StarTeam Extensions User’s Guide

StarTeam®

Borland®
Excellence Endures™

Borland Software Corporation
100 Enterprise Way
Scotts Valley, California 95066-3249
www.borland.com

Borland Software Corporation may have patents and/or pending patent applications covering subject matter in this document. Please refer to the product CD or the About dialog box for the list of applicable patents. The furnishing of this document does not give you any license to these patents.

For third-party conditions and disclaimers, see the Release Notes on your product CD.

Printed in the U.S.A.

Contents

Chapter 6
Using StarTeam Notification Agent 93

Appendix A
Installing StarTeam Workflow Applications 109

Appendix B
Notification Tags and Messages 117

Chapter

1

Preface

This manual contains information for StarTeam administrators and project leaders. It provides instructions on setting up workflows and Alternate Property Editors (APEs) for use with the application Server and clients.

Important The online manuals are distributed in Adobe Acrobat (.pdf) format and require the Adobe Acrobat reader, 4.0 or higher, in order to display them. The installation program for the Adobe Acrobat Reader is located in the \Docs folder of the Installation CD. The reader is also available from the Adobe web site at: www.adobe.com.

Contacting Borland Support

Borland Software Corporation is committed to providing world-class services in the area of consulting and technical support. We have over 15 years of experience in supporting developers and enterprise customers. Our qualified technical support engineers are prepared to handle your support needs on a case-by-case basis or in an ongoing partnership. Borland provides support worldwide, delivering timely, reliable service to ensure every customer's business success.

For more information about Borland’s support services, please see our web site at http://support.borland.com.

From the Web site, you can also access many newsgroups where users exchange information, tips and techniques. See http://info.borland.com/newsgroups/ for the latest list of free product newsgroups. Also, available on the Internet is the Borland Developer Network site at http://community.borland.com. This Borland Community provides access to product specific information, articles, code examples, and news.

When contacting support, be prepared to provide complete information about your environment, the version of the product you are using, and a detailed description of the problem.

For support on third-party tools or documentation, contact the vendor of the tool.

Documentation Conventions

StarTeam documentation uses the following conventions.

Select File > Exit	Indicates a menu selection followed by a submenu selection. The right angle bracket (**>**) separates the commands to be selected from subsequent menus. In this case, select File from the menu bar, then select Exit from the drop-down menu.
Fixed-Space Font	Text appearing in Courier font represents information that you need to enter and messages from the system.
italics	Syntax appearing in italics represents information that you replace with the names of your files, child folders, etc. Italics are also used for the names of dialogs and books and for emphasis.

Note Identifies supplemental information.

Tip Identifies information on alternative procedures or other helpful but nonessential information.

Important Identifies information that is essential to the completion of a task.

Caution Identifies actions that may result in loss of data or procedures that must be followed to ensure that data is *not* lost.

Chapter

2

Understanding StarTeam Extensions Features

Workflow is a comprehensive term to explain a process by which members of an organization complete complex tasks. Whether your organization uses formalized or ad hoc workflows, there is a process or pattern by which work is accomplished. One employee begins a task and passes it along to someone else for review. That person gives the project to yet a third person to finalize and then it goes back to person number two for final review, etc. Workflow software can automate workflow by providing automatic notification of responsibility, by presenting only appropriate options at each step in the process, and by assuring that items are acted upon in a timely manner.

The most difficult part of designing a workflow process is deciding what you want to control and how you want to control it. Once you have determined your needs, the application gives you all the tools you need to create customized workflow models that meet the unique requirements of your organization.

In this guide, the term workflow is used to denote a .Workflow.xml file that indicates step-by-step how items of a particular type will be processed.

You can create a workflow to control any item type within the application—files, change requests, requirements, tasks, or topics. Because teams commonly control processes using change request statuses, the change request management system offers a simple, built-in workflow pattern based on the Status property. It requires a change request to proceed sequentially through a specific number of ordered steps and notifies team members of their responsibilities at each stage of the change request life cycle. This workflow pattern meets the needs of many organizations. (For complete information on the built-in workflow capabilities of the application, see the *StarTeam User's Guide*.)

For some organizations, however, the built-in workflow model is too much. Their workflow models do not require that many steps or that many choices. For other organizations, the built-in workflow model is not enough. They have requirements for complex workflow patterns that may include a time-dependent response from the team member who is responsible for a specific step or may require redirection of the workflow based on specific conditions.

StarTeam's workflow features are sometimes collectively referred to as StarTeam Extensions, a collection of .jar and other files that are updated more frequently than the regularly scheduled application releases. However, customizing the application to meet your workflow needs requires other features, some of which are installed as part of the application Server and clients, and others that come with the product but must be installed separately at locations that meet your needs.

The products, tools, and features used to set up and maintain workflow processes are:

- StarTeam Extensions, which includes:
  - Alternate Property Editors (APEs) and the Java beans that make them easy to create. APEs, which are forms written in a standard programming language such as Java, replace the standard properties dialogs that come with each component of the application.
  - A workflow engine that reads the appropriate .Workflow.xml file and populates the APEs.
  - A toolbar that can be used with the Cross-Platform client to start applications that add new features to the application.
  - Various other utilities and libraries written in Java.

  StarTeam Extensions files are released with the application Server as part of each new release or service pack. There are also interim releases of StarTeam Extensions. In either case, you separately install the runtime files, documentation, and samples.

  The default installation folder is C:\Program Files\Borland\ StarFlow Extensions, but Borland recommends copying these files to a different folder for each server configuration that runs on the same Server. For example, you might use C:\ *config_name*\StarFlow Extensions. In this way, you can ensure that each configuration's StarFlow Extensions project has different working folders.

  Even if you currently run only one server configuration, it is wise to plan ahead. This also ensures that, when you upgrade or reinstall for some other reason, you are unlikely to overwrite any StarTeam Extensions files that you are using.

  See "Using StarTeam Extensions" on page 7.

- StarTeam Workflow Designer

  Workflow Designer outputs .xml files that formalize the steps in a workflow, specifies who will be notified about each step and about exceptions, and so on. Each file is named *item_type*.Workflow.xml and can be used for an entire project or individual views within that project. The StarTeam Extensions workflow engine and Notification Agent read from these files.

  You must use Workflow Designer to develop custom workflows. The resulting .Workflow.xml file can be deployed in StarTeam 5.1 or 5.2 installations—as long as the most recent version of StarTeam Extensions is also installed. Workflows written using the releases of the Workflow Designer prior to 5.2 (which were tightly coupled to Microsoft Outlook) must be re-saved using the new Workflow Designer only if you plan to use Notification Agent or if you want to make changes to the workflow.

  Workflow Designer can be installed separately on any workstation that has a Windows client.

  With the help of Borland consultants, it can be installed on the customer toolbar that can be used with the Cross-Platform client.

  See "Using StarTeam Workflow Designer" on page 29.

- StarTeam Notification Agent

  Notification Agent monitors server configurations to determine who needs to be notified about work to be performed and about exceptions that occur in the workflow process.

It must be installed and set up separately. It is usually installed on the computer that runs the Server, but it can be installed anywhere. When Notification Agent is not on the same computer as the Server, the two computers' clocks must be synchronized. Notification Agent's time cannot be ahead of the Server's time.

See "Using StarTeam Notification Agent" on page 93

- StarTeam Repository Customization

  The application allows you to create custom fields to meet the needs of a workflow process. One enumerated property field controls the workflow process for the item type. The selected field usually requires some customization—even when it is an existing application property field.

  You can customize the property fields in a server configuration for an item type by using the application and, to some degree, by using Workflow Designer. See the customization chapter in the *StarTeam Administrator's Guide* to learn more about creating custom fields from StarTeam. For information about customizing enumerated property fields using Workflow Designer, see "Adding and Modifying Custom Properties" on page 61.

- StarTeam SDK

  The APEs and the applications that run from the StarTeam Extensions toolbar (in the Cross-Platform client) are StarTeam SDK applications.

  See the online documentation for APEs (that can be installed with StarTeam Extensions) and the online documentation that comes with the SDK for more details.

Workflow Roadmap

In general, the steps required to implement a custom workflow in StarTeam are:

1. Install the application Server, StarTeam Extensions, the Windows or Cross-Platform client, Workflow Designer, and Notification Agent in appropriate places. See "Installing StarTeam Workflow Applications" on page 109.
2. Create the necessary projects and folders. See "Creating and Configuring the StarFlow Extensions Project" on page 7 and the *StarTeam Administrator's Guide.*
3. Plan your new workflow.

   As you begin to plan the new workflow, you'll find pencil and paper to be one of your most useful tools. Once you have sketched the basic workflow model, flowcharting software such as Microsoft Visio may help you develop your workflow model. The flow diagram for the example workflow model explained in this chapter was designed using flowcharting software. See "Planning a Workflow" on page 30 for more information.
4. Create a custom enumerated property and values on which the workflow is based. (See "Adding and Modifying Custom Properties" on page 61 and the *StarTeam Administrator's Guide* for more information.)
5. Create any other custom properties to be used.
6. Plan your custom form. You need to decide:
   - What properties to use
   - Where to place them on the form
   - Which properties will be needed, required, hidden, and disabled for each step
   - Which properties will change dynamically as the user edits the change request at each step
7. Create a custom workflow using Workflow Designer to specify the steps and actions of the workflow. See "Using StarTeam Workflow Designer" on page 29 for more

information. In addition to the graphical representation, each step has a properties dialog for options that can be set at each step.

8 Create a custom form that will display all the properties you have decided that you need. See “Creating and Deploying APEs” on page 12 for more information.

9 Test the workflow to ensure that it performs as intended.

- Store the workflow in the Test folder of the StarFlow Extensions project so that it can be used in an application project named Test. (The project name does not have to be Test; this is just an example.)

  See “Adding Folders to the StarFlow Extensions Project” on page 11, “Understanding Locator” on page 10, and “Saving a Workflow” on page 63.

- Use the Test button in JBuilder to test the form with the workflow. This button is on the sample forms and can be added to others. The Test button is set up in the sample so that it is visible only in JBuilder.
- Activate the workflow in the Test project by setting the appropriate APE settings in that project. See “Configuring StarTeam Projects to Use APEs” on page 18.
- Setup and start Notification Agent to monitor the Test project for the correct item type. Notification Agent notifies appropriate users about pending work, item statuses, and exceptions that occur in the process. See “Getting Ready to Use Notification Agent” on page 94 and “Starting Notification Agent and Notification Setup” on page 97.
- Use Notification Agent’s Run Now button to simulate movement from step to step.
- Correct the workflow if necessary.

10 Deploy the workflow

- Store the workflow in the appropriate subfolders of the StarFlow Extensions project. This makes the workflow available for activation.

  See “Adding Folders to the StarFlow Extensions Project” on page 11, “Understanding Locator” on page 10, and “Saving a Workflow” on page 63.

- Activate the workflow for the appropriate projects by setting the appropriate APE settings for those projects. See “Configuring StarTeam Projects to Use APEs” on page 18.
- Setup and start Notification Agent to monitor the project, views, and item types under workflow control. Notification Agent notifies appropriate users about pending work, item statuses, and exceptions that occur in the process. See “Getting Ready to Use Notification Agent” on page 94 and “Starting Notification Agent and Notification Setup” on page 97.
- When users open items under workflow control, StarTeam Extensions displays the APE form.
- The APE calls the workflow engine that examines the item and, based on the information contained in the workflow configuration, displays the relevant workflow information to the user.
- Users perform various workflow-related operations on the items and events are stored in the audit log. The user completes an assigned step by selecting an appropriate next step.

Chapter

3

Using StarTeam Extensions

StarTeam Extensions consist of:

- Sample Alternate Property Editors (APEs) and the JavaBeans that make them easy to create.

  APEs, which are forms written in standard programming languages such as Java, replace the standard properties dialogs that come with each component of the application.
- A workflow engine that applies user-defined workflow rules to the processing of items. The engine can process any item type.
- A custom toolbar that can be used with the Cross-Platform client to start applications that add new features to the application.

  A consultant using the StarTeam Custom Toolbar Framework SDK should set up this feature for your organization.
- Various other utilities and libraries written in Java.

About This Chapter

This chapter explains:

- How to create and configure an Extensions project for each server configuration.
- How to create and deploy APEs.
- How to use various utilities included with Extensions.

Creating and Configuring the StarFlow Extensions Project

The StarFlow Extensions project is used to store the customized workflows and APEs, as well as the workflow engine and other Java utilities and libraries.

Locator, the program used to distribute code and XML updates to client workstations, will look for a project with the name StarFlow Extensions and use the latest revision it finds for the required workflow elements for the APE. Locator provides users with the latest revision of these files whenever the APE is requested. The following procedure explains how to create this project. You need one for each server configuration that will use APEs.

With Locator distributing the workflows and APEs:

- You, as the administrator, are guaranteed that the user has connectivity to these files since they reside on the application Server that the user is currently accessing.
- You can revert to a previous revision of a workflow or APE if necessary.

Do *not* change the name of this project. Locator looks for this project when doing anything regarding APEs and workflow. If this project is renamed or missing, StarTeam Extensions functionality will fail.

Sun's JRE 1.3.1_01 or higher is required for execution of the StarTeam Extensions' workflow engine and utilities. This installation may be found at http://java.sun.com, but it should have been installed when you installed the application Server and clients.

Important All users need read access to the files in the StarFlow Extensions project or they will not be able to use the workflows and APE files. The ability to change the files in this project, however, should be limited to the few users who will be creating or modifying workflows, APEs or custom tools.

Creating StarFlow Extensions Projects

Every server configuration that uses Locator to start APEs or tools must have a StarFlow Extensions project.

To create the StarFlow Extensions project:

1. For each server configuration, copy the StarFlow Extensions folder from the location where it was installed (by default C:\Program Files\Borland\StarFlow Extensions) to a unique location for each server configuration, for example, C:\config_name\StarFlow Extensions.
2. Start the Server with the server configurations that will be using workflow.
3. Start the Windows or Cross-Platform client.
4. Create a StarTeam project named "StarFlow Extensions" in each server configuration, using the unique location for the StarFlow Extensions folder as its working folder:

   a. From My Computer or Windows Explorer, select the folders to be included in your project.

   b. Drag the folders to an empty spot in the application window and release the mouse button. The *New Project Wizard* dialog appears.

      If you drag the folders to an open view window, the application adds the folders to that project view as children of the selected folder. This is not your goal. You want to create a new project.

   c. Do one of the following:

      - Select a server on which you want to store your project (from the server list), and click Next to continue.
      - Click Add Server to add access to another server and follow the steps in "Adding Access to a Server Configuration" on page 9, and then continue with this procedure.

      After you log on or if you are logged on already, the *New Project Wizard: Project Name* dialog appears.

   d. Enter "StarFlow Extensions" in the Project Name text box.

   e. Enter a description of the project and click Next to continue. The *New Project Wizard: Working Folder* dialog appears.

f Click Next. The *New Project Wizard: Child Folders* dialog appears.

g Do not exclude any folders.

h Click Finish to open the project.

5 Repeat step 4 for any other server configurations.

You can add more folders after a project has been created.

Adding Access to a Server Configuration

You can access one or more server configurations from the application. When adding access to a server configuration, you must provide the domain name (DNS) or IP address and appropriate endpoint for that server configuration. After the server configuration is added, users can access whatever projects are available for that server configuration.

You can add server configuration access as part of creating a project or as part of opening an existing project.

1 From the *New Project Wizard* dialog, click Add Server. The *StarTeam Server* dialog appears.

2 Enter a description in the Server Description text box. This is a unique descriptive name for the server. It is not case-sensitive and cannot contain colons (:). It can be any name you want.

3 Enter or browse for the address in the Server Address text box. The address is the computer name or IP address.

4 Enter the correct TCP/IP endpoint in the Endpoint text box. The endpoint is the port number associated with the protocol.

5 (Optional) Select the Compress Transferred Data check box to compress data transferred between your workstation and the server.

6 (Optional) Select a minimum encryption type check box to encrypt data transferred between your workstation and the server. Encryption protects files and other project information from being read by unauthorized parties over unsecured network lines (like the Internet).

The encryption types are ordered (top to bottom) based on speed and security. Each encryption type is slower, but safer, than the type that precedes it in the list.

For more information about encrypting data, see the *StarTeam User's Guide*.

7 Click OK to return to the *New Project Wizard* dialog.

Adding Files to the StarFlow Extensions Project

After you create the StarFlow Extensions project, you must add the files that are "Not in View" so that you can track changes made to these files and support the Locator application.

To add files:

1 In the application, open the StarFlow Extensions project if it is not already open.

2 Select the root folder (named StarFlow Extensions).

3 Select the File tab from the upper pane.

4 Click the All Descendants icon on the toolbar to display all the files that are located in the project's folders.

5 Select the <All Files By Status> or Files Not In View filter from the filter list box on the toolbar.

If you selected the <All Files By Status> filter, click the button in front of the Not In View group to display the files.

Note The Not In View status is assigned to each file in a working folder that is not already added to the project (and, therefore, not yet under version control).

1. Choose Select > Select All from the File or context menu to select all the files simultaneously.
2. Do one of the following:
   - To quickly add all the files, click the Add Files icon on the toolbar to add files without using the *Add Files* dialog.
   - Choose Add Files from the File or context menu. The *Add Files* dialog appears.
     1. Enter a generic description for all files in the *File Description* text box.
     2. From the *Lock Status* group box, make sure that the Unlocked option button is selected so that the files are unlocked.
     3. Make sure that the Delete Working Files check box is cleared so that you keep the working files in the working folder as well as in the repository.
     4. Clear the Link And Pin Process Item check box unless you have set up this project to use process items. If so, see the *StarTeam User's Guide* for more details about selecting a process item while adding files.
     5. (Optional) If you are using the Cross-Platform client, click Advanced to set advanced features. See the *StarTeam User's Guide* for more details.
     6. Click OK.

As you add new files to the application, their statuses change from Not In View to Current.

Depending on the filter you select, the names of the selected files may no longer on the upper pane. For example, if you use the filter Files Not In View, you can see your files again if you select the Files In View filter.

Understanding Locator

StarTeam Extensions includes a Locator for finding and distributing workflow and APE files to workstations running Windows clients.

To use Locator, you enter Locator or LocatorDebug before the name of the APE in the StarTeam Project Properties dialog. See “Configuring StarTeam Projects to Use APEs” on page 18 for more details.

Locator opens the StarFlow Extensions project, finds the correct files automatically, loads them into memory and executes the main entry point for the APE .jar file. The resources required by the APE are specified in the .propertyeditor.xml file.

While Locator only informs the user that the APE is loaded the first time it is called, LocatorDebug can display a console log file during initialization. This console log provides details about the execution and, if an exception occurs during initialization, LocatorDebug displays the location of the exception and its code. When testing changes to the workflow configurations and custom forms, you will want to use LocatorDebug in a test environment before moving the workflow configurations and custom form to your production projects.

The most recent starteam-extensions.jar file must be in the StarTeam installation folder for both the Windows and the Cross-Platform client. The Windows client also needs the most recent Locator.exe and LocatorDebug.exe. Locator.exe, LocatorDebug.exe, and starteam-extensions.jar need to be in the client installation folder.

When you upgrade StarTeam Extensions to a new release, you must check in new files to the StarFlow Extensions project and make sure that the appropriate files are distributed to the clients. For example, if you use the Enterprise Advantage edition of the application with JBuilder, you need to replace JBuilder's version of starflow-extensions.jar with the most recent version of the file.

Using a Specific JRE Causes Dependency on SDK

Starting with the 5.3 release, Locator.exe and LocatorDebug.exe makes use of and become dependent on the launching mechanism added to StarTeam Runtime. This means that you cannot use StarTeam Extensions with older application Servers unless you install the current StarTeam Runtime.

This new mechanism uses a JRE installed with StarTeam Runtime that is private to the SDK and should not be affected by other VMs installed on the system. This solves several problems caused by using a variety of JREs.

Adding Folders to the StarFlow Extensions Project

If you plan to use different APEs and/or different workflows for different projects and/or views in a given server configuration, you need to place those workflows and APE files in different places within the StarFlow Extensions project.

- If you store workflow and APE files in the Projects folder, they can be used by all projects on the server unless overridden by files at a lower level.
- If you create a *myproject* folder under Projects, all workflow and APE files stored within this folder are used for the project named *myproject* that reside in the same server configuration as the StarFlow Extensions project.
- If you create a *myview* folder under *myproject*, the workflow and APE files stored within it are used in the *myview* view for the specified project.

The files that must be checked in to the Projects folder or an appropriate subfolder in the StarFlow Extensions project are *name*.jar, *name*.propertyeditor.xml and *item_type*.Workflow.xml.

If projects and views will use different APEs and/or workflows, then you need to add folders to the StarFlow Extensions project. After you create the folders that you need, you will add workflows and APE files to those folders.

- Use the Projects folder for workflows and APEs to be used as the default for all projects and views.
- Use a folder with the name of a specific project (such as StarDraw) to store workflows and APEs to be used by the views in that project—unless a folder and files exists for a specific view.
- Use a folder with the name of a specific view, such as Beta Release, to store workflows and APEs to be used by that specific view.

Notice that the folder for the root view of a project usually has the same name as the project. In many cases, you would not need both folders. For example, the folder that has the path StarFlow Extensions\Projects\ WebDev\WebDev is currently unnecessary because WebDev, the root view of the WebDev project, is the only view in the project.

Locator use the folder hierarchy to search for resources. If for example, an item in the Release 1.0 Maintenance view is being processed, Locator checks for workflows and APEs in StarFlow Extensions\Projects\ StarDraw\Release 1.0 Maintenance. If it does not find all of the files it needs, it looks in StarFlow Extensions\Projects\StarDraw. Locator continues up the tree until it reaches the StarFlow Extensions\Projects folder. If it still has not located the files it needs, an error message is displayed.

To add a folder to a view using the application:

1. Choose New from the Folder or context menu. The *New Folder Wizard* dialog appears.

2 Select a folder from the folder tree to be the parent of the new folder.

3 Click Next. The *New Folder Wizard: Folder Name* dialog appears.

4 Enter a name for the child folder in the *Folder Name* text box. Use a maximum of 254 characters.

5 Leave the *Working Folder* text box blank. The application creates the working folder using the name of the new folder and the path to its parent's working folder.

6 In the Folder Description text box, enter a description of up to 254 characters for the new folder.

7 Click Next.

The *New Folder Wizard: Child Folders* dialog displays the new folder in the *New Folder's Child Folders* box.

If the working folder has child folders, the application adds an application folder for each of them.

8 Do one of the following:

- To exclude a child folder from your project, select the folder and click Exclude.
- To exclude all child folders, click Exclude All. To redisplay folders you have excluded, click Reset Folders.

9 Click Next. The *New Folder Wizard: StarTeam Folders* dialog displays the view's folder tree with the new child folder.

10 Click Finish.

Setting Access Rights

This section explains the minimum access rights to the StarFlow Extensions project required for users of StarTeam alternate property editors.

