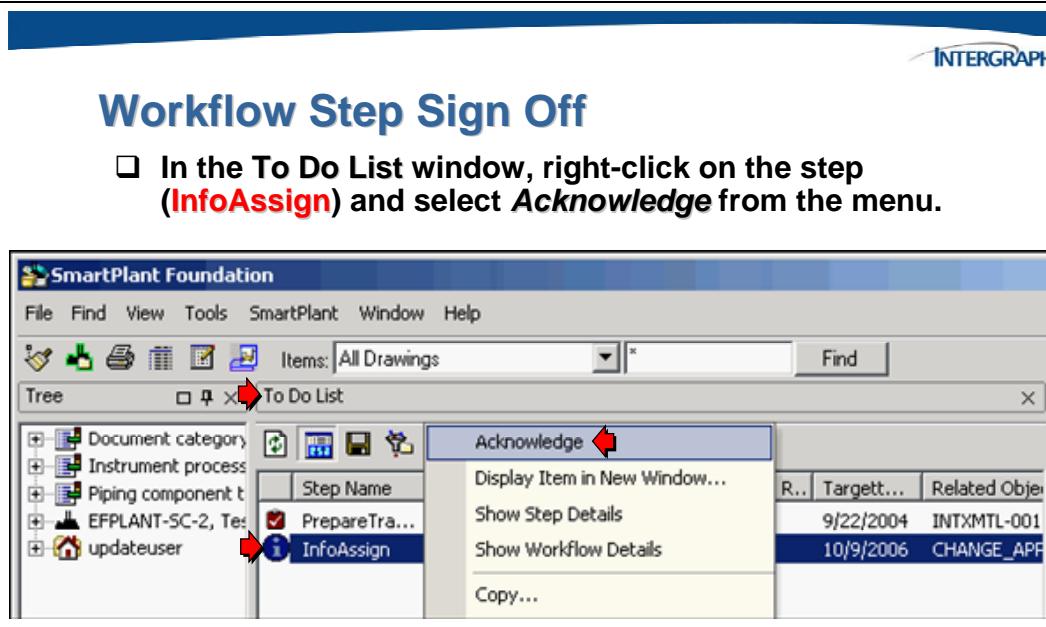
Log back in as user **updateuser**.

Using a Workflow

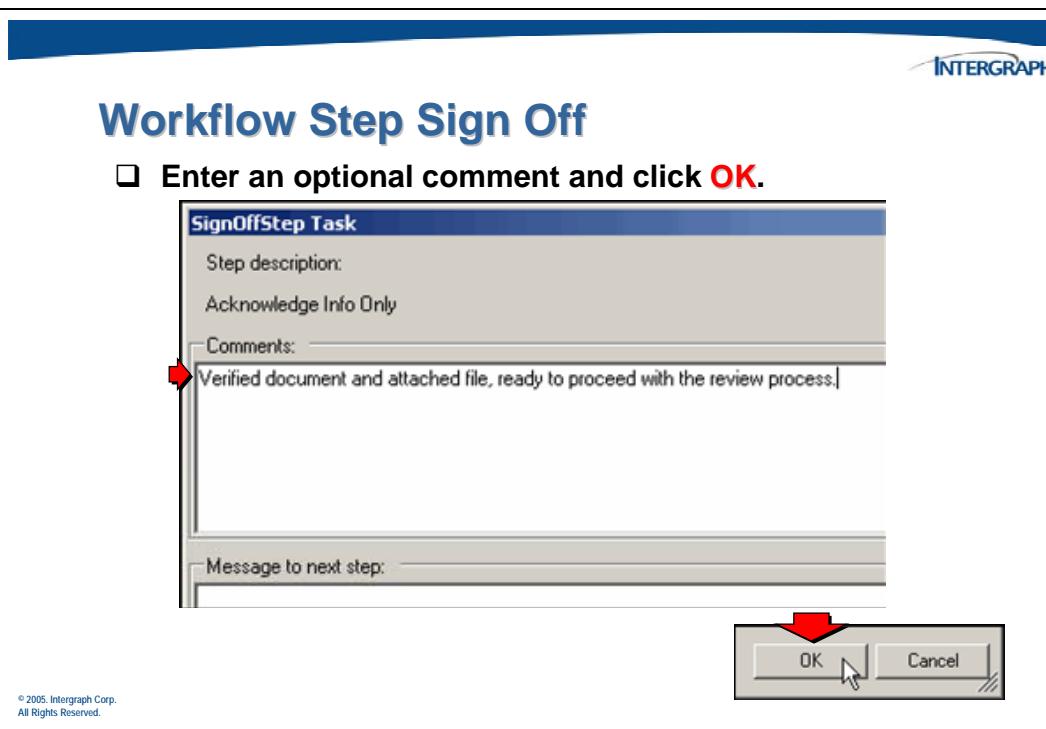
- From the toolbar, choose the **To Do List** icon.



The user will respond to the ***InfoAssign*** step in the *To Do List* by signing off or **Acknowledging** this step.



A *SignoffStep Task* dialog will display that will allow the workflow to capture optional comments as this step is *Acknowledged* by the recipient.

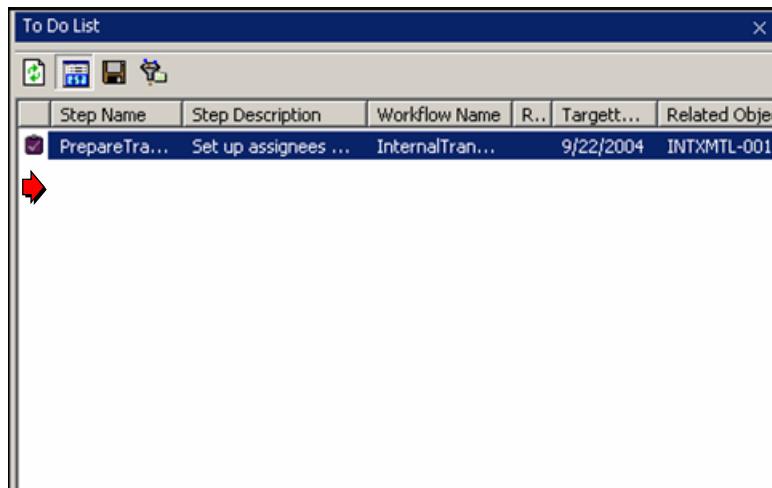


Once the step has been signed off or *Acknowledged*, the recipient no longer has any access to the step and it is removed from the *To Do List*.



Workflow Step Sign Off

- The step will disappear from the To Do List window.



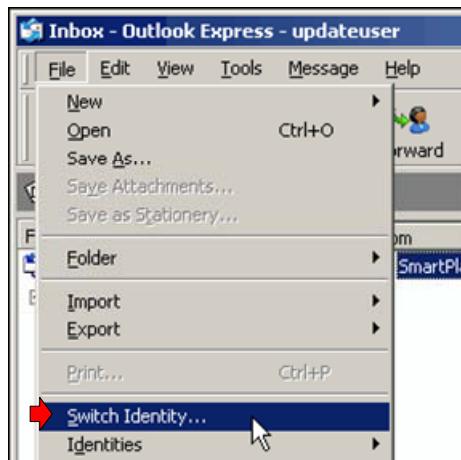
© 2005, Intergraph Corp.
All Rights Reserved.

Restore the email window and change the email account to the next user in this workflow.



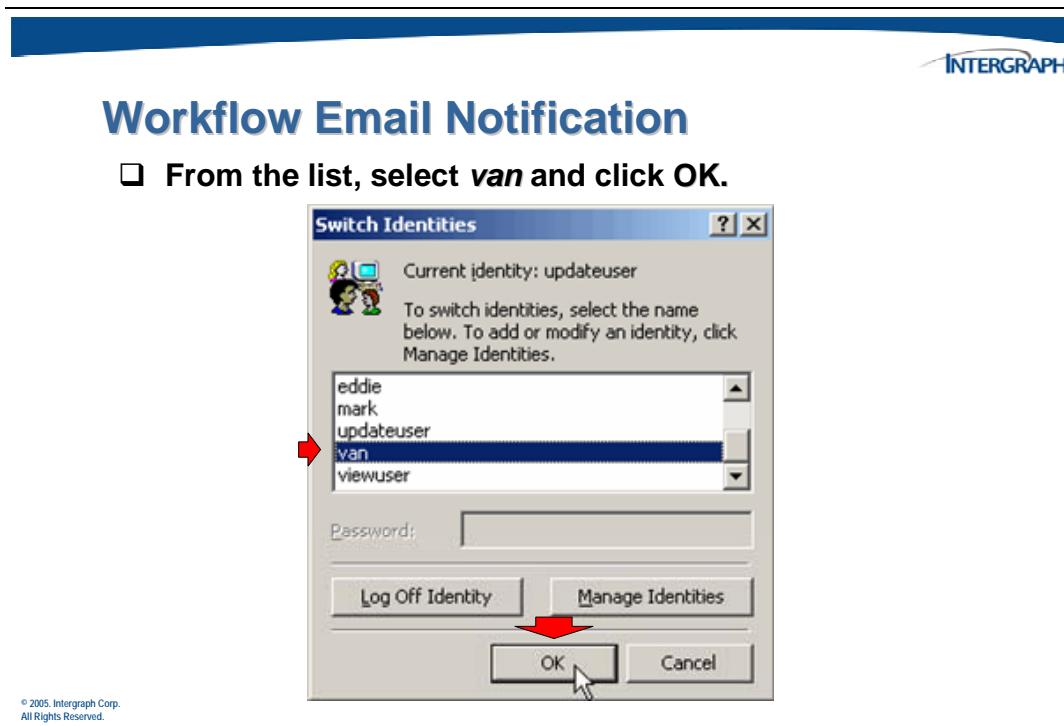
Workflow Email Notification

- Select **File > Switch Identity...** to change the email account being accessed.

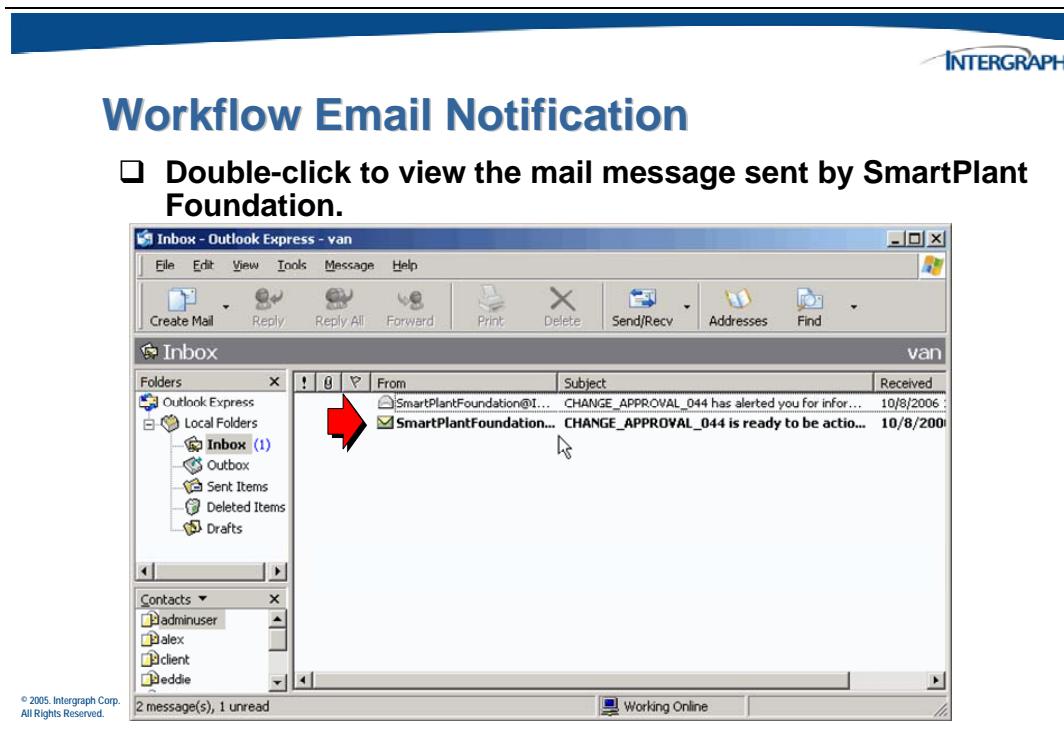


© 2005, Intergraph Corp.
All Rights Reserved.

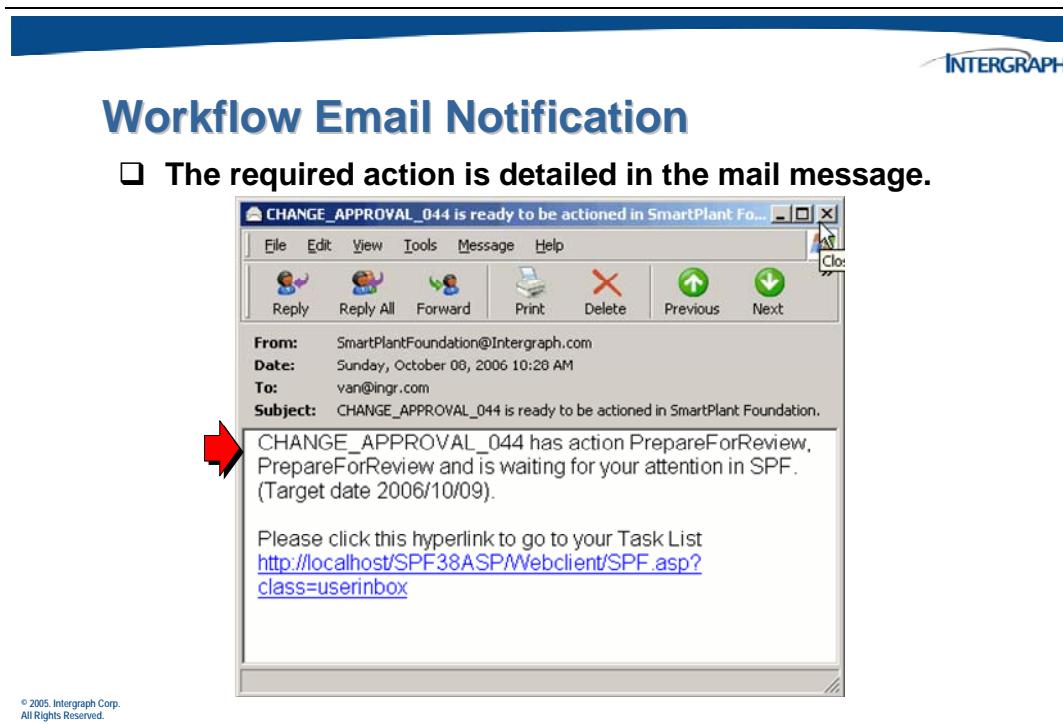
Use the email account **Van View** (van) to review any notifications sent by SPF.



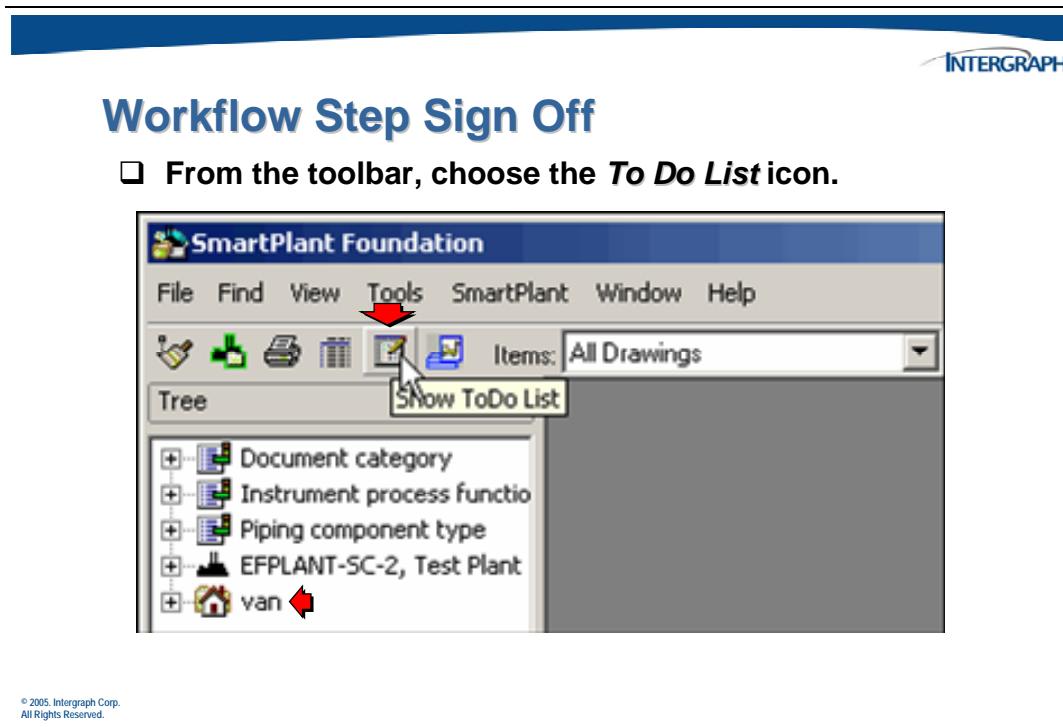
The email *Inbox* for user van will display.



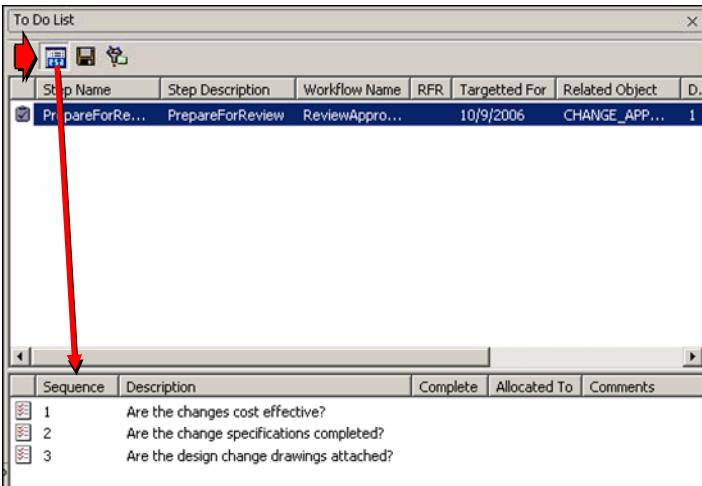
The selected mail message will be opened.



The next recipient in the workflow process will log in to the SPF client. The first action would be to check the *To Do List* for any workflow assignments.



Remember, check lists are a series of tasks or questions that the user can be forced to complete before a step can be completed in a workflow.

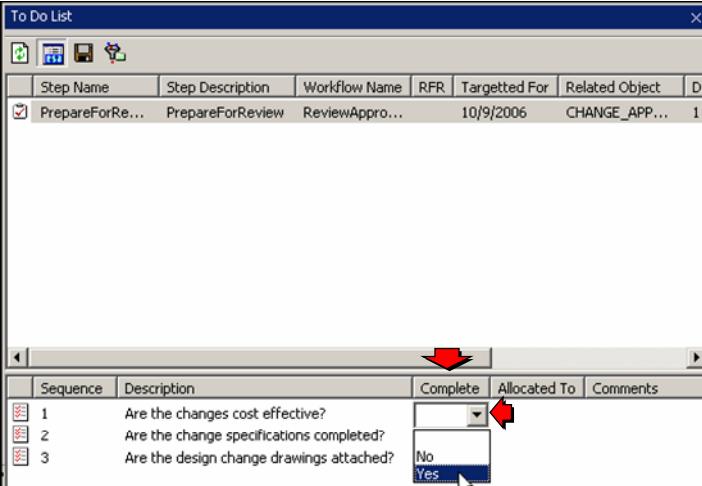


The screenshot shows the 'To Do List' window with a workflow step named 'PrepareForReview'. Below the main grid, a detailed view of the checklist is shown:

Sequence	Description	Complete	Allocated To	Comments
1	Are the changes cost effective?			
2	Are the change specifications completed?			
3	Are the design change drawings attached?			

A red arrow points from the text above to the 'Checklist Preview Pane' icon in the toolbar of the 'To Do List' window.

In this example, the user will complete the check list questions before continuing. In a later example, the check list will be bypassed.



The screenshot shows the 'To Do List' window with the same workflow step and checklist. The first item in the checklist has been checked off:

Sequence	Description	Complete	Allocated To	Comments
1	Are the changes cost effective?	<input checked="" type="checkbox"/>		
2	Are the change specifications completed?			
3	Are the design change drawings attached?			

A red arrow points from the text above to the 'Complete' column of the checklist table, indicating where the user should click to mark items as completed.

Save the results of the completing the checklist.

Completing a Workflow Checklist

- Click the **Save Checklist** icon in the To Do List to save check list results.

Step Name	Step Description	Workflow Name	RFR	Targetted For	Related Object	D.
PrepareForRe...	PrepareForReview	ReviewAppr...	10/9/2006	CHANGE_APP...	1	

Sequence	Description	Complete	Allocated To	Comments
1	Are the changes cost effective?	Yes		Analysis completed
2	Are the change specifications completed?	Yes		Change specs corr
3	Are the design change drawings attached?	Yes		Design drawings n

© 2005, Intergraph Corp.
All Rights Reserved.

Once the check list questions have been answered, the step (*PrepareForReview*) can be signed off. Since the check list was a type of task, the sign off response is **Completed**.

Workflow Step Sign Off

- In the To Do List window, right-click on the step (**PrepareForReview**) and select **Completed** from the menu.

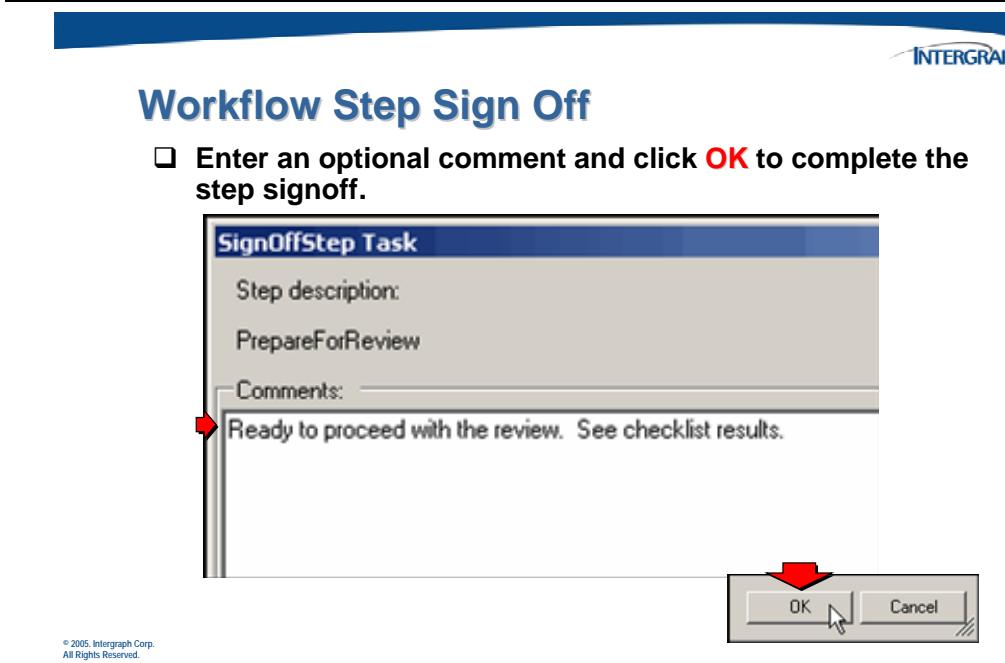
Step Name
PrepareForRe...

Completed

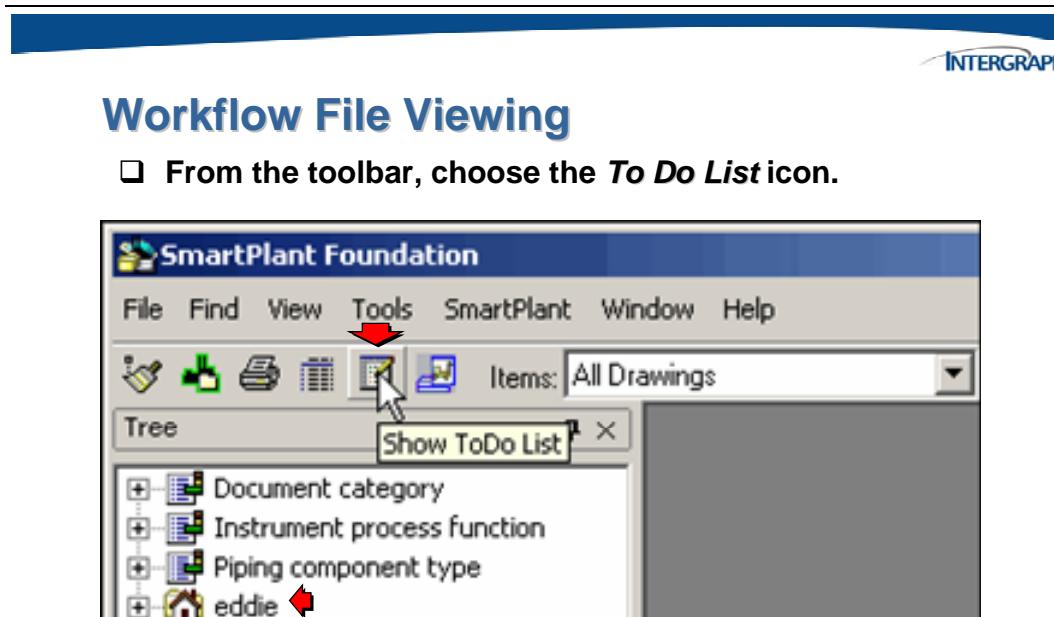
- Display Item in New Window...
- Edit Checklists
- ReAssign
- Show Checklists
- Show Step Details
- Show Workflow Details
- Unable to complete
- Update Workflow...

© 2005, Intergraph Corp.
All Rights Reserved.

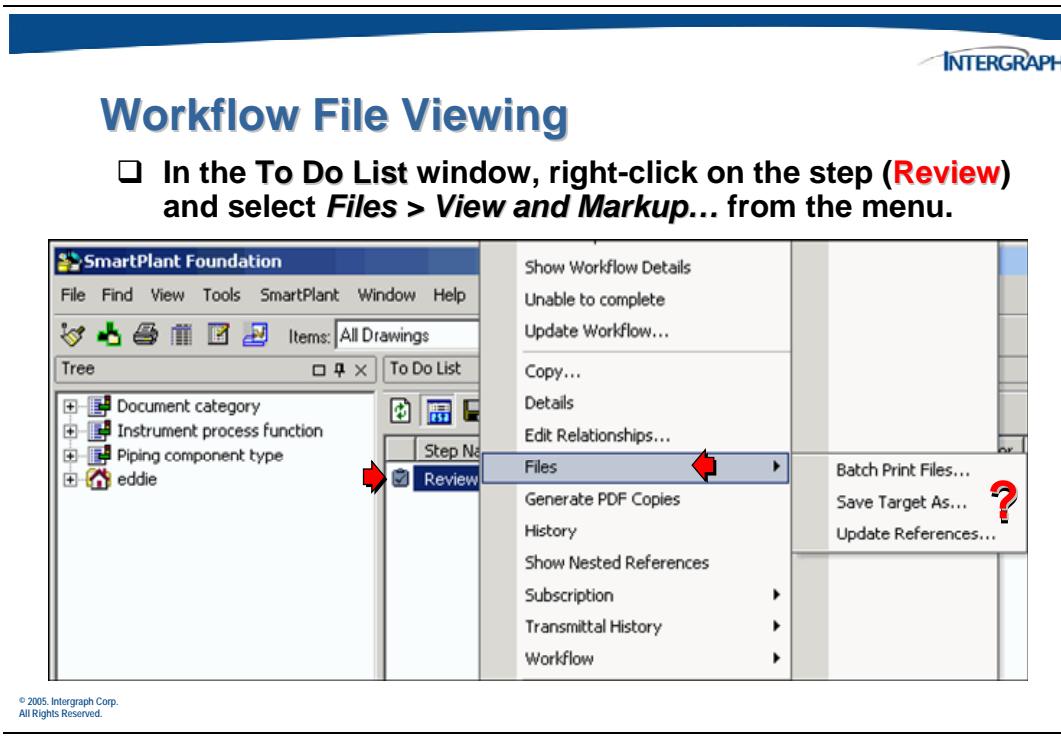
A *SignoffStep Task* dialog will display that will again allow the workflow to capture comments as this step is *Completed* by the recipient.



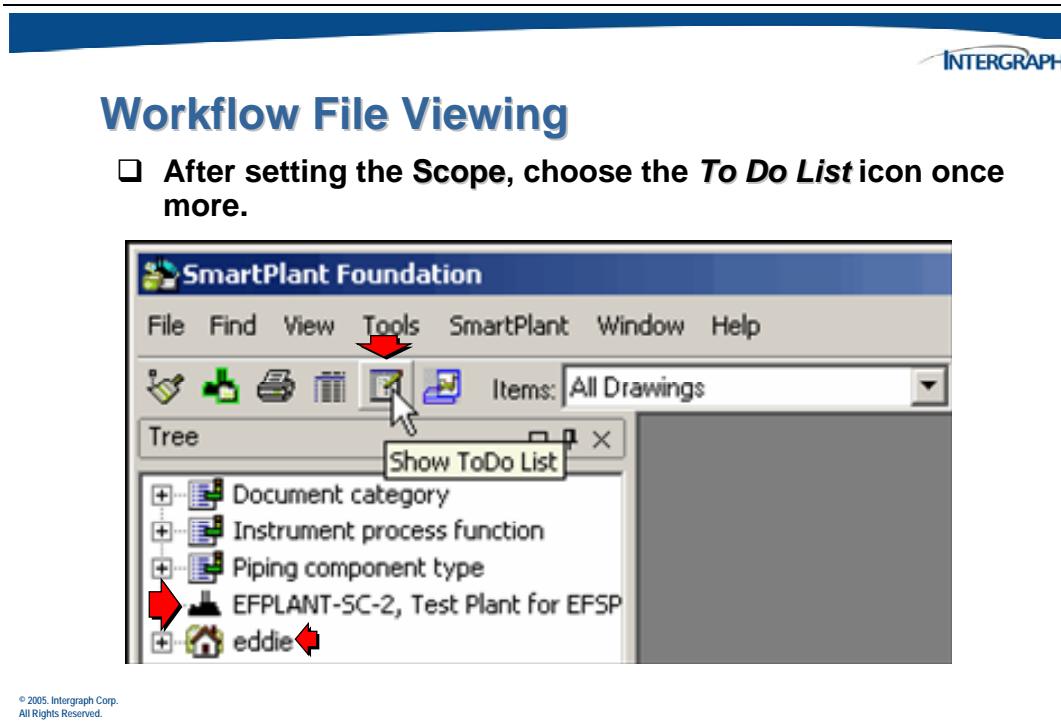
The next recipient in the workflow process will log in to the SPF client. Again, the first action should be to check the *To Do List* for any workflow assignments.



As part of the review process, the user may need to view the attached graphical file.



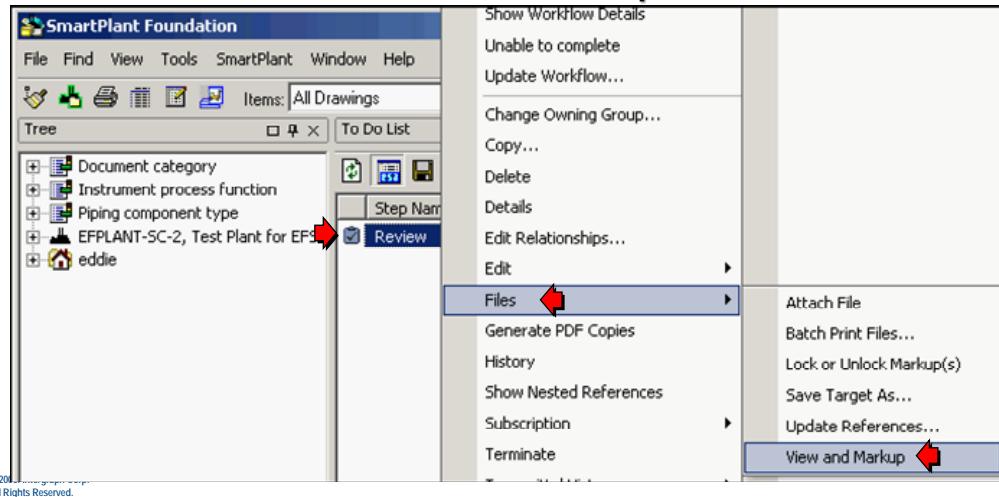
Because the user's **Scope** has not been set (note the absence of a **plant** in the *Tree View*), the **View and Markup** command is missing from the pop-up menu.





Workflow File Viewing

- In the To Do List window, right-click on the step (**Review**) and select **Files > View and Markup...** from the menu.

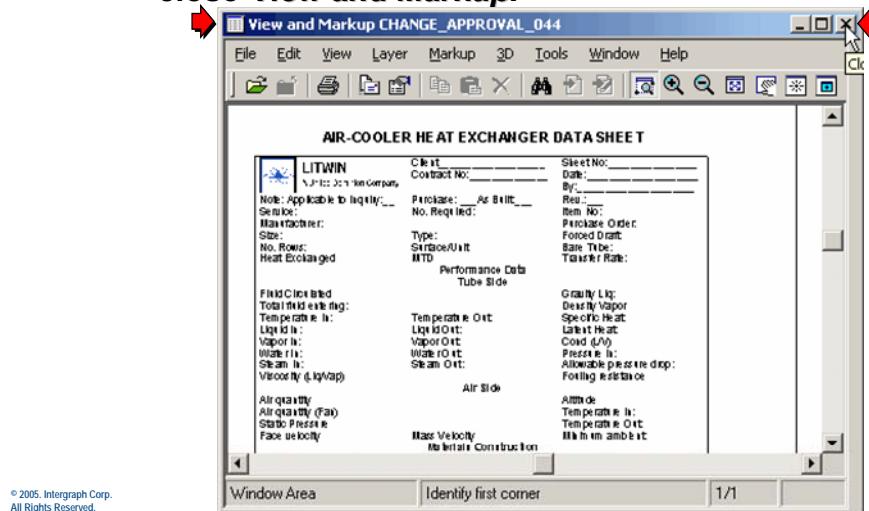


The attached file will display in a *View and Markup* window.

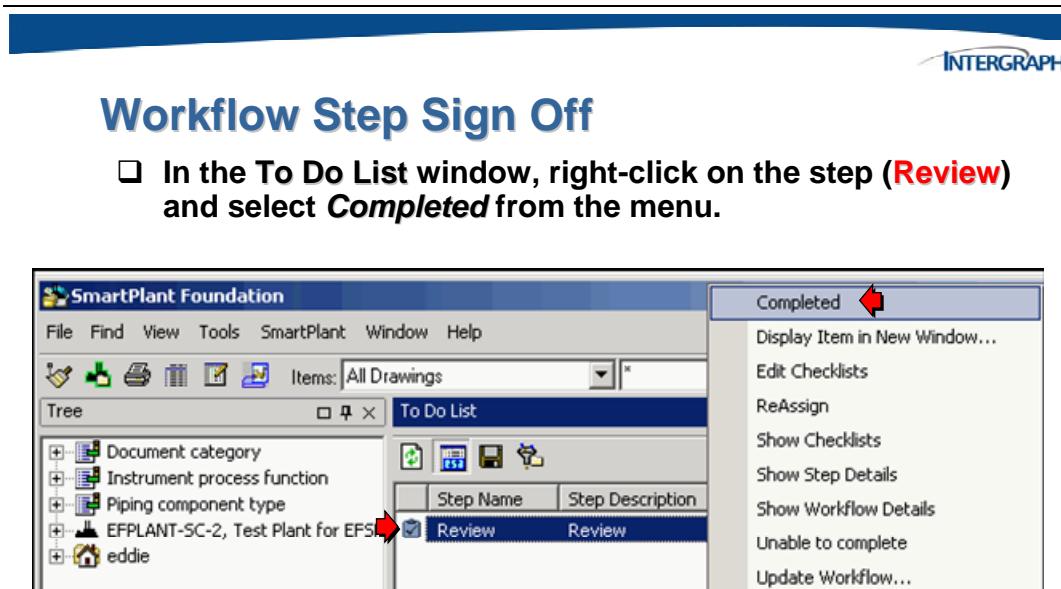


Workflow File Viewing

- Click on the X in the upper right corner of the window to close *View and Markup*.



After the *File Viewer* has been closed, complete the sign off of this workflow step.



© 2005, Intergraph Corp.
All Rights Reserved.

A *SignoffStep Task* dialog will display so that this step can be *Completed* by the recipient.



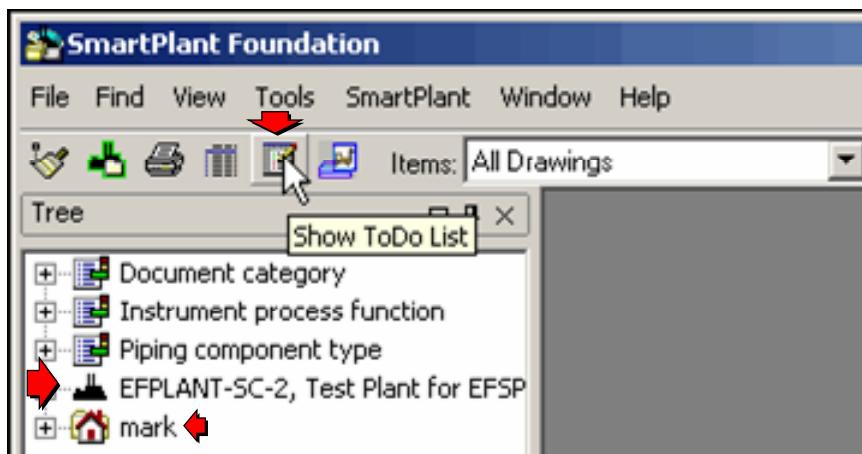
© 2005, Intergraph Corp.
All Rights Reserved.

This step happens to be a parallel step where two (or more) people can review the object at the same time but in separate steps. These steps could be configured so that the first person to signoff on the step, causes the workflow to continue.



Workflow File Viewing

- Again, after setting the Scope, choose the **To Do List** icon.



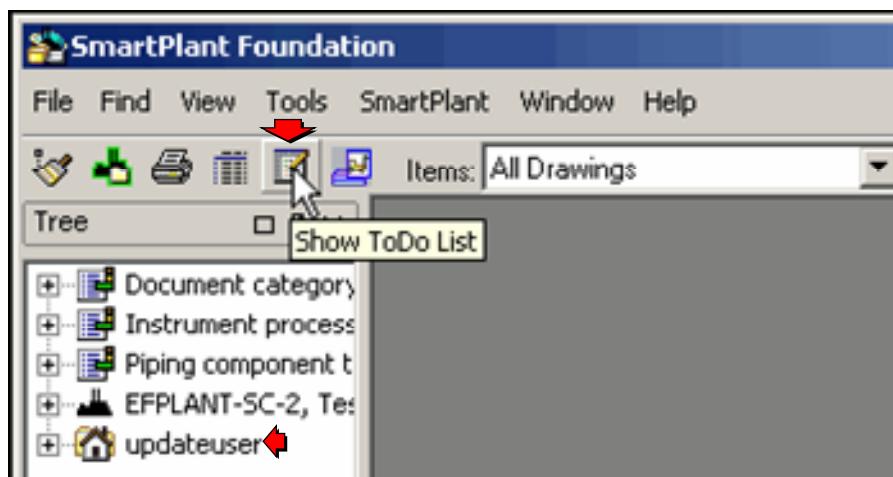
© 2005, Intergraph Corp.
All Rights Reserved.

The next recipient in the workflow process will log in to the SPF client.



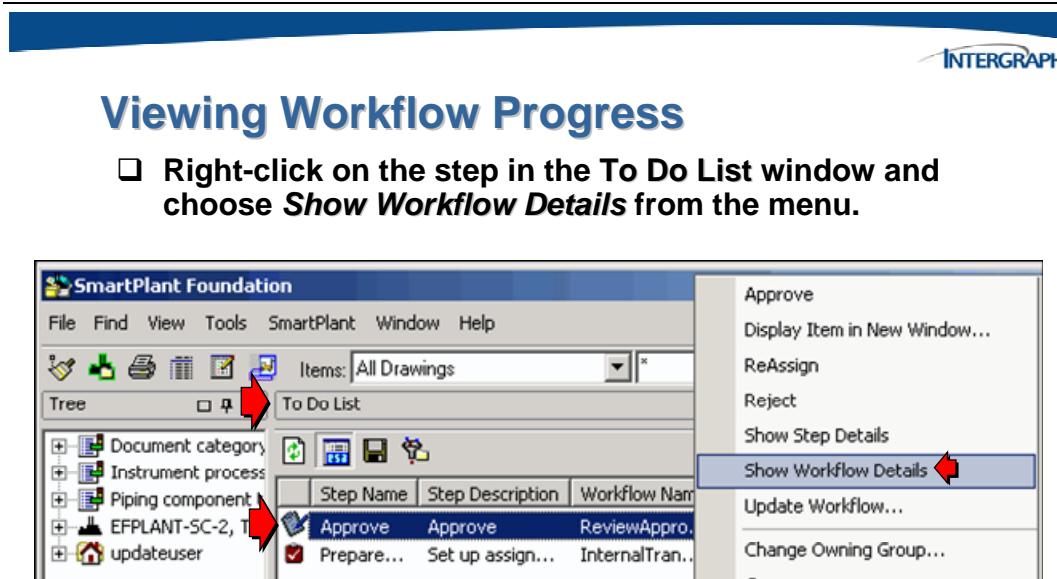
Viewing Workflow Progress

- From the toolbar, choose the **To Do List** icon.



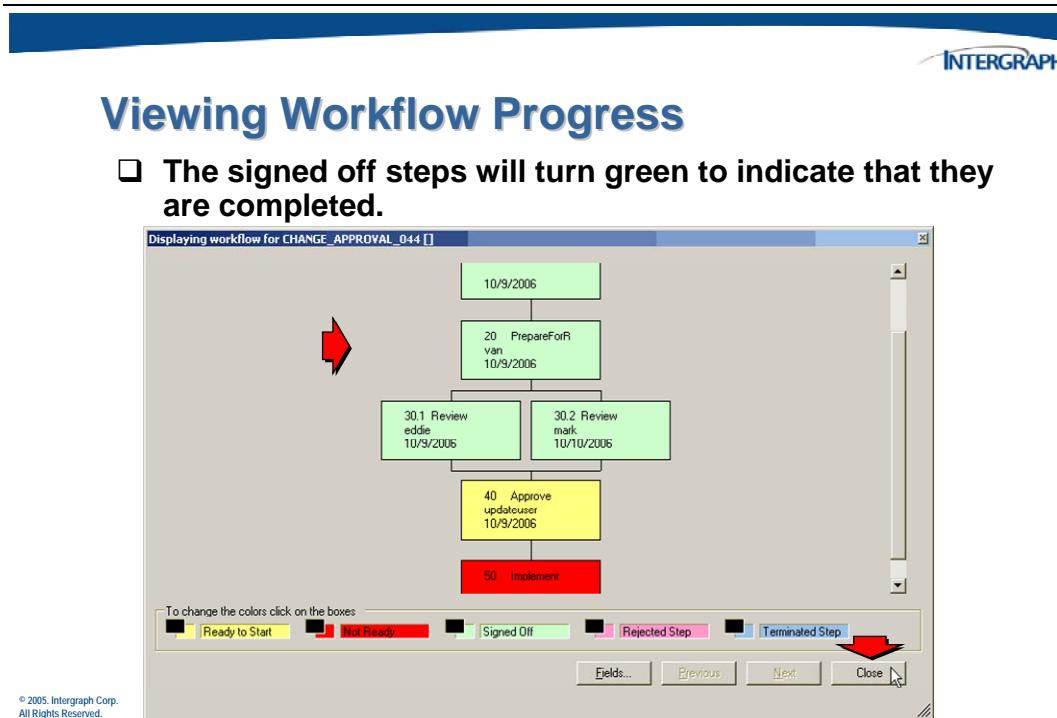
© 2005, Intergraph Corp.
All Rights Reserved.

A graphical view of the workflow progress can be displayed by displaying the workflow details.



© 2005, Intergraph Corp.
All Rights Reserved.

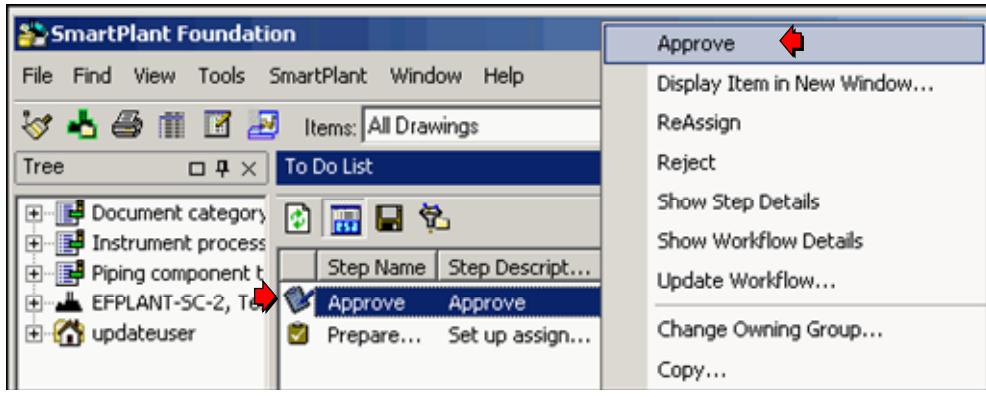
In this example, ***updateuser*** is the user checking the workflow progress.





Workflow Step Sign Off

- ❑ In the To Do List window, right-click on the step (**Approve**) and select **Approve** from the menu.



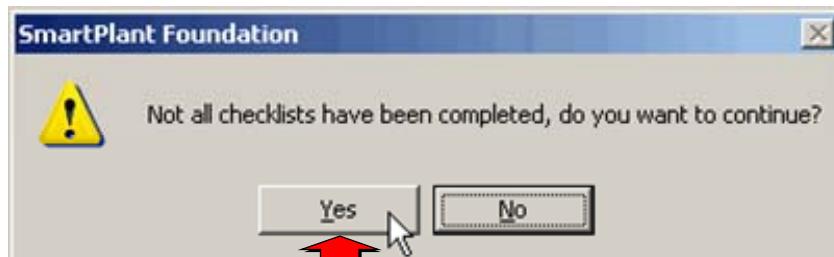
© 2005, Intergraph Corp.
All Rights Reserved.

This is an **Approve** step for a specific recipient. It also has a check list associated with it. The recipient can elect not to complete the check list tasks. If that happens, a warning message is displayed but the recipient can choose to continue with the step sign off.



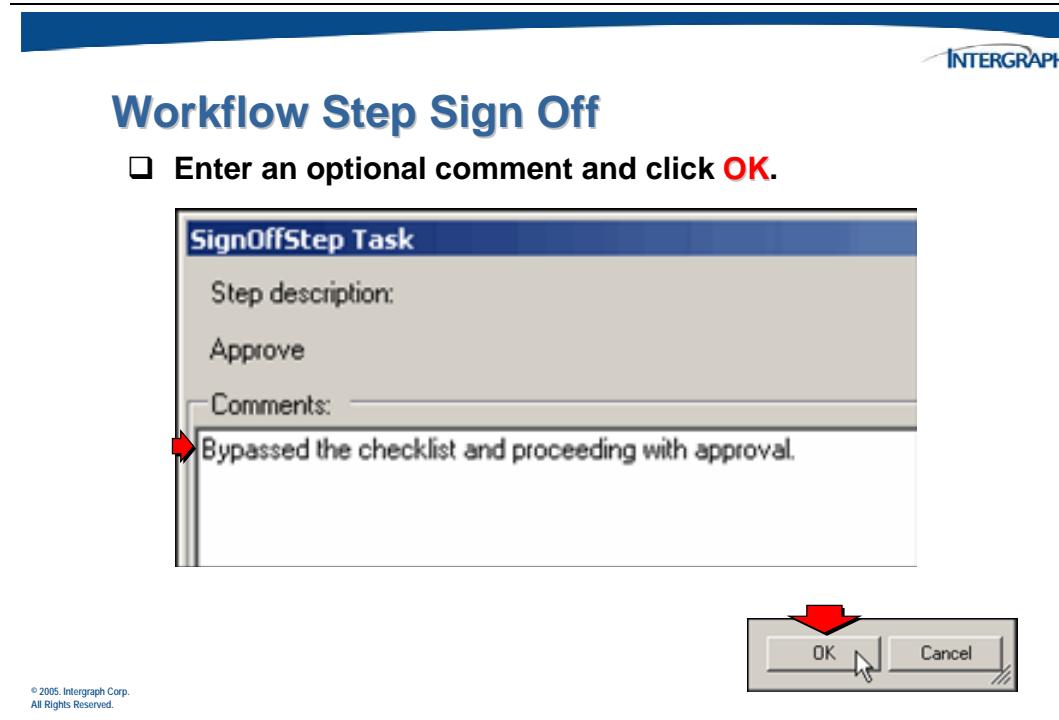
Workflow Step Sign Off

- ❑ Click **Yes** to process the step without completing the check list.

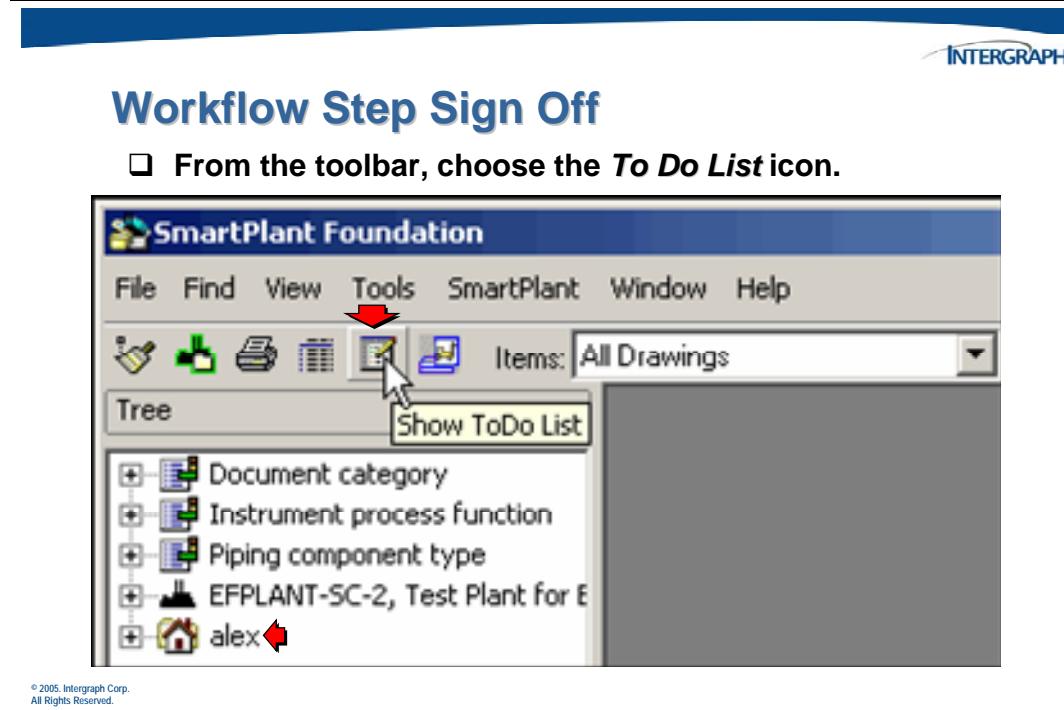


© 2005, Intergraph Corp.
All Rights Reserved.

A *SignoffStep Task* dialog will display so that this step can be *Approved* by the recipient.



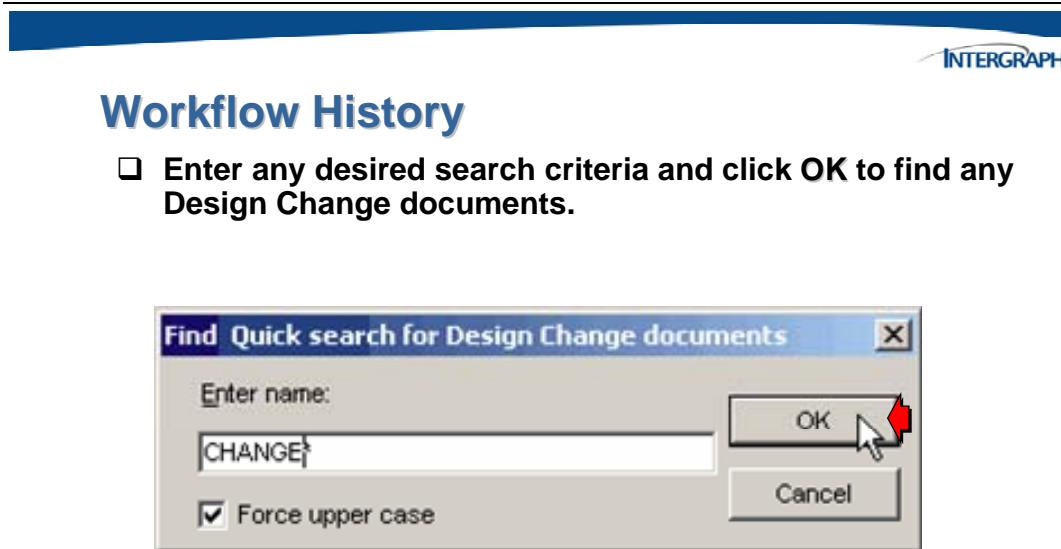
Perform the sign off for the last step in the workflow.



Since this is the last step in this workflow example, there are no more steps to be processed. The workflow is now completed.

6.7 Workflow History

A **Show Workflow** command can be used to once again bring up the workflow graphical display. First, perform a **Find** to locate a document that has completed a workflow.

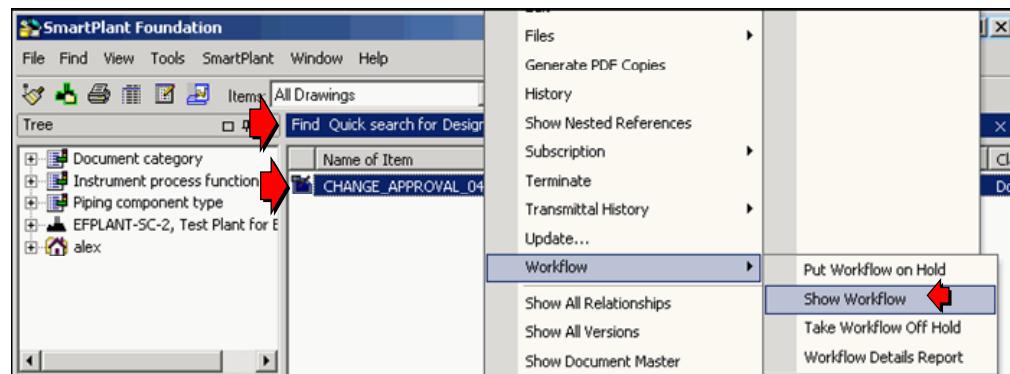


A Find Results window will be displayed.



Workflow History

- ❑ Right-click on the document in the Find results window and choose **Workflow > Show Workflow** from the menu.



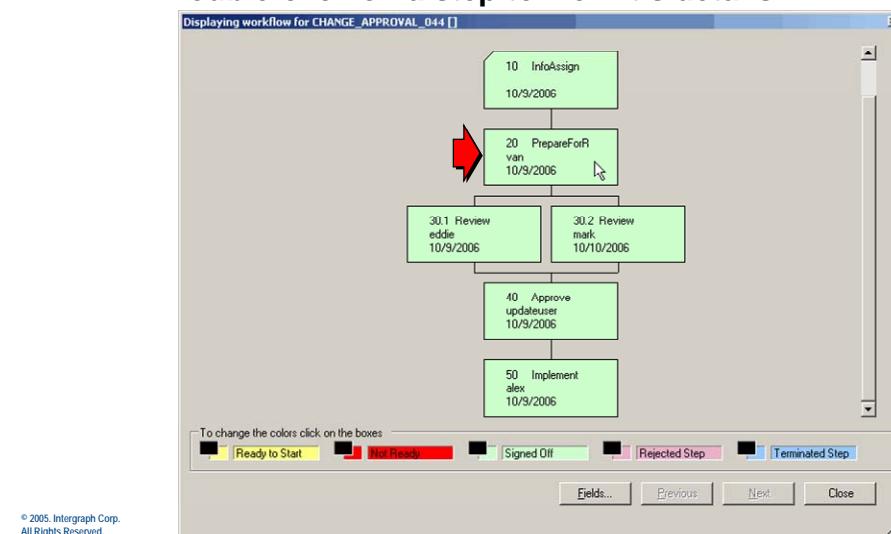
© 2005, Intergraph Corp.
All Rights Reserved.

Note that all steps will show the color green indicating successful completion.



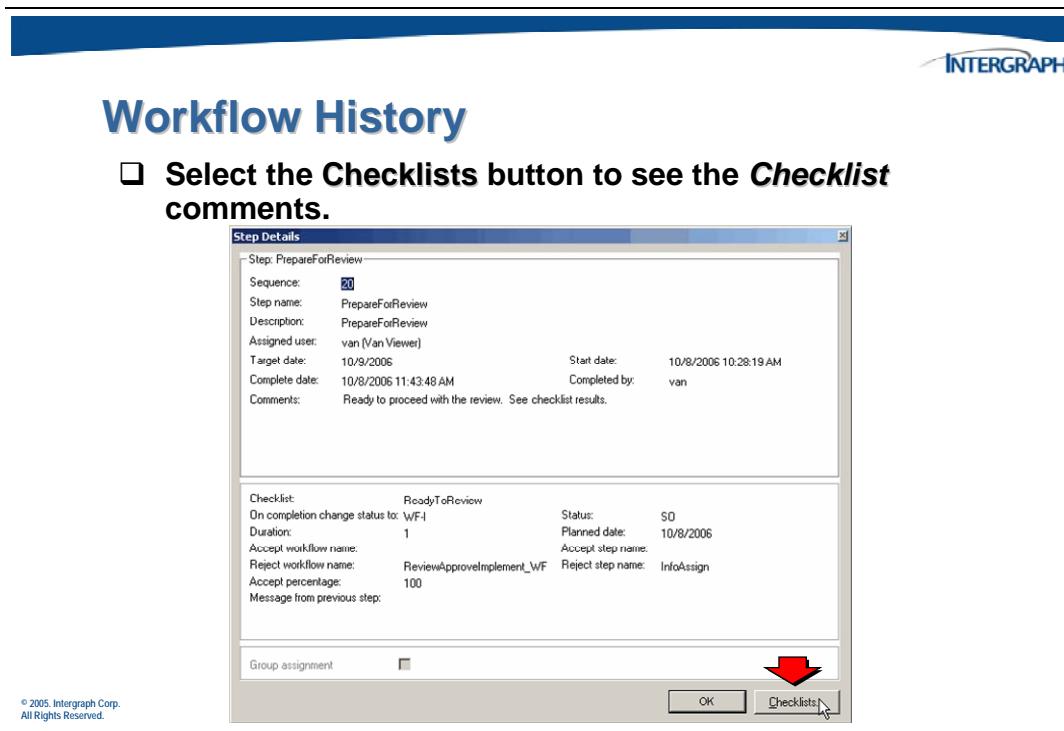
Workflow History

- ❑ Double-click on a step to view its details.

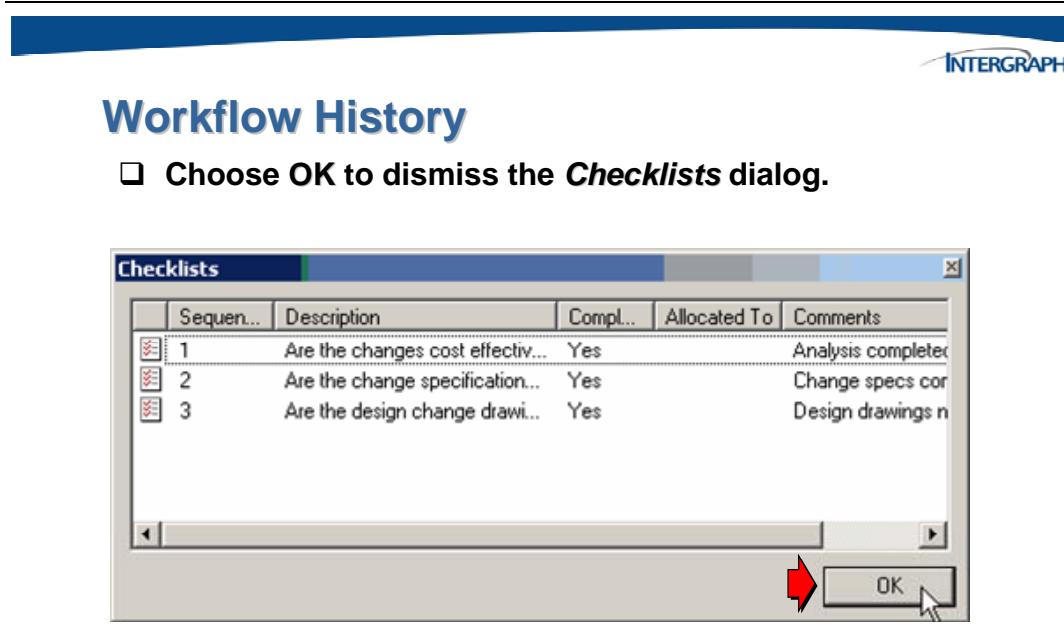


© 2005, Intergraph Corp.
All Rights Reserved.

Sequence 2 is one of the steps that were associated with a checklist.



A *Checklists* dialog window will be displayed showing the results of the associated checklist.

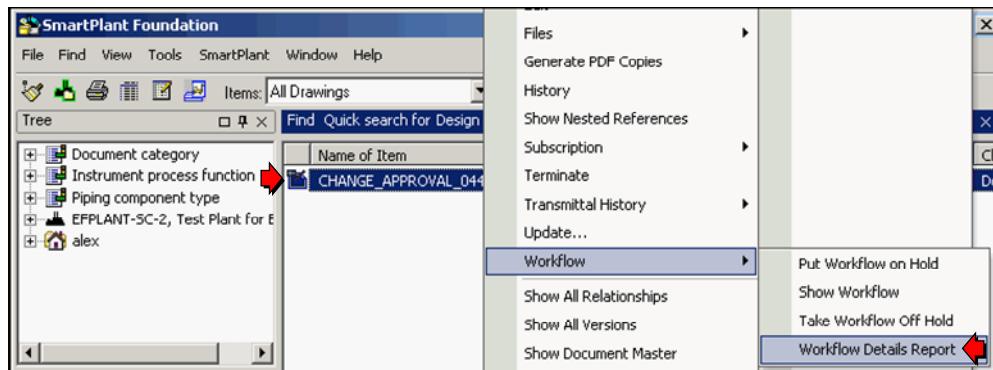


A *Workflow Details Report* can be generated to display a history of the workflow processing.



Workflow History

- Right-click on the document and select **Workflow > Workflow Details Report** from the menu.



© 2005, Intergraph Corp.
All Rights Reserved.

This report will display details about the finished workflow such as the *Name*, *Workflow Status* (at completion), *Step Names*, and *Assignees*.



Workflow History

- Expand sequence 2 to view the **Checklist** comments.

Sequence Number	Name	Description	Status	Duration	Comments
10	InfoAssign	Acknowledge Info Only	SO	1	Verified document and attached file, ready to proceed with the review. See checklist below.
- 20	PrepareForReview	PrepareForReview	SO	1	Ready to proceed with the review. See checklist below.
30.1	Review	Review	SO	1	My review is completed. I recommend re-approval.

Sequence	Description	Complete	Allocated to	Comments
1	Are the changes cost effective?	Yes		Analysis completed
2	Are the change specifications completed?	Yes		Change specs completed by end user
3	Are the design change drawings attached?	Yes		Design drawings not attached

© 2005, Intergraph Corp.
All Rights Reserved.

Comments made by recipients during sign off are displayed in the report. Use the scroll bar to review the details report.

The screenshot shows a Windows application window titled "Workflow Details Report". The main title bar has a blue header with the text "INTERGRAPH" and a white body with the title. Below the title bar is a toolbar with icons for file operations. The main content area is titled "OW HISTORY". It contains two tables: one for "Sign Off History" and another for "Comments". Red arrows point to the "Step Type" column in both tables, highlighting the "I" (Information) and "W" (Work) types. The "Comments" table shows several rows of text messages from users like eddie, mark, updateuser, and alex. At the bottom left of the window, there is a copyright notice: "© 2005. Intergraph Corp. All Rights Reserved."