It is best to set the access rights on the StarFlow Extensions project at the project level. Using the *Project Access Rights* dialog, ensure that the All Users group has the following rights:

On the Project node, users need to be granted the “See project and its properties” right.

On the View node, users need to be granted the “See view and its properties” right.

On the Child Folders node, users need to be granted the “See folder and its properties” right.

On the File node, users need to be granted both the “See file and its properties” right and the “Check out files” right.

On the remaining nodes, (Change Request, Requirement, Task, and Topic), the Administrator should deny access to casual users.

Creating and Deploying APEs

Alternate property editors (APEs) are custom forms created specifically for your company in support of a corporate process. Workflow processes are created for use with the forms. They can be customized easily because they can be implemented in Java. Sample property editors for all application components are included.

However, an APE is not just a form. It is a StarTeam SDK application that is executed instead of the default properties dialog for the item type. The files you must check in to the StarFlow Extensions project or an APE are *name*.propertyeditor.xml and a *name*.jar file.

StarTeam Extensions provide the workflow engine, form customization capability and distribution architecture. You are not required to use this software for APEs because any application can be executed as an APE, allowing maximum flexibility in tailoring the application to work in a manner required by your process needs.

You must define an APE for any project that will use workflows. You can use APEs:

- Without workflow. For example, if you do not like the standard properties dialog for an item type, you can replace it with an APE of your own design. In this way, you can hide properties on the standard properties dialog from view and put all custom properties in easily accessible locations on the form, instead of on the Custom tab.

  To use an APE without workflow, you select an APE for the item type in the project and check in a disabled workflow. See “Disabling a Workflow” on page 64.

- With workflow but without notification. For example, if the workflow is well understood and everyone knows exactly who will be responsible for the next step, notification is unnecessary. Everyone can sort items based on responsibility and creation or entry time, and work on the items for which they are responsible in the order in which they entered the system.

  To use an APE with workflow but without notification, you select an APE for the item type in the project and check in a workflow that has no notification lists.

- With workflow and notification. For example, if you use notification, you can ask several people, in turn, to take responsibility for an item at a given step. You can also notify supervisors if no one accepts responsibility, or if a step is taking longer than expected.

  To use an APE with workflow and notification, you select an APE for the item type in the project and check in a workflow that has notification lists. See “Controlling the Behavior of an Item in a Step” on page 52.

While an APE is for a specific component, you do not have to use the same APE for all the projects or views in a server configuration. You can create different forms per-project or per-view.

StarTeam Extensions simplifies the process of creating custom forms in JBuilder or another Java development environment by providing a variety of JavaBeans. The StarTeam JavaBeans allow you to customize the APEs using a graphical form designer.

In addition, Java code can be added to the form to access the Server via the StarTeam SDK. Such code can provide additions to the form and enforce workflow side-effects as required.

Using Modal or Embedded APEs

A single APE form can be used in a context dialog (as in pre-5.3 releases) and in an embedded context such as in the StarTeam JBuilder integration, where the form is implanted in a document window, rather than in a modal dialog.

Prior to 5.3, APEs consisted of a subclass of com.starbase.starteam.starflow.AbstractLauncher, which launched the form and placed it in a dialog, and a subclass of javax.swing.JPanel which implemented the form UI. APEs now subclass com.starbase.starteam.starflow.BasicLauncher instead of AbstractLauncher. BasicLauncher serves a similar function but works with embedded forms as well.

A new subclass of AbstractItemEditor was added to 5.3 also. Whereas the BasicLauncher subclass is the entry point for launching the editor in a dialog, this subclass becomes the entry point for launching editors embedded in a window.

The form still subclasses JPanel (typically), but the Navigator component, which supplies the Next, Previous, OK, Cancel, Apply, and Reset buttons, is no longer added directly to the form.

Instead, BasicLauncher adds the form to a dialog that already contains the Navigator component, while the AbstractItemEditor subclass merely instantiates the form with no Navigator component. No Navigator component is used in the embedded context because the application, rather than the form, has the responsibility of handling those functions.

The following optional element has been added to the .propertyeditor.xml for an APE:

```
<itemeditor class="com.starbase.starteam.changerequest.ChangeRequestEditor">
       <jar name="ChangeRequest.jar"/>
       <jar name="guicomponents.jar"/>
       <jar name="starflow-extensions.jar"/>
     </itemeditor>
```

The jar elements serve the same function as in the javaclass element, and the class attribute of the itemeditor element serves a similar function as in the javaclass element. It identifies the name of the entry point for the APE as an embedded form (it must identify a subclass of AbstractItemEditor).

Understanding Property Editors

In addition to supporting files as part of its Software Configuration Management (SCM) features, the application also supports change requests, requirements, tasks, and topics. Collectively these are known as item types, and the part of the application that supports a particular item type is known as a component. Each item type is surfaced in the application clients as a separate tab with the component name.

To edit a particular item’s properties, the user selects the item and then chooses *item_type* > Properties from the menu bar. The user can also double-click the item—unless the item is a file. (Double-clicking a file displays its contents rather than its properties; select a file and press Alt+Enter to display the property editor.)

By default, the application displays the standard property dialog.

Many customers do not use all of the standard properties for an item type, and tend to create custom properties for information important to their organization’s internal processes. If a customer creates custom properties, they appear as fields on the Custom tab of the standard property dialog. Many customers would prefer to locate fields for the properties that are most important to them on the first tab and hide the properties that are irrelevant to their process by not having fields for these properties on the form. The application’s alternate property editors (APEs) make this possible. With APEs, the properties can be controlled by a workflow that fits the customer’s needs and allows different properties to be displayed, required, or enabled on the form depending on the step in the workflow that is being performed.

StarTeam Extensions includes sample forms that you can use immediately or customize using your favorite Java compiler and form design product. These forms make use of custom JavaBeans that integrate with the StarTeam SDK and the StarTeam Extensions’ workflow engine. These JavaBeans allow you to drag a new field onto the form and assign an application field name to the JavaBean. All of the access to the field through the StarTeam SDK and the workflow behavior is then handled by the JavaBean.

There are several sample forms included in the StarTeam Extensions installation. These are located in the Samples folder of the StarTeam Extensions installation folder (by default, C:\Program Files\Borland\ StarFlow Extensions\Samples).

Under this folder is a copy of the StarTeam Runtime .jar file for use by the sample application’s JBuilder projects. This is required so that StarTeam Extensions can include a preconfigured project and make it easier for you to get started with customizing the various forms.

Documentation on the StarTeam Extensions API is found in the Program Files\Borland\ StarFlow Extensions\Documentation folder.

Sample Change Request APEs

A sample change request APE is found in the Samples\ChangeRequest folder under the StarTeam Extensions installation folder.

In the JBuilder project named ChangeRequest.jpx, required fields (in this case, Priority, Synopsis, and Responsibility) are indicated in red. The tab name (in this case Synopsis) is in red to indicate that the tab has required property fields on it. The change request cannot be saved until a valid value is supplied for all required fields. The source code for the sample form is provided so you can use this form as a starting point for creating your own custom form.

Another such sample change request APE is found in the Samples\LinkTypeChangeRequest folder under the StarTeam Extensions installation folder. This sample is found in the JBuilder project named LinkTypeChangeRequest.jpx.

In addition to the capabilities of the previous change request APE, this APE can display a corporate logo and access the items to which the change request is linked. Some properties are displayed at all times. Clicking tabs changes only the lower part of the form.

When using StarTeam Extensions for workflow management, the existing Status field found in the change request is not used as the workflow control property. The reasons for this are:

- Existing change requests and the built-in property editor dialog rely on specific status values being used for a change request.
- Changing this status value causes specific side effects. To isolate these side effects from the implementation of a custom workflow, a separate custom property is used.

  In general, you should try to keep the Status property in the state that most closely matches the corresponding Workflow Status. You should also change the public filters to include the Workflow Status field instead of the default Status field.

WorkFlow Status Property

The ChangeRequest.Workflow.xml that ships with the product expects that you have a custom property field name WorkFlow Status with the internal name Usr_StarFlowStatus.

Sample Requirement APEs

StarTeam Extensions includes sample requirements APEs.

These APEs display the selected requirement. If this requirement was originally exported from CaliberRM, it will be displayed in a read-only mode. However, clicking the CaliberRM button or the Launch CaliberRM button starts the CaliberRMViewer, and if the user has a license to CaliberRM, allows the user to start CaliberRM and, using CaliberRM, edit the requirement.

The APE and launcher for the viewer are found in the JBuilder project named Requirement.jpx.

Configuring JBuilder to Use StarTeam JavaBeans

While you can use any full featured Java development environment with StarTeam Extensions, the samples included were created with Borland JBuilder Enterprise Edition. These samples may also be maintained with later releases of JBuilder.

To use the StarTeam JavaBeans in JBuilder you must configure your libraries and palette.

To use the JavaBeans:

1 Configure the JBuilder libraries to include the starflow-extensions.jar, starteam*xx*.jar and guicomponents.jar files.

a Choose Tools > Configure Libraries from JBuilder's menu bar.

b From the resulting *Configure Libraries* dialog, select the User Home folder.

c Click New to create a StarTeam Extensions folder (under the User Home folder) that contains the starflow-extensions.jar and guicomponents.jar files.

Note In earlier editions of this manual, users created a StarFlow folder under the User Home folder. You might already have this set up, but you need to change it.

d In the resulting *New Library Wizard* dialog, enter "StarTeam Extensions" in the Name text box.

e Click Add.

f From the resulting *Select One or More Directories* dialog, select the starflow-extensions.jar and guicomponents.jar files from the StarFlowExtensions\Projects folder at C:\config_name\StarFlow Extensions\ Projects).

g Click OK to return to the *New Library Wizard* dialog.

h Click OK to return to the *Configure Libraries* dialog.

i Click Add again.

j From the resulting *Configure Libraries* dialog, select the User Home folder again.

k Click New to create a StarTeam folder (under the User Home folder) that contains the starteam*xx*.jar file.

Note In earlier editions of this manual, users created a StarGate folder under the User Home folder. You may already have this set up, but you need to change it.

l In the resulting *New Library Wizard* dialog, enter "StarTeam" in the Name text box.

m Click Add.

n From the resulting *Select One or More Directories* dialog, select the starteam*xx*.jar file (usually located at C:\Program Files\Borland\StarTeam SDK *x.x*\Lib\starteam*xx*.jar).

o Click OK to return to the *New Library Wizard* dialog where this file now appears in the Library Paths list.

p Click OK to return to the *Configure Libraries* dialog.

q Click New to create a StarTeam folder (under the User Home folder) that contains the starteam-gui.jar file.

r In the resulting *New Library Wizard* dialog, enter "GUI SDK" in the Name text box.

s Click Add.

t From the resulting *Select One or More Directories* dialog, select the starteam-gui.jar file from the StarFlowExtensions\Projects folder (for example, at C:\config_name\StarFlow Extensions\ Projects).

- u Click OK to return to the *New Library Wizard* dialog where this file now appears in the Library Paths list.
- v Click OK to return to the *Configure Libraries* dialog.

  User Home should now have subfolders named StarTeam Extensions, StarTeam, and GUI SDK.

2 Add a new palette by choosing Tools > Configure Palette from JBuilder's menu bar.

- a In the resulting *Palette Properties* dialog, click Add.
- b If the resulting *Add Page* dialog, enter StarTeam Extensions.
- c Click OK to return to the *Palette Properties* dialog.
- d Select StarTeam Extensions.
- e Click Move Up until StarTeam Extensions is at the top of the list in this tab.
- f Click the Add Components tab.
- g Click Select Library.
- h From the resulting *Select a Different Library* dialog, select the StarTeam Extensions library.
- i Click OK to return to the *Palette Properties* dialog.
- j From the Palette Page list, select the StarTeam Extension palette page.
- k Click Add From Selected Library at the bottom of the dialog.

  As the JavaBeans are added, they are listed in the *Results* dialog.
- l Click OK, and you are ready to use the StarTeam JavaBean components.
- m Click OK again to return to the main window.

3 In JBuilder, open a sample project to be modified for use with StarTeam.

4 Double-click the .java file to be modified.

5 Click the Design tab. The StarTeam Extensions JavaBeans should appear on the StarTeam Extensions page of the toolbar.

Using JBuilder Personal Edition

JBuilder Personal does not include IDE-based support for building Jar files. Building the Jar file is a necessary step for use as an APE. For your convenience, we have included Make*itemtype*Jar.bat in the base of each respective sample project. After you have compiled the project in JBuilder, you can run the batch file to create the Jar file, which will be placed in the Projects folder by default, where it will be ready for check in to the StarFlow Extensions project in the Projects folder. You may need to edit the batch file if JBuilder is not installed in the C:\jbuilder*x* folder.

Editing StarTeam APEs

You can add new fields to or edit the existing fields of a StarTeam Extensions form using your favorite Java development tool. These forms are called alternate property editors (APEs). When you use a StarTeam JavaBean component, you will need to select the correct component that matches the type of field you are using, and after the component is placed on the form, you will need to set the propertyName attribute. The propertyName attribute should have the application field name that you wish to display in this component in the form.

Note In design mode the label on the field will be the value found in the propertyName attribute of the JavaBean, but when executed the label will be the StarTeam Display Name for the field.

Where applicable, the JavaBeans are composed of an editor component and a text label showing the name of the property being edited. By default, the text shown by the label is the display name of the StarTeam property associated with the JavaBean (set via the "propertyName" JavaBean property). When you edit the JavaBean in a visual form designer such as JBuilder, the label's text will be the internal identifier for the property, that is, the value of the propertyName property of the JavaBean. When the form is run, that text is replaced with the display name of the property (defined by StarTeam metadata about the property). For example, the StarTeam Status field has the same display name and internal identifier. However, the custom status field used in the change request samples has the display name “Workflow Status” and the internal identifier Usr_StarFlowStatus. In the latter case, you see Usr_StarFlowStatus in JBuilder and “Workflow Status” when the form runs.

You can control the behavior of this label by doing any of the following:

- Supplying alternate text for the label

  To do this, set the labelTextOverride property of the JavaBean. The text you supply will be seen in JBuilder and when the form runs.

- Turning the label off so that it is not seen at any time.

  To do this, set the showLabel property to false. To turn the label back on later, you would set the showLabel property to true.

Configuring StarTeam Projects to Use APEs

By default, every project uses the standard property dialogs. When an APE is ready for use, you must change the project properties so that all the views in a project will use the APEs.

Once set, every view in the project must use an APE instead of the standard dialog. However, each view can use a different APE that you have checked into the appropriate view-specific subfolder in the Projects folder of StarFlow Extensions project. See “Adding Folders to the StarFlow Extensions Project” on page 11 and “Understanding Locator” on page 10.

Important Take care when setting the APE for a project that is in production because the new setting takes effect immediately. It is important to test any changes that are made to an APE prior to making this editor available to a wide audience.

To use an APE with a StarTeam item type:

1. In the application, choose Project > Properties from the menu bar. The *Project Properties* dialog appears.
2. Select the Editors tab.
3. Select the Use Alternate Property Editor For <item> check box.
4. Enter or browse for the correct APE. Usually you enter the word Locator followed by the name of the APE.

   Locator is a program on each client workstation that distributes resources, such as .jar and .xml files to client workstations. If specified here, whenever the application requests the APE, Locator looks for the StarFlow Extensions project managed by the server configuration. If that project exists and contains a .propertyeditor.xml file with the specified name in either the Projects folder or an appropriate child folder, Locator copies the tip revision of that resources referenced in that file to the client workstation (if they are *not* already there). Then the user edits items using the APE instead of the application’s standard properties dialog for the specified type of item.

Using the propertyeditor.xml File

Locator obtains the details about the resources to load and run from the .propertyeditor.xml file. For example, if you use the APE “Locator ChangeRequest”, the file ChangeRequest.propertyeditor.xml is read by the Locator to determine which resources are deployed for the APE.

Below is a copy of the sample ChangeRequest.propertyeditor.xml file included with the StarTeam Extensions:

```
<?xml version="1.0" encoding="UTF-8"?>

<propertyeditor name="Change Request Editor"
version="1.1">
    <itemeditor
    class="com.starbase.starteam.changerequest.
    ChangeRequestEditor">
        <jar name="ChangeRequest.jar"/>
        <jar name="guicomponents.jar"/>
        <jar name="starflow-extensions.jar"/>
    </itemeditor>
    <javaclass
    name="com.starbase.starteam.changerequest.
    ChangeRequestLauncher">
        <jar name="ChangeRequest.jar"/>
        <jar name="guicomponents.jar"/>
        <jar name="starflow-extensions.jar"/>
    </javaclass>
</propertyeditor>
```

The name of the .propertyeditor.xml file is passed as the only argument to the Locator in the *Project Properties* dialog, for example, “Locator ChangeRequest”. Locator finds this .propertyeditor.xml file and determines which resources are needed for this APE.

The <javaclass> element identifies the Java class that acts as the entry point for the APE. The “name” attribute names the class and the <jar> subelements name the jar files that implement the editor. The named jar files are located in the StarFlow Extensions project in the same manner as the .propertyeditor.xml itself, allowing for overriding in a similar manner. These jars are checked out of StarTeam and loaded into the Java Virtual Machine before the “main” method of the named entry point class is called.

The <external> element contains the default command and any operating specific commands that may be needed. The <command> element includes a platform option that names the operating system and a call option to indicate the executable to call on the target platform. Within the <command> element are the parameters to pass to the executable specified on the call option. The format “${resource:<file>}” is used to tell the Locator that the <file> is to be located by using the standard logic for locating a resource in the StarFlow Extensions project, that is by searching for a folder with the view name and then the project name and, if not found, finally using the default files found in the Projects folder.

Using a Property Other Than Usr_StarFlowStatus

Usr_StarFlowStatus (with Workflow Status as its usual display name) is the property the workflow engine expects to find as the workflow control property for any item. This is easy to change.

To use a property other than Usr_StarFlowStatus:

1 In JBuilder, open the *APEproject*.jpx file for the project (for example, LinkTypeChangeRequest.jpx or ChangeRequest.jpx).

This may be in one of the Samples folders or perhaps you have the source code for this java project in the working folder for an appropriate StarTeam folder in the StarTeam project named StarFlow Extensions.

2 Double-click the *item_type*Launcher.java file to open it for editing.

3 After the file opens, search for Usr_StarFlowStatus:

```
public String getWorkflowStatusPropertyName() {
return "Usr_StarFlowStatus";
```

4 Replace Usr_StarFlowStatus with the internal name for the workflow control property to be used with this *item_type*.

For example, you might be using an existing field such as Status property field for Topics or a Usr_MyStatus for any item type. The only field that should not be used is the Status property field for change requests as the application's built-in workflow involves the use of that property field.

5 Save the file.

6 Recompile the .jar (for example, LinkTypeChangeRequest.jar or ChangeRequest.jar).

7 Check the .jar file into the application's StarFlow Extensions project. Use an appropriate folder.

For example, if this APE will be used by a specific project view, you might check it into the StarFlow Extensions\Projects\Big Project\ Big Project\Web Stuff folder. In this example, there is a project named Big Project with a root view name Big Project and a child view named Web Stuff.

Converting User Groups

Prior to the 5.3 release, you had to use the GroupConverter utility to convert the user and group names found in your server configuration for use with the workflow engine. You had to run the GroupConverter utility initially and every time you modified group membership, because StarTeam Extensions could not retrieve this information from the server via the StarTeam Runtime.

It is no longer necessary to use the utility to maintain a groups.xml file in the StarFlow Extensions project that maps application users to application groups.

Prior to 5.3, only persons with administrator accounts were allowed to directly query the server to determine if a user was a member of a particular group. Now this query is accessible to all user accounts, so the workflow engine queries the server directly.

Using the Form Server

The StarTeam Form Server is a background application that improves the performance and responsiveness of the APE feature. Once loaded by Locator, it keeps in memory the resources that are expensive to reload, so that subsequent access to those resources is faster.

Under normal operation, the user need not do anything to receive the benefit of the Form Server. It automatically starts the first time the user opens an APE. Note that the first time an APE is launched (if the Form Server is not already running), it may take an extended period of time to initialize and display the form.

An icon representing the Form Server appears in the user's system tray indicating that the server is running. The Form Server may be shut down by double-clicking the icon and responding Yes to the confirmation dialog.

However, the form server does not stop when you exit the application.

To stop the form server:

1 Double-click the form server icon in the system tray. A message box asks whether or not you want to shut down the form server.

2 Click Yes.

Using the Custom Toolbar

The custom toolbar can be used with the Cross-Platform client. Buttons on the toolbar start tool applications that are usually written for your organization by Borland consultants using the StarTeam Custom Toolbar Framework SDK. The tools are often time-saving applications that access the application Server and perform multi-step operations that you do routinely.

The toolbar buttons can be displayed at the view level or per-component. For example, you might see buttons while using the change request component but not when using the file component. When you see a component-level button, the tool it represents may also appear on the item menu for that component. For example, if a tool named MyTool appears on the custom toolbar when you use the change request component, the Change Request menu usually has a submenu named Tools with MyTool as one of its menu commands. Some tools have representations that are only appropriate in the toolbar and not the menu.

Which users can see these toolbar buttons is controlled by the access rights on the .tool.xml files (for example, *mytool*.tool.xml) that are checked in to application folders located in the server configuration’s StarFlow Extensions project. The folders use the same folder scheme as the workflow and APE form files.

If changes are made to the tools or the contents of the toolbar, you are unaware of those changes until you reopen the project view in which they are used. If an error occurs as a tool is placed on the toolbar, an error-message toolbar button appears on the toolbar. Clicking the button displays the errors encountered by the toolbar on startup.

Advanced Topics

The following sections explain:

- All the JavaBeans in the guicomponents.jar file
- Passing parameters to an APE
- Using the .propertyeditor.xml file
- How to subclass the ItemWrapper class to support more complex workflow needs

Using the JavaBeans in guicomponents.jar

The JavaBeans included in the guicomponents.jar are:

- AddAndLinkButton

  This button launches a property editor that adds a new item to the repository and automatically links the new item to the item currently displayed by the form in which this button resides.

  Add this button within the scope of a PropogatingPanel and set the itemType and locatorKey properties to the name of the item type and the name of the form to use, respectively. Optionally, set the addToSameFolder property (which defaults to true) and/or the createPinnedLink property (which defaults to false).

- AttachmentsEditor

  This component displays, adds, and removes attachments to an item.

  Add this component within the scope of a PropogatingPanel. Nothing else needs to be done to make this component function. The propertyName is fixed, so setting this property has no effect.

- BooleanEditor

  This component edits properties of type Property.Types.INT32 or Property.Types.ENUMERATED as boolean values.

  Add this component within the scope of a PropogatingPanel and set the propertyName property to the name of the property to be edited. Optionally, set the trueValue and falseValue properties to the integer values that represent true and false. In the case of Property.Types.INT32 properties, this is a pair of arbitrarily designated values. In the case of Property.Types.ENUMERATED properties, this pair corresponds to the code values of the enumerated values that represent true and false.

- ComboEditor

  This component edits properties of type Property.Types.ENUMERATED.

  Add this component within the scope of a PropogatingPanel and set the propertyName property to the name of the property to be edited. Optionally, set the sortOrdering property to make the list of values be sorted in ASCENDING, DESCENDING, or NONE sort order.

- DateEditor

  This component edits properties of type Property.Types.TIME.

  Add this component within the scope of a PropogatingPanel and set the propertyName property to the name of the property to be edited. Optionally, set the displayTimePortion property (which defaults to true) to display the date only or date and time. Optionally set the allowNull property (which defaults to true) to allow or disallow blank values.

- ItemDisplayListener

  This is not a JavaBean; it is an interface that some of the JavaBeans support. It is the listener interface for receiving notifications of a new item to be displayed and to refresh the display of the currently displayed item.

- ItemsPaneComponent

  Displays requirements, tasks, or topics in a folder in tree form. When an item is displayed in this component, the tree will be expanded so that the item is visible and the item can be selected. Most of the normal application item-level functionality is available via the context menu for the selected Item.

  The ItemsPaneComponent depends on a StarTeam library that requires a StarTeam Server 5.3 or higher. On older releases, the component displays a placeholder instead of the component. The placeholder indicates that the component cannot find a 5.3 or higher license.

  This component can be used in one of three ways:

  - If the actAsItemDisplayListener property is set to false (default), then you may set the linksComponent property to create a relationship such that when an Item selected in the LinksComponent, it will be displayed in this component.
  - If the actAsItemDisplayListener property is set to false (default), then you may leave the linksComponent property empty and call the showItem(Item) or setItem(ItemWrapper) methods directly.

  - If the actAsItemDisplayListener property is set to true, then the component will display the ItemWrapper set by the setItem method of the ItemDisplayListener interface.

    Add this component within the scope of a PropogatingPanel, and it will automatically display the Item in the "current scope" according to the PropogatingPanel. Nothing else must be done to make this component function.

- ItemPropertyEditor

  This is not a JavaBean; it is the superclass of most the JavaBeans. This abstract base class is for property editor components. Subclasses of this class allow the display or editing of individual item properties. This class hooks listeners to event sources, such as PropogatingPanel, that control which item to display.

- ItemPropertyLabel

  This component displays a property's display name, optionally allowing an arbitrary name to override the default display name.

  Add this component within the scope of a PropogatingPanel and set the propertyName property to the name of the property. Optionally, set the text property to override the display name of the property. Optionally, set the prefix and suffix properties.

- LinksComponent

  This component displays items linked to an item.

  Add this component within the scope of a PropogatingPanel. Nothing else must be done to make this component function. The propertyName is fixed, so setting this property has no effect. Optionally, set the typeFilter property to the name of an item type to limit the displayed items to items of that type.

- Navigator

  This component controls the navigation through a list of ItemWrapper.

  Add an ActionListener to listen for button events (ok, cancel, apply, and reset), or add a PropertyChangeListener to listen for currentItem events indicating that the currently selected item has changed, and items events indicating that the underlying list of ItemWrappers has changed. Optionally, set the refreshItems property to indicate whether or not the item data should be fetched from the server every time the item selection changes. The refreshItems property defaults to true.

- NumberEditor

  This component edits properties of type Property.Types.INT32 or Property.Types.FLOAT64.

  Add this component within the scope of a PropogatingPanel and set the propertyName property to the name of the property to be edited. The component automatically detects the difference between integer and floating point numbers and behaves appropriately.

- OpenLinkedItemButton

  This button launches a property editor to display an item selected in a LinksComponent.

  Add this button to the form and set the LinksComponent property with a reference to the LinksComponent to find the selected link in.

  Clicking the button performs the same action on the selected item as double-clicking the item in the LinksComponent.

- PropogatingPanel

  This component recursively propagates the currentItem property change events to all the PropogatingPanel, ItemPropertyEditor, and ItemDisplayListener objects contained in this panel. It provides a convenient way to hook all property editors to the Navigator so that they can listen for new current item selections. As a side effect, this component sets tabbed pane tab titles red if the tab contains any required properties. Borland recommends using an instance of this class at the top level somewhere and adding it as a PropertyChangeListener to the Navigator.

- RevisionCommentEditor

  This component displays and edits the revision comment for an item.

  Add this component within the scope of a PropogatingPanel. Nothing else needs to be done to make this component function. The propertyName is fixed, so setting this property has no effect.

- StaticTextEditor

  This component displays any type of property as read-only static text.

  Add this component within the scope of a PropogatingPanel and set the propertyName property to the name of the property to be displayed.

- TaskDependenciesComponent

  This component displays the predecessor and successor relationships for a task.

  Add this component within the scope of a PropogatingPanel. Nothing else needs to be done to make this component function. The propertyName is fixed, so setting this property has no effect.

- TaskWorkEditor

  This component displays, adds, edits, and removes Work Records for a task.

  Add this component within the scope of a PropogatingPanel. Nothing else needs to be done to make this component function. The propertyName is fixed, so setting this property has no effect.

- **TestButton**

  This button populates a form with live data for testing convenience. This button is not intended to be visible on production forms, but may be added to the form so that, during development, live data may be used in the form without requiring that the form be launched from the application. The button should be hidden or removed when used in a production environment. The samples provided with StarTeam Extensions use this button in JBuilder and hide in when APEs are displayed to users.TextAreaEditor

  This component edits properties of type Property.Types.TEXT in a multiline editor.

  Add this component within the scope of a PropogatingPanel and set the propertyName property to the name of the property to be edited.

- TextFieldEditor

  This component edits properties of type Property.Types.TEXT in a single line editor.

  Add this component within the scope of a PropogatingPanel and set the propertyName property to the name of the property to be edited.

- UserListEditor

  This component edits property groups that contain lists of User IDs.

  Add this component within the scope of a PropogatingPanel and set the propertyName property to the name of the primary property of the property group that contains the list of users, the countPropertyName property to the name of the property group’s property that identifies the number of IDs in the list, and the IDsPropertyName property to the name of the property group’s property that identifies the IDs in the list.

Lists of users require implementing a special group of three properties. You must set the property name for all three members of the group.

Passing Parameters to an APE

The Windows client passes information to an APE that uses the StarTeam Runtime to retrieve an item and perform the appropriate operation. One of the following command formats will be passed to the APE depending on whether the user is adding, editing or reading an item from the application:

```
Add -p user@host:port/project[/view/folder/…/folder]
-encryptedpassword password

Edit -p user@host:port/project[/view]  -encryptedpassword password
-current VMID VMID1[,VMID2,…]

Read -p user@host:port/project[/view] -encryptedpassword password
[-cfgl|cfgd|cfgp specifier]
[-vn revision_number]
-current VMID VMID1[,VMID2,…]
```

For the -p option the user, host, port, and project values are always supplied. If the view option is not passed, then the root view for the project is to be used. On the add command, if the folder path is not supplied then the root folder for the view is to be used.

The -encryptedpassword option is used to pass the user’s password in an encrypted format. The password can be decrypted by the StarTeam Extensions library and passed to the StarTeam SDK.

The -current value is required for both the Read and Edit commands and indicates the item currently selected in the application. The VMID (View Member ID) value passed is an internal reference to the application item. The list of VMIDs is an ordered list of references to the items near the current item and are used to support “previous” and “next” functionality in the custom form. To improve performance, only 100 VMIDs before and 100 VMIDs after the current item are passed to the APE by the Windows client. If this list is not passed, then all items for the component are available for display.

The -cfgl, -cfgd, and -cfgp options are used when the view configuration must first be set to a specified state prior to displaying the item.

- The -cfgl option indicates that the view configuration should be rolled back to the label passed in the specifier parameter.
- The -cfgd option indicates that the view configuration should be rolled back to the date passed in the specifier parameter.
- The -cfgp option indicates that the view configuration should be rolled back to the promotion state passed in the specifier parameter.

The -vn option is used when a user is displaying an item from the StarTeam History pane and the parameter passed is the revision number to display. This option is mutually exclusive from the -cfgl, -cfgd and -cfgp options.

Note The Cross-Platform client does not use this parameter format. It loads the Java-based APE directly. As a result, there is no limit to the number of items displayed.

Subclassing the ItemWrapper Class

The ItemWrapper class is the core of the StarTeam Extensions’ workflow logic. There may be situations where the StarTeam Extensions workflow engine cannot support a complex workflow requirement, and in these cases the ItemWrapper can be subclassed and special logic can be implemented by overriding specific class members. The StarTeam Extensions Samples include a Customization Sample that illustrates how to do this.

To subclass the ItemWrapper class:

1. Create your own class to extend ItemWrapper.
2. Override the methods that you need to alter.
3. Change the code in the forms that make use of ItemWrapper to create an instance of your class instead of directly creating ItemWrapper instances.
4. If you continue using the provided “Launcher” framework, you can simply extend AbstractLauncher or one of its subclasses (like ChangeRequestLauncher), and override the createItemWrapper method.

   The framework uses this factory method to create ItemWrapper instances, so once you’ve overridden that one method, your subclass will be created wherever an ItemWrapper would have been created. This can be seen in ChangeRequestLauncher2 in the provided sample.

The methods of ItemWrapper are either public or protected so that they can be overridden. There are just a few methods that you’ll be likely to want to override, however. Following is a brief description of the likely candidate methods to be overridden:

- **put** method

  This method is used to set the value of any property (except the addition of attachments) of an application Item (ChangeRequest, Requirement, etc.). You should not replace this method unless you have an in-depth understanding of the implications. Rather, this is a good place to put pre- and post-processing logic. If you just want to be notified of a property change, you don’t need to subclass ItemWrapper at all; you can add a PropertyChangeListener to the ItemWrapper instance. If you want to block certain values from being “put”, you will want to override the checkValue method instead.

- **checkValue** method

  This method is called within **put** after checks are made to see that the user is authorized to call **put** in the first place, that the Item is not read-only, and that the value being put is different than the existing value. Once these conditions are met, **checkValue** is called with the name of the property and the value that is being set. The default implementation always returns true, but you can override this method to perform any special validation that you want.

- **getAllowedValues** method

  This method returns lists of values that are valid “next” values for a particular property. For instance, this method supplies the values for the drop-down lists used to select values for Responsibility, LastBuildTested, Priority, Severity, etc. Generally speaking, you will most likely want to simply limit the set of values returned by the default implementation based on your own rules, so it would be best to call the superclass’ method first and then rework that list before returning it.

- **isPropertyRequired**, **isPropertyHidden**, and **isPropertyDisabled** methods

  These methods determine whether a particular property requires a non-null value before the Item may be saved, or whether the property’s representation on the form should be hidden or disabled. You might override these if you have special logic that depends on other elements of the Item’s state.

- **isAuthorized** method

  This method determines if the user is allowed to edit the Item. You might override these if you have special logic that depends on other elements of the Item’s state.

- **update** method

  This method causes the changes to the Item to be saved in the repository. The same warnings apply as with **put**. Similar to **put**, if you want to perform some validation before allowing the changes to be saved, you will want to override the **validateProperties** method instead.

- **validateProperties** method

  This method is called within the **update** call to validate that all requirements are met before the changes are saved. If a requirement is not met, the method throws a **RequirementNotFulfilledException**.

- **dynamicDefaults** method

  This method is called with the name of the property that has just been "put" and then sets the default values of other properties based on the workflow rules. You may override this method if you have special logic that cannot be expressed by the workflow rules. The semantics of dynamic defaults is such that dynamic defaults are only set in response to a user changing a property value directly. Therefore, dynamic defaults should not be triggered by the setting of an ancillary property during dynamic defaults processing. In order to allow you to **put** a property value without triggering dynamic defaults processing, there is a protected, overloaded **put** method that takes an additional Boolean parameter indicating whether or not to perform dynamic defaults processing. You should use this overloaded method to put property values if you override **dynamicDefaults**.

Known Issues

Custom Fields not Recognized

The caching of server connections improves the form server's performance, but changes to customizable properties and new properties are not automatically recognized by APEs. Use one of the following workarounds:

- You can force-log-off all users after making the property customization. This causes the cached server connection to be recreated with the new information.
- You can perform the modification after hours if everyone logs off as they leave.
- Tell all users to shut down their form servers. When the APE is restarted, the connection is reestablished with the new information.

Things to Check

- Make sure that the server description's computer name or IP address and endpoint are correct. Also be sure that your username and password are correct.
- The StarFlow Extensions project cannot be renamed. StarTeam Extensions looks for this project when doing anything regarding APEs and workflow. If this project is renamed or missing, StarTeam Extensions functionality will fail.
- Java requires a minimum of 256 colors to start any AWT or Swing-based graphical user interface. When you try to start the application Server from the Start menu on a computer with too few colors, you see a DOS box blink on the screen then go away.

Use of Process Items

Marking process items as fixed, finished or completed as part of the check-in process changes the value of the internal property field named Status for the item type. It works well for users of the standard property dialogs. However, if the item type uses a custom workflow process, this can cause problems and should be avoided. For example, if you mark a change request as fixed as part of check in, the Addressed In property is automatically set to “Next Build”. However, the workflow control property for a change request is never the Status field (because of the application’s built-in workflow). The value of the workflow control property is not changed by this check-in process and the next build label can become attached to the process item erroneously.

Chapter

4

Using StarTeam Workflow Designer

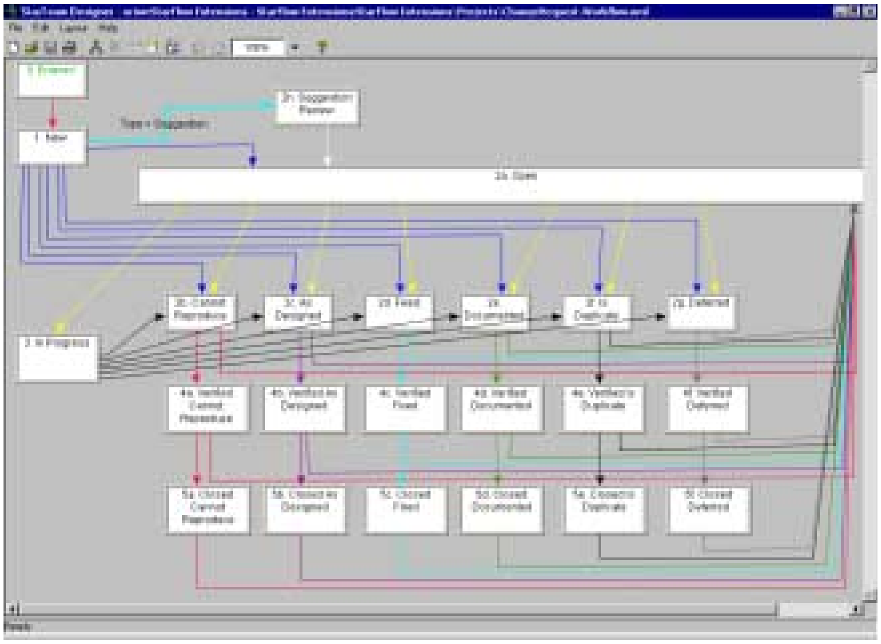
StarTeam Workflow Designer allows you to create workflows using a graphical environment in which you draw boxes that represent the steps of the workflow and connect those boxes with directed lines or arrows (called links). The steps and links represent the paths that an item can take through the workflow.

Workflow Designer meets both the simplest and most complex workflow needs. Using Workflow Designer, you can:

- Create workflow models with any number of sequenced steps.
- Create steps that must begin or be completed before a specific amount of time has elapsed.
- Create steps that redirect workflow based on certain conditions.
- Indicate how the APE will display fields for a step: what fields will be required, disabled, or hidden.
- Set the list of users or user groups to be asked to accept responsibility for an item while that item is in a specific workflow step.
- Set the list of authorized users or user groups for a workflow step. Only authorized team members can edit an item that is at this step in the workflow.
- Specify who will be notified about the status of an item and any exceptions that occur while the item is being processed.
- Define the format of email notifications.
- Create enumerated properties or add values to the workflow control property if necessary.

Note As Workflow Designer looks for server descriptions in the ServerList.st file. The Windows and Cross-Platform clients use the new starteam-servers.xml file. If you update a server description in Workflow Designer, it will not be recognized by the Windows and Cross-Platform clients and vice-versa.

The following figure shows the graphical depiction of the ChangeRequest.Workflow.xml that comes with StarTeam Extensions. The figure shows the workflow as it would appear in the Workflow Designer window when the grid is not shown.



About This Chapter

This chapter lists issues that you need to consider while planning a workflow. Then it explains:

- How to start Workflow Designer.
- The purpose of the commands on the menu bar and the buttons on the toolbar.
- How to display a workflow in the Workflow Designer workspace, set its properties, and save it.
- How to control a workflow at the step level.
- How to extend the XML for a step.

For users who are not new to the application or to the application's workflow features, this chapter provides information about updating items that were created before you used the application's workflow features so that they can be used with a workflow and Notification Agent. See "Changing Workflow Control Property Values Automatically" on page 65.

Planning a Workflow

Before you create a workflow, you should have a plan for the entire workflow. Before you modify a workflow, you should have a clear understanding of what your modification needs to do and how it will fit with what is currently in the workflow.

Walk through the workflow by creating a paper "map", noting any conditions that might change the direction of the item through the workflow and any time constraints that must be applied to items as they progress throughout the workflow.

The steps and links represent the paths that an item can take through the workflow. Both have limitations.

The next few sections cover those limitations and define the terms you need to understand as you create workflows.

General Workflow Information

The more time you spend planning a workflow, the fewer problems you will have during its execution. This section explains issues that you should be aware of during the planning phase.

Workflow Control Property

Each workflow is based on one enumerated property, called the workflow control property. You choose this property from those already available for the item type or create a new property. For example, the sample StarDraw server configuration uses the custom property Usr_StarFlowStatus in its sample workflows for change requests.

If you use an existing property, it will probably need some modifications. In StarTeam, to review the properties for an item type that are available for customization, select *item* > Advanced > Customize. The *Customize* dialog lists the properties. Some of the application's enumerated properties are more customizable than others:

- A property with the red and yellow icon is an application field. It is always an enumerated type and is fully customizable. You can add to, disable, rename, and reorder the enumerated type's values.
- A field with the white and blue icon is a restricted enumerated type. You can change only the names the application displays for the enumerated type's values. Usually these fields have built-in workflow characteristics.

In most cases you will decide to create your own enumerated property for the item. Each step in your workflow takes the name of a value in the workflow control property. Borland recommends you create the workflow first to find out what steps you need, assign names to them, and create the property to have those names as values.

Workflow Responsibility Property

You must also select one of the item type's properties to be used as the workflow responsibility property. This can be a custom property. It must be a user ID property, one that accepts user names as its values.

Notification Agent sends email messages asking users to accept responsibility for each item at each step in the workflow. Then Notification Agent sets the workflow responsibility property to the name of the user who accepts responsibility.

Multiple Workflows for the Same Item Type

Within a server configuration, all projects and views have the same properties for their item types. When you create a property for an item type, that property becomes available in all items of that type in the server configuration, regardless of where the items reside. This is what makes it easy to share and move items, but it can complicate workflow is some circumstances.

For example, suppose that you have two projects, one that uses an APE and one that uses the standard properties dialog for the item type. The standard properties dialog has a Custom tab that will show any fields you have created for the other project's workflow. If any of these properties were created as required fields, the users in the project using the standard properties dialog must provide a value for them. The users in the project with the workflow do not need to be aware of this property. It does not have to appear on the form that they use, and whether or not it was created as a required property does not affect them.

Now suppose that the two projects both use workflow for the same item type, but that the workflows are different. Suppose that the first project’s workflow uses all the values for the workflow control property, but the other workflow uses only a subset of those values. If an item is moved or shared from the first project to the second while it is in a state that is not recognized in the second project, that item is stranded in the second project until its state is changed. You would probably want that change to cause the item to branch.

If you are aware of possible pitfalls, you can eliminate or compensate for them before they occur.

Understanding Steps

Each step has properties that can be set. These properties control what happens to an item under workflow control while it in that step.

Step Names

Each step represents a value in the workflow control property. That value becomes the name of the step.

Time Limits

A step usually indicates the completion of some number of activities. For example, when a developer sets the status of a change request to fixed, team members recognize that, as a result of some defect, that developer has written code, compiled it, tested the fix in debug mode, and checked in one or more files. These activities are not usually reflected in the workflow, but they require some time to complete.

You can allot time for the activities that must be performed at a given step by setting a number of days, hours, or minutes for the step to complete. This time can be relative to the time the steps starts or the time between some previous step and this step. Exception message are sent when deadlines are missed.

Form Control

At each step, you have some control over the APE form. You can:

- Hide property fields.
- Require property fields.
- Disable property fields.
- Limit the values that are displayed in the list box for the workflow control property.

  This controls what “next steps” are available to the user editing the APE.
- Dynamically set values in fields based on the value set in another

  You set “dynamic defaults” so that, if a user selects one value in one field, one or more other fields change their values.

  For example, if you are familiar with the application’s built-in workflow for change requests, you have noticed that when you change the Status to fixed, two dynamic defaults take place:

  - The Responsibility field automatically becomes set to the name of the user who created the defect (the value of the Entered By property).
  - The value of the Addressed In field is automatically set to “Next Build”.

A dynamic default happens in real time and is triggered by some action on the user’s part. It is only a default. The user can change the values in either of these fields.

You cannot use a date/time property as the triggering field for a dynamic default.

Who’s Who

At each step, you create lists of users and groups:

- Those who will be asked to accept responsibility for the work to be done for the step.
- Those who can edit the item at this step. This should include:
  - The previous list of people or at least the one to whom the Notification Agent has assigned to the workflow responsibility property field.
  - Some administrators who need access to the item at all times.
  - Some supervisors who may not be responsible for any items but who will assist those who are.
- Those who will need to be notified when items reach a specific step.
- Those who need to be notified about exceptions.

  Exceptions include missed deadlines, the failure of any user to accept responsibility for an item at this step, and so on.

For the participants, those who are asked to accept responsibility for the step, you can tailor the step notification message’s contents. It can also be sent to users simultaneously or in a specific or random order.

Notification Agent sends status and exception messages to everyone in the lists simultaneously. You can tailor the status message, but the exceptions messages are under Notification Agent’s control.

If you do not want to notify anyone, the notification list should be empty.

Understanding Links

Links are directed lines connecting pairs of steps. They show the routing possibilities within a workflow. Links can be unconditional or conditional. When a link is unconditional, users can select the second step from the first, regardless of the settings of any other properties.

Conditional links limit the possible next steps based on the value of some property other than the workflow control property and other than a date/time property.