Sign Off	Status	Duration	Comments	Assignee	Step	Type
edge Info Only	SO	1	Verified document and attached file, ready to proceed with the review process.	UPDATE	I	
ForReview	SO	1	Ready to proceed with the review. See checklist results.	van	W	

Complete	Allocated to	Comments	Assignee	Step	Type	
active?	Yes	Analysis completed				
ons completed?	Yes	Change specs completed by engineering				
wings attached?	Yes	Design drawings not attached				
	SO	1	My review is completed. I recommend ready for approval.	eddie	W	
	SO	2	Review has been performed. Proceed to approval phase.	mark	W	
	SO	1	Bypassed the checklist and proceeding with approval.	updateuser	W	
	SO	1	Changes are being implemented.	alex	W	

© 2005. Intergraph Corp.
All Rights Reserved.

7

C H A P T E R

Introduction to Transmittals

7. SPF Transmittal Overview

Transmittals are used to distribute documentation around an organization and between different organizations. They record what documents are sent to whom, why, what response is expected of them by when and the response received. In a paper based system, the transmittal is used by the print room to prepare physical prints.

This chapter details how transmittals are set up and configured in the Schema Object model to enable you to change the configuration to match the site requirements.



Overview of Transmittals

What is a Transmittal?

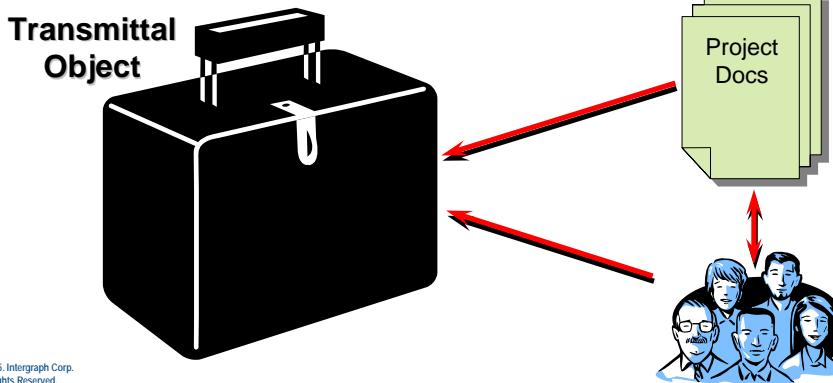
The term transmittal usually refers to the top sheet or report that lists how many copies of each document are sent to whom, and for what reason.



Overview of Transmittals

What is a Transmittal?

In an electronic system such as SmartPlant Foundation, the transmittal becomes a collection object that links a set of documents to a set of recipients with reasons for receipt.



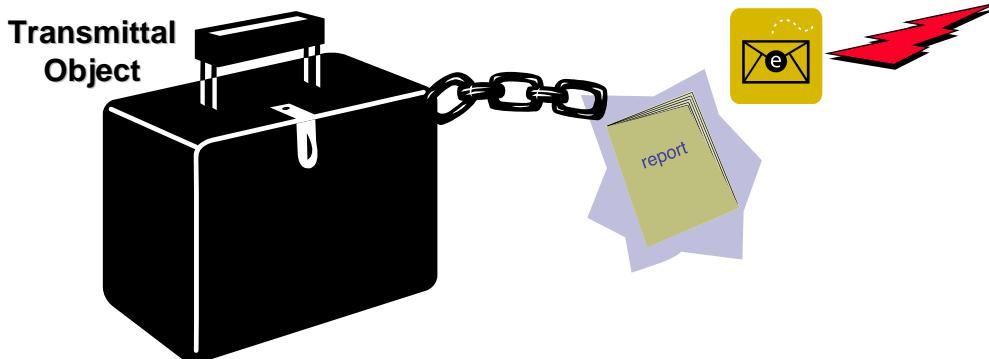
Transmittal object acts as a container for a package of documentation.



Overview of Transmittals

What is a Transmittal?

A transmittal report is attached to the collection object and is emailed to the recipients upon issue of the transmittal.



The Schema Object model has been extended to provide the transmittals functionality.



Overview of Transmittals

What is a Transmittal?

- A formal release and distribution mechanism for document control – a controlled package of documentation**
 - Within internal organizations
 - To external organizations (between different companies)
- Notifies recipients of their involvement via email**
- Allows responses from recipients to be formally recorded**
- Plays an important part in tracking the progress of a project**

© 2005. Intergraph Corp.
All Rights Reserved.

Transmittals play an important role in tracking the progress of a project. Milestone payments from the client are often linked to the distribution of particular documentation on transmittals, the most notable of which is the site construction transmittal when documents are released to the site for construction.

Transmittals are also useful for internal document management, such as the distribution of control documentation, safety procedures, and other documents between internal organizations.

Transmittals can be modeled with or without transmittal sections that group the documents together.

Distribution matrices are used to set up lists of recipients for specific types of documents, such as mechanical P&IDs. Each distribution matrix is created with a *Reason For Issue*, such as *For Design Review* or *For Construction*. Each recipient on the matrix has a **Reason for Receipt** field, such as *For Information*, *For Comment* or *For Work* to specify which action, if any, is required of the user upon receipt of the transmittal.



Overview of Transmittals

- A **Distribution Matrix** is used to set up a list of recipients for specific types of documents.
- Each distribution matrix is set to a **Reason For Issue**, such as *For Design Review* or *For Construction*.
- Each recipient on the matrix has a **Reason for Receipt**, such as *For Comment*, *For Information*, or *For Work*.
- In SPF there are 2 kinds of Distribution Matrices delivered:
 - Internal Matrix
 - External Matrix

© 2005. Intergraph Corp.
All Rights Reserved.

Transmittals often have a controlling recipient, so that the responses from all the recipients are fed back through the controlling recipient rather than individually.

The transmittal is issued and the attached documents are distributed to the list of recipients. In a paper-based world, a transmittal report is printed and sent to the Print Room, where all the relevant documents are printed and packaged for distribution. In a fully integrated transmittal and document management system, the transmittal report is mailed electronically to the recipients, who log in to retrieve the relevant documents and perform whatever actions are necessary, such as comments or changes. If recipients do not have access to the document management system, the documents must be made available through some other mechanism, such as shared folders, FTP sites, or distribution on CD-ROM.

Transmittals that are issued for information are closed out after issue. Transmittals that are issued for comment remain active until all comments are received, at which point the transmittal can be closed, usually by the Document Controller. Transmittals that are issued in error can be cancelled. In this case, a transmittal report with an issue state of **CANCELLED** is redistributed as a form of recall notice.

7.1 Transmittal Project Roles

This section lists typical user roles associated with transmittals. These roles are the *document controller*, *design engineer*, *client*, and *vendors/sub-contractors*.



Transmittal Project Roles

Document Controllers

These personnel are responsible for the integrity of all project documentation and the distribution of that documentation to the relevant project groups at the appropriate time. The tasks that they perform are typically:

- Project document index management
- Transmittal preparation and distribution
- Transmittal distribution management
- Transmittal response tracking and follow up

© 2005, Intergraph Corp.
All Rights Reserved.



Transmittal Project Roles

Design Engineer

These users are responsible for the generation of the project documentation. Although they do not manage transmittals themselves, they typically:

- Review transmittal histories of documents
- Prepare outline transmittals
- Prepare transmittal issue requests
- Receive and respond to transmittals

© 2005, Intergraph Corp.
All Rights Reserved.



Transmittal Project Roles

Client

Client recipients have to be clearly marked and distinguished from the main project recipients on most transmittals. They typically:

- Review transmittal histories of documents
- Receive and respond to transmittals

© 2005. Intergraph Corp.
All Rights Reserved.



Transmittal Project Roles

Vendors and Sub-Contractors

These personnel are usually not connected to the project management system, but still exchange documentation with the project team, and often in large batches.

Transmittals are the mechanism for the exchange, but this process requires packaging the documentation for distribution and access from a separate environment, often outside the firewall. They typically:

- Review transmittal histories of documents
- Receive and respond to transmittals
- Return project documentation on transmittals

© 2005. Intergraph Corp.
All Rights Reserved.

7.2 Transmittal Structure

Transmittal functionality consists of *internal transmittals* and *external transmittals*.



Transmittal Structure

There are 2 types of transmittals delivered in SPF:

- External Transmittals**
- Internal Transmittals**

External transmittals are set up with documents attached directly to the transmittal object.

Internal transmittals are broken down into sections to group documents together, such as those that need editing or comments, or those that are just for reference.

© 2005, Intergraph Corp.
All Rights Reserved.

When you create a transmittal, it is attached to a workflow that manages transmittal *approval, issue, distribution, recipient responses, and completion*.

You can also use template transmittals to create transmittals that already have relevant sections defined, a workflow attached, and optionally some key recipients already identified by their project roles.



Internal Transmittal Structure

Here is an example of an *Internal* Transmittal with Sections:

Name	Rev	Default Status	Classification	LastUpdated C
INT-SG-001		ISSUED - (WF)	SPFInternalTra...	2004/01/29-2
Show Sections				
INT-SG-001-Design		ISSUED -	SPFInternalTra...	2004/01/29-0
Show Documents				
SG-C-544	[A1,1]	CURRENT -	NPD	2004/01/29-0
SG-C-545	[A1,1]	CURRENT -	NPD	2004/01/29-0
INT-SG-001-Reference		ISSUED -	SPFInternalTra...	2004/01/29-0
Show Documents				
SG-C-546	[A1,1]	CURRENT -	NPD	2004/01/29-0

© 2005, Intergraph Corp.
All Rights Reserved.



External Transmittal Structure

Here is an example of an *External* Transmittal, where documents are attached directly to the transmittal object.

Name	Rev	Default Status	Classification	L
EXT_RBGAS REVIEW		WF-S	SPFExternalTransmittal	1
Show Documents				
REQ-PRJ-2001-100	[1A,1]	CURRENT -	DocumentVersion	1
REQ-PRJ-2001-200	[1A,1]	CURRENT -	DocumentVersion	1
REQ-PRJ-2001-300	[1A,1]	CURRENT -	DocumentVersion	1

© 2005, Intergraph Corp.
All Rights Reserved.

7.3 Transmittal Administrative Setup

Before creating Transmittals, some administrative setup tasks must first be accomplished.

Internal Transmittals are used for distribution of documents between the departments, site and affiliate offices of the organization.

External Transmittals are used for the distribution of documents to clients and suppliers.



Administrative Setup for Transmittals

Administrative tasks include creating the following objects before you begin to create Transmittals:

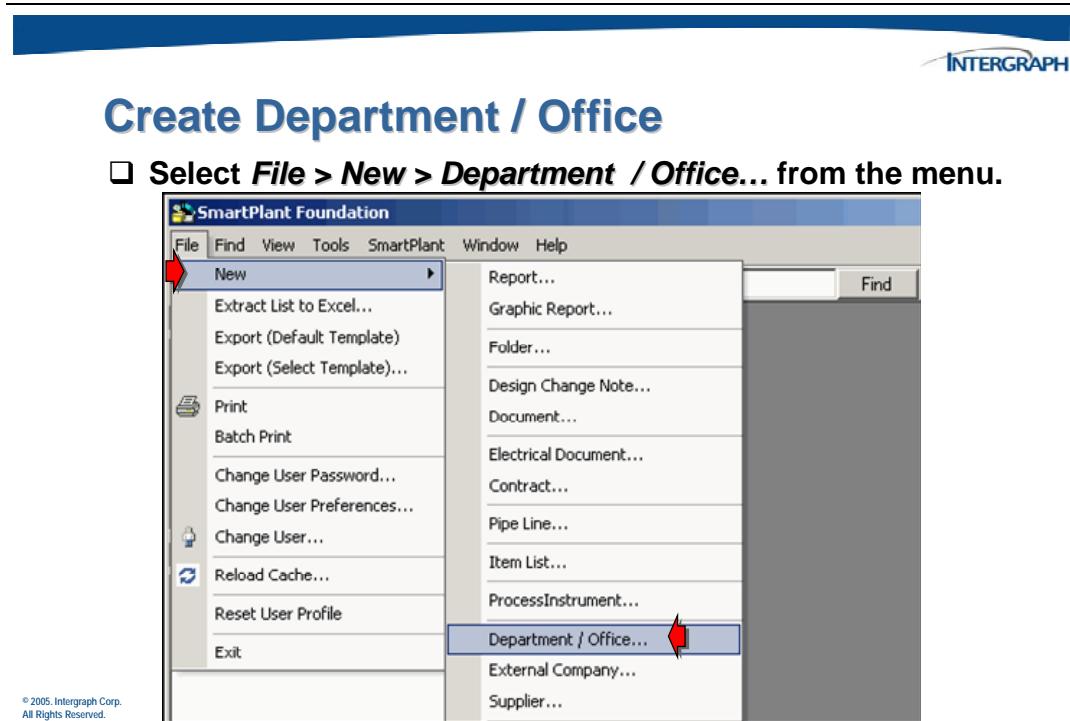
- Department / Office**
- External Company (Client)**
- Supplier**
- Contract**
- Transmittal Distribution Matrices**

© 2005. Intergraph Corp.
All Rights Reserved.

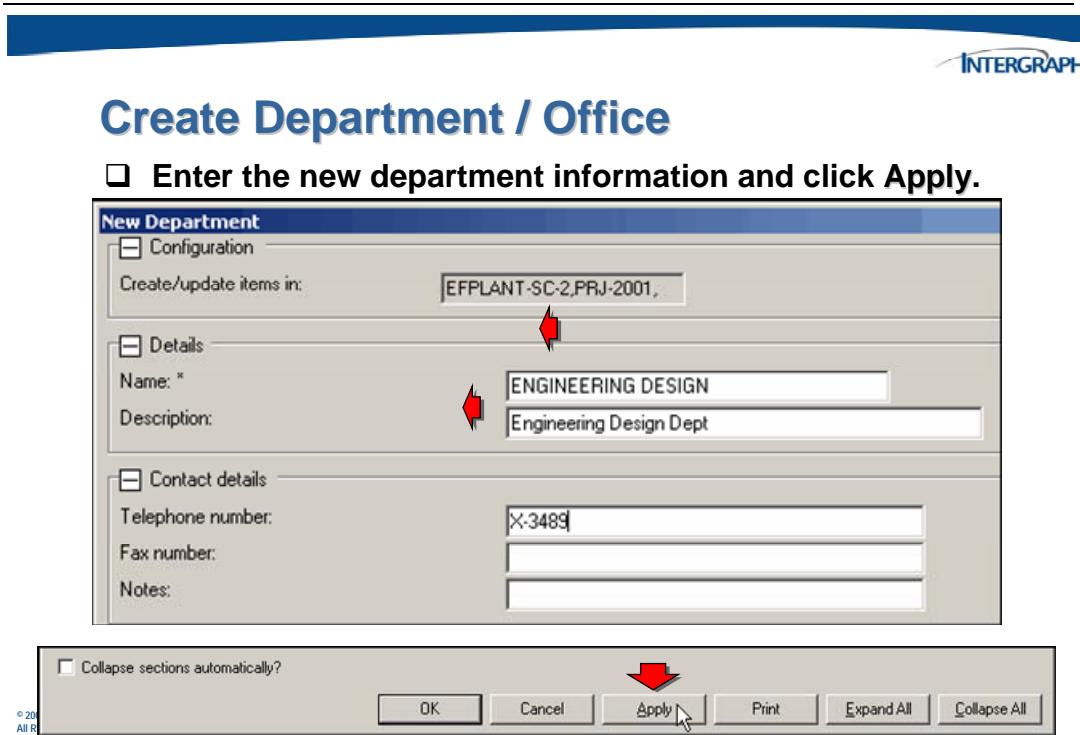
The *Department/Office* is an **Internal Organization** while *External Company* and *Supplier* are **External Organizations**. The picklists for **From Organization** and **To Organization** on a transmittal are set up accordingly.

7.3.1 Creating a Department / Office

The **Department / Office** is classed as an *Internal Organization*. To use *Internal Transmittals* for the distribution of documents between internal *departments*, *site* and *affiliate offices*, you would need to create an object for each of these internal organizations.



The *New Department* dialog will display.



The following fields are displayed on the New Department / Office form:

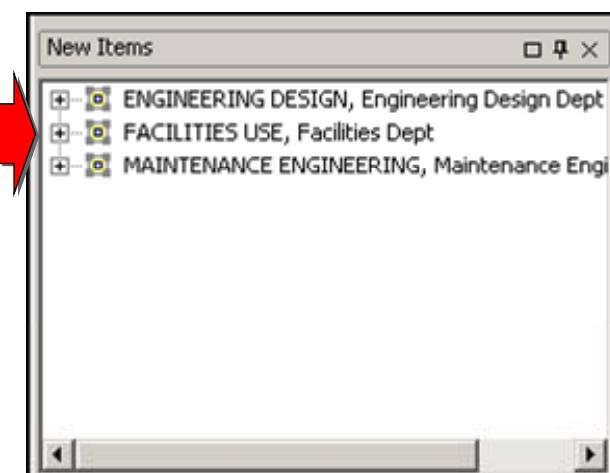
- ❑ **Name** – is a text field that can contain up to 32 characters. It is a required field, denoted by the *.
- ❑ **Description** – is a text field that can contain up to 40 characters. It is an optional field.
- ❑ **Telephone Number** – is a text field that can contain up to 35 characters. It is optional.
- ❑ **Fax Number** – is a text field that can contain up to 35 characters. It is optional.
- ❑ **Notes** – is a text field that can contain up to 35 characters. It is optional.

Click **OK** or **Apply** when the new Department / Office information has been entered. The Apply button can be used when multiple Department / Office objects need to be created.



Create Department / Office

- Repeat this for all internal departments and offices that will be using Internal Transmittals within the same company.



© 2005, Intergraph Corp.
All Rights Reserved.

7.3.2 External Organizations

Once you have created your internal organization objects, which will be used for creating Internal Transmittals, you will need to create any external organization objects such as *External Companies* and *Suppliers*.



External Organizations

There are 2 types of External Organizations that can be created for *External Transmittals*

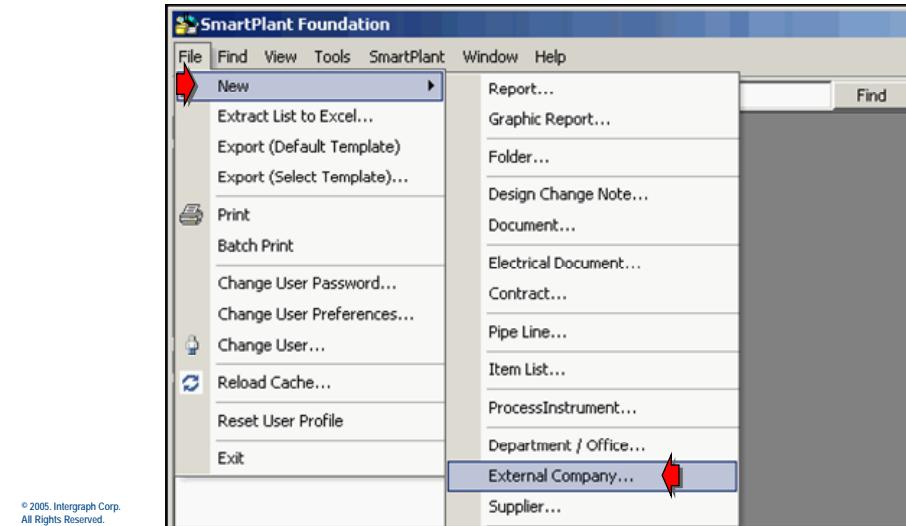
- External Company
- Supplier

External Transmittals can be used for the distribution of documents to external companies (clients) or to suppliers.



Create External Company

- Select **File > New > External Company...** from the menu.

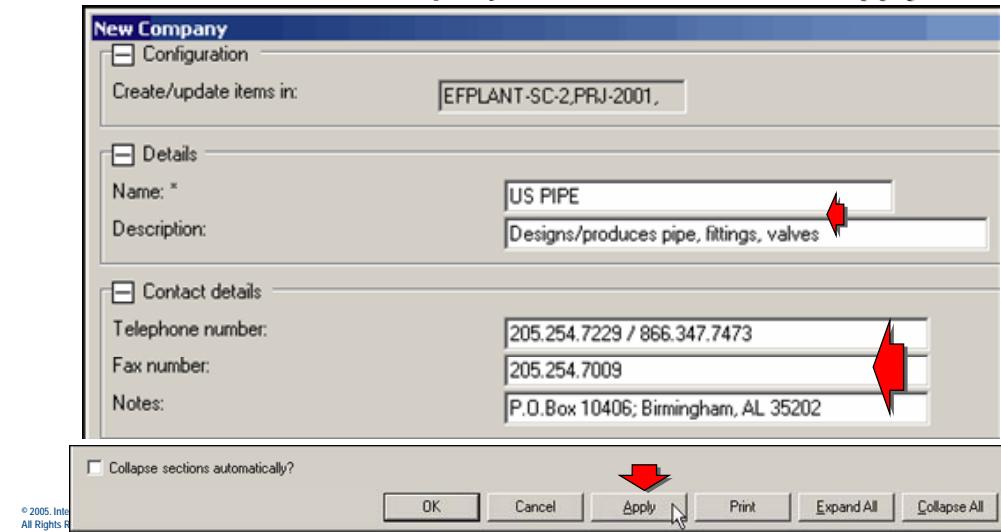


The *New Company* dialog will display.



Create External Company

- Enter the new company information and click **Apply**.

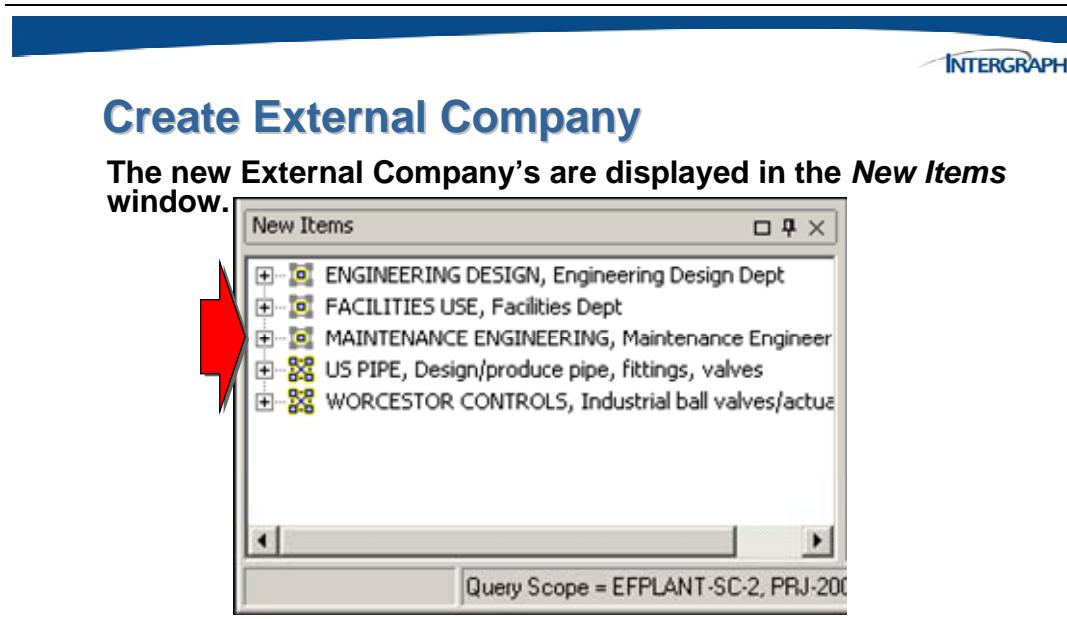


The following fields are displayed on the *New Company* form:

- Create/update items in** – is a read-only field, which displays the plant/project configuration that has been set.
- Name** – is a text field that can contain up to 32 characters. It is a required field, denoted by the *.
- Description** – is a text field that can contain up to 40 characters and is optional.
- Telephone number** – is a text field that can contain up to 35 characters and is optional.
- Fax number** – is a text field that can contain up to 35 characters and is optional.
- Notes** – is a text field that can contain up to 35 characters and is optional.

Click **OK** or **Apply** when the new company information has been entered.

Repeat this for all external companies that would be participating in External Transmittals distributed from your company.



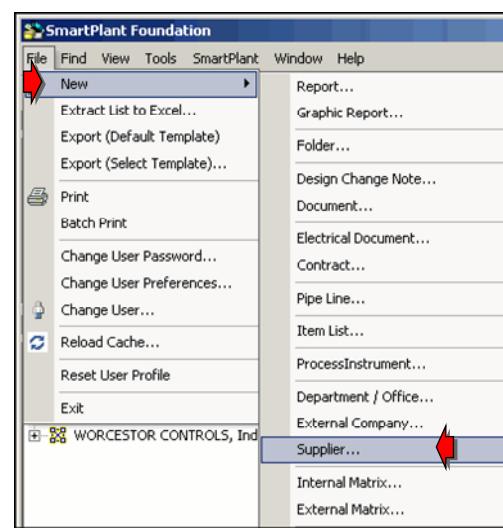
External Company objects will be used in pick list values when creating External Transmittal objects.

7.3.3 Creating a Supplier

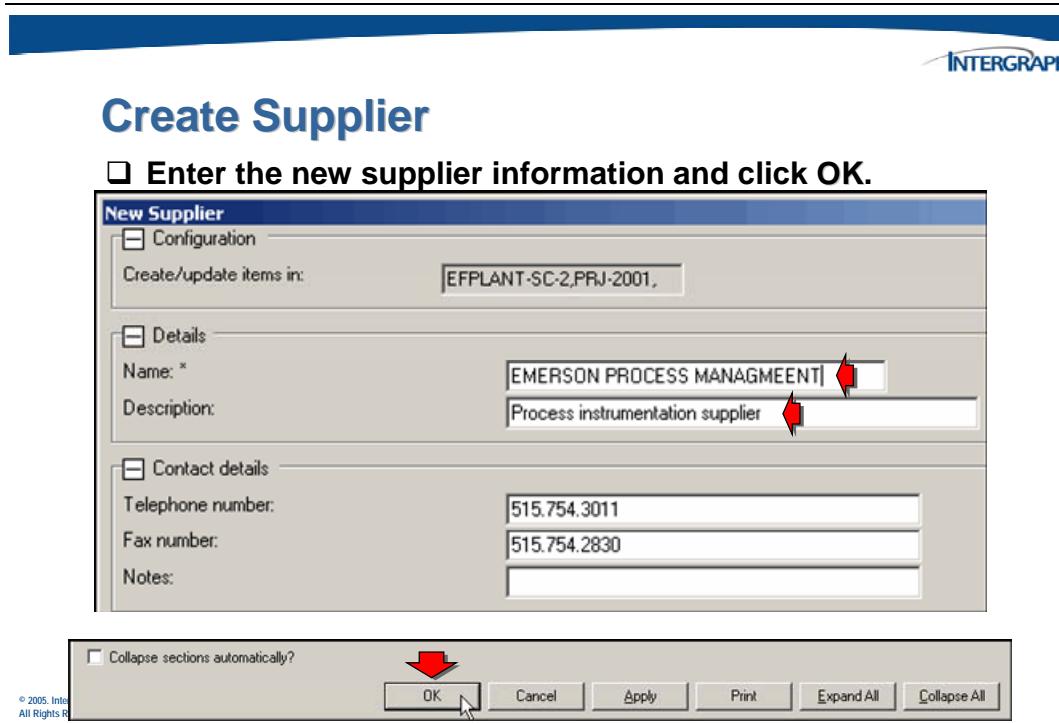
If you plan to distribute documents to suppliers through the use of Transmittals, then you will need to create Supplier objects.

Create Supplier

- Select **File > New > Supplier...** from the menu.



The *New Supplier* dialog will be displayed.



The following fields are displayed on the *New Supplier* form:

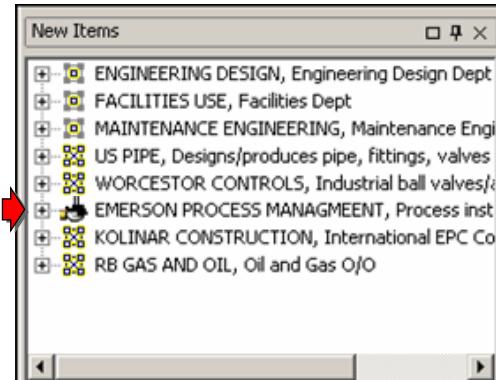
- ❑ **Create/update items in** – is a read-only field, which displays the plant/project configuration that has been set.
- ❑ **Name** – is a text field that can contain up to 32 characters. It is a required field, denoted by the *.
- ❑ **Description** – is a text field that can contain up to 40 characters and is optional.
- ❑ **Telephone number** – is a text field that can contain up to 35 characters and is optional.
- ❑ **Fax number** – is a text field that can contain up to 35 characters and is optional.
- ❑ **Notes** – is a text field that can contain up to 35 characters and is optional.

Click **OK** when the new Supplier information has been entered.



Create Supplier

The new supplier is displayed in the *New Items* window.



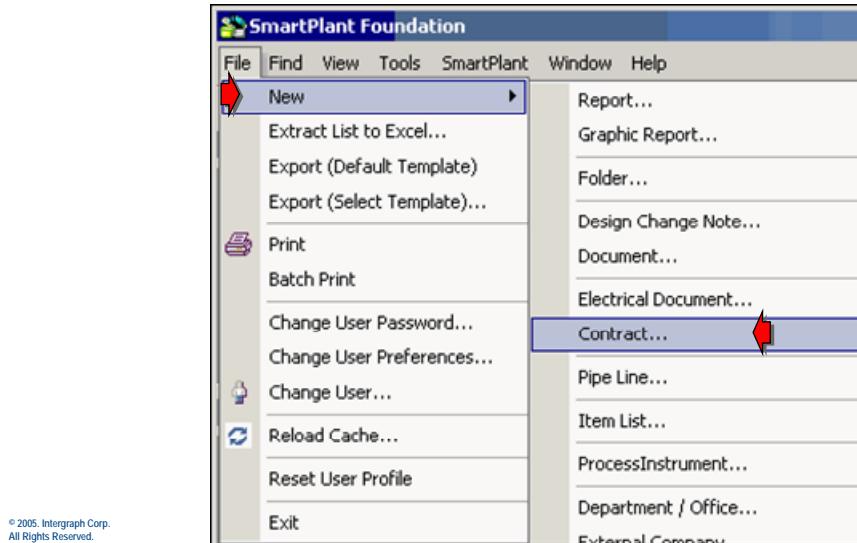
Repeat this for all suppliers that would be participating in External Transmittals distributed from your company.

7.3.4 Creating a Contract

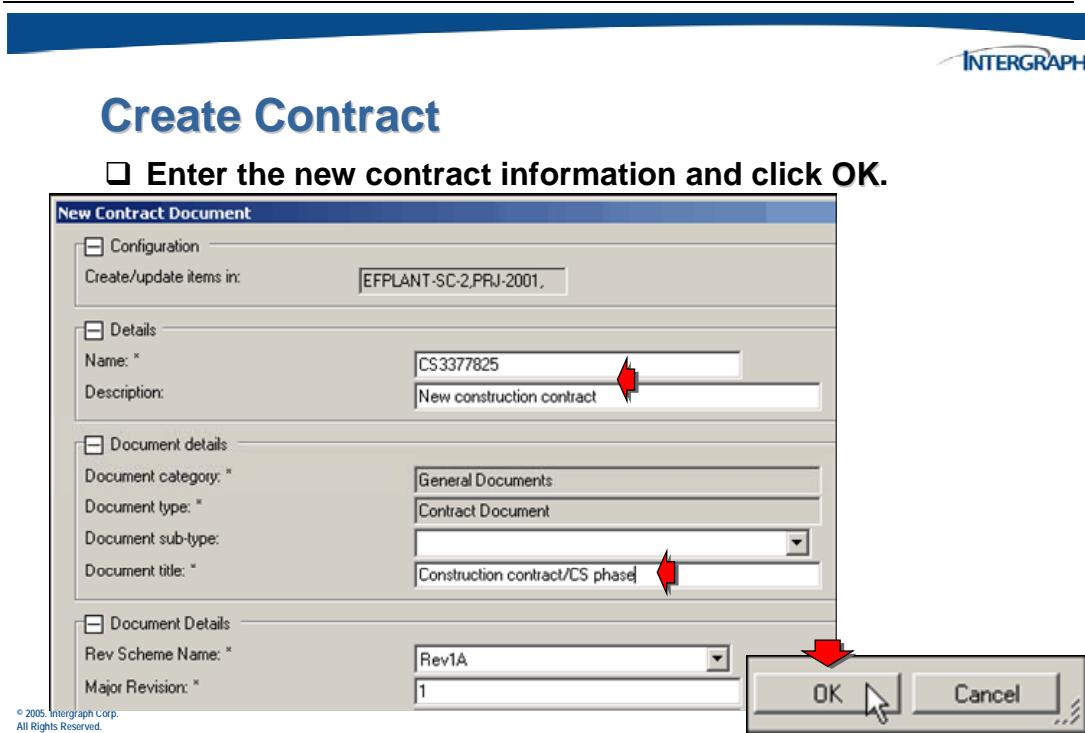
Within a project there may be multiple contracts in use. As part of the Transmittals' administrative setup, you can create **Contracts** and then use them during the creation of Transmittals for tracking purposes.