Keep in mind that:

- On paper, you can draw several links between any two specific steps, but in the Workflow Designer, you see only one link connecting two steps in the workspace.
- On paper you can write any number of simple and complex conditions on a link, but in Workflow Designer, only one property can appear in a conditional link.
- The conditional property cannot be the workflow control property or a date/time property.
- The conditional property can be set to one, more than one, or all the values, including the <otherwise> value with varying consequences depending on each listed value.
- All of the consequences are moves to other workflow steps.
- In most cases, when the user enters the workflow step, the conditional property should be displayed and enabled. It may even be required.

- In most cases, when the user enters the workflow step, the workflow control property must be displayed and enabled.
- The value of the conditional property dictates which values (steps) can be selected from the workflow control property's list.

  For example, suppose a link's condition is equivalent to "When Severity is High" and has the possible next steps of "1. New" and "2a. Open". Then the only values the user can select from the workflow control property's list are "1. New" and "2a. Open" when the condition is true.

Starting Workflow Designer

If you installed Workflow Designer on a Windows platform, Workflow Designer is on your Start menu: Start > Programs > StarTeam > Workflow Designer.

With consulting work, a button for Workflow Designer can be added to the custom toolbar provided with the Cross-Platform client.

Contact Professional Services at www.borland.com/services/.

Using the Menu Bar

Workflow Designer menu bar offers commands that allow you to create, modify and save workflows.

The File Menu

The options on the File menu are:

New
Creates a new workflow on the Server of your choosing. If you are not already logged on to the server configuration that will store this new workflow, you must select the server configuration and provide logon information.

Open
Opens an existing workflow file. If you are not already logged on to the server configuration that stores the workflow, you must select the server configuration and provide logon information.

Close
Closes the current workflow file.

Save
Saves the current workflow file and checks in a new revision. There is no working file. This is equivalent to a force check-in.

Save As
Saves the current workflow file, allowing you to specify a new name or location, and checks in a new revision. If the application already has a file with this name at this repository location, you are asked if you want to overwrite the existing file. If you agree, a force check-in is performed. No working file is affected.

Save As Disabled
Deletes everything in the workflow file, saving a single <Disabled/> XML tag in its place. This tag disables workflow processing. To restore the workflow to its pre-disabled condition, check out a previous revision, import it into Workflow Designer, and save it.

Why do you disable workflows? Suppose you have created a workflow and APE for change requests for an entire project. However, you want one view to have no workflow. In the folder for that one view in the StarFlow Extensions project, you would place a disabled workflow. When the application sees that the workflow has been disabled, it uses the APE without workflow.

Import
Loads a previously exported .Workflow.xml file into Workflow Designer.

Export
Saves the workflow document as a local .Workflow.xml file. Does not check in a new revision. This feature allows you to save a workflow that contains errors.

Check For Errors
Examines the workflow for errors that could prevent it from running and displays the results. Finding no errors does not mean that the workflow will run correctly, but it should run once the errors are corrected.

New Workflow Step
Allows you to draw a new step in the workspace.

Print
Prints the current workflow, reducing the diagram to fit on a single page.

Exit
Closes Workflow Designer.

The Edit Menu

The options on the Edit menu are:

Undo <edit action>
Reverses the last edit made to the current workflow.

Redo <edit action>
Restores a previously undone edit.

Normal Zoom, Zoom In, Zoom Out, Zoom To Fit, Zoom To Box
Adjusts the zoom level of the displayed workflow.

Overview
Creates a thumbnail view of the workflow’s graphical representation within the Workflow Designer window.

Workflow Properties
Displays and allows you to edit the properties of the current workflow.

Step Properties
Displays and allows you to edit the properties of the currently selected step.

Delete
Removes the current selection from the workspace.

Create All Steps
Creates a step for each of the values of the workflow control property that do not already have steps.

Select All
Selects all elements in the current workflow.

Select All Steps
Selects all steps in the current workflow.

Link Options
Allows you to specify colors for the links between steps and to display link text (the properties and values that drive the links).

By default, the link text is cleared whenever you open a workflow. You have to turn it on if you want to see the conditional text for a link. Most customers prefer to hover over the link to display the conditional text as a tool tip.

The Layout Menu

The options on the Layout menu are:

Align
Lines up two or more selected steps vertically or horizontally. Options include Top and Left.

Make Same Size
Adjusts the size of two or more selected steps. Options include Height, Width, and Both.

Show Grid
Displays a grid on which to align steps.

Snap To Grid
Automatically aligns a step on a gridline when you move the step near the line. The gridlines do not have to be displayed.

Set Grid Size
Allows you to adjust the size of the grid.

Automatic Layout
Arranges the workflow steps and links, usually in landscape orientation.

Apply FLW
Imports the layout instructions for workflows that were created in the Outlook-based versions of this product (Workflow Designer 5.2 was the first release that was not Outlook-based.)

Displaying a Workflow in the Workspace

To display a workflow in the Workflow Designer workspace, you can:

- Create a new workflow “from scratch”.
- Open a workflow from the server configuration in which it resides.
- Import a .Workflow.xml file from your workstation or a network location.

Every workflow has properties that specify:

- The type of item that the workflow controls.
- One of the item type’s properties as the field that will indicate who is currently responsible for an item. This field is called the workflow responsibility property.
- One of the item type’s properties as the workflow control property on which the workflow is based. This field is called the workflow control property because its values become the names of the steps in the workflow.
- The entry step for the workflow.

Workflow properties must be established when the workflow is created but can be changed at any time.

Caution Editing or changing the workflow control property or workflow responsibility property after a workflow is deployed when items are already under workflow control may yield undesirable results.

For more information about workflow properties, see “Viewing and Editing Workflow Properties” on page 39.

You can create custom properties for an item type or add values to an existing property using either Workflow Designer or StarTeam. See “Adding and Modifying Custom Properties” on page 61 or the customization chapter in the *StarTeam Administrator’s Guide*.

Creating a Workflow

When you create a new workflow, Workflow Designer asks you to select (and, if necessary, log on to) the server configuration that will store the new workflow. Workflow Designer displays the *Workflow Properties* dialog so you can set properties for the new workflow. After you provide the required information, Workflow Designer automatically creates the entry step in the top left corner of the Workflow Designer workspace.

To create a workflow:

1. Do one of the following:
   - Select File > New from the Workflow Designer menu bar.
   - Click the New Workflow toolbar button.
2. From the resulting *New* dialog:
   a. Select a server configuration.
   b. If the *Log On* dialog appears, log on using a valid user name and password.
   c. Click New.
3. From the resulting *New Workflow Properties* dialog:
   a. In the Item Type list, select the item type on which the workflow is based. You can create a workflow for the following item types: file, change request, requirement, task, and topic.
   b. In the Workflow Responsibility list, select the item property that indicates responsibility for a step.

      If you are creating a workflow for change requests, the default is Responsibility. You may change this, for example, if you have created a custom property to track responsibility or choose to use another user ID field.
   c. From the Workflow Control Property list, select the item type’s property that will control the workflow.

      This property is always an enumerated type. It may be a custom property or an existing application property that is customizable.

      If you are creating a workflow for change requests and have created a custom property with the internal name Usr_StarFlowStatus, the list defaults to that property’s display name (usually Workflow Status).
   d. From the Entry Step list, select the name of the enumerated value from the workflow control property that will be the name of the entry step, the step at which each item enters the workflow.

      The entry step is usually named “0. Entered”. Steps with that name define the constraints for an item as it is initially created.
   e. Click OK.

   Workflow Designer displays the workspace and the entry step. That step, represented in the workspace by a rectangle has the name of the enumerated value you selected from the Entry Step list. The name is in green. Other step’s names will be black. This makes it easy to distinguish the entry step. All links created from the entry step must be directed towards other steps. No links can be directed to the entry step. Normally, there is only one unconditional link from the entry step to the second step.

   If and when you save the new workflow, it will be checked into the selected server configuration’s StarFlow Extensions project.

Note The *New* dialog has its own toolbar:

Opening a Workflow Stored in a Server Configuration

Workflow files must be stored in the StarFlow Extensions project of any server configuration in which workflows are used. You may have several server configurations, any or all of which may have a StarFlow Extensions project, although the paths to their working folders should be different. See “Creating and Configuring the StarFlow Extensions Project” on page 7 for more information.

To open an existing workflow file in a StarFlow Extensions project:

1. Do one of the following:
   - Select File > Open from the Workflow Designer menu bar.
   - Click the Open Workflow toolbar button.
2. If you have not already logged on to a server configuration, the *Open* dialog displays a list of servers.

   If you have previously logged on to a server configuration, the *Open* dialog displays any .Workflow.xml files found in the Projects folder of the StarFlow Extension project on that server.

   If you want to select a different server configuration to log on to, you can open the Look In list, and select one from the tree it displays. You can also navigate to other projects, views, the Projects folder’s child folders, and so on.

   a. If the dialog displays a list of server configurations or if you want to log on to a different server configuration, select one.

   b. If the *Log On* dialog appears, log on using a valid user name and password.

      The *Open* dialog replaces the list of server configurations with the list of *item_type*.Workflow.xml files in the selected configuration’s StarFlow Extensions project’s Projects folder. If you had previously logged on to a server configuration during this session, the *Open* dialog showed these files initially.

   c. Locate the file to be worked on.

      You may have several project and view folders to choose from, many of which have workflow files.

   d. Click Open.

   Workflow Designer displays the workflow steps from the selected file in the workspace.

Note The *Open* dialog has its own toolbar:

If this workflow was created using a release of Workflow Designer prior to 5.2, a pile of linked steps appears in the upper left corner of the workspace. Within seconds, Workflow Designer applies its automatic layout routine, but you will probably prefer to apply an appropriate .flw file to this workflow so that it has the same look as in the old Workflow Designer.

Importing a Workflow File

You can export workflow files from one server configuration and then import them into another. You can also import a workflow file that has been checked out to your workstation or an accessible network location.

To import a workflow file:

1. Select File > Import from the menu bar.
2. From the resulting *Import Workflow Document* dialog:

   a. Locate a .Workflow.xml file.

b Click OK.

3 From the resulting *Select Server* dialog:

a Select the server configuration that will store this workflow file.

b If the *Log On* dialog appears, log on using a valid user name and password.

c Click Select.

Workflow Designer displays the workflow steps from the selected file in the workspace.

Note The *Open* dialog has its own toolbar:

If this workflow was created using a release of Workflow Designer prior to 5.2, a pile of linked steps appears in the upper left corner of the workspace. Within seconds, Workflow Designer applies its automatic layout routine, but you will probably prefer to apply an appropriate .flw file to this workflow so that it has the same look as in the old Workflow Designer

Viewing and Editing Workflow Properties

Whenever you open a workflow, Workflow Designer lets you know if the workflow properties need to be specified. Even when they are already specified, you may choose to change them.

Caution Editing or changing the workflow control property or workflow responsibility property after a workflow is deployed and there are items under workflow control may yield undesirable results.

To specify workflow properties:

1 Do one of the following:

- Select Edit > Workflow Properties from the Workflow Designer menu bar.
- Click the View/Edit Workflow Properties toolbar button.

2 From the resulting *Workflow Properties* dialog:

a From the Item Type list, select the item type on which the workflow is based. You can create a workflow for the following application item types: file, change request, requirement, task and topic.

b From the Workflow Responsibility list, select the item property that indicates responsibility for a step.

If you are creating a workflow for change requests, the default is Responsibility. You may change this, for example, if you have created a custom property to track responsibility or choose to use another user ID field.

c From the Workflow Control Property list, select the item type’s property that will control the workflow.

This property is always an enumerated type. It may be a custom property or an existing application property that is customizable.

If you are creating a workflow for change requests and have created a custom property with the internal name Usr_StarFlowStatus, the list defaults to that property’s display name (usually Workflow Status).

d From the Entry Step list, select the name of the value from the workflow control property that will be the name of the entry step, the step at which each item enters the workflow.

The entry step is usually named “0. Entered”. Steps with that name define the constraints for an item as it is initially created.

e Click OK.

Note You can create a custom enumerated property from the *Workflow Properties* dialog. You can also add values to that property or any other enumerated property that is either a custom property or a customizable application property. For more information about custom and customizable fields, see the *StarTeam Administrator's Guide*. For information about creating and modifying custom fields from Workflow Designer, see "Adding and Modifying Custom Properties" on page 61.

Creating and Linking Steps

The steps in a workflow and the links between those steps indicate all the possible paths than an item can take through a workflow.

Every step in a workflow has the same name as an enumerated value in the workflow control property. Normally every enabled enumerated value appears as a step in the workflow, but this is not enforced.

You can:

- Create steps manually by drawing them in the workspace
- Create steps automatically by listing them in the Next Steps tab of the *Edit Step* dialog
- Use standard graphical features such as aligning, resizing, and repositioning to give steps an acceptable look

You link two steps to show that items can transition from the first step to the second in the direction of the link. When an item's workflow control property is set to the value represented by the first step, the user who is editing the item can select the value represented by the second step. For example, suppose the workflow control property contains steps named "1. New" and "2a. Open". If you link the step "1. New" to the step "2a. Open", then when an item has the value "1. New", the user can select "2a. Open" as the next value.

You can:

- Create links from an entry step to other steps. You cannot reenter the step designated as the entry step.
- Create links between any two non-entry steps in either direction.
- Change the step to which the link points.
- Delete the link.
- Reposition the link so that you can more easily see its path in the workspace. This includes adjusting the location where the ends of the link contact the steps as well as adding link points to bend the link as it travels from one step to the other.
- Apply a color to the link.

There are no alignment and sizing features available for links.

Creating Steps

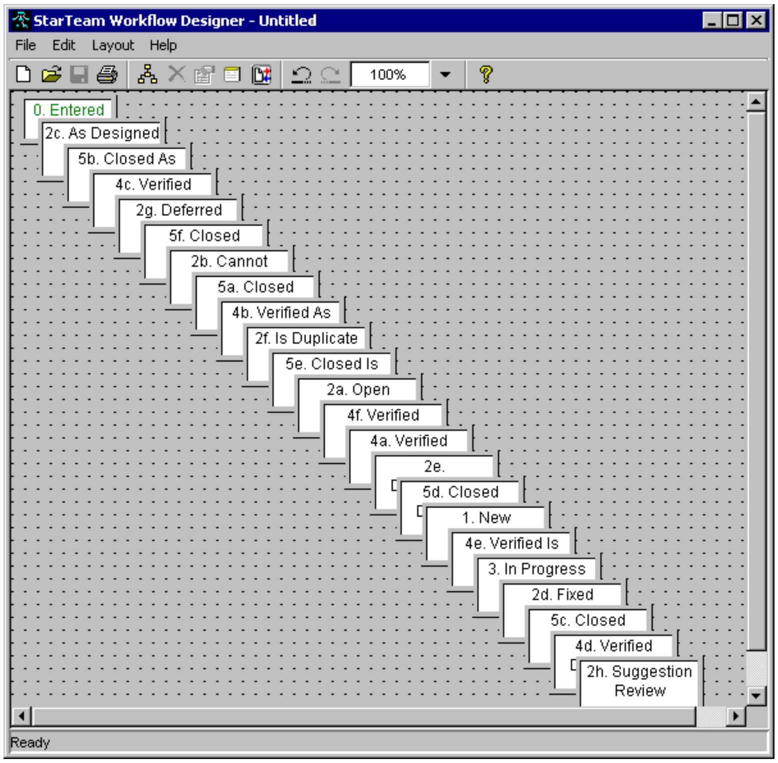
You can draw steps manually in the workspace. Borland recommends drawing all of your steps manually as you start your workflow. You can position the steps in an appropriate manner. This makes it easier to create links, either manually or automatically.

Workflow Designer draws the entry step for you when you create a new workflow. The rest of the steps can be created automatically all at one or manually step-by-step.

To create all the steps at once:

1 Select Edit > Create All Steps from the menu bar.

Workflow Designer draws one step for each of the values in the workflow control property as a cascade.



To create one step at a time:

1 In the workspace, do one of the following:

- Select File > New Workflow Step from the menu bar.
- Click the Add New Workflow Step toolbar button.

The mouse pointer changes to a cross-hair.

2 Using the mouse, click and drag in the workspace to define a rectangle. When you release the mouse button, the *New Step* dialog opens.

3 From the *New Step* dialog:

a Use the Step Name list to select the name of one of the workflow control property’s enumerated values as the name of this step.

b Click OK.

A new step with the selected name appears in the workspace. You can resize and align the step at any time.

You can also use the Add Value button to add values to the workflow control property from this dialog. See “Adding a Value to an Enumerated Property” on page 62.

You can copy a previously created step—except for its links (the information on the Next Steps tab).

To copy a step:

1 Right-click the step.

2 Select Copy Step from the right-click menu.

3 Position the mouse where you want the new step to be located and right-click.

4 Select Paste Step from the right-click menu.

5 Add links (Next Steps) out of this step as appropriate.

6 Be sure to create at least one link into this step from another step as well.

When you copy a step, you can paste what you copied just once. To paste the step a second time, you must recopy it.

Linking Two Steps Manually

When you draw links manually in the workspace with your mouse, you can start and end the link at a variety of locations on any edges of the steps.

Manually adding a link from one step to another automatically affects the information in the Next Steps tab of the *Edit Step* dialog for the step being transitioned from. The name of the step being transitioned to is added to the list of steps for the <otherwise> value.

The <otherwise> value specifies the steps to which the item is linked (and, therefore, the values to which the item can transition) when no conditions exist or no conditions are met.

When the link between two steps is created solely because the next step is available only as a setting for <otherwise>, the link is not dotted. That transition is unconditional, even if there are conditional links also originating from the first of the two steps.

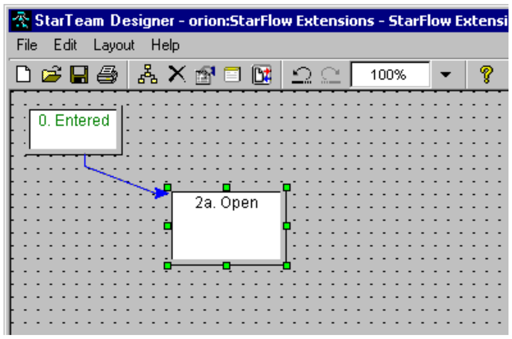
To manually create a link from one step to another:

- In the workspace, do one of the following:
  - Click the edge of a step. This immediately links the selected step with the step closest to it—unless that step is the entry step.
  - Click-and-drag a link from the edge of one step to the edge of the next. When you release the mouse button, the link appears. Again, you cannot drag a link to the entry step.

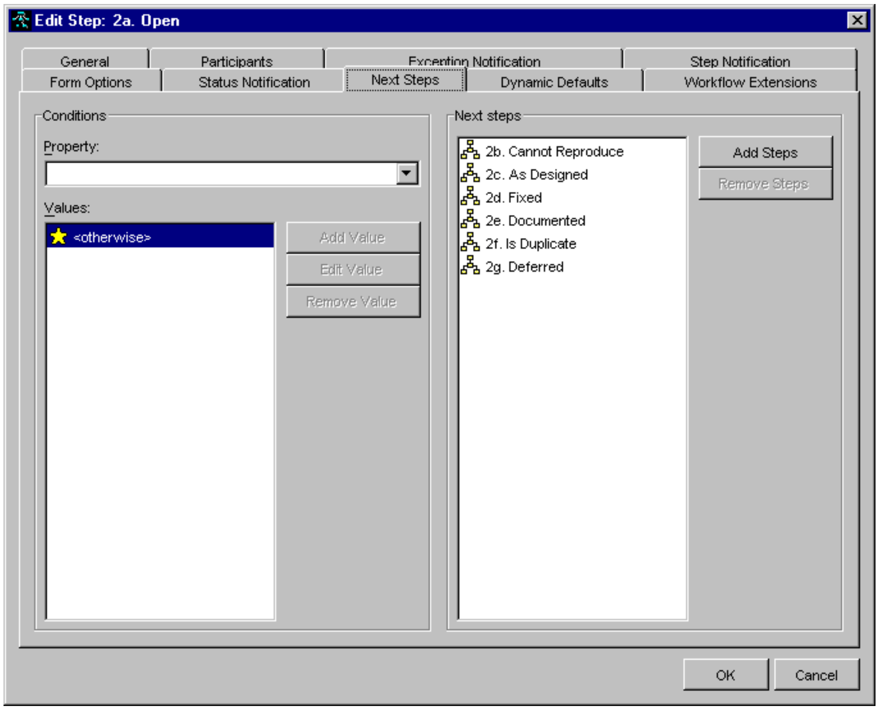
Creating and Linking Steps Automatically

After you select a step, you can automatically create all the links from it to other steps using the Next Step tab of the *Edit Step* dialog. If the connected steps have not already been drawn, this procedure draws them as well.

The following figure shows what the workspace might have looked like prior to opening the *Edit Step* dialog.


StarTeam Designer - orion:StarFlow Extensions - StarFlow Extensi
File
Edit
Layout
Help
100%
0. Entered
2a. Open

The next figure shows an example of a workflow in which the “2a. Open” step will be automatically linked unconditionally to several other steps.



Conditions

Property:

Values:

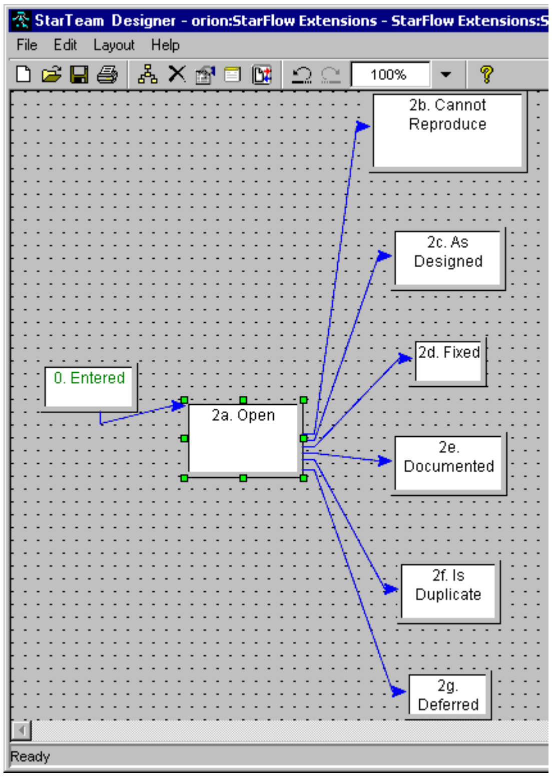
<otherwise>

Next steps

- 2b. Cannot Reproduce
- 2c. As Designed
- 2d. Fixed
- 2e. Documented
- 2f. Is Duplicate
- 2g. Deferred

When you click OK to close the *Edit Step* dialog, the new steps and/or links appear at the top left corner of the workspace. Use the mouse to drag the steps to more appropriate positions, or use the Layout > Automatic Layout command to arrange the steps.

The next figure shows these steps and links after automatic layout has been applied to the workspace.



If the steps were already drawn manually, but you defined links using the Next Step tab of the *Edit Step* dialog, Workflow Designer draws the arrow from the first step's left edge to the second step's right edge, regardless of their locations in the workspace.

This procedure is not recommended initially because the rearranging may be as much work as drawing the steps and links manually, but it can be used to make a few changes automatically afterwards.

Conditional links can be created only by using the Next Steps tab of the *Edit Step* dialog. See "Conditionally Linking Steps" on page 45 for details.

To create and link steps automatically:

1 Do one of the following to select the step from which the new steps will be linked, and open the *Edit Step* dialog:

- Double-click an existing step.
- Select an existing step and then select the View/Edit Step Properties toolbar button.
- Right-click inside an existing step and select Step Properties from the context menu.
- Select an existing step and then select Edit > Step Properties from the menu bar.

2 From the resulting *Edit Step* dialog, click the Next Steps tab.

3 Select <otherwise> in the Conditions column to specify links for this step that do not have conditions.

4 Click Add Steps.

5 From the resulting *Select Next Steps* dialog:

a Select the steps to *transition to* from the list of values.

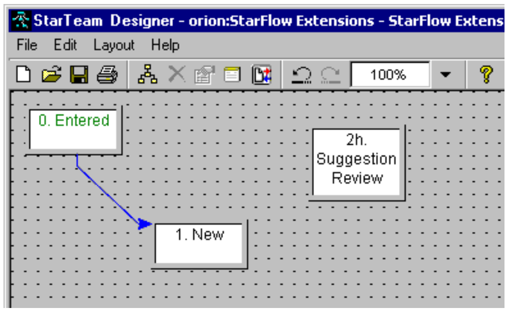
b Click OK.

Tip If you change your mind about some of the steps selected, highlight the unwanted steps and click Remove Steps.

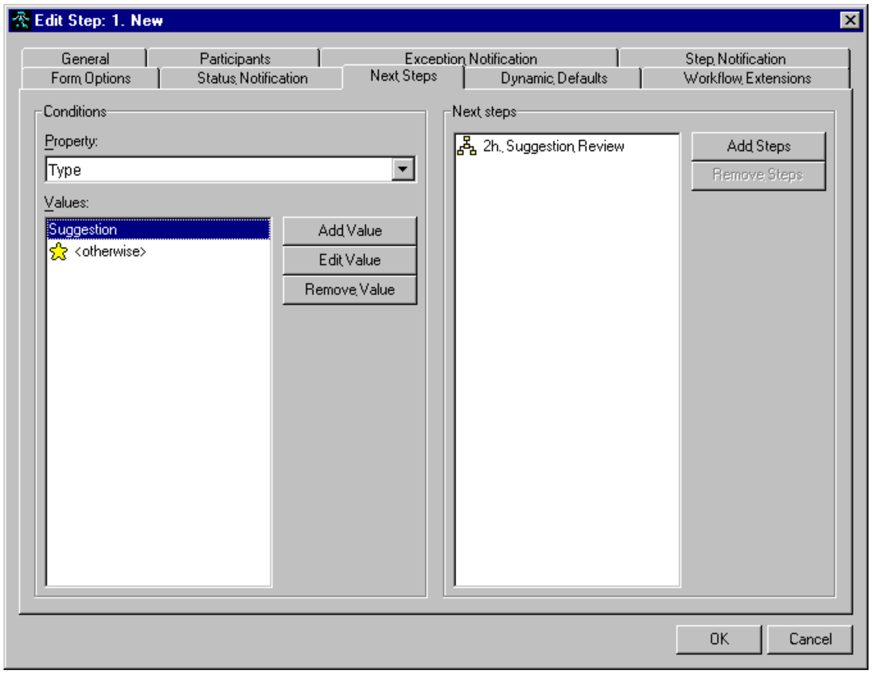
Conditionally Linking Steps

A link can indicate that a transition from one step (or enumerated value) to another is valid only when an item property other than the workflow control property has specific values. For example, suppose you have two steps named “1. New” and “2h. Suggestion Review”. Suppose that the user can select “2h. Suggestion Review” as the next value for an item’s workflow control property only when the Type property has the value “Suggestion”. To show this in the workflow, you connect the two steps with a conditional link. A conditional link is a dotted arrow pointing to the step that is available only when the condition is true.

The following figure shows the workspace with the “0. Entered”, “1. New”, and “2h. Suggestion Review” steps, but no link has yet been created between the “1. New” and “2h. Suggestion Review” steps.


StarTeam Designer - orion:StarFlow Extensions - StarFlow Extens
File
Edit
Layout
Help
100%
0. Entered
2h. Suggestion Review
1. New

The next figure shows how this conditional transition would be displayed in the Next Steps tab of the *Edit Step* dialog. You must use this dialog to create conditional links. They cannot be created manually within the workspace.



Conditions

Property: Type

Values:
Suggestion
<otherwise>

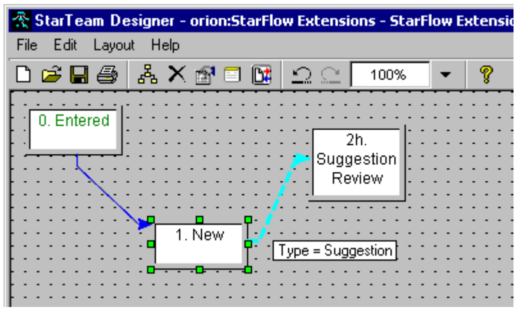
Next steps
2h. Suggestion Review

When you specify a condition, you select a property and one or more values. For each value, you select one or more steps that are appropriate transitions for an item whose property is set to that value—an item that meets that condition. In this example, the conditional property is Type. When the value of that property is Suggestion, the user can select “2h. Suggestion Review” as the next value of the workflow control property.

If the steps and links have not already been drawn manually, Workflow Designer creates a step for each step listed in the Next Steps list. It also creates links from the selected steps to those steps. The steps and links are drawn automatically and layered in the upper left corner of the workspace. Use Automatic Layout to reposition the steps or drag each step to an appropriate location using the mouse.

Hovering over the link displays these values. They can also be displayed as text in the workspace. As you open existing workflows, this text is on by default.

The next figure shows these steps and links after the mouse has hovered over the new conditional link long enough to display the conditional information.



If the steps were already drawn manually, but not the links, Workflow Designer draws the links. In each case, the link starts from the first step’s left edge and ends at the second step’s right edge, regardless of the locations of these steps in the workspace.

If the links were already drawn, there already are steps listed as possible transitions for <otherwise>. If you add a condition, they remain in place but they become conditional links as well as <otherwise> links. Only one link is ever drawn between two steps, although the link may have more than one value (including <otherwise>) for the conditional property.

If all the possible values for the conditional property appear in the Next Steps tab, the <otherwise> condition may be left blank. Borland does not recommend this because if an additional value is added to the conditional property, an item with that new value as its setting becomes stranded. It has no next step to go to.

To conditionally link steps:

1. In the workspace, double-click an existing step.
2. From the resulting *Edit Step* dialog, click the Next Steps tab.
3. From the Properties list, select the property on which to base the condition (for example, the Type property).
4. Click Add Value.
5. From the resulting *Select a Value* dialog:

   a. Do one of the following:

      - Select a value for the condition (for example, Suggestion), then click OK. The value appears in the Values list and is selected.
      - Select a property from the Value of Property list so that the condition becomes dependent upon the value of that property.

        For example, suppose the property in the Properties list is *Entered By*. You can select another property of the same type, such as *Locked By* from the

Value of Property list. Your condition is equivalent to “When the value of these two properties is the same.”

b Click OK to return to the Next Steps tab of the *Edit Step* dialog.

6 Click Add Steps.

7 From the resulting *Select Next Steps* dialog:

a Select one or more next steps.

b Click OK.

8 (Optional.) Repeat steps 4 through 7 of this procedure to select additional values of the specified conditional property and the steps available for transition when the conditional property has those values.

9 Select <otherwise> to specify the workflow steps to which the item is linked (and, therefore, the values of the workflow control property to which the user can transition) when no conditions exist or no conditions are met. Repeat steps 6 and 7 of this procedure to select those workflow steps.

10 (Optional) Use the Edit Value, Remove Value, and Remove Steps buttons to make corrections as needed.

11 When you click OK to close the *Edit Step* dialog, any steps that were specified as next steps and were not yet drawn appear in the upper left corner of the Designer window. Note that the link appearance is different when the link is conditional.

When Editing an Item

Suppose that you are editing an item at “1. New” step and the item’s Severity is High. Further suppose that, in the workflow, the step “1. New” has a conditional link to the step “2a. Open” and another conditional link to the step “2b. Emergency”. The condition attached to the link between “1. New” and “2b. Emergency” is equivalent to “When Severity is High”. The condition attached to the link between “1. New” and “2a. Open” is equivalent to “When Severity is Low”. Then the only value you can choose for the workflow control property at this step is “2b. Emergency”.

If you change the value of Severity to Low, the APE changes the value (step) that can be selected from the workflow control property’s list to “2a. Open”.

Editing a Link

You can edit the links for a step by double-clicking one of the links that originates from that step.

To edit a link:

- Double-click the link.

  Workflow Designer displays the *Edit Step* dialog for the step from which the link originates. The dialog is open to the Next Step tab so you can easily change the link’s destination step or the conditions required for a transition to that or other destination steps.

Displaying Link Conditions as Text

You can display a link’s conditions as text in the workspace. When you do not display the text in the workspace, it appears as a tip when you hover the mouse pointer over a link. Whenever you open a workflow, the link text is turned off. You must turn it on or use the tooltip.

To display or hide the conditions in the workspace:

- Select Edit > Link Options > Display Link Text.

Aligning Steps

You can align two or more selected steps horizontally or vertically.

To align steps:

1. In the workspace, select the step that the other steps must match.
2. Select one or more additional steps.

   The selection marks for the additional steps are a different color: light blue instead of green.
3. Select Layout > Align > Top for vertical alignment, or select Layout > Align > Left for horizontal alignment.

Resizing Steps

You can resize steps by dragging an edge or corner of the steps. You can make one step's size the same as another's in height, width, or both using menu commands.

To resize steps:

1. In the workspace, select the step that the other steps must match.
2. Select one or more additional steps.

   The selection marks for the additional steps are a different color: light blue instead of green.
3. Select one of the following:
   - Layout > Make Same Size > Height
   - Layout > Make Same Size > Width
   - Layout > Make Same Size > Both

Repositioning Links

When selected, each link has at least two link points. The initial two link points mark each end of the link.

To reposition a link in the workspace:

- Drag the link point nearest the arrow to reposition it on the current step.
- Drag the link point nearest the arrow to change the step to which the link points.
- Drag the link point farthest from the arrow to a new position on the same step only. You cannot change the step from which the link is drawn.
- Add bends and turns to the link by adding link points and dragging them to appropriate spots in the workspace.

To add a link point:

1. In the workspace, right-click the location on the link where you want to create a new link point.
2. From the resulting right-click menu, select Add Link Point.

   A new link point appears on the link.

Changing the Default Link Colors

Links are color-coded, one color for links with no conditions and another color for links with conditions. By default, Workflow Designer displays unconditional links in dark blue and conditional links in cyan. Conditional links are dotted lines. You can easily change the default colors. You cannot change the default line style.

To change the default link color:

1. Choose Edit > Link Options > Default Link Colors from the menu. A color selection dialog opens.
2. Select the new default link color by selecting a color swatch or entering RGB or HSB values, then click OK.

To change the default conditional link color:

1. Choose Edit > Link Options > Default Conditional Link Colors from the menu. A color selection dialog opens.
2. Choose the new default link color by selecting a color swatch or entering RGB or HSB values, then click OK.

Changing the Color of Individual Links

In complicated workflows, too many overlapping links can make it difficult to follow a link from its source to its destination. You can change the color of individual links within a workflow to make it easier to see where the link lines start and end.

To change the color of an individual link:

1. In the workspace, point to the link for which you want to change the color and right-click to display the context menu.
2. From the resulting right-click menu, select one of the following:
   - Select Edit Link Color > Use Default to return to the default color for the type of link.
   - Select Edit Link Color > Customize. From the resulting color selection dialog, select the new color for the link line by selecting a color swatch or entering RGB or HSB values, then click OK.

Controlling the Contents of the APE Form

When an APE is created, it is a composite of all the fields that will be needed to perform any of the steps in the workflow.

You can specify:

- What properties can be viewed/edited from the APE by the user modifying an item that is at a specific step.
- What values can be listed for the workflow control property using the Next Steps. For an explanation of this feature, see “Conditionally Linking Steps” on page 45.
- Dynamic changes to the item’s properties that occur while the user modifies the item at that step.

Controlling What Properties the Form Displays

Use the Form Options tab of the *Edit Step* dialog to specify which property fields, if any, are required, disabled, or hidden on the item's custom form at a given step in the workflow process. These field attributes are mutually exclusive; each field may be assigned only a single attribute per step.

By default, none of the fields are required, disabled, or hidden. Click the column header to sort the fields based on that column header. For example, clicking Required results in listing required fields first, then the fields that are not required. Use *Shift+click* to reverse the sort.

Be aware that as soon as a user selects a new value for the workflow control property, the on-form fields change based on the required, disabled, and hidden settings for the newly selected status. This happens regardless of whether the user is authorized to edit an item with the new status. Authorized participation is enforced only when a user opens an item.

Important The Form Options tab lists every possible property for the item type. The APE form usually does not display all these properties. Be careful not to designate a field as required that is not available on the APE form.

To specify required, disabled, or hidden fields:

1. In the workspace, double-click an existing step.
2. From the resulting *Edit Step* dialog, click the Form Options tab.
3. In the Field Name column, locate the name of the property field that needs to have its attributes changed. This column lists all of the fields that an item of the selected type can have. Ignore fields that are not on the form.
4. Do one of the following:
    - Select the Required check box to designate the field as *required* at this step in the process. If your APE form is based on a sample form that comes with the product, required fields probably appear in red (although the color can be changed programatically).
    - Select the Disabled check box to prevent input to this field at this step in the workflow. Disabled fields are usually dimmed in appearance.
    - Select the Hidden check box to stop the APE form from displaying the field at this step in the process. The area that the field occupies on the form is left blank.
5. Click OK or another tab.

Controlling Dynamic Changes to Forms

Use the Dynamic Defaults tab of the *Edit Step* dialog to set the value of one property based on the value the user selects for another property. For example, using dynamic defaults, if the user sets the Severity of a change request to High, the value of Responsibility can be set to Jack automatically.

For each property in the property list, you can place one or more of its values in the When Equals Value list. The value <otherwise> substitutes for any values not explicitly specified in the When Equals Values list.

For each value in the list, you can specify one or more other properties and the values to which they should be set. For example, suppose you add Severity to the Property list. While it is selected, you add High and Low to the When Equals Value List. While High is selected, you add Responsibility/Jack as the property/value pair to the Assign Values To Other Properties list; while Low is selected, you add Responsibility/Jill and a Short Comment “As time permits” to the Assign Values To Other Properties list.

The result is that when a user sets the Severity set to High, the Responsibility property is automatically set to Jack; when the user sets the Severity to Low, the responsibility is automatically set to Jill and the Short Comment property becomes “As time permits”.

A common use of dynamic defaults is to set a revision comment for the revision created at each step. For example, when the workflow control property for a change request is set to “1. New”, the comment can be set to “Status: 1. New”. Users can then see the steps in the history pane when they select the item in the application and review the revision comments.

The user can change the settings created by dynamic defaults. The idea is that, normally, these are changes that the user would be making and, therefore, setting them dynamically saves the user time.

You cannot use a date/time property as the triggering field for a dynamic default.

To set properties based on the user’s changes to other properties:

1. In the workspace, double-click an existing step.
2. From the resulting *Edit Step* dialog, click the Dynamic Defaults tab.
3. Select a property that the user would be expected to change at this step:
   a. Click the Add button beneath the Property list.
   b. From the resulting *Select Property* dialog, select the property.
   c. Click OK to return to the Dynamic Defaults tab.
   d. Select the property from the Property list. The <otherwise> value appears in the When Equals Value list.
4. Select a value to which the user might set the selected property.
   a. Click the Add button beneath the When Equals Value list.
   b. From the resulting *Select Condition Value* dialog (which can vary depending on the type of the selected property), select one of the following:
      - A specific value for this property from the Value list.

        For example, if the property is Severity, you might select High. If the user changes Severity to High, dynamic changes will occur.
      - Another property of the same type from the Value Of Property list. If the user changes the selected property so that it matches the value of this property, dynamic changes will occur.

        For example, if the selected property is Severity, an enumerated field, only enumerated properties appear in the Value Of Property list. If the user selects High for the Severity property and this property, for example, the Priority property, is also set to High, then the dynamic changes will occur. These same changes would also occur if the both the Severity and Priority properties were set to Low.
      - The current user (available for user ID properties).
      - None (available for enumerated types that have an empty value).
   c. Click OK to return to the Dynamic Defaults tab. The selected value or the property to be matched appears in the When Value Equals list.
5. Specify a dynamic change:
   a. Select the value in the When Value Equals list.
   b. Click the Add button beneath the Assign Values To Other Properties list.
   c. From the resulting *Select Property and Value to Assign* dialog, select a property (such as Responsibility) to be changed and the value (such as Jack) to which it should be set.

This dialog varies depending on the type of the property. The property types are User ID, Date/Time and Enumerated.

d Click OK to return to the Dynamic Defaults tab. The property to be changed and its value appear in the Property and Value columns of the Assign Values To Other Properties list.

6 Optional) Repeat step 5 for other dynamic changes to be made if the property selected from the Property list has the value selected from the When Value Equals list.

7 (Optional) Repeat steps 4 through 5 for other values of the property selected from the Property list.

8 (Optional) Repeat steps 3 through 5 for other properties.

9 Click OK or another tab.

Tip Use the Edit and Remove buttons to change and delete properties and values.

Controlling the Behavior of an Item in a Step

Workflow Designer allows you to control the behavior of an item at a particular step. You can specify:

- Who will be asked to accept responsibility for an item that has reached a particular step.
- Who will be allowed to modify that item. For example, while everyone who is asked to accept responsibility for an item must be allowed to edit it, you might want additional personnel, such as administrators or team leaders, to be able to edit that item at this step.
- How long a user will have to accept responsibility for an item at a given step. For example, if this step is a critical one and the first user does not accept responsibility in a specified amount of time, another user will be asked to accept responsibility.
- How long a user will have to perform the tasks necessary at a given step before editing the item. For example, source code may need to be modified before a change request can be marked fixed.
- Who monitors the progress of an item as it reaches a particular step.
- Who is notified when exceptions occur in the workflow process. For example, an item at a particular step may exceed the time allowed for the step to complete or all participants may decline to accept responsibility for the item at that step.

You specify these settings using Workflow Designer, but Notification Agent must be running to enforce your notification settings in real time.

Listing the Participants

Each step has its own set of participants. For example, someone in development may have to address a change request during the implementation step while someone in QA may have to address that same change request during the testing step. You identify the team members who are allowed to modify the item at a given step in the workflow. You also determine whether they are asked simultaneously or one-at-a-time to accept responsibility for a particular item as it reaches a particular step.

The Participants tab on the *Edit Step* dialog has two lists. The Step Notification List specifies team members who will receive notification that the workflow has transitioned to the current step. Members of these lists receive the message defined on the Step Notification tab or a default message from Notification Agent if no message is provided. See “Using StarTeam Notification Agent” on page 93.

The Authorized List specifies team members who are authorized to modify an item at this step in the workflow. For example, suppose you are authorized to edit change requests with the Open status but not with the Fixed status. If you select an Open change request and change its status to Fixed, you can still click OK because the authorization for Fixed does not go into effect until a new revision is created with that status. After you click OK, you cannot reopen the change request because you are not authorized to modify an item that is fixed.

For proper operation of the workflow, everyone on the Step Notification List must also be on the Authorized List. In addition, you might want to add some supervisory team members to the Authorized List. For example, a QA team leader might not be assigned items to test, but might need to assist other testers as they perform their jobs. Having the ability to edit items during the testing steps may make both persons' jobs easier.

To specify who can edit an item at a particular step:

1. In the workspace, double-click an existing step.
2. From the resulting *Edit Step* dialog, click the Participants tab.
3. Select one of the following types of notification from the Notification Type list.
   - Blast instructs the Notification Agent to notify everyone on the list at once. If a step is critical and must be done in a very short time period, you might prefer that everyone be notified immediately.

     If Notification Agent will be sending a responsibility notification message that requires no response, the Blast option is recommended. Otherwise, after the amount of time set for users to accept responsibility passes, the next participant will receive the message. This goes on until all participants receive the message, and then an exception notification is sent.
   - Ordered, the default, instructs the Notification Agent to notify users one-at-a-time in the order they appear in the list. By default, users in a group are listed alphabetically by their logon names. If certain users are better equipped to perform this step than others, you might want the users asked to accept responsibility for an item at this step to be listed in a different order.
   - Random instructs the Notification Agent to randomize the list and then notify users one-at-a-time in the randomized order. If anyone in the list can perform this step, you may not want the first person is the list to be bombarded with requests while a user at the bottom of the list is rarely asked to perform this step. If this is the case, random comes closest to balancing each user's workload.
4. Before you create the Step Notification List, select the Use Same Names As Step Notification List check box.

   Important: This ensures that all the users and groups in the Step Notification List also appear in the Authorized List.
5. Add any number of individual users, user groups, and references to the Step Notification List. Specifying a reference adds to the list one or more user ID fields. The user whose ID appears in the referenced field at the time an item enters this step will be added to the list. These users and groups are from the server configuration.

   To add one or more users:

   a. Click the Add Users button next to the Step Notification List to add users to the list.

   b. From the resulting *Select Notified Users* dialog, select one or more users.

   c. Click OK to return to the Participants tab. The selected users appear in both the Step Notification List and the Authorized List.

   To add one or more user groups:

d Click the Add Groups button next to the Step Notification List to add a group to the list.

e From the resulting *Select Notified Groups* dialog, select one or more groups.

f Click OK to return to the Participants tab. The selected groups appear in both the Step Notification List and the Authorized List.

To add one or more users by reference:

g Click the Add References button next to the Step Notification List to add users to the list.

h From the resulting *Select Referenced Fields* dialog, select one or more user ID fields.

i Click OK to return to the Participants tab. The referenced user ID fields appear in both the Step Notification List and the Authorized List.

6 (Optional) If you want to add additional users or groups to the Authorized List so that those users can also edit items at this step, clear the Use Same Names As Step Notification List check box.

The users, groups and references currently in the Authorized List remain there, but the Add Users, Add Groups, and Add References buttons beside the list become enabled.

- Use these buttons to add any number of additional individual users, user groups, and references to the Authorized List. It is common to add a reference to the responsibility field to ensure that anyone who has been assigned responsibility to the item at this step, for whatever reason, can edit the item.

7 Click OK or another tab.

Tips You can use drag-and-drop to rearrange users, groups and references in the Step Notification List if you are using Ordered notification. Use the Remove buttons to delete users, groups or references from either list.