Create Contract

- Select **File > New > Contract...** from the menu.



The *New Contract* dialog will be displayed.



The following fields are displayed on the *New Contract* form:

- Create/update items in** – is a read-only field, which displays the plant/project configuration that has been set.
- Name** – is a text field that can contain up to 32 characters. It is a required field, denoted by the *.
- Description** – is a text field that can contain up to 40 characters and is optional.
- Document category** – is a required field that is automatically populated.
- Document type** – is a required field that is automatically populated.
- Document sub-type** – is an optional field that may or may not contain picklist values associated with the Document category.
- Document title** – is a text field that can contain up to 40 characters. It is a required field, denoted by the *.
- Rev scheme name** – is a pick list of revision schemes available. It is a required field, denoted by the *.
- Major revision** – is a field automatically populated with the major revision of the revision scheme selected above. It is a required field, denoted by the *.

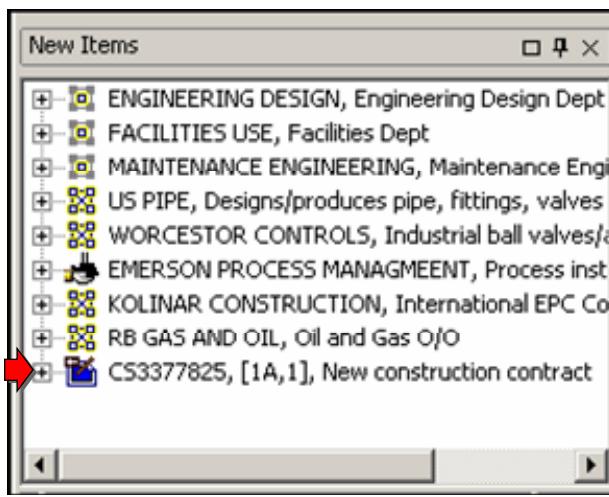
- Minor revision** – is a field automatically populated with the minor revision of the revision scheme selected above. It is a required field, denoted by the *.
- Owning Group** – is a pick list of available user groups. It is a required field, denoted by the *.

Click **OK** when the new Contract information has been entered.



Create Contract

The new contract is displayed in the **New Items** window.



Contract objects are used as pick list values when creating Internal Transmittal objects.

7.3.5 Transmittal Users and User Groups

As part of the administrative setup for transmittals, the users who will be participating in the transmittal process will need to be modified and the **Company name** field updated. This can only be done after the *departments, offices, external companies and suppliers* have been added to the system.

If a needed user doesn't exist, use the **System Administration** interface to add the account. Use the **File > New > User** menu command to display the *New user* form.



Creating Transmittal Users

- Click **OK** once all new user information has been entered

The screenshot shows the 'New user' dialog box. It contains the following fields:

Field	Value
Login name:	dawn
Password:	[redacted]
Full name:	Dawn document controller
Company name:	ENGINEERING DESIGN
E-mail address:	doccontroller@ingr.com
Warning dialog count:	[redacted]
Maximum SQL limit:	[redacted]
Inbox refresh:	600

Also use the System Administration interface to search for any existing users. Then select the user and use the **Update** menu command.



Updating Transmittal Users

- Update each internal user and choose one of the recently added departments.**

Update alex

Login name:	alex
Password:	<input type="password"/>
Full name:	Alex Admin
Company name:	<input type="button" value="ENGINEERING DESIGN"/> ENGINEERING DESIGN FACILITIES USE KOLINAR CONSTRUCTION MAINTENANCE ENGINEERING RB GAS AND OIL Supplier US PIPE WORCESTOR CONTROLS
E-mail address:	<input type="text"/>
Warning dialog count:	<input type="text"/>
Maximum SQL limit:	<input type="text"/>
Inbox refresh:	<input type="text"/>
Expand to:	NewWindow
Days between password changes:	<input type="text"/>
Default configuration:	EFPLANT-SC-2,PRJ-2001,
Default query configuration:	EFPLANT-SC-2,PRJ-2001,

© 2005, Intergraph Corp.
All Rights Reserved.

Select a **Company name** value from the displayed list. Continue to update the appropriate user accounts.



Updating Transmittal Users

- Update each internal user and choose one of the recently added departments.**

Update alex

Login name:	alex
Password:	<input type="password"/>
Full name:	Alex Admin
Company name:	<input type="button" value="ENGINEERING DESIGN"/> ENGINEERING DESIGN FACILITIES USE KOLINAR CONSTRUCTION MAINTENANCE ENGINEERING RB GAS AND OIL Supplier US PIPE WORCESTOR CONTROLS
E-mail address:	<input type="text"/>
Warning dialog count:	<input type="text"/>
Maximum SQL limit:	<input type="text"/>
Inbox refresh:	<input type="text"/>
Expand to:	NewWindow
Days between password changes:	<input type="text"/>
Default configuration:	EFPLANT-SC-2,PRJ-2001,
Default query configuration:	EFPLANT-SC-2,PRJ-2001,

© 2005, Intergraph Corp.
All Rights Reserved.

If any new user groups are needed, add them using System Administration. Use the **File > New > User Group** menu command to display the *New user group* form.

The screenshot shows the 'Update client' dialog box. It contains fields for Login name (client), Password, Full name, Company name (dropdown menu showing 'RB GAS AND OIL'), E-mail address (client@ingr.com), Warning dialog count, Maximum SQL limit, and Inbox refresh. A red arrow points to the 'Company name' dropdown menu. At the bottom left, there is a copyright notice: © 2005, Intergraph Corp. All Rights Reserved.

If any other groups will be needed, such as a group that can participate in workflows for the purposes of receiving email, add them now.

The screenshot shows the 'New user group' dialog box. It includes fields for User group name (EMAILONLY, highlighted with a red arrow), User group description (Used to provide project filtering and workflow acc), User group type (CM, highlighted with a red arrow), Lead user, Vault name, and Owning group indicator. At the bottom left, there is a copyright notice: © 2005, Intergraph Corp. All Rights Reserved.

Any users that have been added will need to be added to the new groups.



Creating Transmittal User Groups

- Repeat for the next new group name and group type.

New user group	
User group name: *	DOCCONTROL
User group description:	Access to manipulate transmittal documents
User group type:	CM
Lead user:	<input type="button" value="▼"/>
Vault name:	<input type="button" value="▼"/>
Owning group indicator:	<input type="checkbox"/>

© 2005, Intergraph Corp.
All Rights Reserved.

Make sure that the **client** is added to the **EMAILONLY** group.



User/User Group Association

- Enter or accept the default for **User group user name** field.

New relationship	
User group name (Parent):	EMAILONLY
Login name (Child):	client
Plant list name: *	EFPLANTS
Project list name:	<input type="button" value="▼"/>
User group user name: *	client (EMAILONLY) EFPLANTS

© 2005, Intergraph Corp.
All Rights Reserved.

Also make sure that **dawn** gets added to the **DOCCONTROL** group.

User/User Group Association

- Enter or accept the default for **User group user name** field.



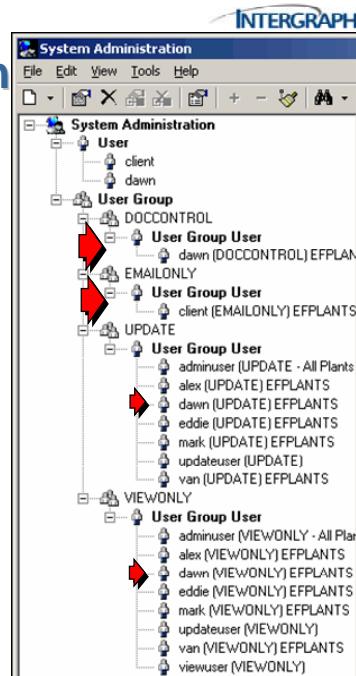
The screenshot shows the 'New relationship' dialog box. It has fields for 'User group name (Parent)' set to 'DOCCONTROL', 'Login name (Child)' set to 'dawn', 'Plant list name:' dropdown set to 'EFPLANTS' (with a red arrow pointing to it), 'Project list name:' dropdown set to 'EFPROJECTS', and 'User group user name:' text box containing 'dawn (DOCCONTROL) EFPLANTS'.

© 2005, Intergraph Corp.
All Rights Reserved.

This part of the setup is necessary in order to use the automated transmittal workflows.

User/User Group Association

- The new user group relationships are displayed in the *Tree View*.

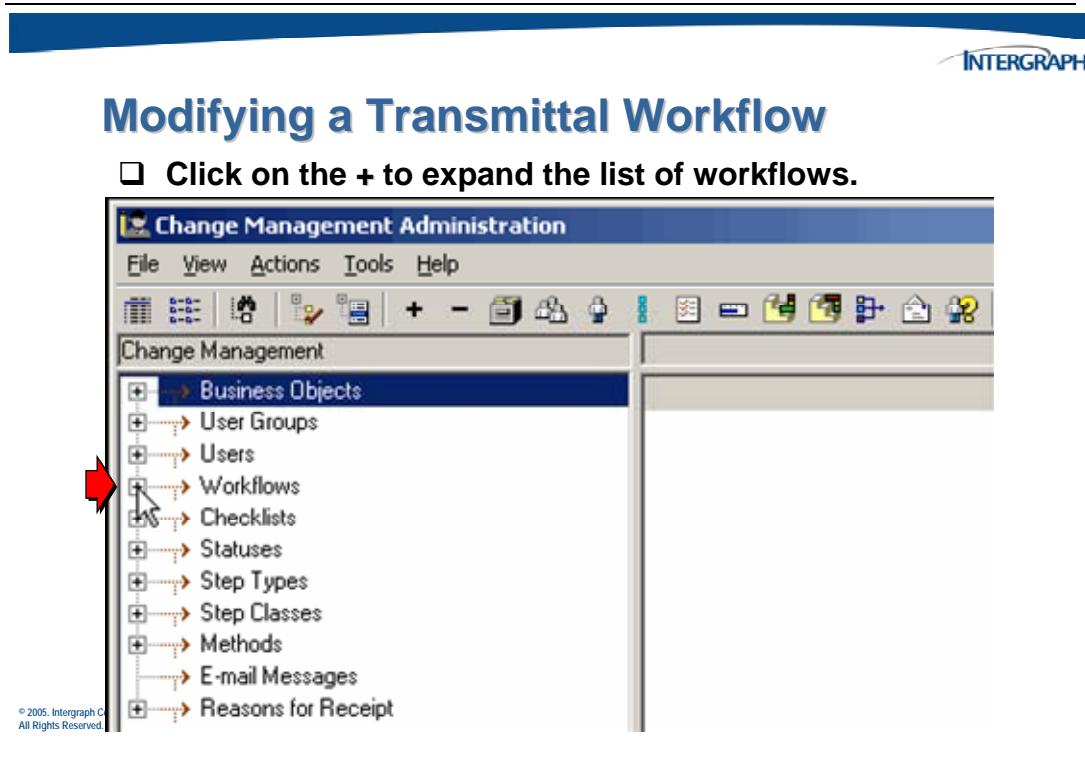


© 2005, Intergraph Corp.
All Rights Reserved.

7.4 Modifying Transmittal Workflows

There are two pre-defined workflows delivered with transmittals, one for internal transmittals and one for external transmittals. These are example workflows and can be modified as needed using the **Change Management Administration** utility.

Once logged in to the Change Management Administration utility, locate the transmittal workflows in the *Tree View* window.

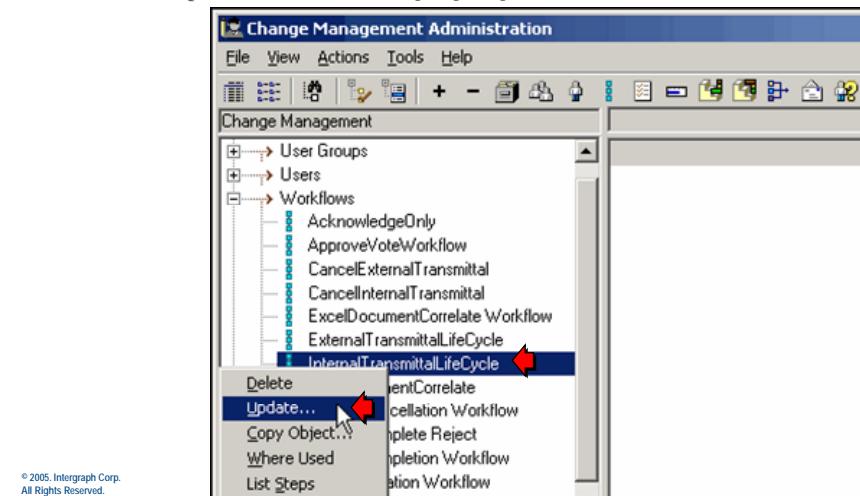


The two delivered workflows are the **ExternalTransmittalLifeCycle** workflow and the **InternalTransmittalLifeCycle** workflow.



Modifying a Transmittal Workflow

- Right-click on the *InternalTransmittalLifeCycle* and choose **Update** from the pop up menu.

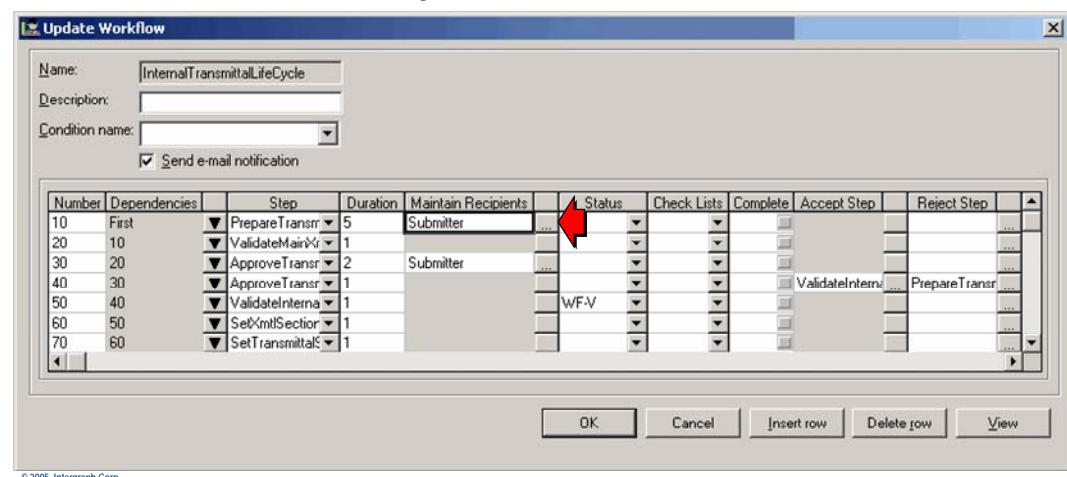


You can change the recipient (user) who will be responsible for the work in this step of the workflow.

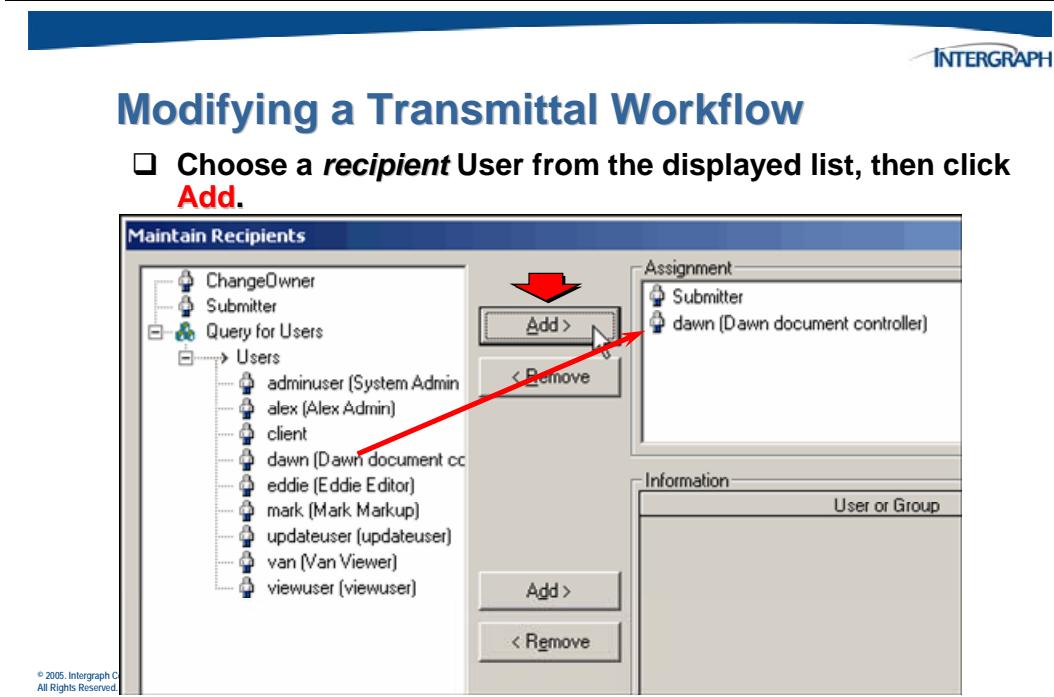


Modifying a Transmittal Workflow

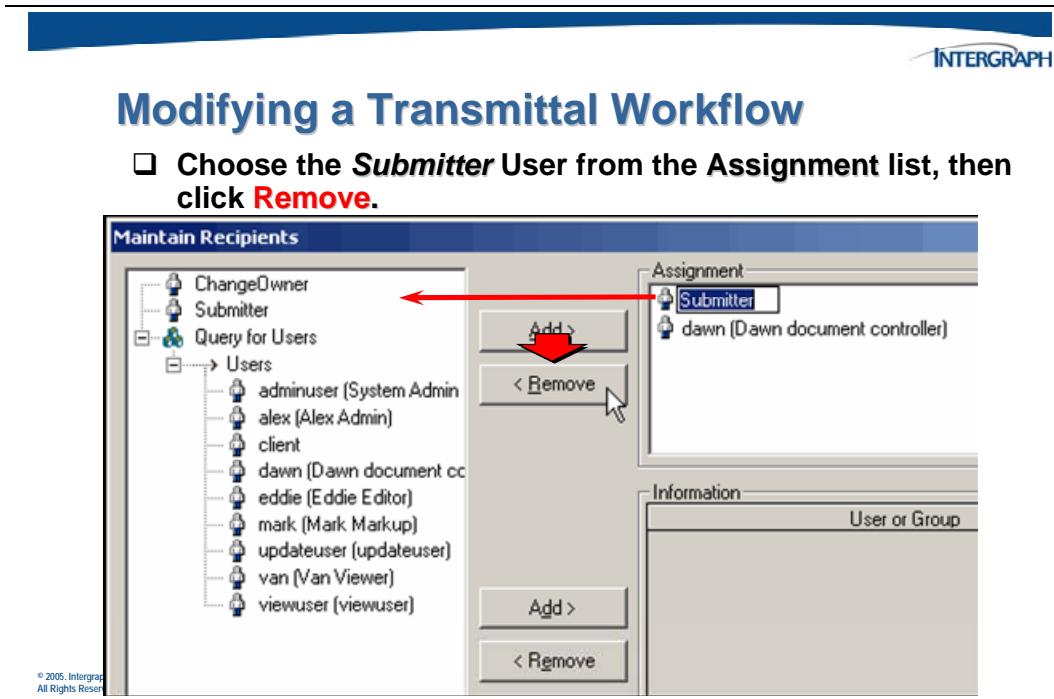
- Choose the ellipses to select the **recipient** that will perform this workflow step.



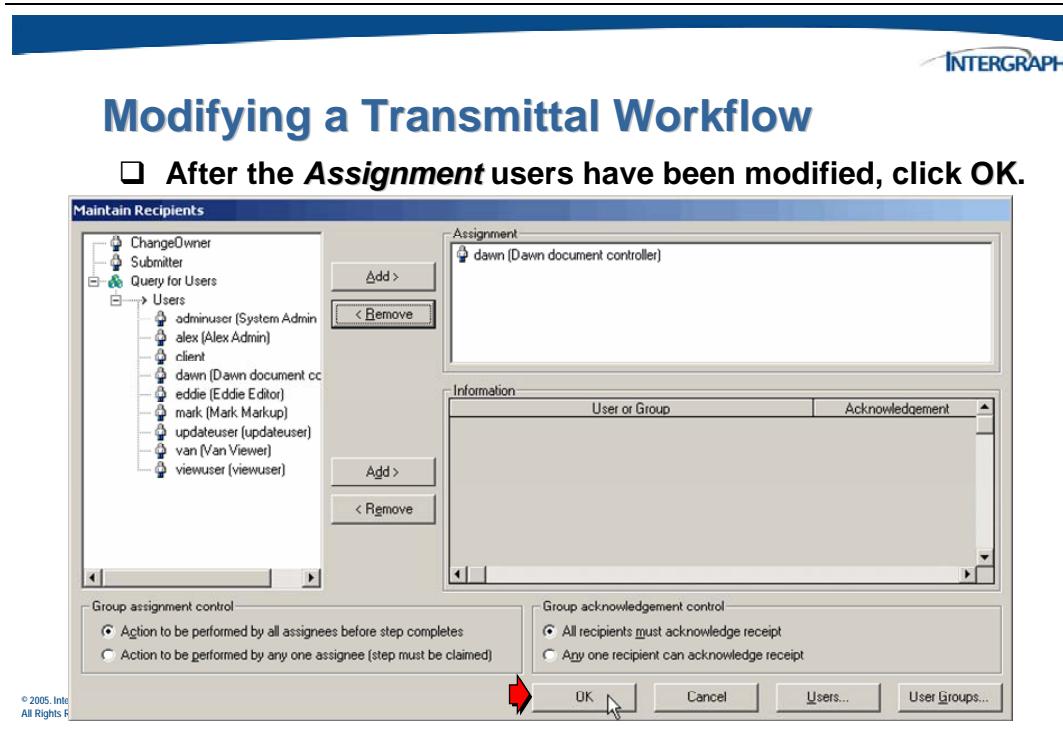
Perform a query to show the **Users** or **User Group** in the *Tree View* of the *Maintain Recipients* dialog. From the *Maintain Recipients* dialog, users can be added and removed from the **Assignment** and **Information** fields.



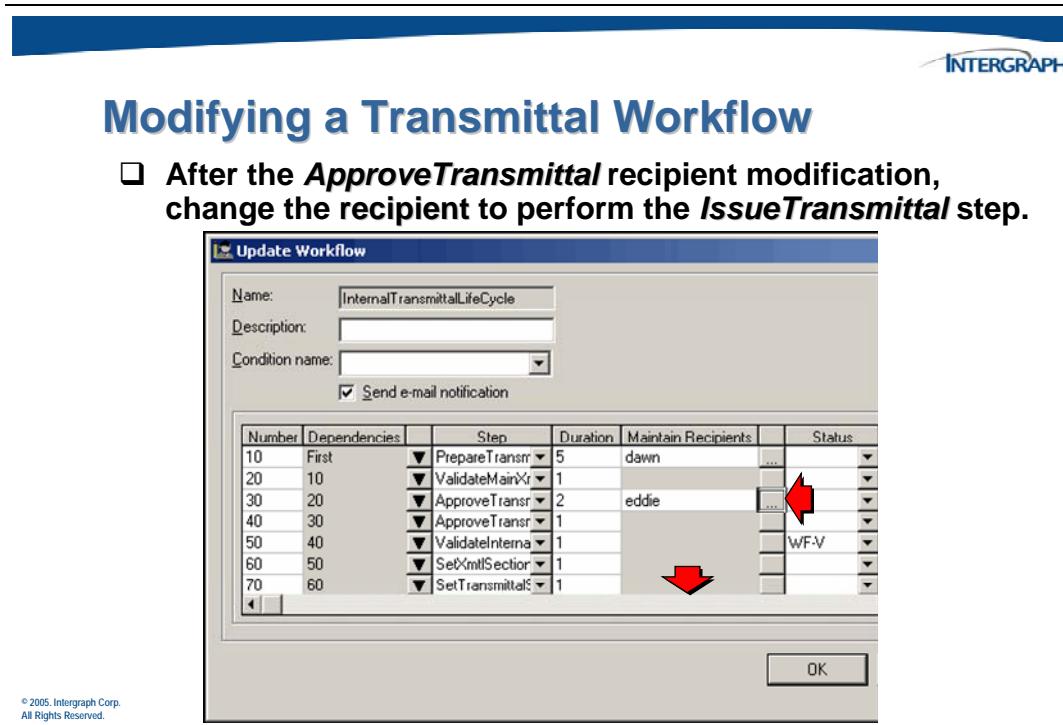
In these examples the default participant **Submitter** is being removed from the step *Assignment* and user **dawn** (document controller) is being added.



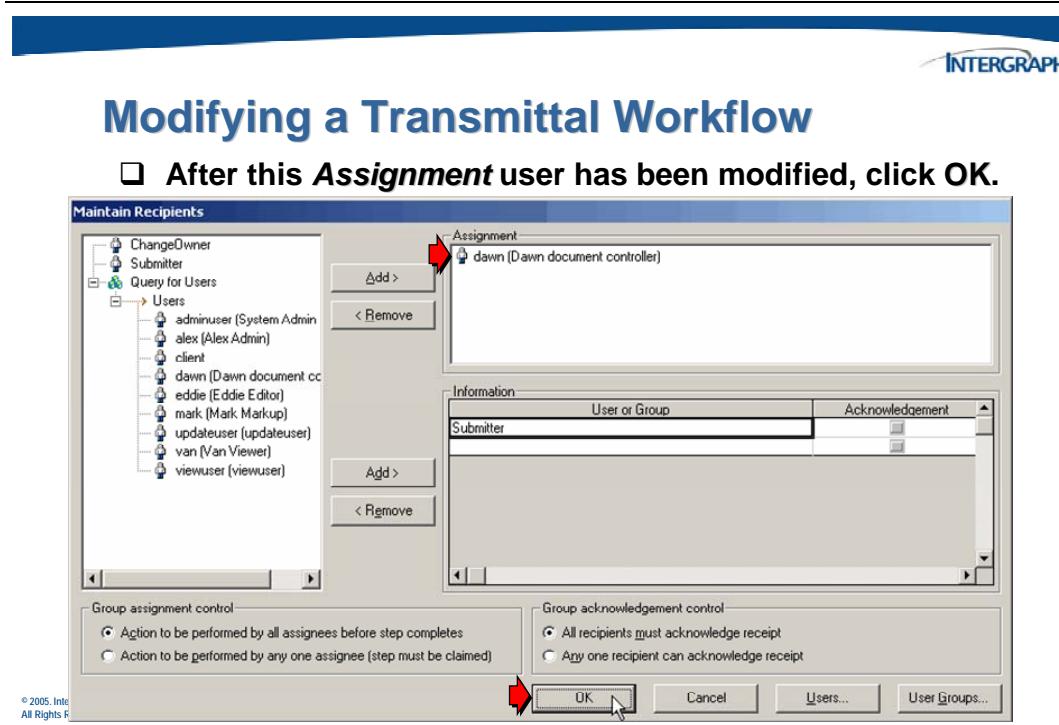
This completes the updates for this workflow step.



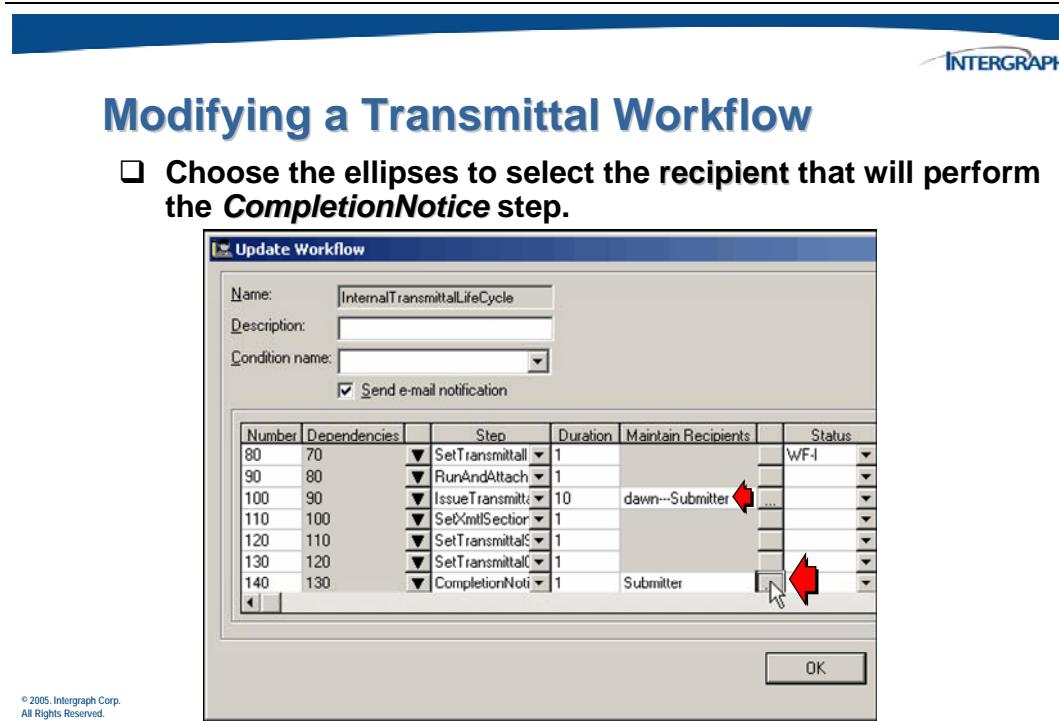
Make any changes for the other steps in the *InternalTransmittalLifeCycle* workflow.



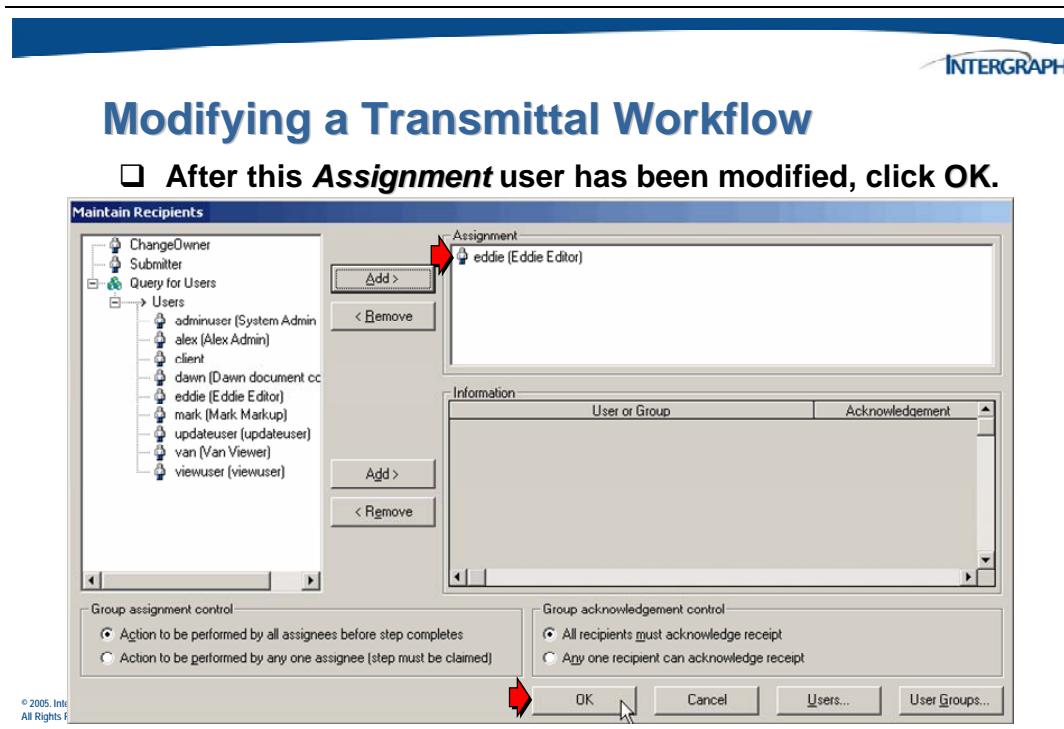
In this step the default participant **Submitter** has been supplemented with the user **dawn** (document controller) being added.