Specifying a Notification Message and Interval

Use the Step Notification tab to:

- Specify the message that asks users to accept responsibility for a given step
- Limit the amount of time a user has to accept that responsibility
- Indicate what importance should be attached to the message

Note Setting intervals can result in time limits that end during the night, over the weekend, or on holidays. That means that an exception message may be sent if no one accepts responsibility for an item when no one was at work to make such an acceptance.

Setting a 24-hour or greater interval should allow the users’ decisions to accept or decline responsibility to control when the next user is notified in most cases. Exception notifications would usually occur only when no one accepted responsibility.

When notification is ordered or random, if a participant sends an “I decline...” response or does not respond within the interval set on this tab, the message is sent to the next name on the Step Notification List.

If no participant accepts responsibility for the item, Notification Agent generates an exception notification message.

To specify a notification message and the interval at which it is sent to team members listed in the Step Notification List:

1 In the workspace, double-click an existing step.

2 From the resulting *Edit Step* dialog, click the Step Notification tab.

3 From the Set The Importance To list, select Low, Normal (the default), or High. This determines what importance a mail message has attached to it. For example, a message of high importance would have an exclamation point in front of it in an Outlook inbox.

4 Specify the time interval between notification messages in minutes, hours, or days using the Notification Interval text and list boxes. The default is one hour.

Notification Agent uses this interval both to determine when to send the next message and, finally, when everyone has been contacted, to determine that an exception message needs to be sent because no participant has accepted responsibility for a specific item at this step. Setting this interval is important even when the message is "blasted" simultaneously to everyone, because at the end of the interval, an exception message is sent if no one has accepted responsibility.

5 (Optional) If you prefer not to use the default step notification message subject line, enter a subject for the email notification message. Leave the Subject text box blank to use the default subject line for step notification messages (see "Default Notification Messages" on page 125).

6 (Optional) If you prefer not to use the default step notification message body, specify the body of the notification message.

The body of the message may be in plain text or HTML format. If you do not specify the message body, Notification Agent sends the default body for step notification messages (see "Notification Tags and Messages" on page 117).

This message is sent to team members listed on the Participants tab in the Step Notification List.

7 (Optional) Add the current values of specific item property fields to the subject line or message body using the appropriate Insert Reference button.

a Click Insert Reference.

b From the resulting *Select a Reference* dialog, select one or more property field names.

c Click OK to return to the Step Notification tab.

For each property field you selected, a reference appears in the subject line or message body. For example, if you selected the Entered By field, the reference token {{property name="EnteredBy"}} appears. (EnteredBy is the internal name for the Entered By property.) When the message is sent, the token is replaced with the name of the team member who entered the item.

8 (Optional) Click Preview HTML to see how the body of your message would appear in a browser window.

9 Click OK or another tab.

Understanding Notification List Operation

Notification Agent is responsible for processing the Step Notification List and sending notification messages appropriately. Names on the notification list are processed as follows:

- Leaving the notification list blank creates a silent step; no notification is sent. This is useful when:
  - The workflow contains a number of consecutive steps for which the same user is responsible.
  - There is only one user who performs this step and that user prefers not to get email whenever an item enters this step. For example, the user may perform the

tasks associated with this step only once a month and prefers to use a query in the application to locate the items that need to be processed.

- Entering a single name on the notification list results in the notification of:
  - The one user who always completes this step.
  - A single user who manually assigns responsibility for the step to one of several appropriate users in order to balance their workloads. Note that all these appropriate users must be entered on the Authorized List for that step.
- Entering two or more individual names or a group name on the Step Notification list instructs Notification Agent to send messages as specified in the Notification Type field. If Blast is selected, the system notifies everyone on the participants list at once. If either Random or Ordered are specified, the system sends notification messages to one individual at a time. If a notified user accepts the task, no other messages are sent. If a notified user declines a task or does not respond before the notification interval expires, a notification message is sent to the next name on the list.

  Users may manually assume responsibility for a task after the notification interval has expired as long as no other user has already accepted the task. Users may not, however, use the automated accept/decline links within a notification message after the interval has expired; the system informs the user that the message was tardy and discards these messages without processing them.
- If there is a mixture of users and groups in the list, Notification Agent resolves duplicate names so that users are notified only once, the first time they appear on the list.

  For example:
  - Ellen, John, Mary, and Scott are in Engineering
  - Ellen, Joan, and Tom are in Management
  - Elizabeth and Scott are in Marketing

  The table below illustrates notification resolution for a mixed list that is set to Ordered and uses the above names and groups.

Step Notification List	Resolution
Mary	A single message is sent to Mary. If she accepts, the notification process is over; no additional messages are sent. If she declines or fails to respond before the notification interval expires, the agent moves on to the next entry in the list.
Engineering	Ellen, John, Mary, and Scott are in Engineering. Mary has already received a message, so messages are sent *sequentially* to the remaining members of the group, namely Ellen, John, and Scott. Each team member is given the length of the notification interval (set on the Step Notification tab of the *Edit Step* dialog) to respond before the next message is sent. Only the *active* team member, that is, the one with the most recent notification message, may respond using the automatic links within the notification message. If no one in Engineering accepts before their notification interval expires, Notification Agent moves on to the next entry in the list.
Tom	A single message is sent to Tom. If Tom declines or does not reply before the notification interval expires, the agent moves on to the next entry in the list.

Step Notification List	Resolution
Marketing	Elizabeth and Scott are in Marketing. Scott has already received notification as a member of Engineering, so a single message is sent to Elizabeth. If Elizabeth declines or does not reply before the notification interval expires, the agent moves on to the next entry in the list.
Management	Ellen, Joan, and Tom are in Management. Ellen has already received notification as a member of Engineering and Tom has already received notification individually, so a single message is sent to Joan. If Joan declines or does not reply before the notification interval expires, the agent generates an exception, following the instructions on the Exceptions tab.

When a user accepts responsibility for an item at a particular step, Notification Agent sets the workflow responsibility property to that user's name.

Limiting the Time Allowed to Complete the Task

When you create a timed step, you specify a duration for the step and a base step for the duration. The base step for the duration may be the step for which the duration is specified or any preceding step. For example, if you want to specify that a step be accomplished in 24 hours, you set a duration of either 24 hours or 1 day for the step and base the duration on nothing or on the step itself. If, however, you require that step 3 end within 3 days after step 0 started, you would set a duration of 72 hours or 3 days on step 3, based on step 0.

If a step is not accomplished within the specified duration, an exception occurs. The Exceptions tab of the *Edit Step* dialog allows you to specify recipients for exception messages that are generated at this step. Exception messages are generated by Notification Agent and cannot be changed.

Note Setting durations can cause problems—unless your organization operates 24-7—because the duration time continues to be calculated overnight, during the weekend, and on holidays. That means that steps will expire and exception messages will be sent earlier than anticipated. For example, if on a Friday, a step has 24 more hours before it expires, exception messages may be sent on Saturday.

To create a timed step:

1. In the workspace, double-click an existing step.
2. From the resulting *Edit Step* dialog, click the General tab.
3. In the Step Duration group, select the Limit Step Duration check box.
4. Specify a duration for the step by typing an integer value in the Duration text box and selecting days, hours, or minutes from the list.
5. From the Relative To list, select the step on which the duration is based.
   - To specify a duration for the step from its beginning to its end, set Relative To to the current step or leave it blank.

     Setting a duration *relative to* nothing (blank) or the current step specifies that the current step must be performed in the specified time, independent of other steps in the process.
   - To specify a duration for a task that spans two or more steps, set Relative To to the first step in the task.

     Setting a duration *relative to* a preceding step specifies that the duration begins when the relative step begins and ends when the current step ends, no matter how many steps intervene.

     The Relative To list contains the names of all the steps in the workspace. Be sure to select a step that precedes the current step in the workflow.

6 Click OK or another tab.

Monitoring an Item’s Status

Use the Status Notification tab on the *Edit Step* dialog to specify recipients and a message for notification of an item’s transition to this step in the process. Status notification messages are informational in nature and recipients are usually team members other than those listed on the Participants tab for the step. For example, Customer Support supervisors might need to be aware that a change request has reached an escalation step.

Notification Agent sends one message to all the users in the status notification list simultaneously.

To create a status notification list and message:

1 In the workspace, double-click an existing step.

2 From the resulting *Edit Step* dialog, click the Status Notification tab.

3 Add any number of individual users, user groups, and references to the Status Notification List. Specifying a reference adds to the list one or more user ID fields. The user whose ID appears in the referenced field at the time an item enters this step will be added to the list. These users and groups are from the server configuration.

   a To add all users simultaneously:

      1 Click the All Users button to add all users to the list.

      2 Click OK to confirm this decision because it will empty any current entries in the Status Notification List.

      3 The word Everyone appear in the Status Notification List. There is no need to add individual users, groups, or references.

   b To add one or more users:

      1 Click the Add Users button next to the Status Notification List to add users to the list.

      2 From the resulting *Select Notified Users* dialog, select one or more users.

      3 Click OK to return to the Status Notification tab. The selected users appear in the Status Notification List.

   c To add one or more user groups:

      1 Click the Add Groups button to add groups to the list.

      2 From the resulting *Select Notified Groups* dialog, select one or more groups.

      3 Click OK to return to the Status Notification tab. The selected groups appear in the Status Notification List.

   d To add one or more users by reference:

      1 Click the Add References button to add users to the list.

      2 From the resulting *Select Referenced Fields* dialog, select one or more user ID fields.

      3 Click OK to return to the Status Notification tab. The referenced user ID fields appear in the Status Notification List.

   Use the Remove button to delete users, groups or references from the list.

4 (Optional) If you prefer not to use the default status notification message subject line, enter a subject for the email notification message. Leave the Subject text box

blank to send a message with the default subject line for status notification messages (see “Default Notification Messages” on page 125).

5 (Optional) If you prefer not to use the default status notification message body, specify the body of the notification message.

The body of the message may be in plain text or HTML format. If you do not specify the message body, Notification Agent sends the default body for status notification messages (see “Default Notification Messages” on page 125).

6 (Optional) Add the current values of specific item property fields to the subject line or message body using the appropriate Insert Reference button.

a Click Insert Reference.

b From the resulting *Select a Reference* dialog, select one or more property field names.

c Click OK to return to the Status Notification tab.

For each property field you selected, a reference appears in the subject line or message body. For example, if you selected the Entered By field, the reference token {{property name="EnteredBy"}} appears. (EnteredBy is the internal name for the Entered By property.) When the message is sent, the token is replaced with the name of the team member who entered the item.

7 (Optional) Click Preview HTML to see how the body of your message would appear in a browser window.

8 Click OK or another tab.

Managing Exceptions

Use the Exception Notification tab on the *Edit Step* dialog to specify the recipients for notification messages to be sent if:

- The current step does not reach completion as expected (for example, if a timed step is not finished within the specified duration).
- No one accepts responsibility for an item at a particular step.

The exception messages are generated by Notification Agent based on the above and on system errors. These messages vary according to the exception that occurred, and cannot be modified.

Notification Agent sends one message to all the users in the exception notification list simultaneously. See “Default Notification Messages” on page 125 for more information.

To create an exception notification list:

1 In the workspace, double-click an existing step.

2 From the resulting *Edit Step* dialog, click the Exception Notification tab.

3 Add any number of individual users, user groups, and references to the Exception Notification List. Specifying a reference adds to the list one or more user ID fields. The user whose ID appears in the referenced field at the time an item enters this step will be added to the list. These users and groups are from the server configuration.

a To add all users simultaneously:

1 Click the All Users button to add all users to the list.

2 Click OK to confirm this decision because it will empty any current entries in the Exception Notification List.

3 The word Everyone appear in the Exception Notification List. There is no need to add individual users, groups, or references.

- b To add one or more users:
    1. Click the Add Users button next to the Exception Notification List to add users to the list.
    2. From the resulting *Select Notified Users* dialog, select one or more users.
    3. Click OK to return to the Exception Notification tab. The selected users appear in the Exception Notification List.
- c To add one or more user groups:
    1. Click the Add Groups button to add groups to the list.
    2. From the resulting *Select Notified Groups* dialog, select one or more groups.
    3. Click OK to return to the Exception Notification tab. The selected groups appear in the Exception Notification List.
- d To add one or more users by reference:
    1. Click the Add References button to add users to the list.
    2. From the resulting *Select Referenced Fields* dialog, select one or more user ID fields.
    3. Click OK to return to the Exception Notification tab. The referenced user ID fields appear in the Exception Notification List.

4 Click OK or another tab.

Tip Use the Remove button to delete users, groups or references from the list.

Extending the Workflow's XML

The Workflow Extensions tab allows you or Borland consultants to extend the workflow with complex logic for any of the following:

- The entire workflow using the Workflow Extensions tab in the *Workflow Properties* dialog
- A particular step using the Workflow Extensions tab in the *Edit Step* dialog

For example, you might add features that are not directly supported by Workflow Designer. This XML is inserted into the XML output for the workflow. With additional coding, the new XML elements can be used by StarTeam Extensions features and the custom APE forms.

Any XML tags you add will not conflict with those being used by the application. Your tags are scoped within the Workflow Extensions XML element so there is no chance of a conflict.

To add XML elements: for a step

1. Do one of the following:
    - For a step: In the workspace, double-click an existing step.
    - For the entire workflow: Select Edit > Workflow Properties.
2. From the resulting *Workflow Properties* or *Edit Step* dialog, click the Workflow Extensions tab.
3. Type the new elements in the text box.
4. Click OK to check the new XML for errors.

Adding and Modifying Custom Properties

Often the workflow control property is a custom property that has been designed to fit your organization's process. You can create and modify custom properties from Workflow Designer if you have the correct access rights.

Note It is easier to create custom properties when you are using the application because they can be edited while you create them. See the *StarTeam Administrator's Guide* for more information about creating custom property fields.

Adding a custom property field or making changes to an existing property field can cause confusion for users who do not use APEs. For example, APEs and your workflows control what property fields users see and what workflow property values (next steps) they can select, but users of the standard property dialogs see every field and every possible value for that field.

The initial default that you set for a new workflow control property is very important if you have existing items to be put under workflow control. For example, if you have been using the application, but without its workflow features, you may have hundreds of items that have not previously been under workflow control. As you create a new property, these existing items acquire the initial default value for that new property. See "Changing Workflow Control Property Values Automatically" on page 65 for more information.

You can create a custom enumerated property from the *New Workflow Properties* dialog and the *Workflow Properties* dialog. These dialogs appear as part of the process of creating a new workflow or editing an existing workflow's properties.

You can add a value to an enumerated property from the following dialogs:

- *New Workflow Properties* dialog (displayed by selecting File > New).
- *Workflow Properties* dialog (displayed by selecting Edit > Workflow Properties).
- *New Step* dialog (displayed by selecting File > New Workflow Step).
- *Edit Step* dialog's General tab (displayed by double-clicking a step in the workspace).

Creating a New Enumerated Property

After you have diagrammed a new workflow, you may need to add a enumerated property to the item type. Usually, the new property is the workflow control property. For convenience, Workflow Designer allows you to create it from within the designer.

Caution Before you click OK, be sure that the field name and codes you select are the ones that you want. They cannot be changed after you exit this dialog. The display name and the values can be changed from StarTeam, but not from Workflow Designer.

To create an enumerated custom field:

1. From the *New Workflow Properties* dialog or the *Workflow Properties* dialog, click Add Property.
2. In the resulting *Add Enumerated Property* dialog, enter the name for the user-defined field to be used by the database in the Field Name text box. Use only alphanumeric characters and no spaces in this name.

   The name should be fewer than 31 ASCII characters (including the Usr_) and not contain these following characters that are not accepted by one or more of the databases that the application supports:
   `= \\.^$@,;!:#*&<>?-//%|[](())+"`
3. Type the name that StarTeam will display to users in the Display Name text box.

4 Click Add to enter the first value for this enumerated type. The *Add Value* dialog appears. The application reserves the numeric codes from 0 to 99, so this dialog shows numbers starting with 100.

5 Use the displayed code or enter another in the Code text box.

6 Type the name for this enumerated value in the Name text box.

7 Click OK.

8 Repeat Steps 4-7 to add additional values.

Tips If you create an enumerated code or value incorrectly, click Edit to repair or Delete to remove that value from the list. When you are finished adding values, you can use drag-and-drop to rearrange the values in the Possible Values list box.

9 Select one of the enumerated values as a default from the Default Value list box.

This value automatically becomes the value for this field in all existing items for which this field is a property. It does not become the default value for newly created items—unless they are created when APEs are disabled or an APE is used with a disabled workflow. See "Changing Workflow Control Property Values Automatically" on page 65 for more information about the importance of the initial default setting for this new field.

10 Click OK.

Adding a Value to an Enumerated Property

Occasionally, you need to add a value to an enumerated property. Usually, the new value is an addition to the workflow control property. For convenience, Workflow Designer allows you to add that value from within the designer.

To add an enumerated value to a custom field:

1 From the *Workflow Properties* dialog, *New Step* dialog or *Edit Step* dialog's General tab, click Add Value.

2 In the resulting *Add Enumerated Value* dialog, click Add. The *Add Value* dialog appears.

3 Use the displayed code or enter another in the Code text box.

4 Type the name for this enumerated value in the Name text box.

5 Click OK.

Caution Before you click OK, be sure that the code and value are correct. You cannot change them from Workflow Designer after you exit this dialog. The value can be changed from the application.

Changing the Name of the Step

Use the General tab to change the name of the step, that is, to change the value of the workflow control property that the step represents. For example, if you have just added a new value to the workflow control property, you might want change the name of an existing step to match the new property name.

To change a workflow step's name:

1 In the workspace, double-click an existing step.

2 From the resulting *Edit Step* dialog, click the General tab.

3 From the Step Name list, select another value as the name of this step.

4 Click OK or another tab.

Checking for Errors

When you open a workflow that contains errors or when you select File > Check For Errors from the menu bar, Workflow Designer checks for any and all errors that could prevent the workflow from executing. First it checks that all of the workflow properties are set and valid. Then it checks each step to verify that all properties are valid. A message box lists the errors and red questions marks are used to guide you to the problems in specific property dialogs.

For example, if you import a workflow that specifies a workflow control property that does not exist for the current server configuration, it places a red question mark on the Form Options tab of the *Edit Step* dialog.

Specific fields display question marks when they are in error in the *Workflow Properties* dialog. The *Edit Step* dialog displays question marks on individual tabs as well as specific fields.

Saving and Exporting Workflows

Saving a workflow to the Projects folder or one of its subfolders in the server configuration’s StarFlow Extensions project is equivalent to deploying that workflow. Borland recommends that you save workflows to a Test folder in a Test project until it is ready for general use.

Saving the workflow to the Projects folder or one of its subfolders makes that workflow available for deployment, so long as:

- The name of its parent folder is the name of an application project or view
- The path from the Projects folder to the parent folder represents the project/view hierarchy within the server configuration.

Making a workflow available for deployment can provide workflow steps or stop the use of the workflow. If the workflow was disabled via Workflow Designer, then the APE will appear but no workflow will be enforced. Using both enabled and disabled workflows makes it possible to turn on workflow for an entire project but stop it for specific views.

Exporting a workflow allows you to use it in another server configuration. This feature also allows you to save a workflow that contains errors.

For more information about deploying workflows, see “Deploying the Workflow” on page 91.

Saving a Workflow

When you save a workflow for deployment, it is output as *item_type*.Workflow.xml and checked in to the server configuration you select. You must store it in that configuration’s StarFlow Extensions project. The default location for workflows is the Projects folder, but you can select another folder.

No working file is created for a .Workflow.xml file. The file is checked directly into the application repository.

To save a new revision of an existing workflow:

1. Do one of the following:
   - Select File > Save from the menu bar.
   - Click the Save Workflow toolbar button.

   No *Save* dialog appears as the location of this workflow was specified when you opened it.

2 In the resulting *Revision Comment* dialog, enter a comment for this workflow that will appear in the new revision's comment field.

3 Click OK. This causes a forced check-in.

To save a workflow to a new location:

1 Select File > Save As from the menu bar.

2 In the resulting *Save As* dialog:

   a Type a name for this workflow.

   b Select an appropriate folder to store it in.

3 Click OK.

4 In the resulting *Revision Comment* dialog, enter a description for this workflow.

   Your description/comment appears in both the description and comment fields of the saved revision if this revision is the first revision of a new file.

   It appears only in the comment field of the saved revision of this workflow file is not a new file.

5 Click OK again.

   If a workflow with that name already exists in that location, you are asked whether the existing file can be overwritten. Clicking yes causes a forced check-in.

Disabling a Workflow

When you disable a workflow, it is emptied of all XML tags except for a single <Disabled/> tag. A disabled workflow turns off the workflow for a specific view when workflow has been enabled for the project in which the view resides and when the disabled workflow is checked in to that view's folder in the StarFlow Extensions project.

If you reopen a disabled workflow, you must confirm that you want to create a new workflow in this now empty .Workflow.xml file.

To create a disabled workflow:

1 Select File > Save As Disabled from the menu bar.

2 Click Yes when a message asks you to confirm.

3 In the resulting *Save As* dialog:

   a Type a name for this workflow.

   b Select an appropriate folder to store it in.

4 Click OK.

5 In the resulting *Revision Comment* dialog, enter a description for this workflow.

   Your description/comment appears in both the description and comment fields of the saved revision if this revision is the first revision of a new file.

   It appears only in the comment field of the saved revision of this workflow file is not a new file.

6 Click OK again.

   If a workflow with that name already exists in that location, you are asked whether the existing file can be overwritten. Clicking yes causes a forced check-in.

Exporting a Workflow

You can export a file to use with another server configuration. This feature also allows you to save a workflow that contains errors.

To export a workflow:

1 Select File > Export.

2 In the resulting *Export Workflow Document* dialog, select a file name and location.

3 Click OK.

Changing Workflow Control Property Values Automatically

You can convert items, such as change requests, that existed prior to your use of the application's custom workflow features for use with the new workflow and Notification Agent by writing a StarTeam SDK script or having the Professional Services Organization from Borland create a script for you. Using change requests as an example, such a script would determine the value of the Status property for each change request and, based on that value, assign an appropriate value to the new workflow control property.

Chapter

5

Walking Through a Workflow

Workflow Designer provides a graphical environment in which to create and modify workflows. Workflow Designer allows you to quickly and easily create complex customized workflows for your organization.

This chapter walks you through the creation of a workflow for change requests. The example illustrates a workflow that deals with information captured from a feedback form on a company's web site.

It is written primarily for new users of the application who have no items that were created prior to the use of workflows and Notification Agent. If you are adding the use of these features or upgrading from a release of StarTeam Extensions, Workflow Designer, and Notification Agent prior to 5.2, see "Changing Workflow Control Property Values Automatically" on page 65, "Upgrading StarTeam Extensions" on page 131, and "Differences in the New and Old Designers" on page 135.