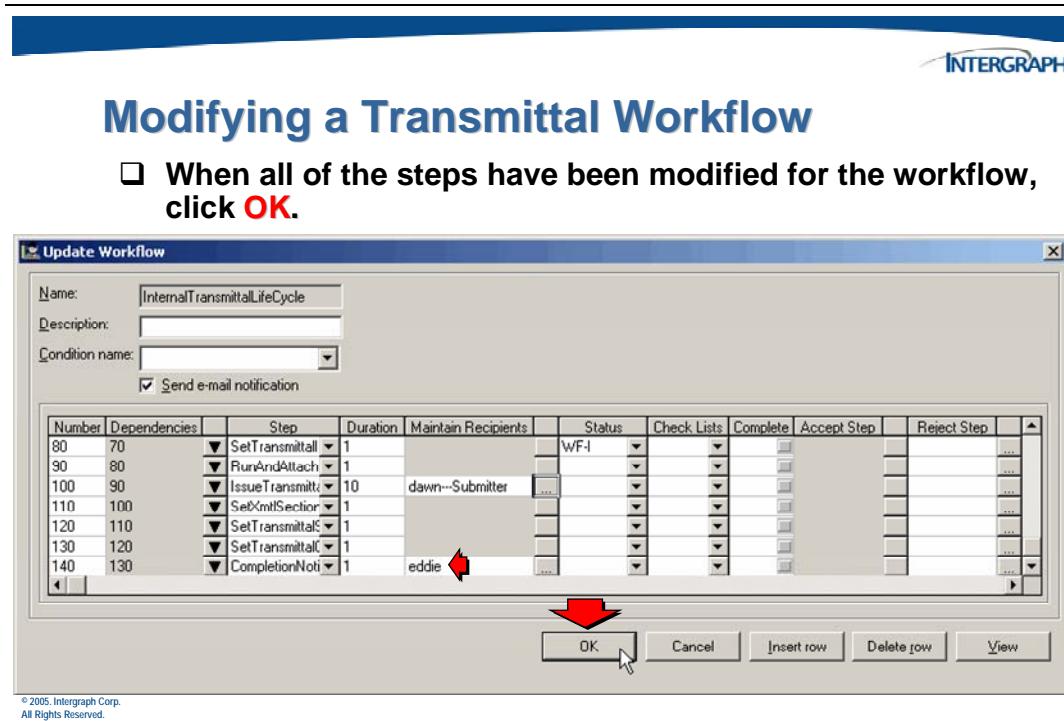
Also make a change to the *CompletionNotice* step by changing the participant.



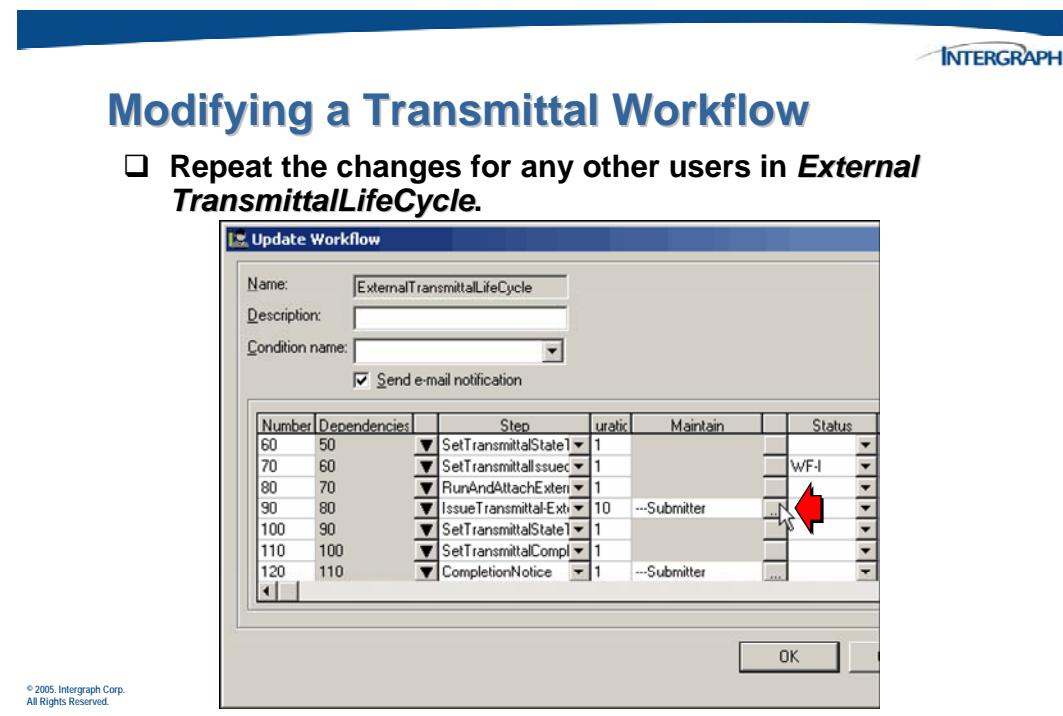
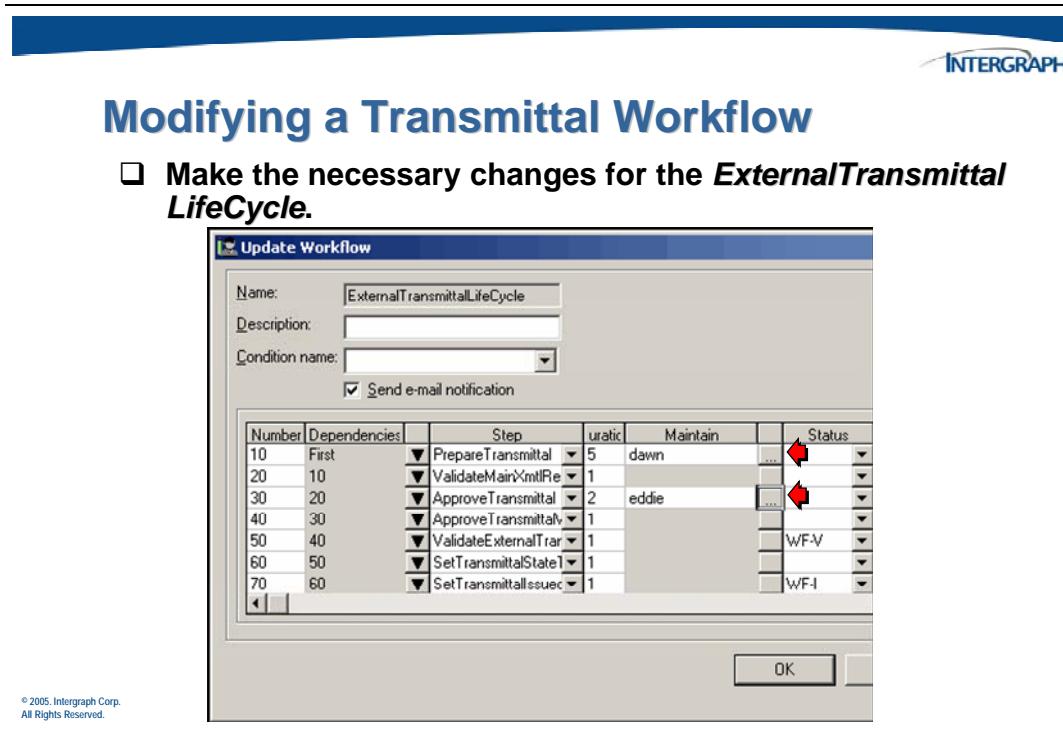
In this step the default participant **Submitter** has been removed from the step *Assignment* and user **eddie** has been added.



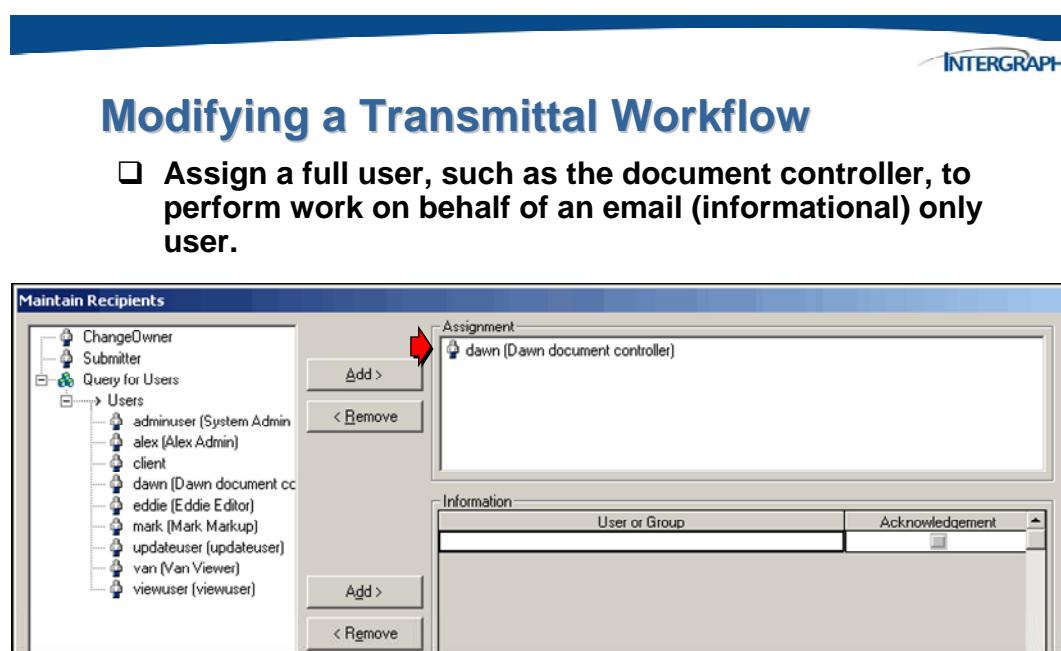
Now that the step participants have been changed, **OK** and save the workflow updates.



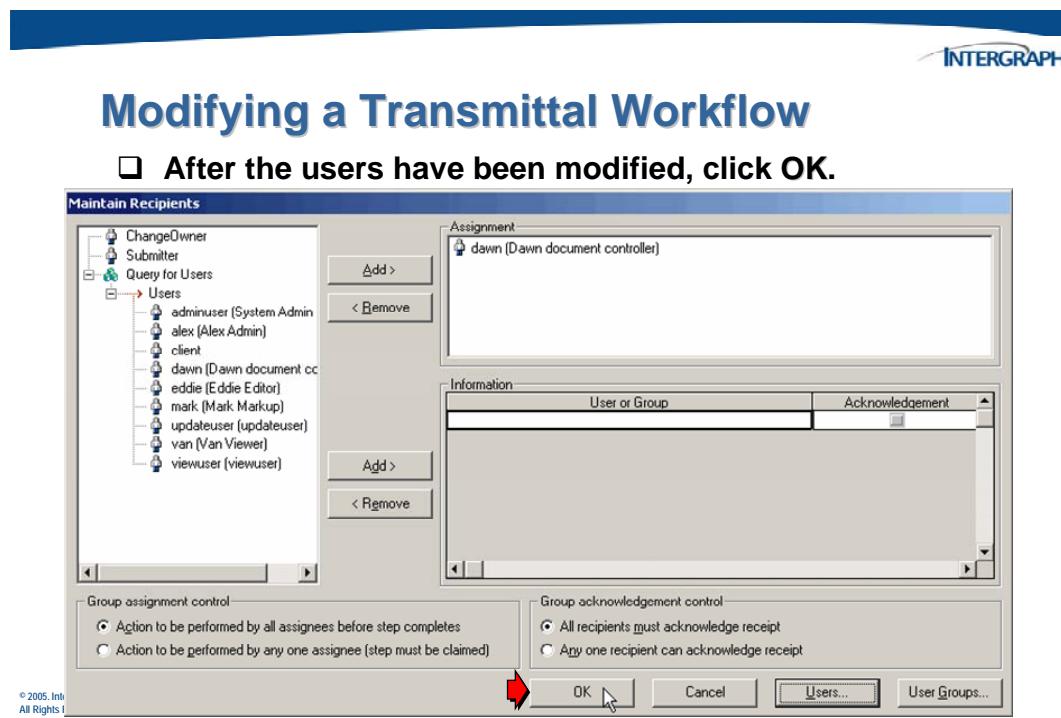
The previous modifications apply to the **internal** transmittal workflow. The same kinds of changes can be made to the **external** transmittal workflow.



In this step the default participant **Submitter** has been removed and user **dawn** (document controller) has been added along with the external **client** user.



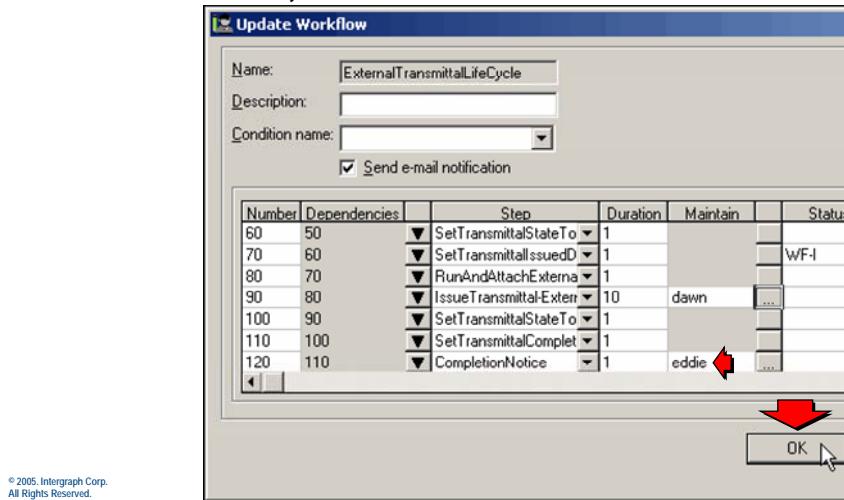
© 2005, Intergraph Corp.
All Rights Reserved.



Now that the step participants have been changed, **OK** and save the workflow updates.

Modifying a Transmittal Workflow

- When all of the steps have been modified for this workflow, click **OK**.



© 2005. Intergraph Corp.
All Rights Reserved.

7.5 Distribution Matrix Overview

One of the most difficult tasks performed by the Document Controller is the management of the project personnel index and the Transmittal Distribution Matrices.



Distribution Matrix Overview

There are 2 kinds of Distribution Matrices required for the Internal Transmittal and External Transmittal, because they are categorized by different properties:

- Internal Matrix**
- External Matrix**

Each distribution matrix is created for a specific Transmittal Reason For Issue. The recipients are added into the matrix for different document types and contracts (for Internal Transmittals) and companies (for External Transmittals).

The Reason For Issue is termed a header property as it identifies the complete matrix. The document type and contract are row properties that are set up in the body of the matrix.

© 2005. Intergraph Corp.
All Rights Reserved.

The Transmittal Distribution Matrix is used to determine who receives documentation via Transmittals and what is required of them. The term *matrix* is applied to the table with recipients along the top and transmittal categories down the side. Transmittal categories could be based on document type, project, contract, etc. Each 'cell' is marked with the *Reason For Receipt* for each recipient. There is usually more than one matrix for each type of Transmittal with the unique factor being the *Reason For Issue*.



Distribution Matrix Overview

When a Transmittal is created, the selected **Reason For Issue** determines which matrix to utilize.

To determine the recipients that are to be obtained from the matrix, the row properties are matched to extract the correct row from the matrix.



Distribution Matrix Overview

The **Internal Transmittal** matrix information to use is determined by the **Reason For Issue**. It is further determined by:

- Contract
- Document Category

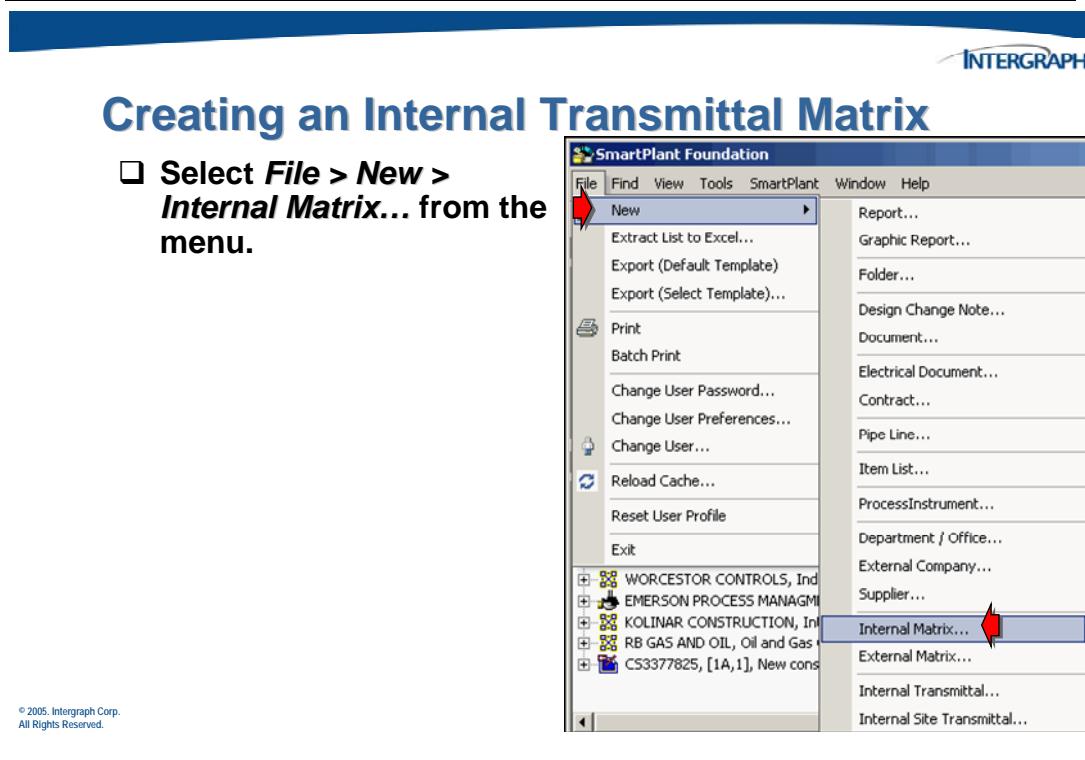
The **External Transmittal** matrix information to use is determined by the **Reason For Issue**. It is further determined by:

- Document Category
- External Organization

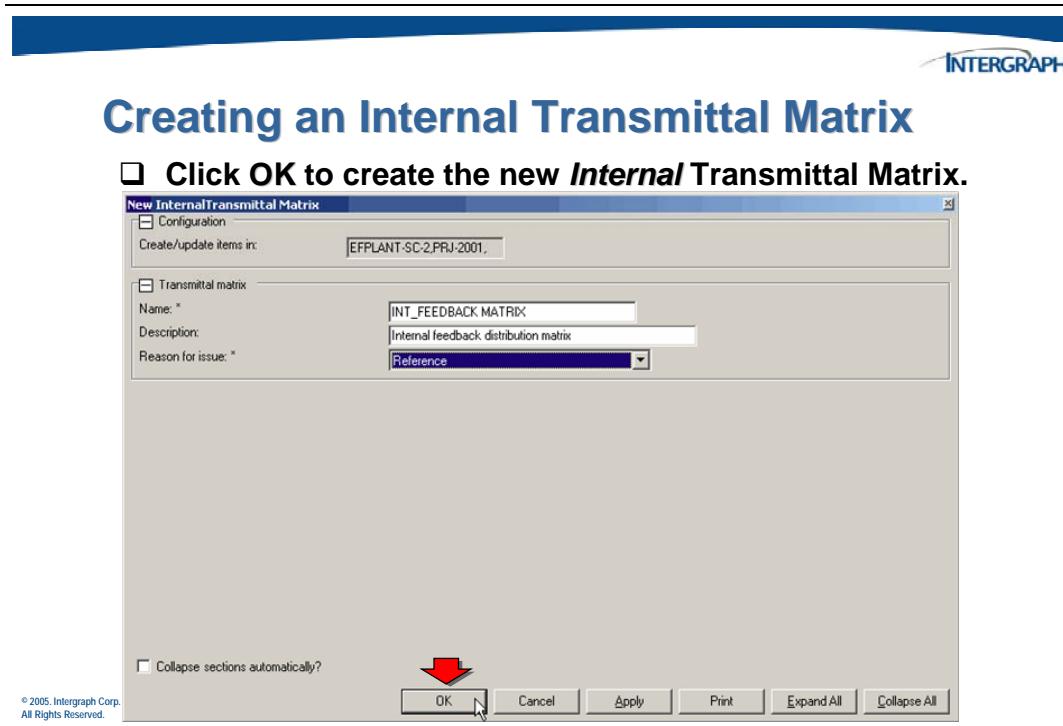
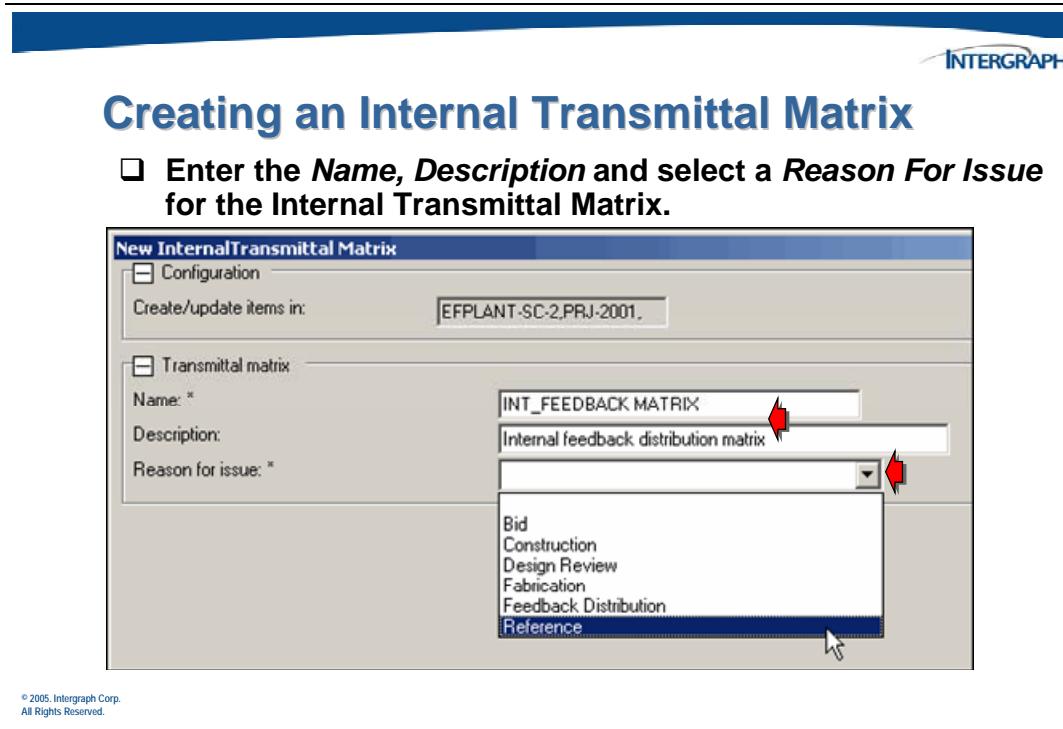
To determine the recipients that are to be extracted from the matrix, these additional properties on the transmittal are matched to those in the rows of the matrix.

7.5.1 Creating an Internal Transmittal Matrix

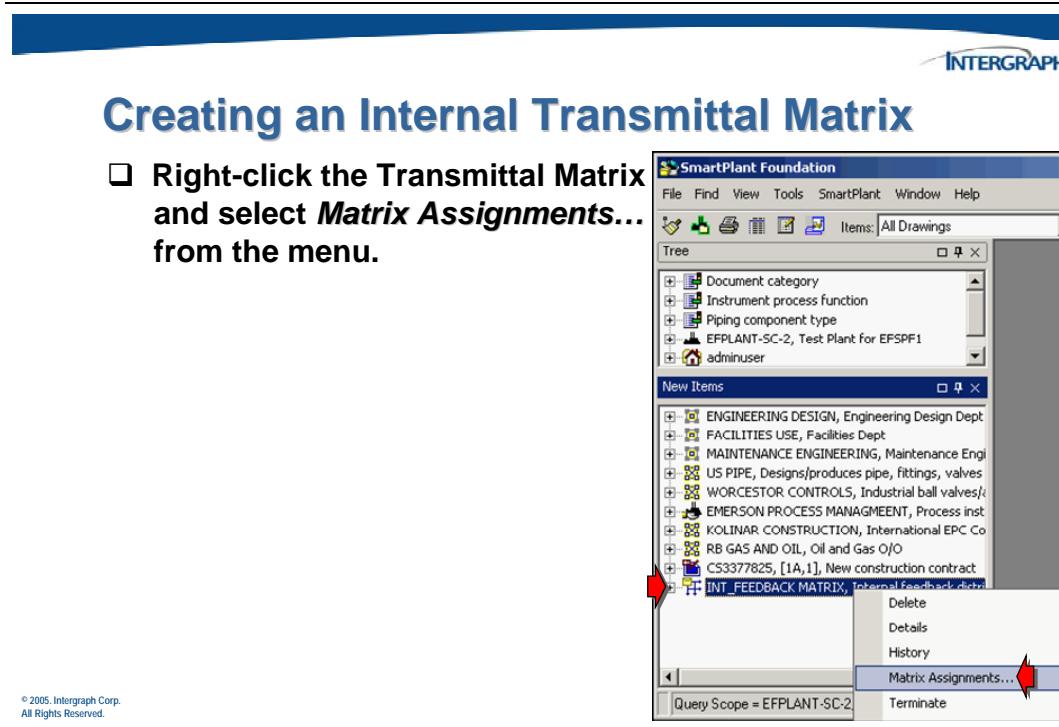
The following section will demonstrate how to create an **Internal Transmittal Matrix**.



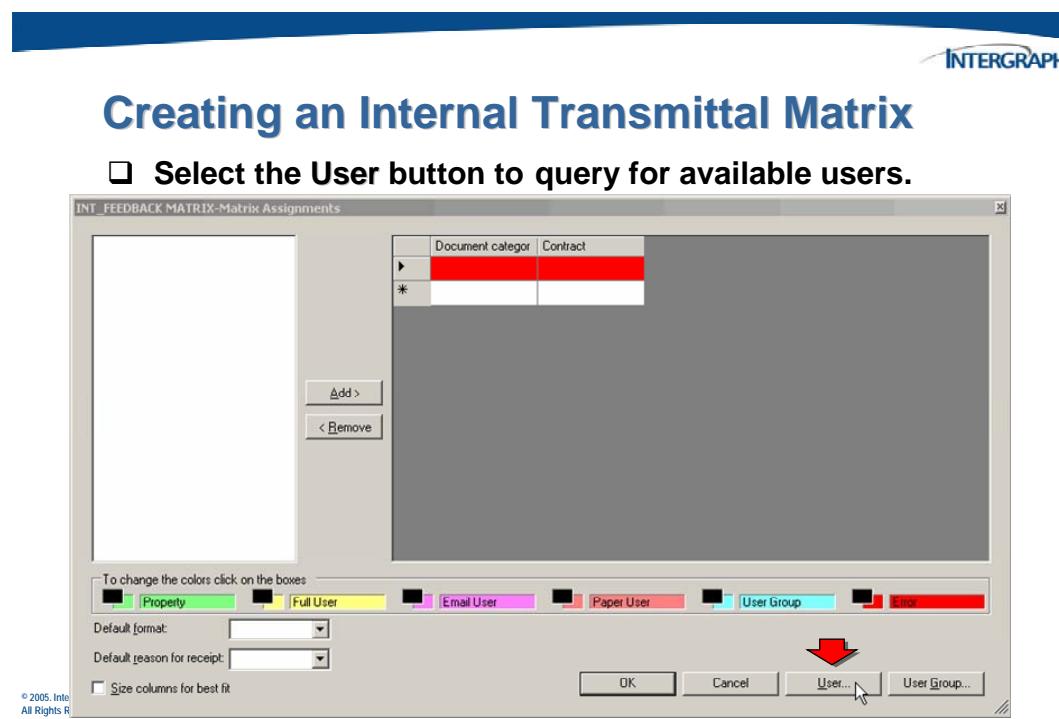
The *New Internal Transmittal Matrix* dialog will display.



Once the *Internal Transmittal Matrix* has been created, the next step is to add the users that will be associated with this matrix.



The *Matrix Assignments* dialog will display.

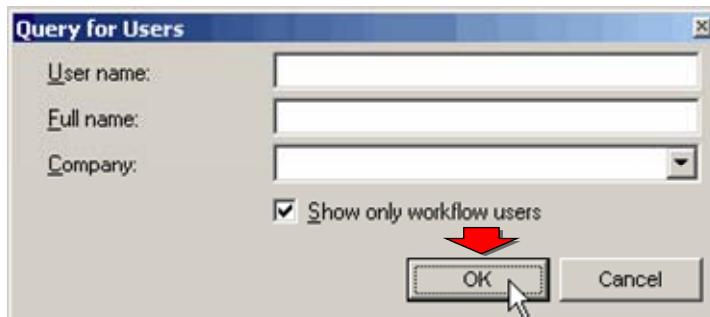


Users must be members of a User Group of type **CM** for the Workflow functionality to be available to them. This is done in System Administration for each User Group.



Creating an Internal Transmittal Matrix

- ❑ Enter the search criteria to locate users you want to assign to the *Internal Transmittal Matrix*.



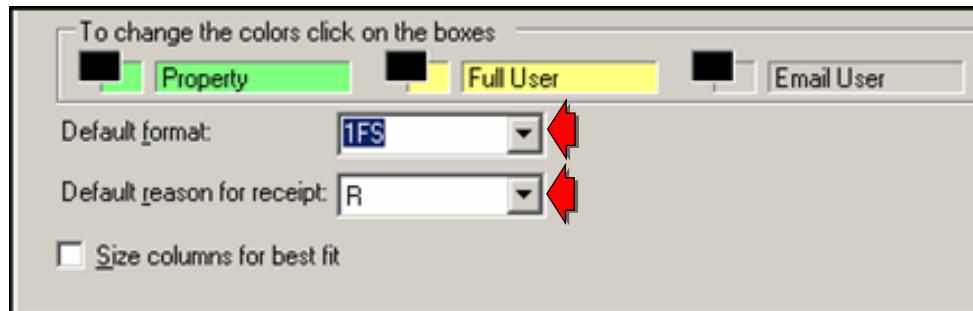
© 2005, Intergraph Corp.
All Rights Reserved.

On the *Matrix Assignments* dialog, there is a **Default format** and **Default reason for receipt**. Select the values that will be default settings for the Internal Matrix.



Creating an Internal Transmittal Matrix

- ❑ Set the **Default reason for receipt** and the **Default format** on the Internal Transmittal Matrix.



© 2005, Intergraph Corp.
All Rights Reserved.



Creating an Internal Transmittal Matrix

- Select a user from the tree view and click the Add button to add the user to the Matrix.

INT_FEEDBACK MATRIX-Matrix Assignments

Query for Users

- adminuser (System Admin L)
- alex (Alex Admin)**
- client
- dawn (Dawn document cor
- eddie (Eddie Editor)
- mark (Mark Markup)
- updateuser (updateuser)
- van (Van Viewer)
- viewuser (viewuser)

	Document category	Contract
*		

© 2005, Intergraph Corp.
 All Rights Reserved.

Users are added to the Transmittal Matrix. After setting the *Reason for Receipt* and *Format* for each recipient, these settings appear in the matrix. The Internal Transmittal matrix information to use when a transmittal is created is determined by *Document Type* and *Contract*.



Creating an Internal Transmittal Matrix

- Click in the *Document category* field to display a picklist of values.

INT_FEEDBACK MATRIX-Matrix Assignments

Query for Users

- adminuser (System Admin L)
- alex (Alex Admin)**
- client
- dawn (Dawn document cor
- eddie (Eddie Editor)
- mark (Mark Markup)
- updateuser (updateuser)
- van (Van Viewer)
- viewuser (viewuser)

	Document category	Contract	alex
*	<input type="button" value="▼"/>	R-1FS	

3D Documents
Dimensional Data
Electrical Docu
Equipment Docu
General Docume
Instrument Docu
P&ID Document

© 2005, Intergraph Corp.
 All Rights Reserved.



Creating an Internal Transmittal Matrix

- Click in the **Contract** field to select a valid contract number.

INT_FEEDBACK MATRIX-Matrix Assignments

Document categor	Contract	alex
P&ID Documents	<input type="text"/>	R-IFS
*	<input type="text"/>	CS3377825

Query for Users

- adminuser (System Admin L)
- alex (Alex Admin)
- client
- dawn (Dawn document cor
- eddie (Eddie Editor)
- mark (Mark Markup)
- updateuser (updateuser)
- van (Van Viewer)
- viewuser (viewuser)

Add > < Remove

© 2005, Intergraph Corp.
All Rights Reserved.

When the Internal Transmittal Matrix is complete, select the **OK** button on the *Matrix Assignments* dialog.



Creating an Internal Transmittal Matrix

- Click **OK** to complete the Internal Matrix Assignments.