Workflow Roadmap

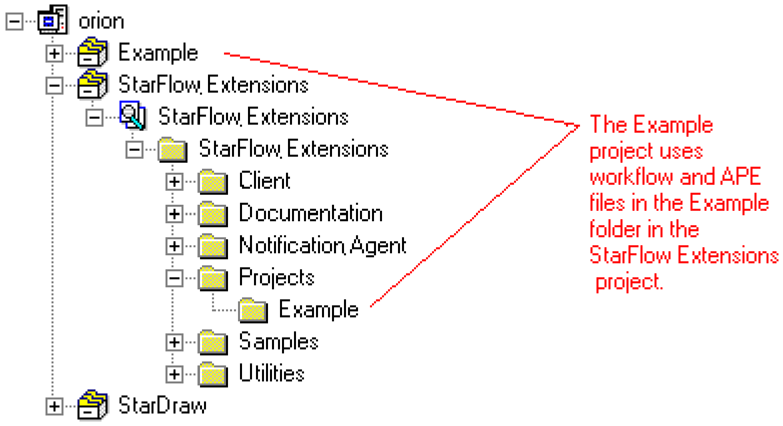
For this example, it is assumed that you have installed or have access to an application Server, a server configuration with a StarFlow Extensions project, StarTeam client, Workflow Designer, and Notification Agent.

1. Create the necessary StarTeam projects and folders.
2. Set up appropriate users and groups for the Server.
3. Plan your new workflow.
4. Create a custom enumerated property and values on which the workflow is based.
5. Plan and create your custom APE form.
6. Create a custom workflow using Workflow Designer to specify the steps and actions of the workflow.
7. Test the workflow to ensure that it performs as intended:
   - Use the Test button in JBuilder to test the form with the workflow.
   - Use the Run Test button in Notification Agent to check the notifications.

Setting Up Projects and Files

When you store custom forms and workflows in the application repository, they must be stored in specific projects and folders. The Locator locates these projects and folders and uses the appropriate files to apply your customizations to the appropriate items.

In this example, you create an Example project and an Example folder in the StarFlow Extensions project to store the Example project's workflow and APE files.



The Example project uses workflow and APE files in the Example folder in the StarFlow Extensions project.

To create a StarTeam project named Example:

1 Use the new project wizard to create the Example project in your server configuration.

To create an Example folder in the StarFlow Extensions project:

1 Open the StarFlow Extensions project.

2 Create an Example folder as a child of the Projects folder. You will store the custom forms and workflows to be used in the Example project here.

The project in which the workflows and APE forms are to be used and the folder in which they are stored must have the same name.

Setting Up Users and Groups

For this example, you will need some specific group and user names. You can create users of your choice to populate the new groups; however, all of them must have a resolvable email address. If you need more details about creating users and groups, see the *StarTeam Administrator's Guide*.

Since you probably have only one email address at your disposal, all of these users will need to have your email address. You will be able to identify who has been notified because their full names will appear in the email.

Create the following users and groups in User Manager:

1 From StarTeam or its Administration Tool, open the *User Manager* dialog.

2 Create the following groups and users:

Groups:

Customer Support group
Marketing group
Product Maintenance group
Product Management group

Research and Development group
Testing group

Users with the following full names:

Put these users in the groups that match their names.

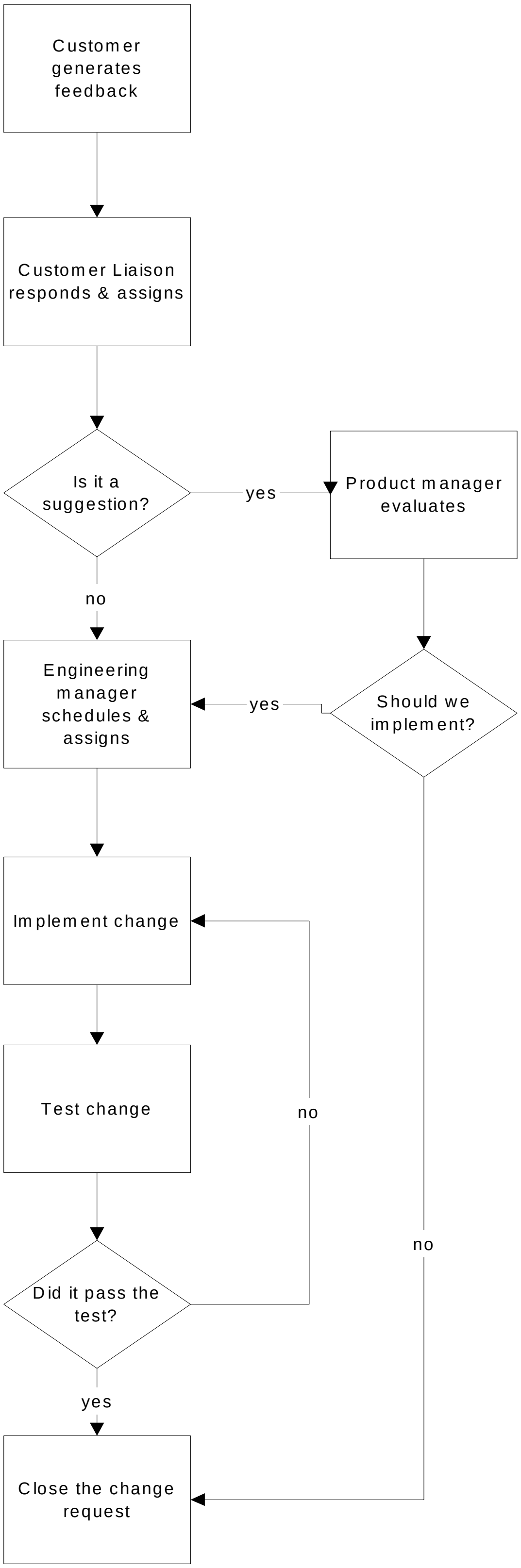
Customer Support Person 1
Marketing Person 1
Product Maintenance Person 1
Product Management Person 1
Research and Development Person 1
Testing Person 1
Testing Person 2

Additional users with the following full names:

Do *not* put these users in any group other than All Users.

Customer Liaison
EM
General Manager
Notification Agent
PM
QA Manager

Understanding the Example Use Case


Customer generates feedback
Customer Liaison responds & assigns
Is it a suggestion?
yes
Product manager evaluates
no
Engineering manager schedules & assigns
yes
Should we implement?
Implement change
Test change
no
no
Did it pass the test?
yes
Close the change request

The example illustrates a workflow to deal with information captured from a feedback form on a company's web site.

- When feedback comes in, team leaders must be notified that new feedback has entered the system.
- If the feedback is a suggestion, it is forwarded to the product manager (PM) for review. The PM decides whether to implement the suggestion.
  - If yes, the feedback is forwarded to the engineering manager (EM) for assignment.
- If the feedback is to report a defect in the product, it is forwarded to the EM for assignment.
- The EM assigns the feedback to an engineer. If the engineer does not acknowledge the assignment within 24 hours, the feedback must be assigned to a different engineer.
- When the engineer is finished with the assignment, the EM is notified and the product change is sent to quality assurance (QA) for testing.
  - If the change does not pass testing, it is sent back to the engineer.
  - If the enhancement passes QA testing, the customer, engineering, and marketing are notified and the change request is closed.

The example workflow process controls a change request (CR) item and has several possible steps:

1. **New.** The customer generates feedback, which is entered into the system. The team must be notified that a CR has entered the system. The CR is assigned to the Customer Liaison for review. The Customer Liaison responds to the customer and evaluates the CR.
2. **PM Evaluation.** If the CR is a suggestion for product enhancement, it is forwarded to the PM for review. If the PM decides to implement the suggestion, the CR is sent to the EM for scheduling and assignment. Otherwise, the CR is closed.
3. **EM Processing.** All CRs that are to be acted upon must go to the EM for scheduling and assignment.
4. **Implementation.** This step has two possible solutions.
   a. Product enhancements proceed to the research and development (R and D) group for implementation
   b. Defect reports go to Product Maintenance to be fixed.
5. **Testing.** All changes to the product must be tested by the QA department. If the change does not pass testing, it's returned to the engineer for more work; otherwise, the CR is closed.
6. **Closed.** There are three possible states for closed CRs.
   a. If the CR is a suggestion that has not been implemented, the team is notified that the CR is closed and the customer is sent an appropriate thank you note.
   b. If the CR is a suggestion that has been implemented, the team is notified that the CR is closed, the customer is sent an appropriate thank you note, and Marketing is notified of the product's new feature.
   c. If the CR is a defect that is fixed, the team is notified that the CR is closed, Customer Support is notified that the problem has been fixed, and the customer is sent an appropriate thank you note.
7. **Customer Letter.** After the Customer Liaison has sent the appropriate thank you letters, the item's workflow control property (Workflow Status) is set to Customer Letter.

In addition to these steps, a special entry step must be created for use during the creating of new items. This step is usually named “0. Entered”.

For newly created items, the entry step controls what user sees and can do. In this example, all new change requests will automatically start with a Workflow Status of “1. New” because that is what the “0. Entered” step will dictate.

Deciding Which Properties to Use

Before starting a new workflow, you need to decide which properties will be required to process change requests using your workflow.

Directly tied to the workflow are a workflow control property and a workflow responsibility property.

- The workflow control property has your workflow steps as its values.

  Workflow control properties are always enumerated properties. This property can be created from the application or as part of the process of creating a new workflow in Workflow Designer.

- The workflow responsibility property contains user IDs and allows the work to progress from person to person through the workflow.

The Status property for change requests most closely matches our workflow steps. However, that property controls the application’s built-in workflow for change requests, and cannot be used for a custom workflow. A sample change request APE uses Usr_StarFlowStatus, which has the display name Workflow Status. For this example, we will use a similar property. It will have the same name but different values.

The Responsibility property for change requests most closely fits the needs of the workflow responsibility property, and it is not part of the application’s built-in workflow for change requests. It is the property to use.

Note When you create a workflow for the change request item type, the sample APEs that come with StarTeam Extensions expect the workflow control property to be a custom property named Usr_StarFlowStatus. You can use other properties for change requests, but it requires additional APEs or additional customization of a sample APE.

Numbering the Example Steps

To customize the change request item in the application, add a custom change request property with the name Usr_StarFlowStatus, which has the following values:

0. Entered

1. New

2. PM Evaluation

3. EM Processing

4a.R and D Implementation

4b.Maintenance Fix

5. Testing

6a.Not implemented

6b.Feature Implemented

6c.Fix Complete

7. Customer Letter

8. Customer Letter Sent

While numbering the values that drive the process in this way is not necessary, it improves the readability and maintainability of the workflows you design.

Creating a Custom Property to be the Workflow Control Property

The workflow control property must be an enumerated property. In this example, it is a custom property which is substituting for the Status property.

You can create a custom property in StarTeam or in Workflow Designer. You are more likely to create the property from StarTeam because it offers more support for creating custom properties.

Important The initial default that you set for a new workflow control property is very important if you have existing items to be put under workflow control. For example, if you have been using the application, but without its workflow features, you may have hundreds of items that have not previously been under workflow control. As you create a new property, these existing items acquire the initial default value for that new property. See “Changing Workflow Control Property Values Automatically” on page 65 for more information.

To create a workflow control property:

1. In StarTeam, open a project in the server configuration.
2. Select the Change Request tab.
3. Select Change Request > Advanced > Customize.

   The *Customize* dialog lists the fields that can be customized:
4. Click Add. The *Add Field* dialog appears.
5. Type StarFlowStatus to create a custom field with the name Usr_StarFlowStatus.
6. Type “Workflow Status”, the name that StarTeam and Workflow Designer will display to users in the Display Name text box.
7. Click Add to enter the first value for this enumerated type. The *Add Value* dialog appears. The application reserves the numeric codes from 0 to 99, so this dialog shows numbers starting with 100.

8. Use the displayed code.
9. Type “0. Entered” in the Name text box.
10. Click OK to return to the *Add Field* dialog.
11. Repeat steps 3–6 using the following code and name combinations:

110	1. New
120	2. PM Evaluation
130	3. EM Processing
140	4a. R and D Implementation
150	4b. Maintenance Fix
160	5. Testing
170	6a. Not Implemented
180	6b. Feature Implemented
190	6c. Fix Complete
200	7. Customer Letter
210	8. Customer Letter Sent

12. Select “0. Entered” as a default from the Default Value list box.

    This value automatically becomes the value for this field in all existing items for which this field is a property. It does not become the default value for newly created items—unless they are created when APEs are disabled or an APE is used with a disabled workflow.

13. Click OK.

Caution Before you click OK, be sure that the field name and code numbers are the ones that you want. They cannot be changed after you exit this dialog. Later, you can change only the display name and the names of the enumerated values.

Creating the Custom APE Form

This section provides an overview of the procedure for creating the form used in the example workflow. Before you begin, verify that JBuilder (or the Java development environment of your choice) is installed and properly configured to work with the application. See “Configuring JBuilder to Use StarTeam JavaBeans” on page 16 for more information on using JBuilder to customize application forms.

To create a form:

1. Copy the files located in C:\Program Files\Borland\ StarFlow Extensions\ Samples\ ChangeRequest folder on your workstation to the working folder for the StarFlow Extensions\Projects\Example folder (for example, C:\config_name\StarFlow Extensions\Projects\Example).
2. Start JBuilder and open the ChangeRequest.jpx JBuilder project file now located in the Example folder. View ChangeRequestForm.java (double-click the file name in the pane on the upper left of the window).
3. Click the Design tab at the bottom of the central pane. A sample Change Request form appears in the Design window.
4. Experiment with adding one property that is not already there.
5. Choose File > Save from the menu and save the form with the same name (ChangeRequestForm.java)
6. Create the .jar file by selecting the Project > Make *project_name* command using the menu bar.
7. If you are using JBuilder Professional Edition, run the Make*project_name*jar.bat file to complete the creation of the .jar file.

8 Check in the new form .jar and .propertyeditor.xml to the StarFlow Extensions\Projects\Example folder.

Creating a Workflow

Now that you have the form, groups, users, and properties that you need, you are ready to create the workflow.

To create the workflow:

1 Start the Workflow Designer by choosing Start > Programs > StarTeam > Workflow Designer from the Start menu.

2 In the Workflow Designer window, choose File > New from the menu bar.

3 From the resulting *New* dialog:

  a Select a server configuration.

  b From the *Log On* dialog, log on using a valid user name and password.

  c Click New.

4 From the resulting *New Workflow Properties* dialog:

  a From the Item Type list, select ChangeRequest.

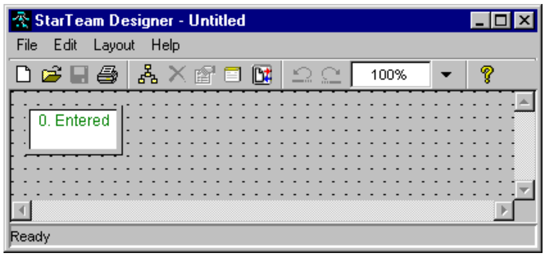
  b From the Workflow Responsibility list, select Responsibility, which is the default.

  c In the Workflow Control Property list, select “Workflow Status”, which is the default.

5 From the Entry Step list, select “0. Entered”.

6 Click OK.

The “0. Entered” step has been drawn for you in the upper left corner of the workspace, but you must set properties for that step.



Creating the Steps and Links

In the next few sections, you create the steps of your workflow and link them. The following table summarizes the step properties for the “0. Entered” step.

Because the entry step is numbered and named appropriately, the item will be automatically forwarded to the next step in the workflow by the workflow engine. No participant list or notification list is necessary. In general, you will only set form options and specify the next step for the entry step. You may also set dynamic defaults in the entry step.

Drawing the “1. New” Step

You can draw steps manually in the workspace. In the following procedure, you draw the “1. New” step.

To create a step:

1 In the workspace, do one of the following:

- Choose File > New Workflow Step from the menu bar.
- Click the Add New Workflow Step toolbar button.

The mouse pointer changes to a cross-hair.

2 Using the mouse, click and drag in the workspace to define a rectangle. When you release the mouse button, the *New Step* dialog opens.

3 From the *New Step* dialog:

a Use the Step Name list to select the name of one of the workflow control property's enumerated values as the name of this step. In this case, you select "1. New".

b Click OK.

A new step with the name "1. New" appears in the workspace. You can resize and align the step at any time.

Tips To draw a step for each value in the workflow control property all at once, use Edit > Create All Steps. The steps form a cascade in your workspace.

You can also copy an existing step (except for its links) by right-clicking the step to display a right-click menu and selecting Copy Step. To place the new step, right-click a location in the workspace and select Paste Step from the right-click menu.

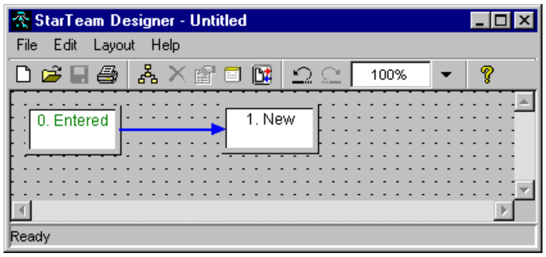
Linking the Two Steps Manually

When you draw links manually in the workspace with your mouse, you can start and end the link at a variety of locations on any edges of the steps.

Manually adding a link from one step to another automatically affects the information in the Next Steps tab of the *Edit Step* dialog for the step being transitioned from. The name of the step being transitioned to is added to the list of steps for the <otherwise> value.

The <otherwise> value specifies the steps to which the item is linked (and, therefore, the values to which the item can transition) when no conditions exist or no conditions are met.

When only an <otherwise> value exists in the Next Steps tab, the link is not dotted because the link is not conditional.



To manually create a link from one step to another in the workspace, do one of the following:

- Click the edge of the "0. Entered" step. This immediately links the selected step with the step closest to it—unless that step is the entry step. In this case, it creates a link to "1. New".
- Click-and-drag a link from the edge of the "0. Entered" step to the edge of "1. New". When you release the mouse button, the link appears. Again, you cannot drag a link to the entry step.

Editing the “0. Entered” Step

The entry step is the point at which the item to be controlled enters the workflow. Each workflow must contain an entry step; and the entry step should be named “0. Entered” to be processed properly by the workflow engine. The name of this (and any other) step is one of the possible values for the workflow control property.

The entry step is a special *pass-through* step. That is, as a user creates an item, the APE uses the step properties for the “0. Entered” status to control what the user sees and can do.

Entry steps can have no links directed to them—only from them. In this case, you will construct your workflow to provide a single link out of the entry step.

Because this is an unconditional link from “0. Entered” to “1. New”, you can draw it manually. Creating the link manually automatically sets the <otherwise> value to have the next step of “1. New”.

In this example, there is only one possible way for the workflow item to move off of the entry step (that is, only one next step).

Table 5.1 0. Entered (Summary)

Field	Value/Notes
General Tab	
Step Name	0. Entered
Limit Step Duration	cleared
Participants Tab	
To	<leave blank>
Exception Notification Tab	
To	<leave blank>
Step Notifications Tab	
To	<leave blank>
Form Options Tab	
Field Name	Type
Setting	Required
Field Name	Severity
Setting	Required
Status Notification Tab	
To	<leave blank>
Next Steps Tab*	
Condition	<otherwise>
Next Step	1.New
Dynamic Defaults Tab	
Property	Severity
When equals value	High
Assign Property: Value:	Priority Yes
Property	Severity
When equals value	<otherwise>
Assign Property: Value:	Priority: No
Property	Workflow Status
When equals value	1. New
Assign Property: Value:	Short Comment Status is 1. New
Workflow Extensions Tab	
Workflow Extensions	<leave blank>

To edit the step:

1 Double-click in the interior of the step. The *Edit Step* dialog opens.

2 Click the Form Options tab and set the Severity and Type fields to Required by activating the check box in the Required column for each field.

3 Click the Next Steps tab and set the <otherwise> condition to step *1.New*.

   a Select <otherwise> in the Values list.

   b Click Add Steps. The *Select Next Steps* dialog opens.

   c Select “1. New” and click OK.

4 Click the Dynamic Defaults tab and specify that if the Severity property is set to High, then the value of the Priority property should be set to Yes.

   a In the Property list, click Add. The *Select Property* dialog opens.

   b Select Severity from the list of properties and click OK.

   c Select Severity in the Property column.

   d In the When Equals Value column, click Add. The *Select Condition Value* dialog opens.

   e Select High as the value and click OK.

   f Select High in the When Equals Value list.

   g In the Assign Values To Other Properties column, click Add. The *Select Property and Value to Assign* dialog opens.

   h Select Priority as the property and Yes as the value, then click OK.

5 Click OK to close the dialog when you are done.

Notice that the “1. New” step and the link between “0. Entered” and “1. New” have been drawn for you. You could have drawn the two steps and linked them by clicking the edge of the “0. Entered” step and dragging the link to the “1. New” step because the two are linked unconditionally. However, conditional links must be created in the Next Steps tab of the *Edit Step* dialog.

Editing the “1. New” Step

Step “1. New” is always processed by someone in the product management group. The user who accepts responsibility at this step must set the Type to either suggestion or defect.

From step “1. New”, there are two possible transitions: defects go directly to the engineering manager (EM) for assignment, while suggestions go to the product manager (PM) for evaluation first.

In this example, the transition to step “2. PM Evaluation” is conditional, based on Type = Suggestion and using the default <otherwise> condition for the transition to step **“**3. EM Processing”. Implementing an <otherwise> condition assures that an item can always transition to some next step.

At runtime, the user will be presented with a single choice for the next step. If the item is a suggestion, the only choice will be to proceed to step 2; if the item is anything other than a suggestion (a defect), the only choice will be to proceed to step 3. Both of these are set as dynamic defaults.

Try editing this step on your own from the following summary.

Table 5.2 1. New (Summary)

Field	Value
General Tab	
Step Name	1. New
Limited Step Duration	Selected
Duration	4 days
Relative To	<leave blank>
Participants Tab	
Notification Type	Random
To	Product Management group
Authorized	Product Management group PM Administrators group Reference to current value of Responsibility property
Exception Notification Tab	
To	PM
Step Notification Tab	
Importance	Normal
Notification Interval	12 hours
Subject	Use the default step notification message.
Body	Use the default step notification message.
Form Options Tab	
	<leave blank>
Status Notification Tab	
To	<leave blank>
Next Steps Tab	
Condition	Type
Value	Suggestion
Next Steps	2. PM Evaluation
Condition	<otherwise>
Next Steps	3. EM Processing
Dynamic Defaults Tab	
Property	Workflow Status
When equals value	2. PM Evaluation
Assign Property: Value:	Short Comment Status is 2. PM Evaluation
Property	Workflow Status
When equals value	3. EM Processing
Assign Property: Value	Short Comment Status is 3. EM Processing
Workflow Extensions Tab	
Workflow Extensions	<leave blank>

Creating Next Steps for “1. New”

Use the Next Steps tab to specify one or more steps to follow the current step in the workflow and the conditions that govern the movement of the item from this step to the next. Specifying conditions for a step’s transition governs the choices that appear to users when they complete the step.