INT_FEEDBACK MATRIX-Matrix Assignments

Document categor	Contract	alex	eddie	van
P&ID Documents	CS3377825	R-IFS	R-IFS	R-IFS
*				

To change the colors click on the boxes

Default format: IFS
Default reason for receipt: R

Size columns for best fit

OK Cancel User... User Group...

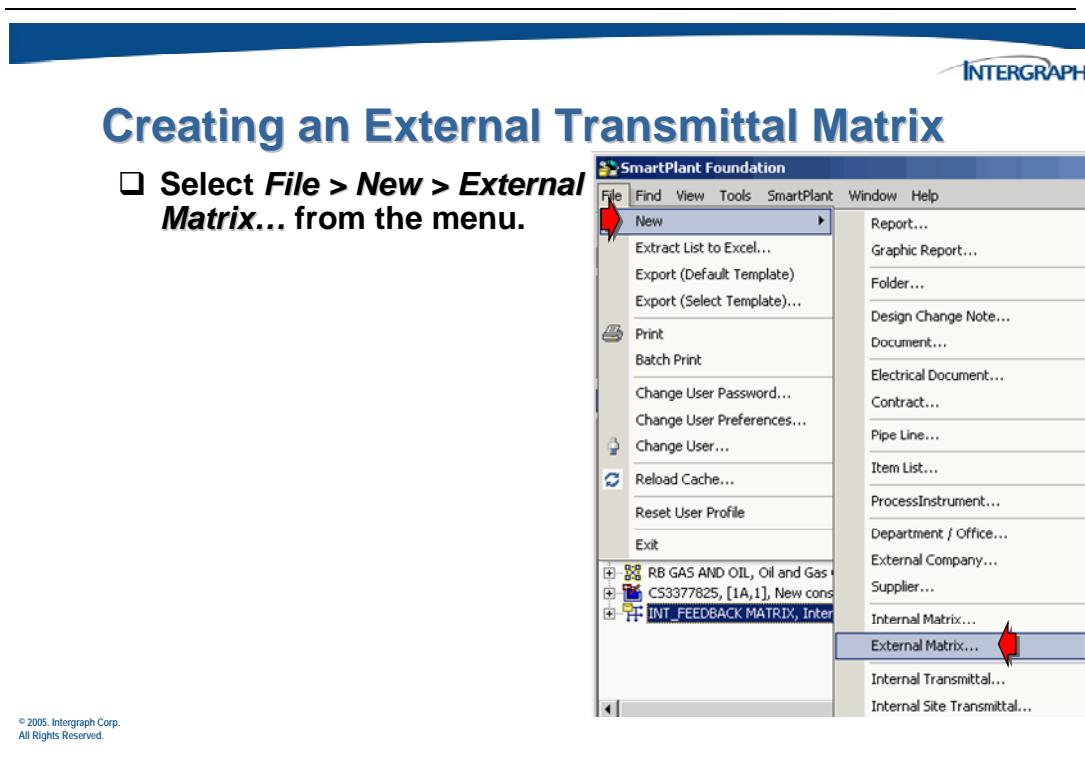
© 2005, Intergr
All Rights Rese

Other actions that can be performed on the Transmittal Matrix:

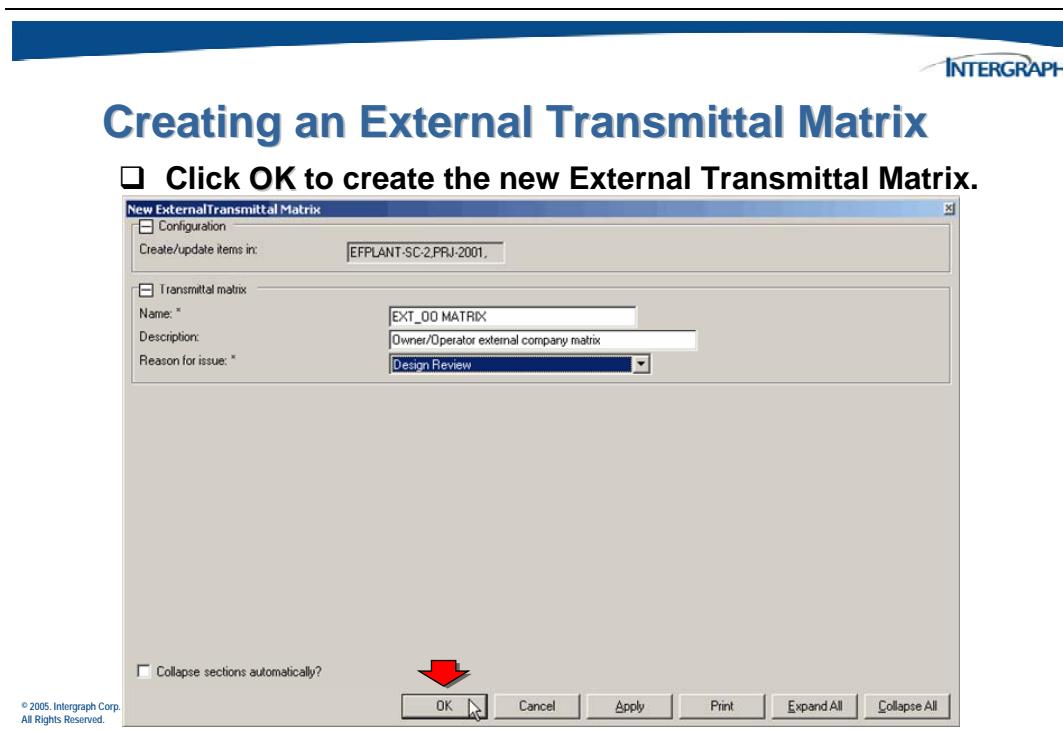
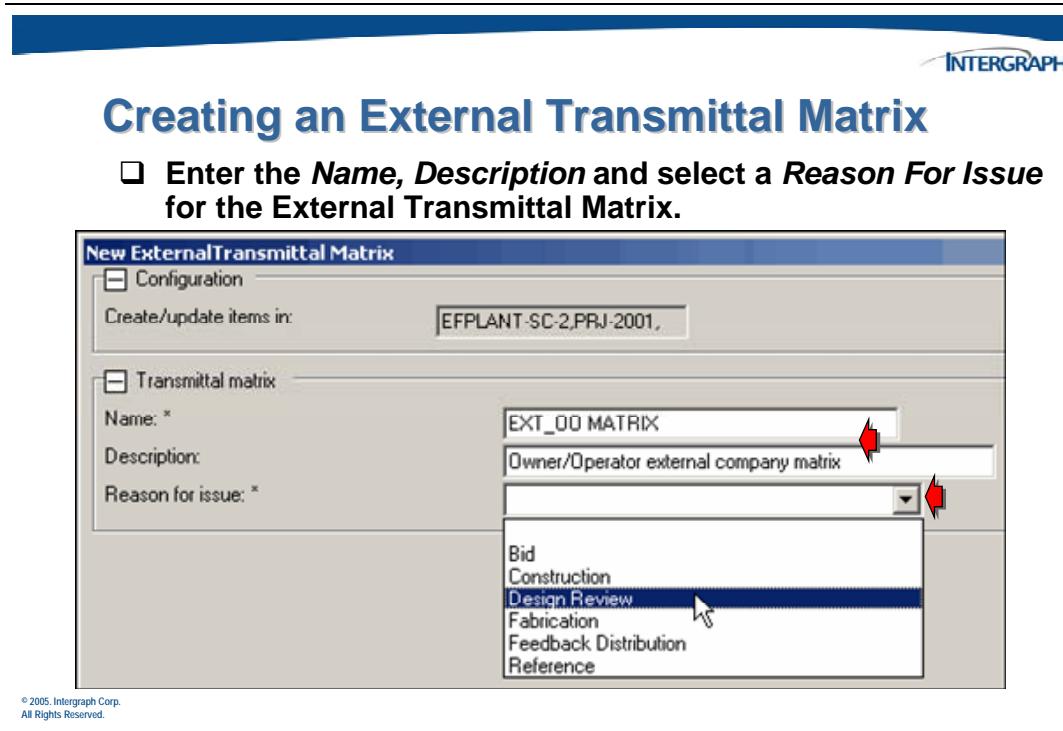
- Add a new matrix row** - Select the * on the next row or right-click to add a new Document Type or Contract.
- Duplicate a recipient or row** - right-click in the column or row header.
- Export to spreadsheet** - right-click in the top left cell of the border and paste from Clipboard into spreadsheet.
- Import from spreadsheet** – copy selected area from spreadsheet into Clipboard. Then right-click in the top left cell of the border.
- Data Validation** - It is valid to enter values for the row attributes that are not listed in the picklists, such as a *Contract* that does not yet exist. However, it is not valid to enter a *Reason for Receipt* that is not yet available in its picklist. The validation takes place on selection of the OK button to capture any cut-n-paste errors in the data.

7.5.2 Creating an External Transmittal Matrix

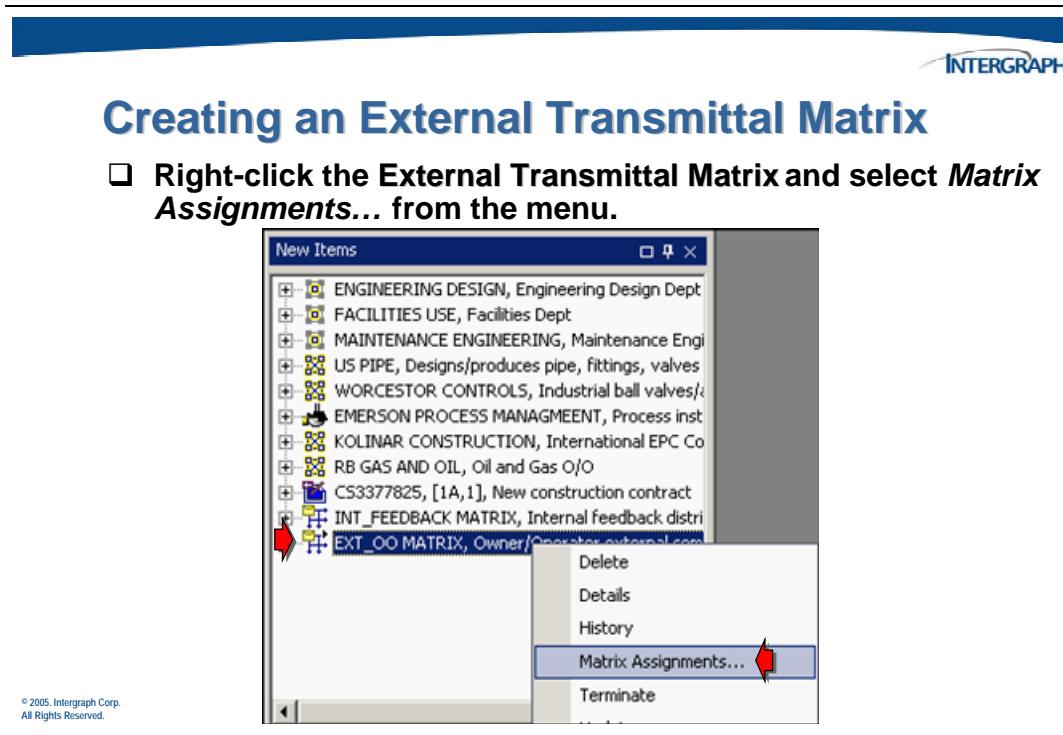
This section demonstrates how to create an **External Transmittal Matrix**. The procedure for creating it is similar to the *Internal Transmittal Matrix*.



The *New External Transmittal Matrix* dialog will display.

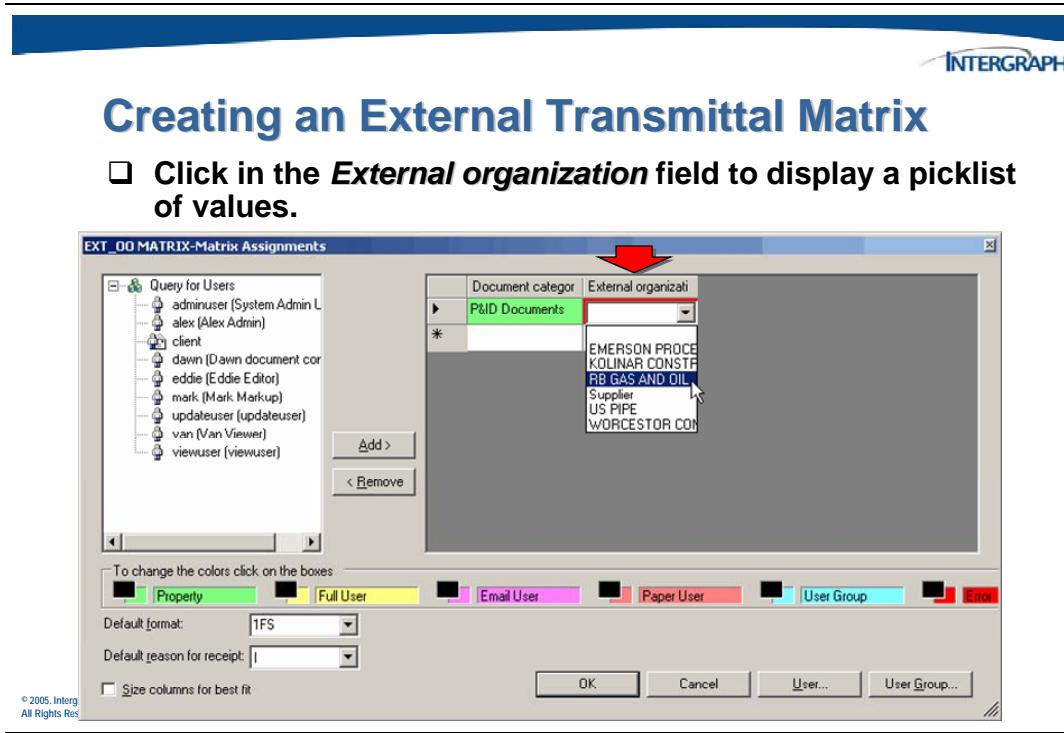


Next, add the users that will be associated with the External Transmittal Matrix.

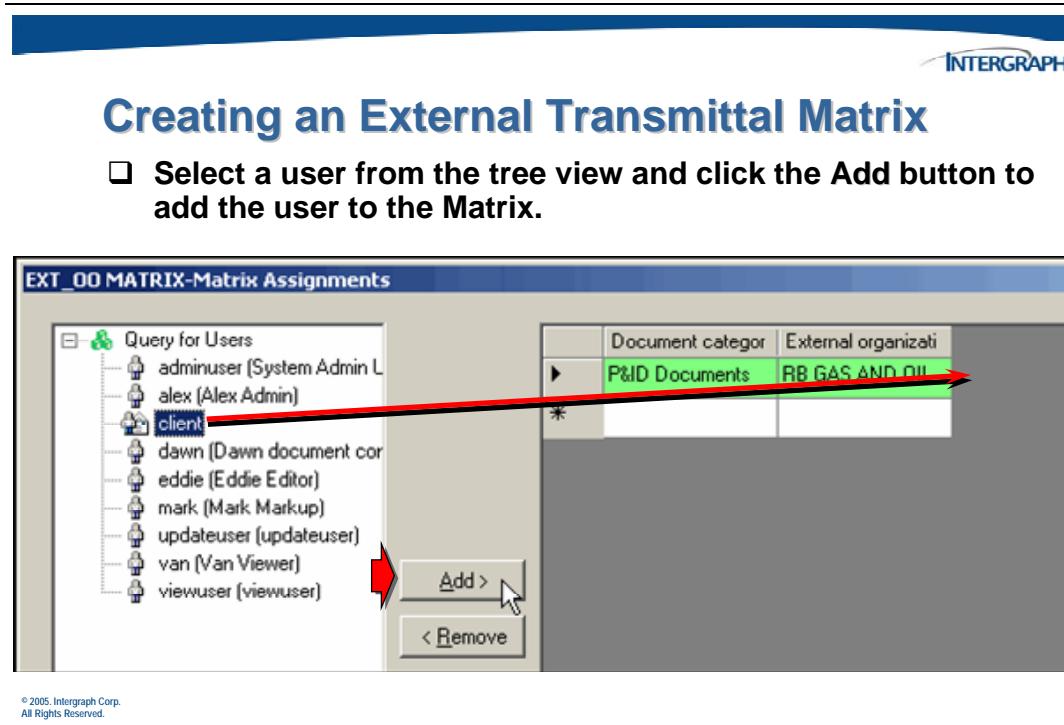


The External Matrix is slightly different than the Internal Matrix. Where the Internal Matrix contained columns for *Document category* and *Contract* along with the recipients associated to them, the External Matrix contains columns for *External organization* (Company) and *Document category*.

The *Matrix Assignments* dialog displays.



Setup the Matrix Assignments for the External Transmittal Matrix in much the same way as for the Internal Transmittal Matrix.



The *document controller* user (**dawn**) will actually be responsible for updating the feedback from the external recipient.

Document category	External organization	client	dawn
P&ID Documents	RB GAS AND OIL	I-1FS	
*			

To change the colors click on the boxes
Property Full User Email User Paper User User

Default format: IFS
Default reason for receipt: W

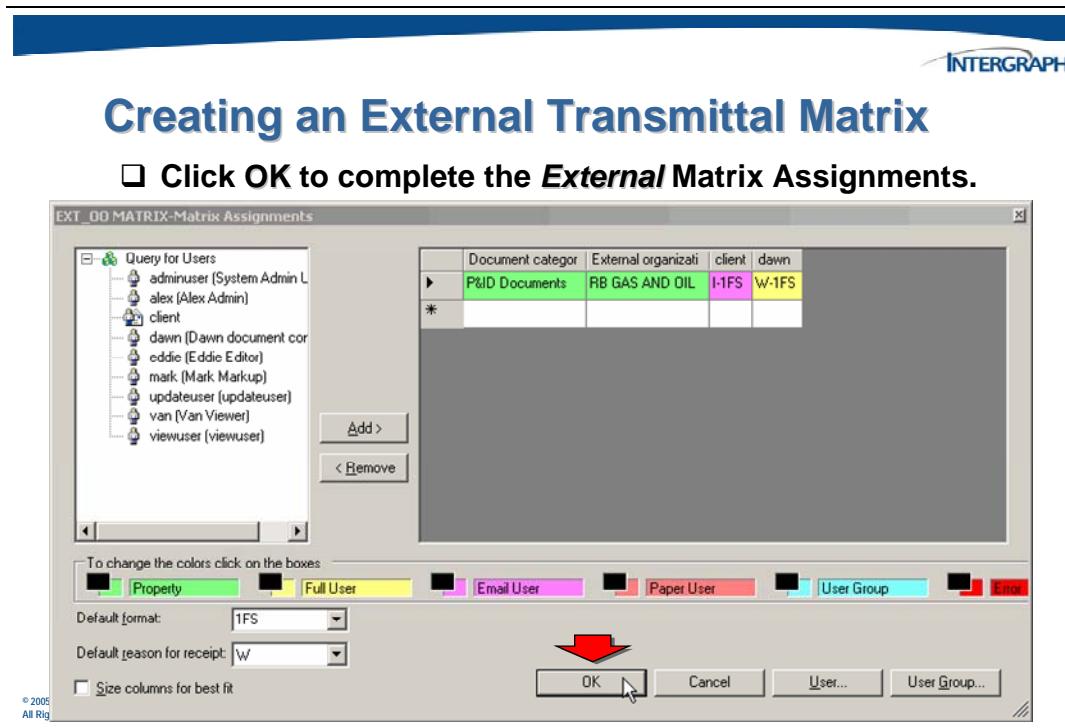
© 2005, Intergraph Corp.
All Rights Reserved.

On the Matrix Assignments dialog, there is a *Default format* and *Default reason for receipt*. These values will be default settings for the users added to the External Matrix.

Document category	External organization	client	dawn
P&ID Documents	RB GAS AND OIL	I-1FS	W-1FS
*			

© 2005, Intergraph Corp.
All Rights Reserved.

When the External Transmittal Matrix is complete, select the **OK** button on the *Matrix Assignments* dialog.

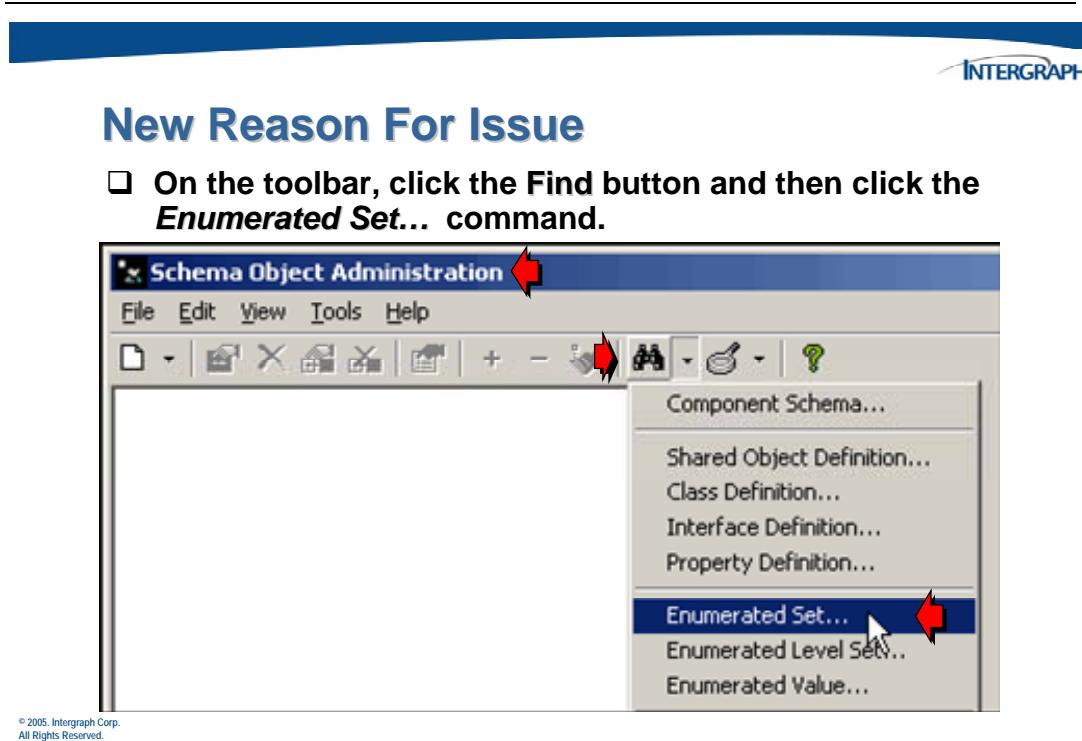


7.6 Creating a New Reason For Issue

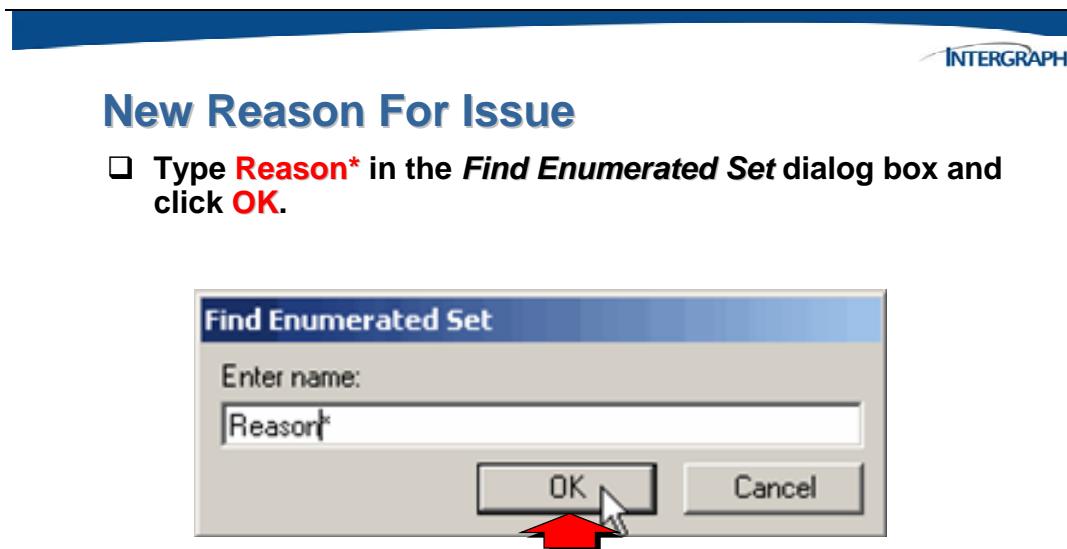
One of the fields used in the creation of a transmittal object is the **Reason For Issue**. This field is used to determine which set of recipients will be populated on the maintain recipients dialog when the transmittal is created.

The **Reason For Issue** is an *Enumerated Set* (Enumerated List Type) that can be customized by adding additional *Enum Details* (EnumEnums).

In the following example, adding a new Reason For Issue entry is demonstrated. Start by locating the existing **Reason For Issue** *enumerated set*.

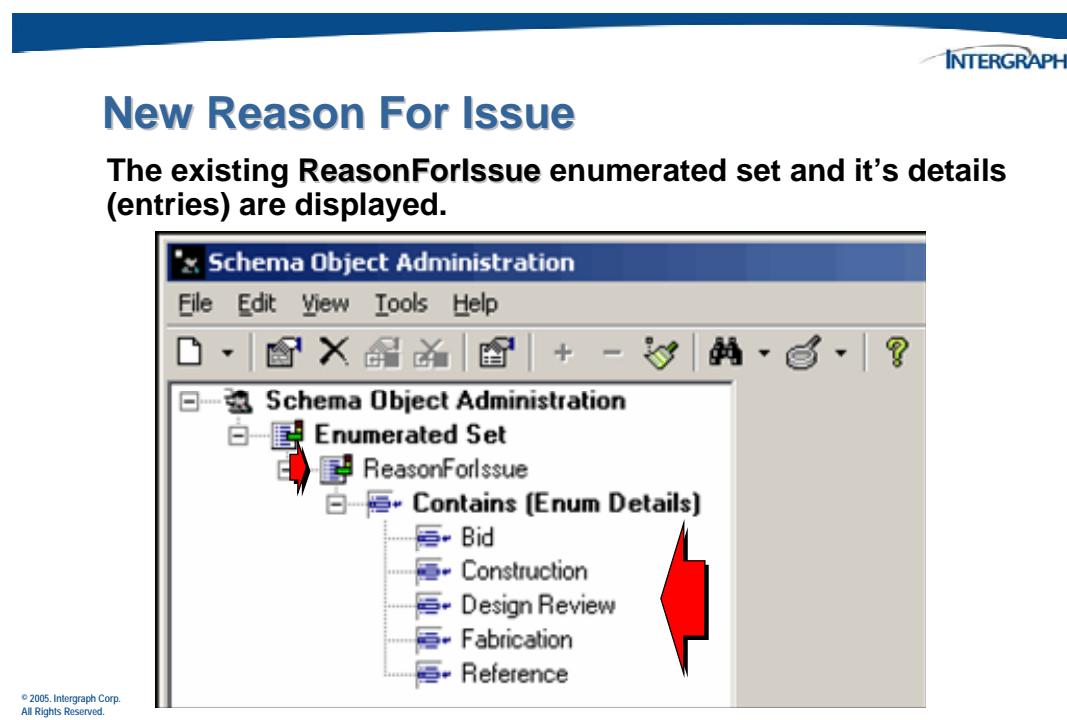


The *Find Enumerated Set* search dialog will display.

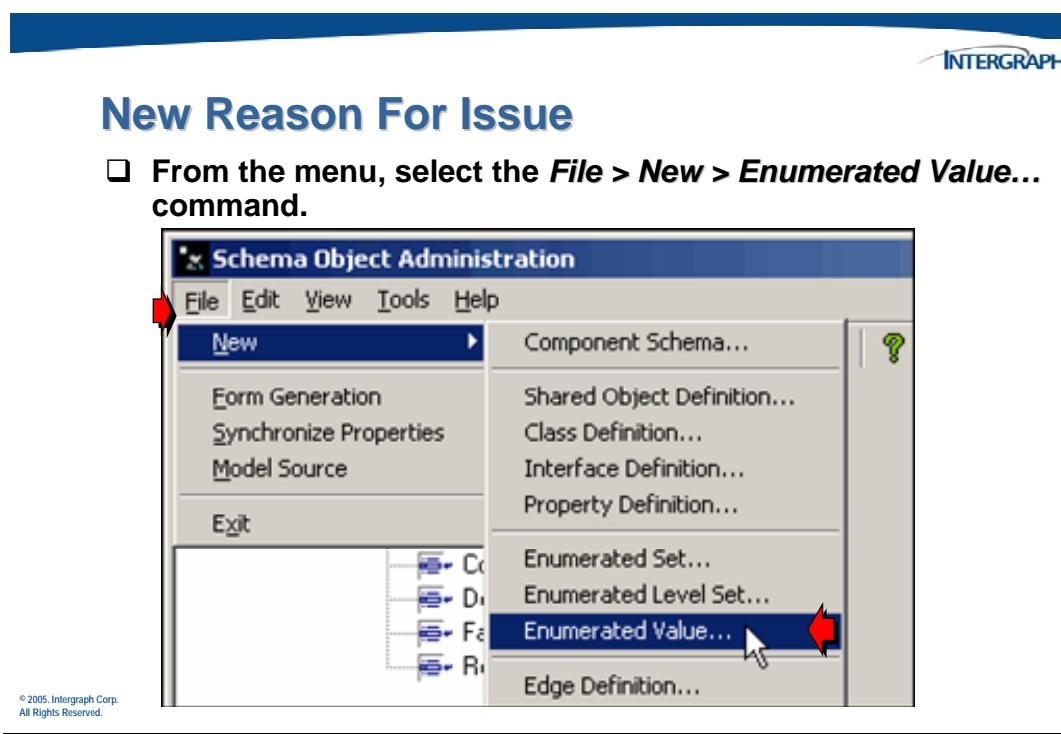


© 2005, Intergraph Corp.
All Rights Reserved.

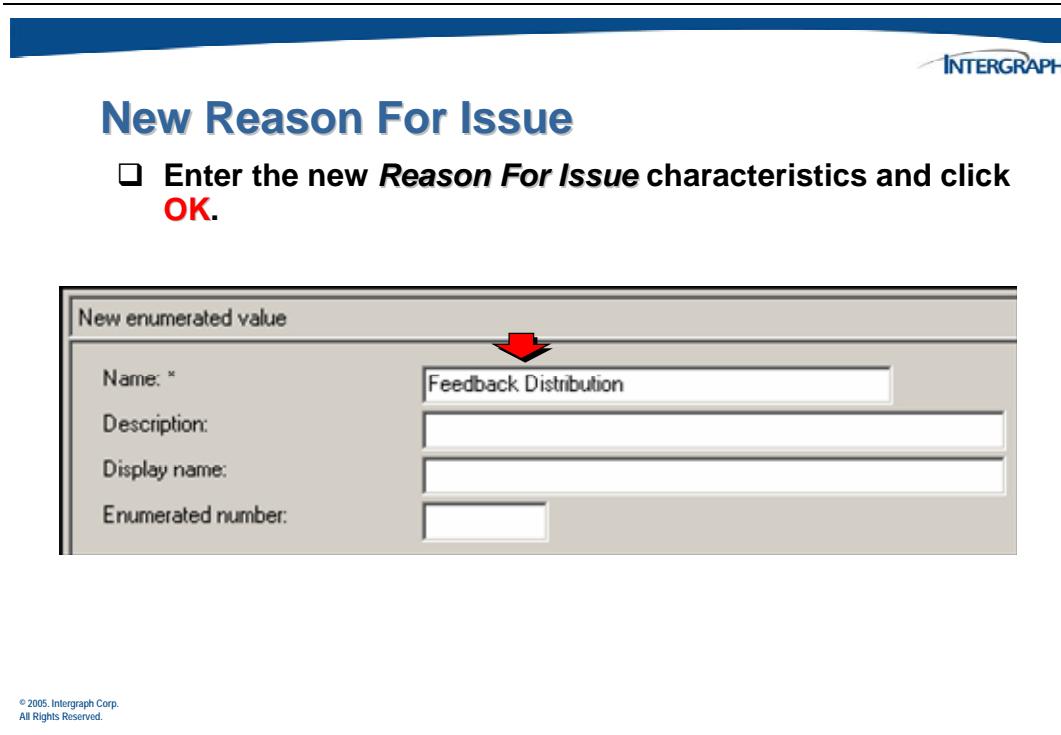
Expand the **ReasonForIssue** enumerated set in the *Tree View*.



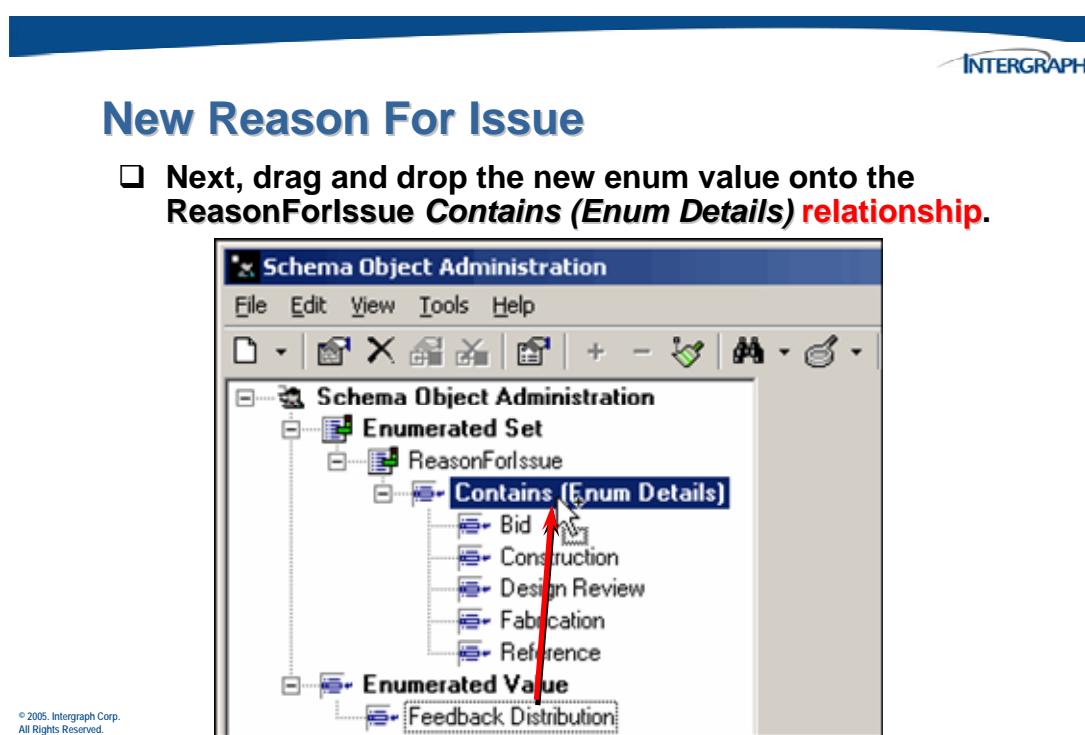
Now, add any new desired entries to this enumerated set.



The *New enumerated value* dialog will display.



Finally, associate the new entry with the **ReasonForIssue** enumerated set.



7.7 Activity 1 – Transmittal Administrative Setup

In this activity you will be performing some administrative setup that will be needed by Transmittals. First, you will add the necessary Transmittal user, user group and workflow modification. Then you will add Contract, Department, External Company and Supplier objects to be used by the browse buttons when creating Transmittals. Finally you will create Transmittal Matrices for both internal and external transmittals.

If you are not currently logged into your machine:

1. Log on to your operating system (if not already logged in):
spfuser with no password

2. Start the *Desktop Client interface* by selecting **Start > All Programs > Intergraph SmartPlant Foundation > SmartPlant Foundation Desktop Client**.

3. When the *Login* dialog window appears, use the *User name* **adminuser** with no password and click on **OK**.

4. Create a Contract object using the **File > New > Contract...** menu command. When the *New Contract* dialog displays enter the following information:

<input type="text"/> Name:	CS3377825
<input type="text"/> Description:	New construction contract
<input type="text"/> Document title:	Construction contract for CS phase

Click **OK** to create the new Contract object.

5. Create an Internal Organization object using the **File > New > Department / Office** menu command. When the *New Department* dialog displays enter the following information:

<input type="text"/> Name:	ENGINEERING DESIGN
<input type="text"/> Description:	Engineering Design Dept
<input type="text"/> Telephone Number:	X-3489

Click **Apply** to create the new Department object but leave the dialog displayed.

Create another Internal Organization object using the following information:

<input type="checkbox"/> Name:	FACILITIES USE
<input type="checkbox"/> Description:	Facilities Dept
<input type="checkbox"/> Telephone Number:	X-1140

Click **Apply** to create the new Department object but leave the dialog displayed.

Create a final Internal Organization object using the following information:

<input type="checkbox"/> Name:	MAINTENANCE ENGINEERING
<input type="checkbox"/> Description:	Maintenance Dept
<input type="checkbox"/> Telephone Number:	X-8040

Click **OK** to create the new Department object.

6. Create an External Organization object using the **File > New > External Company** menu command. When the *New Company* dialog displays enter the following information:

<input type="checkbox"/> Name:	U.S. PIPE
<input type="checkbox"/> Description:	Design/produce pipe, fittings, valves
<input type="checkbox"/> Telephone Number:	205.254.7229
<input type="checkbox"/> Fax Number:	205.254.7009

Click **Apply** to create the new Company object but leave the dialog displayed.

Create another External Organization object using the **File > New > External Company** menu command. When the *New Company* dialog displays enter the following information:

<input type="checkbox"/> Name:	WORCESTOR CONTROLS
<input type="checkbox"/> Description:	Industrial ball valves/actuators
<input type="checkbox"/> Telephone Number:	44 1444 314400
<input type="checkbox"/> Fax Number:	44 1444 314401

Click **Apply** to create the new Company object but leave the dialog displayed.

Create another External Organization object using the **File > New > External Company** menu command. When the *New Company* dialog displays enter the following information:

<input type="checkbox"/> Name:	RB GAS AND OIL
<input type="checkbox"/> Description:	Oil and Gas O/O
<input type="checkbox"/> Telephone Number:	877.932.2324
<input type="checkbox"/> Fax Number:	205.945.3999

Click **Apply** to create the new Company object but leave the dialog displayed.

Create another External Organization object using the **File > New > External Company** menu command. When the *New Company* dialog displays enter the following information:

<input type="checkbox"/> Name:	KOLINAR CONSTRUCTION
<input type="checkbox"/> Description:	International EPC Corp
<input type="checkbox"/> Telephone Number:	244.221.1212
<input type="checkbox"/> Fax Number:	244.221.4141

Click **OK** to create the new Company object.

7. Create a Supplier object using the **File > New > Supplier** menu command. When the *New Supplier* dialog displays enter the following information:

<input type="checkbox"/> Supplier:	FISHER ROSEMONT
<input type="checkbox"/> Description:	Process instrumentation supplier
<input type="checkbox"/> Telephone Number:	515.754.3011
<input type="checkbox"/> Fax Number:	515.754.2830

Click **OK** to create the new Supplier object.

8. Create a new user account: From the System Administration menu, select **File > New > User...** A *New user* form will display.

After entering the values for the new user click **OK**.

Document Controller user

<input type="checkbox"/> Login name * -	<i>deb</i>
<input type="checkbox"/> Password -	leave blank
<input type="checkbox"/> Full name-	<i>Deb Document Controller</i>
<input type="checkbox"/> Company Name -	ENGINEERING DESIGN
<input type="checkbox"/> Email Address -	<i>doccontroller@ingr.com</i>
<input type="checkbox"/> Warning dialog count -	leave blank
<input type="checkbox"/> Maximum SQL limit -	leave blank
<input type="checkbox"/> Inbox refresh -	leave blank
<input type="checkbox"/> Expand to -	leave blank
<input type="checkbox"/> Days between password changes -	leave blank
<input type="checkbox"/> Default configuration -	leave blank
<input type="checkbox"/> Business object maintenance -	leave blank
<input type="checkbox"/> Form maintenance -	leave blank
<input type="checkbox"/> Core maintenance -	leave blank
<input type="checkbox"/> User maintenance -	leave blank
<input type="checkbox"/> Left company -	leave blank
<input type="checkbox"/> UOM display set name -	leave blank
<input type="checkbox"/> Signoff expiration -	take the default
<input type="checkbox"/> Show hidden files -	leave blank
<input type="checkbox"/> User type -	Full

9. Update the user accounts to set the *Company name* field in order to participate in a transmittal.

alex

<input type="checkbox"/> Company Name -	ENGINEERING DESIGN
<input type="checkbox"/> Email Address -	<i>alex@ingr.com</i>
<input type="checkbox"/> User type -	Full

eddie

Company Name - **MAINTENANCE ENGINEERING**
 Email Address - *eddie@ingr.com*
 User type - **Full**

mark

Company Name - **ENGINEERING DESIGN**
 Email Address - *mark@ingr.com*
 User type - **Full**

van

Company Name - **FACILITIES USE**
 Email Address - *van@ingr.com*
 User type - **Full**

client

Company Name - **RB GAS AND OIL**
 Email Address - *client@ingr.com*
 User type - **E-mail**

10. Create a new group called the *DOCCONTROL* group. Use the **File > New > User Group...** command and the following information:

User group name * - **DOCCONTROL**
 User group description - *Access to manipulate released documents*
 User group type - **CM**
 Lead user - leave blank
 Vault name - leave blank
 Owning group indicator - leave blank

Click **OK** after entering the values for the new group.

11. Create a new group called the *EMAILONLY* group. Use the **File > New > User Group...** command and the following information:

User group name * - **EMAILONLY**

<input type="checkbox"/> User group description -	<i>Used to provide project filtering and workflow acc</i>
<input type="checkbox"/> User group type -	CM
<input type="checkbox"/> Lead user -	leave blank
<input type="checkbox"/> Vault name -	leave blank
<input type="checkbox"/> Owning group indicator -	leave blank

Click **OK** after entering the values for the new group.

12. You are now going to add the users that you have created to the **DOCCONTROL** and **EMAILONLY** groups to allow them system functionality.

- Right click on the two group names and select **Show Users** to view the relationship.
- Drag and drop your user objects to the default group objects relationship.
 - *deb* to the **DOCCONTROL**, **UPDATE** and **VIEWONLY** groups
 - In the *New relationship* form for user *doccontrol*, enter the following:
 - Plant list name - **EFPLANTS**
 - Project list name - **EFPROJECTS**
 - User group user name - **deb (DOCCONTROL) EFPLANTS**
 - User group user name - **deb (UPDATE) EFPLANTS**
 - User group user name - **deb (VIEWONLY) EFPLANTS**
 - *client* to the **EMAILONLY** group
 - In the *New relationship* form for user *client*, enter the following:
 - Plant list name - **EFPLANTS**
 - Project list name - **EFPROJECTS**
 - User group user name - **client (EMAILONLY) EFPLANTS**

13. Modify the **ExternalTransmittalLifeCycle** workflow using the *SmartPlant Change Management Administration* utility.

- Expand the list of Workflows from the *tree*.
- Update the **ExternalTransmittalLifeCycle**

- Change the *Recipient* (Maintain Recipients) for **step 10** to be **deb**.
- Remove **Recipient submitter** from step 10.
- Change the *Recipient* (Maintain Recipients) for **step 30** to be **eddie**.
- Remove **Recipient submitter** from step 30.
- Change the *Recipient* (Maintain Recipients) for **step 90** to be **deb**.
- Remove **Recipient submitter** from step 90.
- Change the *Recipient* (Maintain Recipients) for **step 120** to be **eddie**.
- Remove **Recipient submitter** from step 120.

Click **OK** to save the workflow changes.

14. Modify the **InternalTransmittalLifeCycle** workflow using the *SmartPlant Change Management Administration* utility.

- Expand the list of Workflows from the *tree*.
- Update the **InternalTransmittalLifeCycle**
- Change the *Recipient* (Maintain Recipients) for **step 10** to be **deb**.
- Remove **Recipient submitter** from step 10.
- Change the *Recipient* (Maintain Recipients) for **step 30** to be **eddie**.
- Remove **Recipient submitter** from step 30.
- Change the *Recipient* (Maintain Recipients) for **step 100** to be **deb**.
- Remove **Recipient submitter** from step 100.
- Change the *Recipient* (Maintain Recipients) for **step 140** to be **eddie**.
- Remove **Recipient submitter** from step 140.

Click **OK** to save the workflow changes.

15. Create a new *Internal Transmittal Matrix*. Use the **File>New>Internal Matrix** command. When the *New Internal Transmittal Matrix* dialog displays, enter the following:

- | | |
|--|---|
| <input type="checkbox"/> Name- | INT_GENERAL MATRIX |
| <input type="checkbox"/> Description- | Internal doc distribution reference matrix |
| <input type="checkbox"/> Reason for issue- | Reference |

Click **OK** to create the Internal Matrix.

16. Add matrix assignments to your new Internal Matrix. From the *New Items* window, right-click and use the **Matrix Assignments...** command.

- Select the **User** button at the bottom of the dialog to query for workflow users.
- Set the *Document category* value to **P&ID Documents**.
- Set the *Contract* value to **CS3377825** (the one that you added earlier).
- Set the *Default format* to **1FS** at the bottom of the dialog.
- Set the *Default reason for receipt* to **C** at the bottom of the dialog.
- Drag and drop **alex**, **van** and **eddie** onto the first row in the matrix.

Click **OK** to save the Matrix Assignments.

17. Create a new *External Transmittal Matrix*. Use the **File > New > External Matrix** command. When the *New External Transmittal Matrix* dialog displays, enter the following:

- | | |
|--|---|
| <input type="checkbox"/> Name- | EXT_OO MATRIX |
| <input type="checkbox"/> Description- | Owner/Operator external company matrix |
| <input type="checkbox"/> Reason for issue- | Design Review |

Click **OK** to create the External Matrix.

18. Add matrix assignments to your new External Matrix. From the *New Items* window, right-click and use the **Matrix Assignments...** command.

- Select the **User** button at the bottom of the dialog to query for workflow users.
- Set the *Document category* value to **P&ID Documents**.
- Set the *External organization* value to **RB GAS AND OIL**.
- Set the *Default format* to **1FS** at the bottom of the dialog.
- Set the *Default reason for receipt* to **I** at the bottom of the dialog.
- Drag and drop **client** and **deb** onto the first row in the matrix.
- Right click on doccontroller from the matrix and change the *Default reason for receipt* to **W**.

Click **OK** to save the Matrix Assignments.

19. You may take a short break until class is ready to resume with part 2 of this chapter.

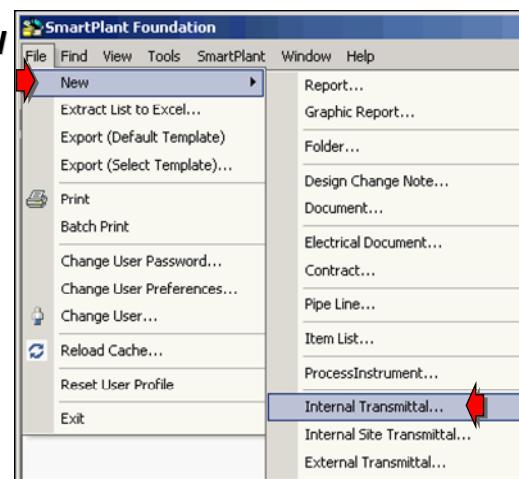
7.8 Creating an Internal Transmittal

Now that some administrative steps are complete, transmittals can now be created.

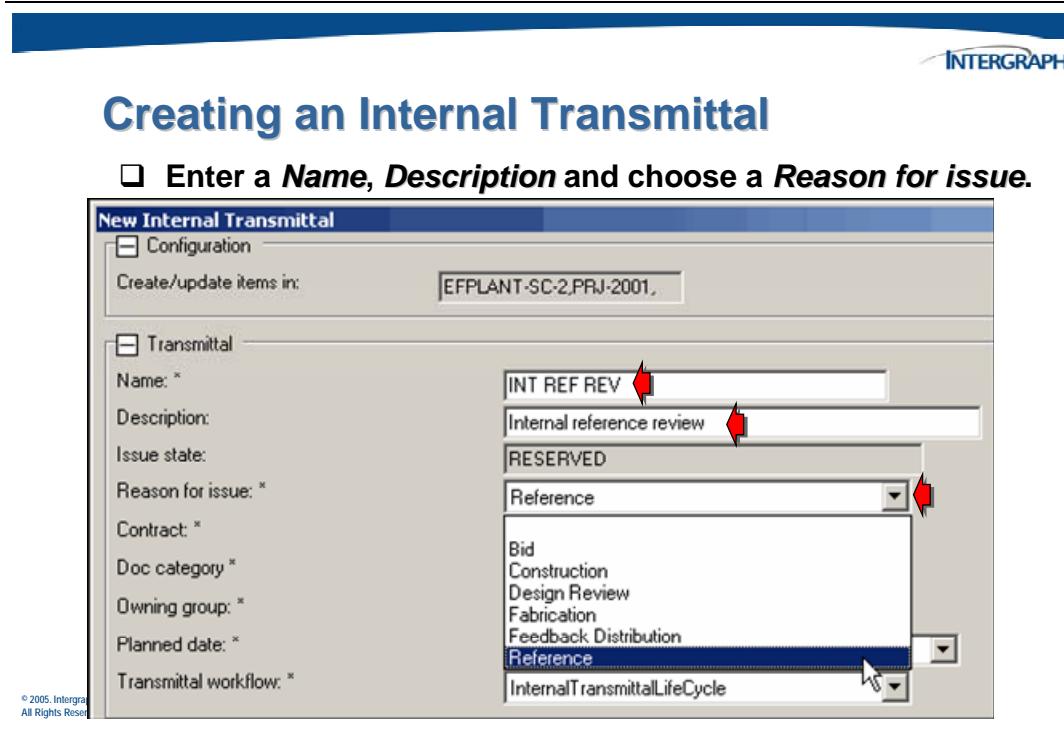
An *Internal* transmittal is used to distribute documents within internal organizations. To create an internal transmittal, begin by creating the transmittal object.

Creating an Internal Transmittal

- Select **File > New > Internal Transmittal...** from the menu.



The *New Internal Transmittal* dialog displays.



The following information is entered or populated on this form, with an * denoting required fields:

- **Name** – the name for the internal transmittal. ENS can be configured and used for this field.
- **Description** – description of the transmittal
- **Issue State** – a read-only field which defaults to RESERVED
- **Reason For Issue** – a configured pick list



Creating an Internal Transmittal

- Choose a **Contract** number.

New Internal Transmittal

Configuration	
Create/update items in: EFPLANT-SC-2,PRJ-2001,	
Transmittal	
Name: *	INT REF REV
Description:	Internal reference review
Issue state:	RESERVED
Reason for issue: *	Reference
Contract: *	<input type="button" value="▼"/>
Doc category *	CS3377825
Owning group:	UPDATE
Planned date: *	10/ 8/2006 4:09:52 PM
Transmittal workflow: *	InternalTransmittalLifeCycle

© 2005, Intergraph Corp.
All Rights Reserved.

- **Contract** – a field that maps to the distribution matrix



Creating an Internal Transmittal

- Choose a **Doc category** value.

New Internal Transmittal

Configuration	
Create/update items in: EFPLANT-SC-2,PRJ-2001,	
Transmittal	
Name: *	INT REF REV
Description:	Internal reference review
Issue state:	RESERVED
Reason for issue: *	Reference
Contract: *	CS3377825
Doc category *	<input type="button" value="▼"/>
Owning group:	3D Documents
Planned date: *	Dimensional Data Documents
Transmittal workflow: *	Electrical Documents
Distribution	
From organization:	P&ID Documents

© 2005, Intergraph Corp.
All Rights Reserved.

- **Document category** – a field that maps to the distribution matrix



Creating an Internal Transmittal

- ❑ Choose a *Planned date* from the displayed calendar.

New Internal Transmittal

Configuration
Create/update items in: EFPLANT-SC-2.PRD-2001,

Transmittal
Name: * INT REF REV
Description: Internal reference review
Issue state: RESERVED
Reason for issue: * Reference
Contract: * CS3377825
Doc category: * P&ID Documents
Owning group: * UPDATE
Planned date: * 1/12/2007 4:09:52 PM
Transmittal workflow: *

Distribution
From organization:
From user:
To organization:

© 2005, Intergraph Corp.
All Rights Reserved.

January, 2007

Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Today: 10/8/2006

- ❑ Owning group – used for ownership and vaulting
- ❑ Planned date – planned date of completion which can also be set by the right mouse menu



Creating an Internal Transmittal

- Choose a value for the *From organization*.

The screenshot shows the 'Transmittal' creation interface. In the 'Distribution' section, the 'From organization' dropdown menu is open, displaying a list of internal organizations: 'ENGINEERING DESIGN', 'FACILITIES USE', and 'MAINTENANCE ENGINEERING'. A red arrow points to the dropdown button. The 'From user' field below it is empty.

© 2005, Intergraph Corp.
All Rights Reserved.

- From organization** – a pick list of internal organizations



Creating an Internal Transmittal

- Choose a value for the *From user*.

The screenshot shows the 'Transmittal' creation interface. In the 'Distribution' section, the 'From user' dropdown menu is open, displaying a list of users from the selected organization: 'alex', 'dawn', and 'mark'. A red arrow points to the dropdown button. The 'From organization' field above it contains 'ENGINEERING DESIGN'.

© 2005, Intergraph Corp.
All Rights Reserved.

- From user** – a pick list containing users from the selected organization

Creating an Internal Transmittal

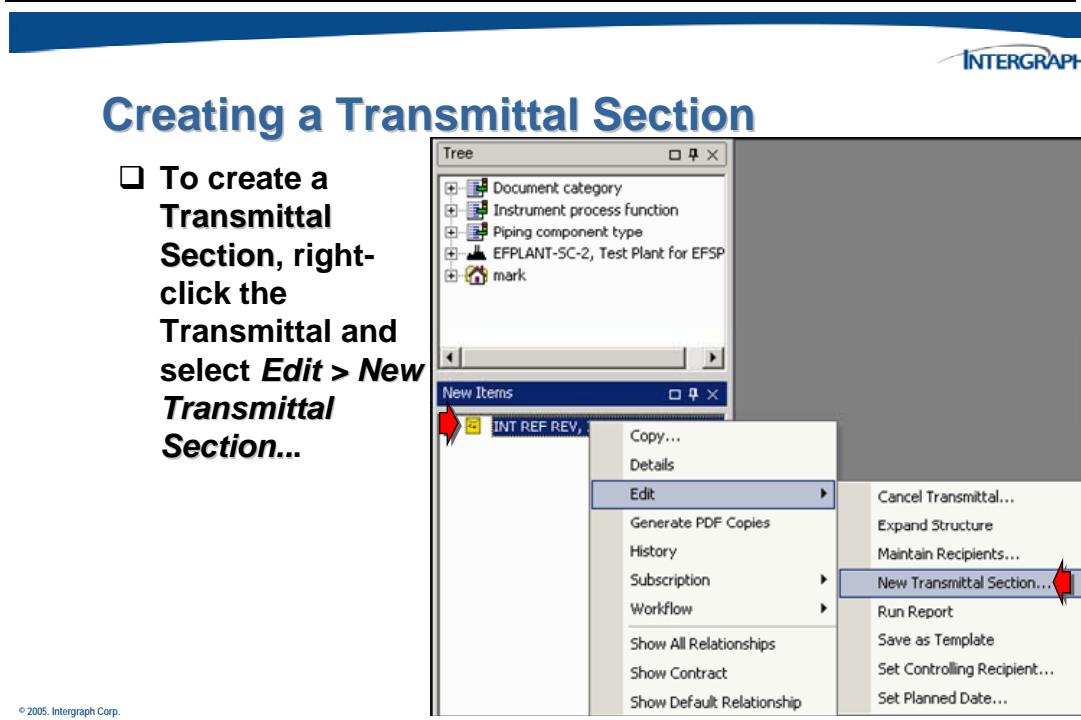
Choose a **To user** value from the list and click **OK** to create the new *Internal Transmittal*..

The screenshot shows the 'New Internal Transmittal' dialog box. In the 'Distribution' section, the 'To user' dropdown is open, showing 'mark' and 'eddie'. A red arrow points to the 'eddie' entry. At the bottom right of the dialog box, there is an 'OK' button with a red arrow pointing to it.

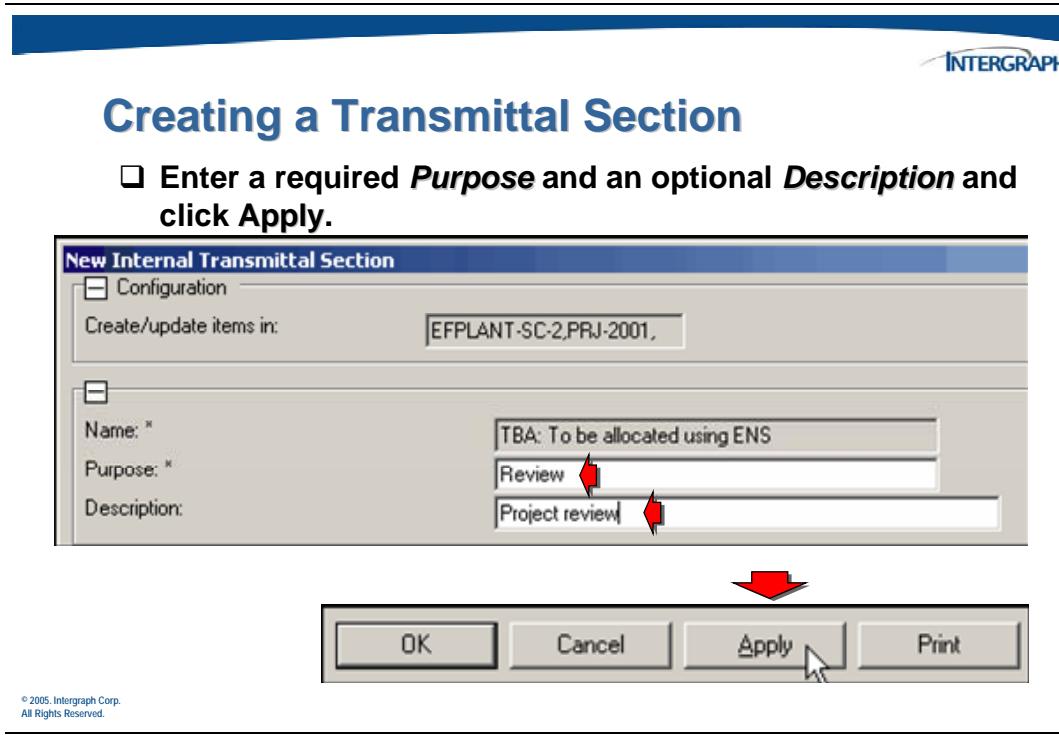
- To organization** – a pick list of internal organizations
- To user** – a pick list containing users from the selected organization
- Populate automatically from matrix** – an option to automatically populate the transmittal with recipients from matching matrices. When this option is selected, all recipients from the matrix are automatically added.

7.8.1 Creating a Transmittal Section

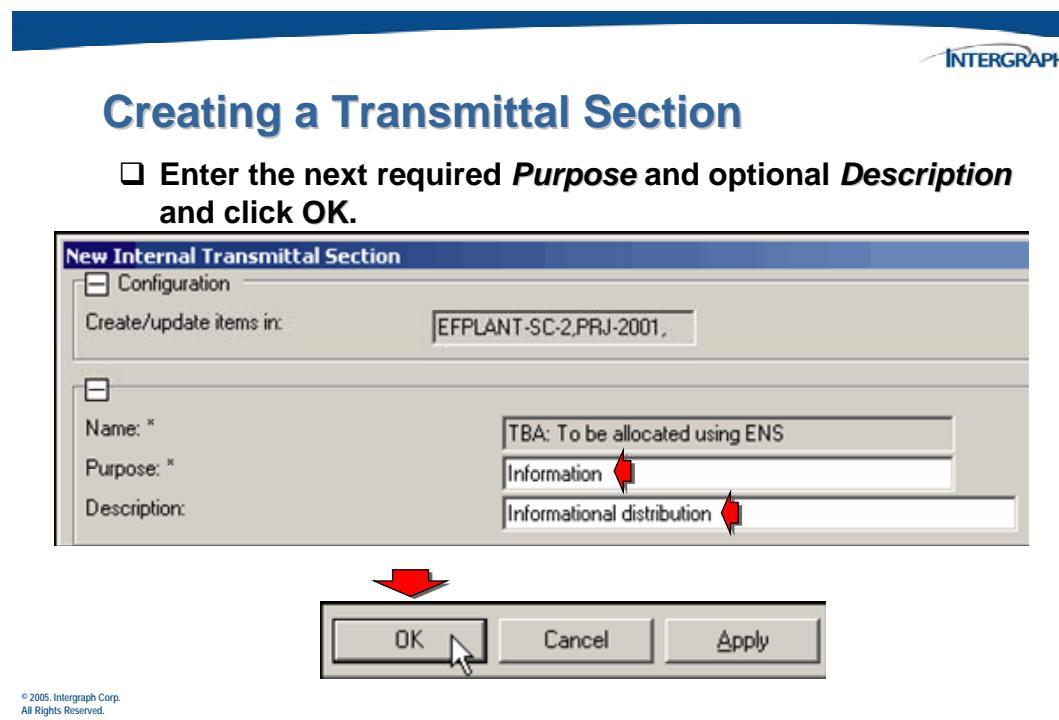
As part of the creation of an Internal Transmittal, the *New Transmittal Section* command is available on the Transmittal's right mouse *Edit* menu. In the delivered model of SPF, Internal Transmittals are configured with sections whereas the External Transmittals are configured with documents attached directly to the transmittal object.



The *New Internal Transmittal Section* dialog is displayed.



Repeat this for all needed sections for this specific transmittal.

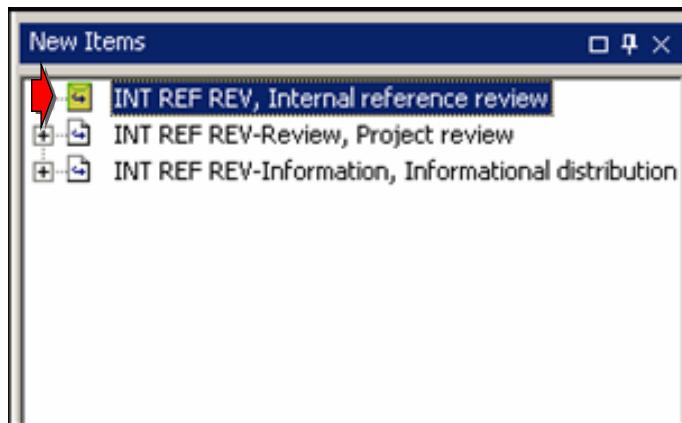


The new transmittal section is named after the parent internal transmittal (the transmittal that was selected) along with the name using this format **Transmittal Name – Section Purpose**.



Creating a Transmittal Section

- Multiple sections can be created for the selected Transmittal.**

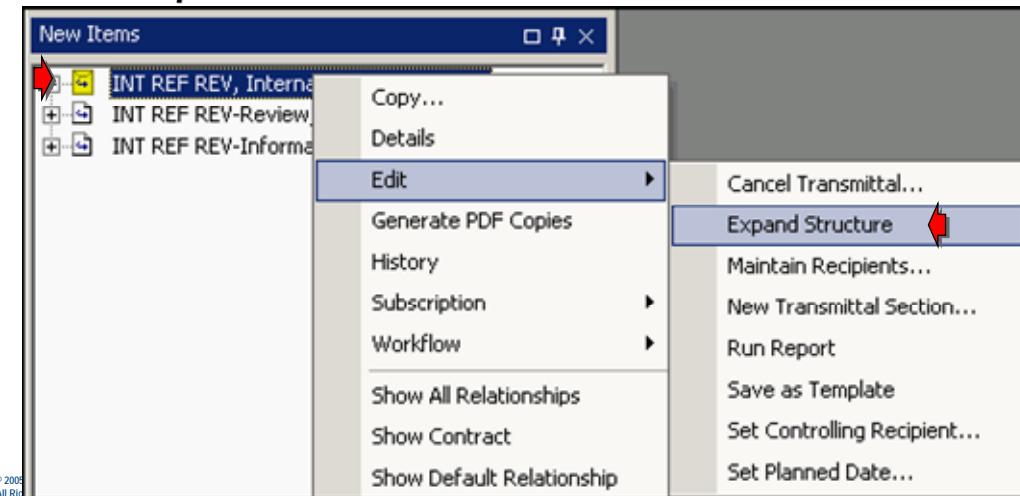


7.8.2 Expanding the Transmittal Structure

Now that an internal transmittal has been created and one or more sections have been added, the transmittal structure can be viewed.

Expanding the Transmittal Structure

- Right-click on the Internal Transmittal and select **Edit > Expand Structure...**



A *Transmittal Structure* window will be created.



Expanding the Transmittal Structure

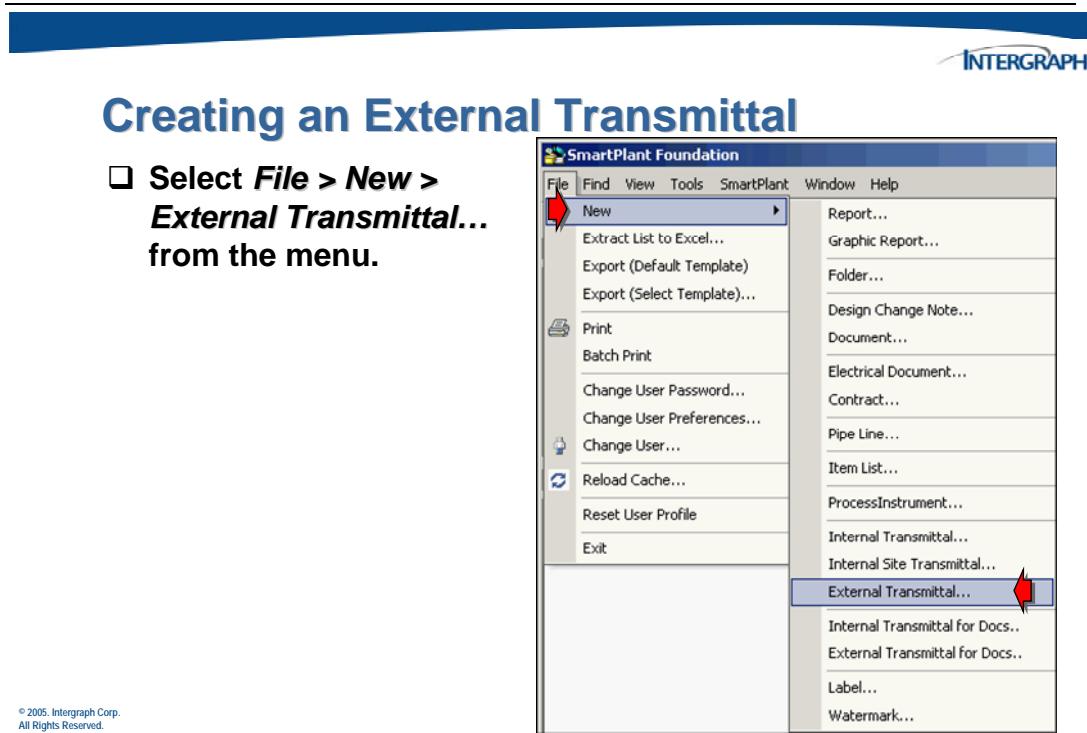
The Transmittal Structure is expanded to show its Sections.

A screenshot of the "Transmittal Structure" window. The title bar has a red arrow pointing to the title "Transmittal Structure". The main area is a table with columns: Name, Rev, Default St..., Classification, and LastUpdated Date. There are two rows. The first row is for "INT REF REV" with Rev "WF-S", Classification "SPFInternalTransmittal", and Date "10/8/2006". The second row is for "Show Sections" which contains two items: "INT REF REV-In..." and "INT REF REV-Re...". Each item has a red arrow pointing to it, indicating they are expandable sections. The "INT REF REV-In..." section expands to show "Show Document..." and "INT REF REV-Re..." expands to show "Show Document...".

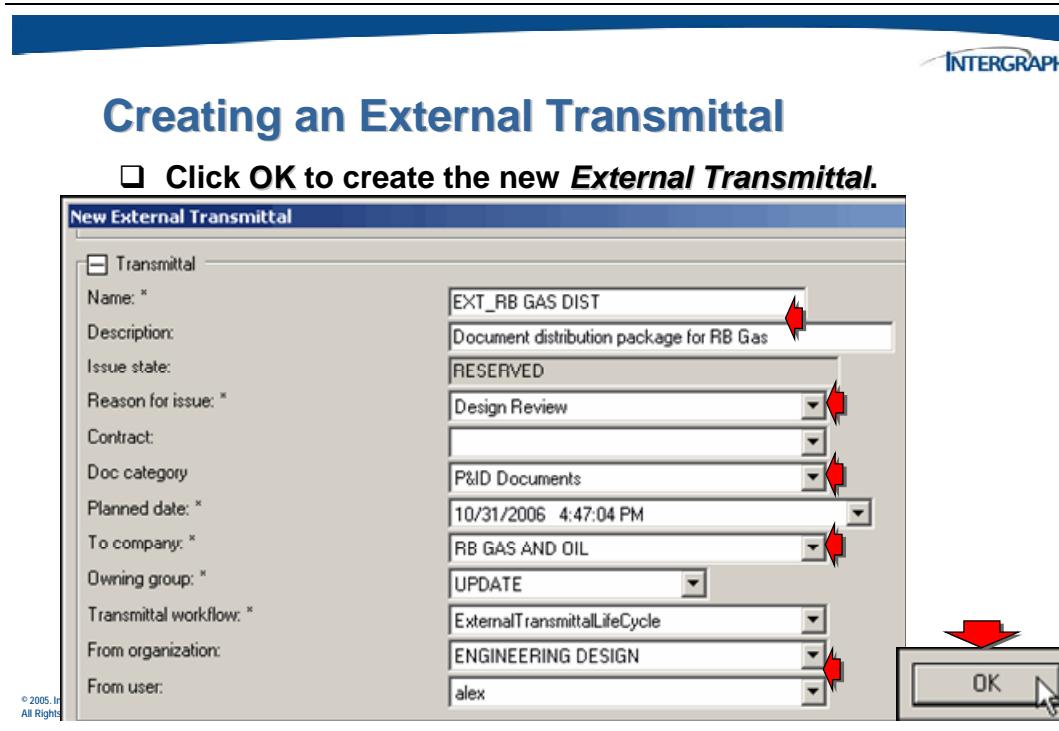
Name	Rev	Default St...	Classification	LastUpdated Date
INT REF REV	WF-S		SPFInternalTransmittal	10/8/2006
Show Sections				
INT REF REV-In...	RESERVED -		SPFInternalTransmittal	10/8/2006
Show Document...				
INT REF REV-Re...	RESERVED -		SPFInternalTransmittal	10/8/2006
Show Document...				

7.9 Creating an External Transmittal

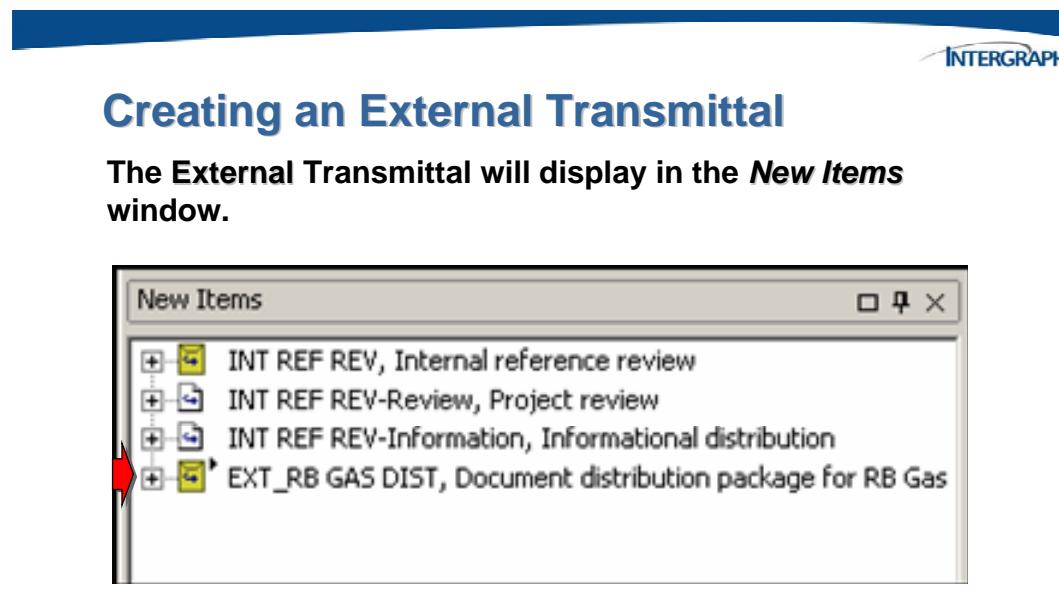
The steps to create an *External Transmittal* are identical to those used in creating an Internal Transmittal, except External Transmittals do not contain transmittal sections. The main transmittal object is created, and documents are attached directly to it.



The *New External Transmittal* dialog displays.



The new External Transmittal object will be created.



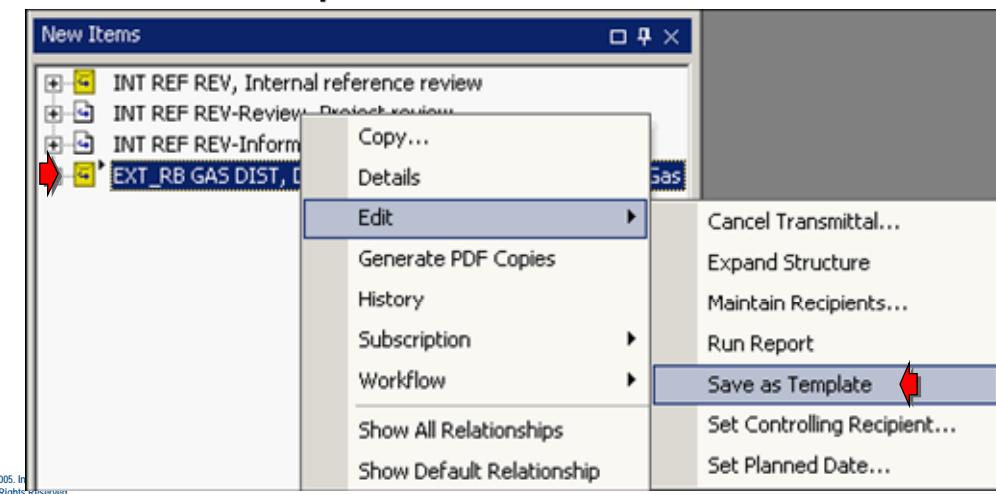
7.10 Saving a Transmittal Template

Any Transmittal can be stored and used as a template. The workflow is terminated and the state of the Transmittal is changed to TEMPLATE effectively freezing it from modification. All section information, relationships to documents and workflow set up is frozen.



Saving a Transmittal Template

- Right-click on an External Transmittal and select **Edit > Save as Template.**



© 2005. In
All Rights Reserved.



Saving a Transmittal Template

- Click Yes to make the selected Transmittal a template.



7.11 Create a Transmittal for a Selected Set of Documents

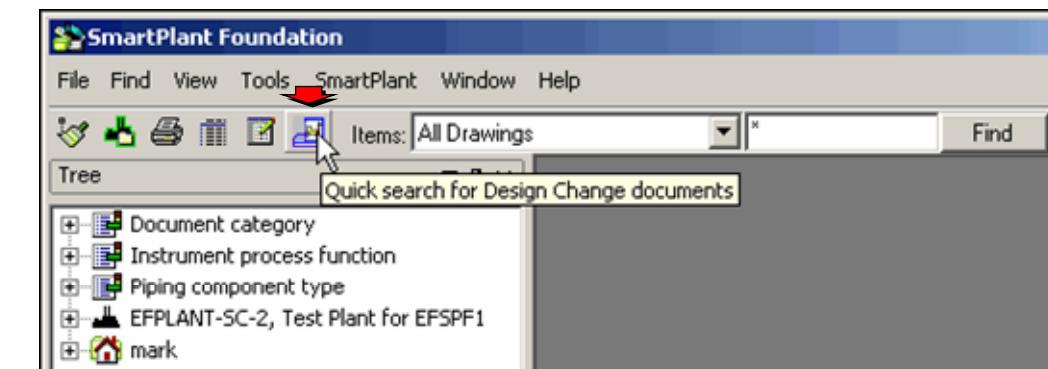
It is possible to select a set of documents and generate a new Transmittal for them. When the documents are selected, choose either the ***Internal Transmittal for Docs*** command or the ***External Transmittal for Docs*** command from the menu.

Begin by searching for a set of documents to be added to the transmittal.

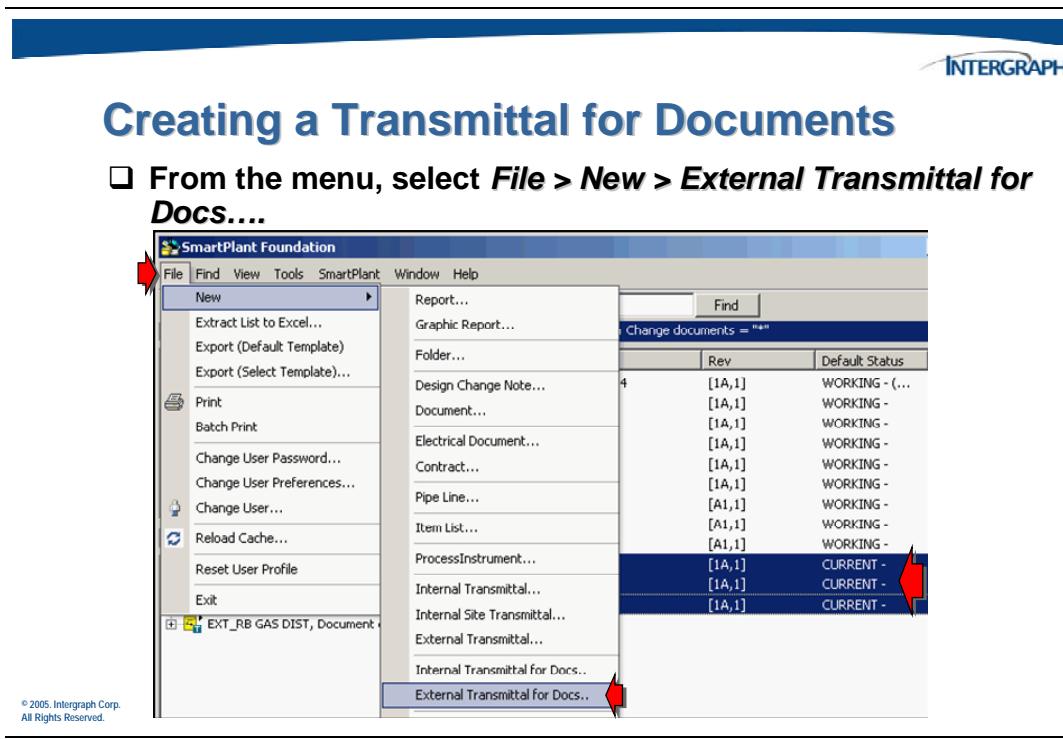


Creating a Transmittal for Documents

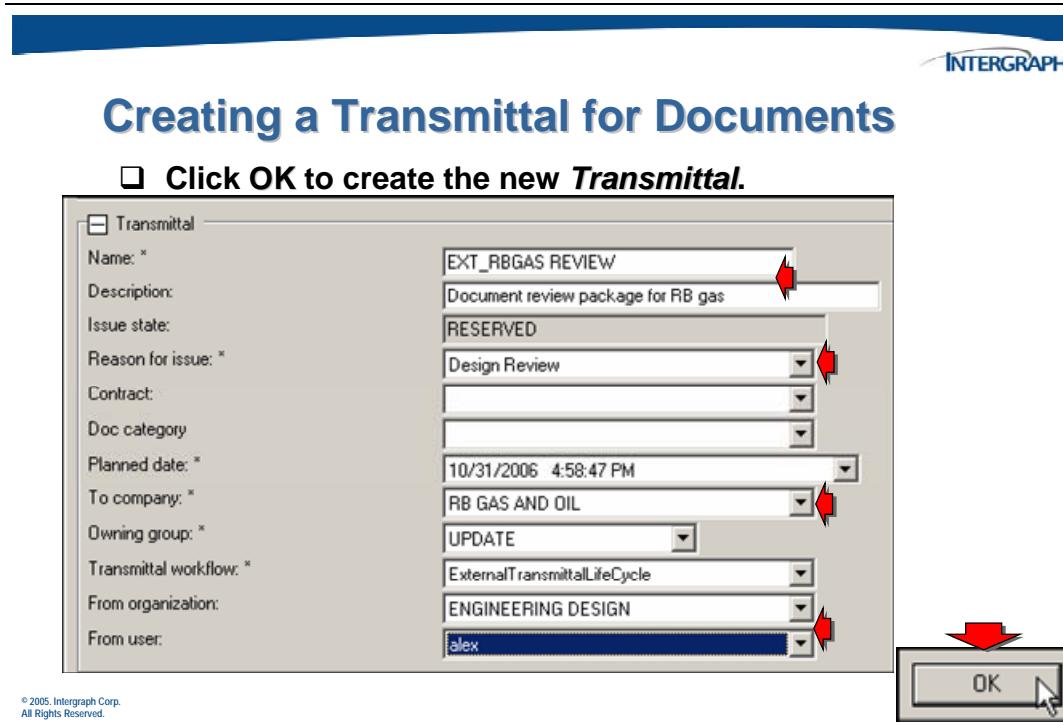
- Select the ***Find Design Change*** tool from the toolbar to locate a set of documents.



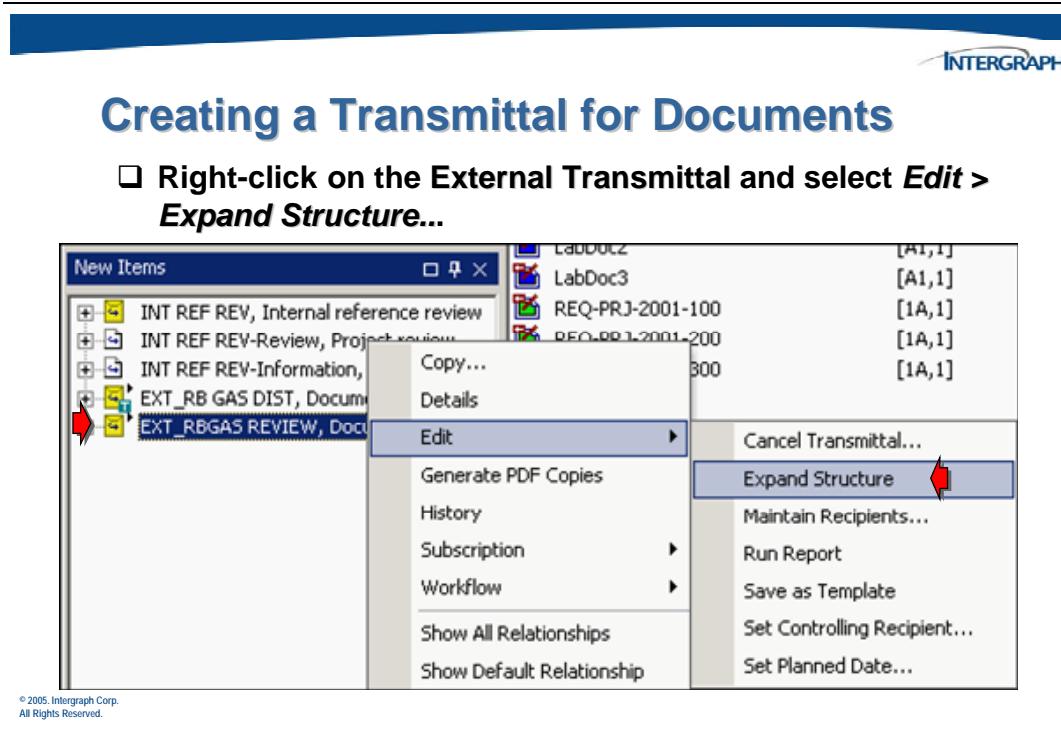
Next, highlight a set of documents from the *Find results* window.



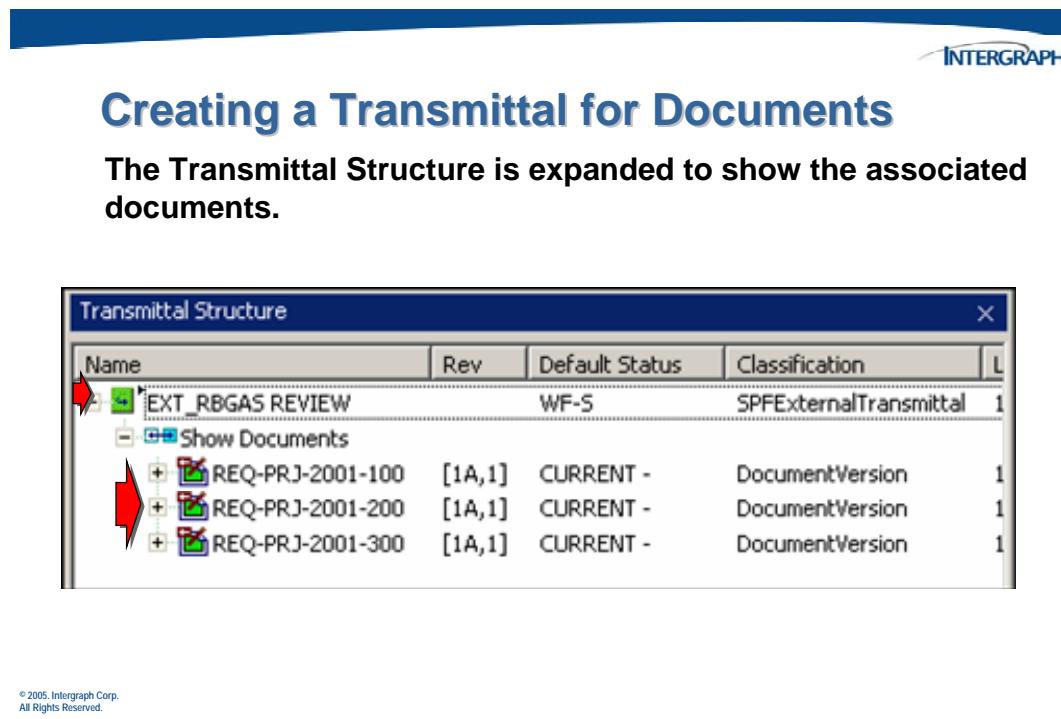
A New External Transmittal dialog will display. Enter the information necessary to create the new transmittal object.



Expand the transmittal structure to verify that the documents were attached to the new transmittal.



A *Transmittal Structure* window will be created.

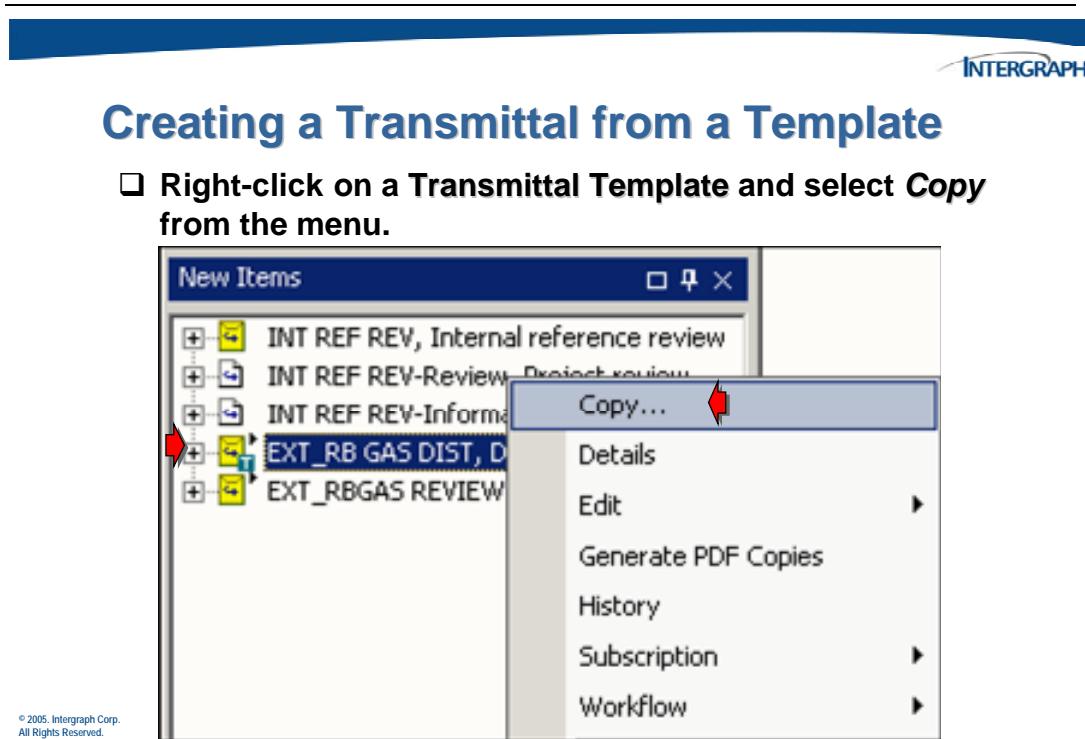


7.12 Creating a Transmittal from a Template

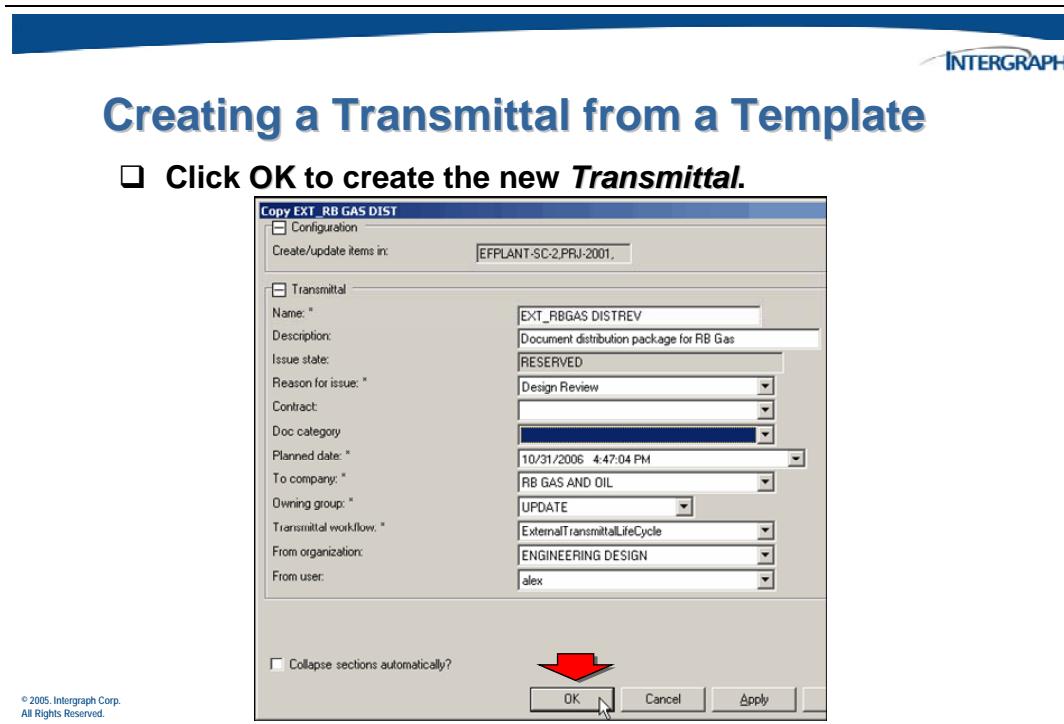
A Transmittal can be stored as a template and used to create new transmittals. There are two ways to create a transmittal from a template:

- Use the **Find > Transmittals > Transmittal Templates** to retrieve the templates. Right-click a template to display the right mouse menu to use the Copy command.
- Use dedicated menu options. In the delivered SPF model there is an example to create a Site Transmittal at the File > New > Internal Site Transmittal.

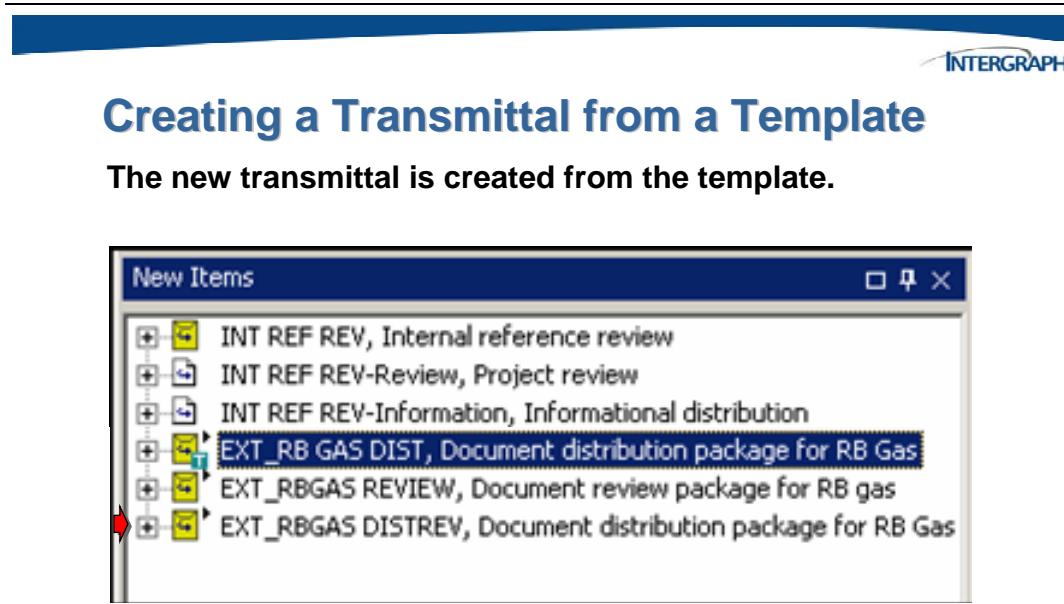
Highlight a *Transmittal Template* to be used for creating a new transmittal.



A *Copy* transmittal dialog will be displayed.



Enter a new name for the transmittal and change any of the other values.



7.13 Activity 2 – Creating Transmittals

In this activity you will create an Internal and an External Transmittal along with Transmittal Sections.

If you are not currently logged into your machine:

1. Log on to your operating system (if not already logged in):
spfuser with no password

2. Start the *Desktop Client interface* by selecting *Start > All Programs > Intergraph SmartPlant Foundation > SmartPlant Foundation Desktop Client*.

3. When the *Login* dialog window appears, use the *User name* **alex** with no password and click on **OK**.

4. Create an *Internal Transmittal* object. Use the *File > New > Internal Transmittal* command. When the *New Internal Transmittal* dialog displays, enter the following:

<input type="checkbox"/> Name-	INTDESIGNREVIEW
<input type="checkbox"/> Description-	Internal design review package
<input type="checkbox"/> Reason for issue-	Design Review
<input type="checkbox"/> Contract-	CS3377825
<input type="checkbox"/> Doc category-	P&ID Documents
<input type="checkbox"/> Owning group-	UPDATE
<input type="checkbox"/> Planned date-	<some date in the future>
<input type="checkbox"/> Transmittal work flow-	InternalTransmittalLifeCycle
<input type="checkbox"/> From organization-	ENGINEERING DESIGN
<input type="checkbox"/> From user-	alex
<input type="checkbox"/> To organization-	MAINTENANCE ENGINEERING
<input type="checkbox"/> To user-	eddie

Click **OK** to create the new Internal Transmittal.

5. Add a couple of Transmittal Sections to your new Internal Transmittal. From the *New Items* window, right-click and use the **Edit > New Transmittal Section...** command.

When the *New Internal Transmittal Section* dialog displays enter the following:

<input type="checkbox"/> Purpose-	Design
<input type="checkbox"/> Description-	Maintenance Design Section

Click **OK** to create the new Internal Transmittal Section.

Again, right-click and use the **Edit > New Transmittal Section...** command.

When the *New Internal Transmittal Section* dialog displays enter the following:

<input type="checkbox"/> Purpose-	Review
<input type="checkbox"/> Description-	Informational Review Only

Click **OK** to create the new Internal Transmittal Section.

6. Expand the transmittal structure of the new Internal Transmittal to verify the association to the new sections.
7. Create an *External Transmittal* object. Use the **File > New > External Transmittal** command. When the *New External Transmittal* dialog displays, enter the following:

<input type="checkbox"/> Name-	RBGASDISTREV
<input type="checkbox"/> Description-	Document distribution package for RB Gas
<input type="checkbox"/> Reason for issue-	Design Review
<input type="checkbox"/> Doc category-	P&ID Documents
<input type="checkbox"/> Planned date-	<some date in the future>
<input type="checkbox"/> To company-	RB GAS AND OIL
<input type="checkbox"/> Owning group-	UPDATE
<input type="checkbox"/> Transmittal work flow-	ExternalTransmittalLifeCycle
<input type="checkbox"/> From organization-	ENGINEERING DESIGN
<input type="checkbox"/> From user-	alex
<input type="checkbox"/> Populate automatically from matrix-	enabled

Click **OK** to create the new External Transmittal.

8. You may take a short break until class is ready to resume with the next chapter.

DSPF1-TP-1000010A

SmartPlant Foundation Configuration and Administration II

Course Guide Volume 1