For example, the example workflow specifies that all suggestions are to be reviewed by the product manager before they are sent to engineering, while all defects are sent directly to the engineering manager. The logic for the *Next Steps* dialog is:

```
If Type = Suggestion
then Next Step = 2. PM Evaluation
<otherwise> Next Step = 3. EM Processing
```

If the Type is set to Suggestion, the only possible next step, and the only choice presented to the user, is step 2. For all other values of Type, the only possible next step is step 3.

In this simple example, the next steps could just as easily be specified as:

```
If Type = Suggestion
then Next Step = 2. PM Evaluation
If Type = Defect
then Next Step = 3. EM Processing
```

yielding the same result from the user’s perspective. However, if values are ever added to the Type property and corresponding conditions are not added to the workflow step, an item with the new value for Type will not be able to transition from this step to any other.

So, it is not necessary to specify an <otherwise>, or default, condition for the transition if the values of the conditional property are accounted for. However, specifying an <otherwise> condition safeguards against the possibility of an item becoming stranded in a step.

Borland recommends that you specify a next step for the <otherwise> condition. If you do not specify an <otherwise> condition, Borland recommends that you set a duration limit for the step so that an exception will be raised if the step does not transition within the specified duration. Duration is set on the General tab.

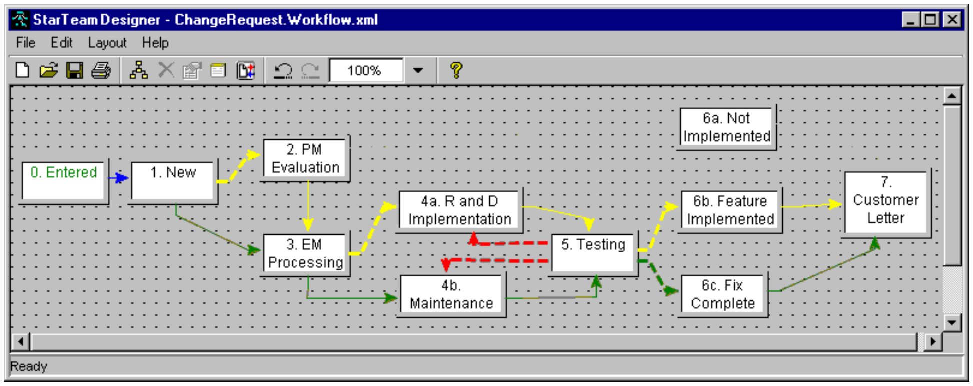
If the steps specified as next steps have not yet been drawn, Workflow Designer draws rectangular objects in the workspace to represent these steps automatically. Workflow Designer deposits the graphical steps in the upper left corner of the workspace, one on top of another. You can move the steps manually or select Layout > Automatic Layout to rearrange them.

Drawing the Rest of the Steps

As you progress through the assignment, rearranging automatically generated steps becomes tedious. It is best to draw a step for each value now, before proceeding with the rest of the Example exercises.

Do not draw the last step (8. Customer Letter Sent) as there are no actions taken after this workflow status is assigned to an item.

See “Drawing the “1. New” Step” on page 75 for the directions to draw a step.



Summary of Other Step Properties

The following sections contain tables that summarize the properties for all the other steps in this example. While the workflow control property is named Usr_StarFlowStatus, its display name Workflow Status is what you see in property lists.

Exception notification should be set up at every step that has participants, so that someone is notified when no one takes responsibility for an item at a particular step. An exception is also generated if a step does not transition to the next step in the set time limit (duration).

Try editing these steps entirely on your own. Some steps are given different properties to suggest alternate ways of doing the same thing, depending on your needs.

2. PM Evaluation

Note that neither of the two possible transitions from step 2 are conditional. The PM will always be presented with the same two choices for the next step.

If the PM selects the next step “6a. Not Implemented”, the Responsibility property is automatically filled using dynamic defaults. In this case, assume that the Customer Liaison user, by choice, receives no email notifications from Notification Agent. This user prefers to sort the change requests by responsibility to find items that require letters. This is done every few days on a regular schedule. Notice that in the Testing step, when the next step is set to “6b. Feature Implemented” or “6c. Fix Complete”, the Customer Liaison is similarly set as the responsible user.

Table 5.3 2. PM Evaluation

Field	Value
General Tab	
Step Name	2. PM Evaluation
Limit Step Duration	selected
Duration	7 days
Relative To	<leave blank>
Participants Tab	
Notification Type	Ordered (does not matter because there is only one participant)
To	PM
Authorized	PM Administrators group Reference to current value of Responsibility property value

Table 5.3 2. PM Evaluation

Field	Value
Exception Notification Tab	
To	PM General Manager
Step Notification Tab	
Importance	Normal
Notification Interval	4 days
Subject	<leave blank> to use the default step notification message.
Body	<leave blank> to use the default step notification message.
Form Options Tab	
	<leave blank>
Status Notification Tab	
To	<leave blank>
Next Steps Tab	
Condition	<otherwise>
Next Steps	3. EM Processing 6a. Not Implemented
Dynamic Defaults Tab	
Property	Workflow Status
When equals value	6a. Not Implemented
Assign Property: Value:	Short Comment Status is 6a. Not Implemented
Assign Property: Value:	Responsibility Customer Liaison
Property	Workflow Status
When equals value	3. EM Processing
Assign Property: Value:	Short Comment Status is 3. EM Processing
Workflow Extensions Tab	
Workflow Extensions	<leave blank>

3. EM Processing

Both of the transitions from step 3 are conditional. If the change request is a suggestion, the EM’s only option is to forward the item to R and D; if the change request is a defect, the only option is to forward the item to maintenance.

In this case, the EM can either set only the next step or both set the next step and directly assign responsibility to a specific team member. If this is common practice, you might want to turn off email notification.

Table 5.4 3. Processing

Field	Value
General Tab	
Step Name	3. EM Processing
Limit Step Duration	selected
Duration	7 days
Relative To	<leave blank>
Participants Tab	
Notification Type	Ordered (does not matter because there is only one participant)
To	EM
Authorized	EM Research and Development group Product Maintenance group Administrators group Reference to current value of Responsibility property
Exception Notification Tab	
To	EM PM
Step Notification Tab	
Importance	Normal
Notification Interval	4 days
Subject	<leave blank> to use the default step notification message.
Body	<leave blank> to use the default step notification message.
Form Options Tab	
	<leave blank>
Status Notification Tab	
To	<leave blank>
Next Steps Tab	
Condition	Type
Value	Suggestion
Next Steps	4a. R and D Implementation
Condition	<otherwise>
Next Steps	4b. Maintenance Fix
Dynamic Defaults Tab	
Property	Workflow Status
When equals value	4a. R and D Implementation
Assign Property: Value:	Short Comment Status is 4a. R and D Implementation
Property	Workflow Status
When equals value	4b. Maintenance Fix
Assign Property: Value:	Short Comment Status is 4b. Maintenance Fix
Workflow Extensions Tab	Workflow Status
Workflow Extensions	<leave blank>

4a. R and D Implementation

When completed, the user would select Testing as the next step.

Table 5.5 4a. R&D Implementation

Field	Value
General Tab	
Step Name	4a. R and D Implementation
Limit Step Duration	Selected
Duration	10 day
Relative To	<leave blank>
Participants Tab	
Notification Type	Ordered
To	Reference to current value of Responsibility property Research and Development group
Authorized	Reference to current value of Responsibility property Research and Development group Administrators group
Exception Notification Tab	
To	EM
Step Notification Tab	
Importance	Normal
Notification Interval	24 hours
Subject	<leave blank> to use the default step notification message.
Body	<leave blank> to use the default step notification message.
Form Options Tab	
	<leave blank>
Status Notification Tab	
To	<leave blank>
Next Steps Tab	
Condition	<otherwise>
Next Steps	5. Testing
Dynamic Defaults Tab	
Property	Workflow Status
When equals value	5. Testing
Assign Property: Value:	Short Comment Status is 5. Testing
Workflow Extensions Tab	
Workflow Extensions	<leave blank>

4b. Maintenance Fix

When completed, the user would select Testing as the next step.

Table 5.6 4.b. Maintenance Fix

Field	Value
General Tab	
Step Name	4b. Maintenance Fix
Limit Step Duration	Selected
Duration	5 day
Relative To	<leave blank>
Participants Tab	
Notification Type	Ordered
To	Reference to current value of Responsibility property Product Maintenance group
Authorized	Reference to current value of Responsibility property Product Maintenance group Administrators group
Exception Notification Tab	
To	EM
Step Notification Tab	
Importance	Normal
Notification Interval	24 hours
Subject	<leave blank> to use the default step notification message.
Body	<leave blank> to use the default step notification message.
Form Options Tab	
	<leave blank>
Status Notification Tab	
To	<leave blank>
Next Steps Tab	
Condition	<otherwise>
Next Steps	5. Testing
Dynamic Defaults Tab	
Property	Workflow Status
When equals value	5. Testing
Assign Property: Value	Short Comment Status is 5. Testing
Workflow Extensions Tab	
Workflow Extensions	<leave blank>

5. Testing

All of the transitions from step *5. Testing* are conditional. If the item is a suggestion, the user may either close the item or send it back to Research and Development for additional work. If the item is a defect, the user may either close the item or send it back to maintenance for additional work.

Dynamic defaults are used to direct a faulty implementation or fix back to the engineer who sent it to test. This is another case in which users would prefer to have email notification turned off to avoid redundant messages.

Table 5.7 5. Testing

Field	Value
General Tab	
Step Name	5. Testing
Limit Step Duration	Selected
Duration	2 days
Relative to	<leave blank>
Participants Tab	
Notification Type	Random
To	Testing group
Authorized	Testing group Administrators group Reference to current value of Responsibility property
Exception Notification Tab	
To	QA Manager EM
Step Notification Tab	
Importance	Normal
Notification Interval	12 hours
Subject	<leave blank> to use the default step notification message.
Body	<leave blank> to use the default step notification message.
Form Options tab	
	<leave blank>
Status Notification Tab	
To	QA Manager
Next Steps Tab	
Condition	Type
Value	Suggestion
Next Steps	4a. R and D Implementation 6b. Feature Implemented
Condition	Type
Value	Defect
Next Steps	4b. Maintenance Fix 6c. Fix Complete
Dynamic Defaults Tab	
Property	Workflow Status
When equals value	4a. R and D Implementation
Assign Property: Value:	Short Comment Status returns to 4a. R and D Implementation
Assign Property: Value:	Responsibility Reference to current value of Modified By property
Property	Workflow Status
When equals value	4b. Maintenance Fix
Assign Property: Value:	Short Comment Status returns to 4b. Maintenance Fix
Assign Property: Value:	Responsibility Reference to current value of Modified By property

Table 5.7 5. Testing

Field	Value
Property	Workflow Status
When equals value	6b. Feature Implemented
Assign Property: Value:	Short Comment Status is 6b. Feature Implemented
Assign Property: Value:	Responsibility Customer Liaison
Property	Workflow Status
When equals value	6c. Fix Complete
Assign Property: Value:	Short Comment Status is 6c. Fix Complete
Assign Property: Value:	Responsibility Customer Liaison
Workflow Extensions	
Workflow Extensions	<leave blank>

6a. Not Implemented

If a suggested change request has not been implemented, the appropriate users and groups are notified that the change request is closed and the customer is sent an appropriate thank you note.

Letters to be generated to customers are not sent from the application, although the templates for them may be stored in the application. The Customer Liaison notices this step’s completion and is expected to send the appropriate letter and move to the final step, which is Customer Letter.

Table 5.8 6a. Not Implemented

Field	Value
General Tab	
Step Name	6a. Not Implemented
Duration	2 days
Relative to	<leave blank>
Participants Tab	
Notification Type	<leave blank>
To	<leave blank>
Authorized	Customer Liaison Administrators group Reference to current value of Responsibility property
Exception Notification Tab	
To	PM General Manager
Step Notification Tab	
To	<leave blank>
Form Options Tab	
	<leave blank>
Status Notification Tab	
To	Research and Development group EM Product Management group PM Customer Liaison
Subject	<leave blank> to use the default step notification message.
Body	<leave blank> to use the default step notification message.

Table 5.8 6a. Not Implemented

Field	Value
Next Steps Tab	
Condition	<otherwise>
Value	7. Customer Letter
Dynamic Defaults Tab	
Property	Workflow Status
When equals value	7. Customer Letter
Assign Property: Value	Short Comment Status is 7. Customer Letter
Workflow Extensions Tab	
Workflow Extensions	<leave blank>

6b. Feature Implemented

If a suggestion change request has been implemented, the appropriate users and groups are notified that the CR is closed, the customer is sent an appropriate thank you note, and Marketing is notified of the product's new feature.

Letters to be generated to customers are not sent from the application, although the templates for them may be stored in the application. The Customer Liaison is notified about this step's completion and is expected to send the appropriate letter and move to the final step, which is Customer Letter.

Table 5.9 6b. Feature Implemented

Field	Value
General Tab	
Step Name	6b. Feature Implemented
Duration	2 days
Relative to	<leave blank>
Participants Tab	
Notification Type	<leave blank>
To	<leave blank>
Authorized	Customer Liaison Administrators group Reference to current value of Responsibility property value
Exception Notification Tab	
To	PM EM General Manager
Step Notification Tab	
To	<leave blank>
Form Options Tab	
	<leave blank>

Table 5.9 6b. Feature Implemented

Field	Value
Status Notification Tab	
To	Research and Development group EM Product Management group PM Customer Liaison Marketing group
Subject	<leave blank> to use the default step notification message.
Body	<leave blank> to use the default step notification message.
Next Steps Tab	
Condition	<otherwise>
Value	7. Customer Letter
Dynamic Defaults Tab	
Property	Workflow Status
When equals value	7. Customer Letter
Assign Property: Value:	Short Comment Status is 7. Customer Letter
Workflow Extensions Tab	
Workflow Extensions	<leave blank>

6c. Fix Complete

If a defect has been fixed, the appropriate users and groups are notified that the CR is closed, Customer Support is notified that the problem has been fixed, and the customer is sent an appropriate thank you note.

Letters to be generated to customers are not sent from the application, although the templates for them may be stored in the application. The Customer Liaison is notified about this step’s completion and is expected to send the appropriate letter and move to the final step Customer Letter.

Table 0.1 6c. Fix Completed

Field	Value
General Tab	
Step Name	6c. Fix Complete
Duration	2 days
Relative to	<leave blank>
Participants Tab	
Notification Type	<leave blank>
To	<leave blank>
Authorized	Customer Liaison Administrators group Reference to current value of Responsibility property
Exception Notification Tab	
To	PM EM General Manager
Step Notification Tab	
To	<leave blank>

Table 0.1 6c. Fix Completed

Field	Value
Form Options Tab	
	<leave blank>
Status Notification Tab	
To	Product Maintenance group Product Management group Customer Liaison Customer Support group
Subject	<leave blank> to use the default step notification message.
Body	<leave blank> to use the default step notification message.
Next Steps Tab	
Condition	<otherwise>
Value	7. Customer Letter
Dynamic Defaults Tab	
Property	Workflow Status
When equals value	7. Customer Letter
Assign Property: Value:	Short Comment Status is 7. Customer Letter
Workflow Extensions Tab	
Workflow Extensions	<leave blank>

7. Customer Letter

This is the final step. The General Manager (who is Customer Liaison’s supervisor) is notified when an item moves to this step.

Table 5.10 7. Customer Letter

Field	Value
General Tab	
Step Name	7. Customer Letter
Participants Tab	
To	<leave blank>
Exception Notification Tab	
To	<leave blank>
Step Notification Tab	
To	<leave blank>
Form Options Tab	
	<leave blank>
Status Notification Tab	
To	General Manager
Next Steps	
	<leave blank>
Dynamic Defaults Tab	
Property	Workflow Status
When equals value	8. Customer Letter Sent
Assign Property: Value	Short Comment Customer Letter has been sent
Workflow Extensions	
Workflow Extensions	<leave blank>

Testing the Workflow

To test a workflow, you need to:

1. In StarTeam, create a Test project in your server configuration. (The project name does not have to be Test; this is just an example.)
2. Create a Test folder as a child of the Projects folder in the StarFlow Extensions project.

   See "Adding Folders to the StarFlow Extensions Project" on page 11, "Understanding Locator" on page 10, and "Saving a Workflow" on page 63.
3. Copy the ChangeRequest.Workflow.xml, ChangeRequest.jar and ChangeRequest.propertyeditor.xml files to the Test folder's working folder.
4. Check them in.
5. Activate the APE form:
   a. In StarTeam, open the Test project and choose Project > Properties from the menu.
   b. From the resulting *Project Properties* dialog, click the Editors tab.
   c. Select the Use Alternate Property Editor For Change Requests check box.
   d. Type "Locator ChangeRequest" in the text box under the check box.
   e. Click OK.
6. Setup and start Notification Agent to monitor the Test project for the correct item type.

   See "Getting Ready to Use Notification Agent" on page 94 and .
7. Use Notification Agent's Run Now button to simulate movement from step to step.
8. Correct the workflow if necessary.

Deploying the Workflow

Perform the following steps to activate your workflow in the Example project now that it has been tested:

1. In StarTeam, copy the ChangeRequest.Workflow.xml, ChangeRequest.jar and ChangeRequest.propertyeditor.xml files to the StarFlow Extensions project's Example folder's working folder.
2. Check them in.
3. Activate the APE form for the Example project using the *Project Properties* dialog.
4. Setup and start Notification Agent to monitor the Example project for the correct item type.

   Notification Agent notifies appropriate users about pending work, item statuses, and exceptions that occur in the process. See "Getting Ready to Use Notification Agent" on page 94 and "Starting Notification Agent and Notification Setup" on page 97.

Chapter

6

Using StarTeam Notification Agent

StarTeam Notification Agent watches the projects, views, and items controlled by workflows and performs the following functions:

- Status notification

  If a workflow step requires status notification, Notification Agent sends messages simultaneously to all the users in the status notification list, letting them know that an item has reached that workflow step. The notification messages informing users about item status can be customized.

- Responsibility notification

  Notification Agent ensures that users are notified of pending work. Notification messages are sent whenever an item under workflow control enters a step that has a list of participants. The users who are listed as participants for that step are notified, all at once or in the specified order, depending on that step's properties in the workflow. When a participant accepts a workflow request, the responsibility notification process for that step ends. The notification messages asking users to accept responsibility for the work can be customized.

  Responsibility notification messages now come in the following types:

  - Those that require a response

    Notification Agent always sends email messages using the SMTP protocol. It receives messages using one of the following:

    - The Apache HTTP Server
    - IMAP protocol (as in previous releases)
    - POP3 protocol (as in previous releases)

    You must provide either an Http Server Address or a Mail Store Address so that Notification Agent can process the responses. You can provide both addresses.

    If you provide only an Http Server Address, Notification Agent sends a default email message that uses http links for accepting and declining responsibility.

    If you provide only a Mail Store Address, you must also provide Notification Agent with a mailbox for your email system. Notification Agent sends a default email message with email links for accepting and declining responsibility.

You can provide both an Http Server Address and a Mail Store Address. However, this does not much sense unless you also plan to provide a custom message that will use both http and email links. In this case, by default, Notification Agent will send the responsibility message that includes email links.

See “Setting Up Notification Agent to Monitor a Server Configuration” on page 98 for more information about setting address options.

You can provide a custom message using no links, email links, or http links. Custom messages for responsibility notification must go in the *Edit Step* dialog on the Step Notification tab. See “Specifying a Notification Message and Interval” on page 54.

- Those that require *no* response

  Notification Agent sends mail messages using the SMTP protocol.

  If Notification Agent notices that you have provided no Http Server Address and no Mail Store Address, it sends a default message that requires no response.

  You should probably designate that messages to participants be blasted. This is a Workflow Designer setting. See “Listing the Participants” on page 52 for details.

- Exception notification

  When a workflow step is not completed as expected—for example, when a step is not completed within the specified duration, or when no participant accepts responsibility for an item at this step—Notification Agent generates an exception message. This message goes simultaneously to the users in the exception list for that workflow step. The exception message is formed by Notification Agent based on the type of exception.

Notification information is logged in files with names based on when the files were created, for example, NA-2002-9-25-21-12-10.log. The log files are stored in Notification Agent’s installation folder.

Messages can be sent as plain text for email or as html (the TomCat http server is installed along with Notification Agent).

About This Chapter

This chapter explains how to:

- Ensure that Notification Agent will run successfully by creating mailboxes and user names and setting access rights in advance.
- Set up and run Notification Agent.
- Change the http serverport number, if necessary. For example, you may be using the default port (9000) for another purpose or you might want to run multiple http servers.

Getting Ready to Use Notification Agent

Each instance of Notification Agent can monitor only one server configuration. That means that you must run Notification Agent once for each server configuration to be monitored.

Before you can run an instance of Notification Agent, you must:

- Create a mailbox in your mail system for the sole use of the Notification Instance instance, if you will be sending email notification messages that require a response. Thus, you will use a different email address for each server configuration, as one instance of Notification Agent can monitor only one server configuration. You may need to request that this mailbox be created by an MIS representative. If you run Notification Agent on more than one server configuration, each server configuration will need its own mailbox.

- Decide what StarTeam user name and password will be used by Notification Agent. Borland recommends dedicating a user license to Notification Agent. However, you can use an existing user, if you prefer. If you do this:
  - The user must have an email address listed in the server's User Manager. Notification Agent does not have to use that user's email address, but the user must have one.

    If the user name to be used by Notification Agent is that of a team member and you plan to send email notification messages that require a response, Notification Agent must use a different email address. You will not want to fill the team member's mailbox with email from the Notification Agent.

    For details about adding users, see the *StarTeam Administrator's Guide.*

    Be sure that the user who runs Notification Agent has adequate access rights. See "Setting Access Rights for Notification Agent" on page 96.
- Consider whether you want other forms of notification to be enabled at the same time as Notification Agent. For example, you may want to turn off email notification or status tray notification.
  - Email notification per server configuration

    If you enable e-mail notification, all users will receive email when:

    - A change request, requirement, or task for which the team member is responsible is created or modified. For example, a user may need to fix it or verify a change request.
    - A topic that has the team member as a recipient is created or modified. (If a topic has no recipients listed, no one receives notification.)

    This type of notification is either on for all users or off for all users.

    This type of notification can be redundant or conflict with Notification Agent email messages.

    Redundant example: Suppose Notification Agent has been set up to request users to accept or decline responsibility for an item at a particular step. Suppose that one user has been sent email to that effect. The user accepts responsibility and receives both confirmation email from Notification Agent and notification email from the Server.

    Conflict example: Suppose Notification Agent has been set up to manage responsibility. Suppose that one user has been sent email to that effect. Suppose that another user forgets to let Notification Agent manage the responsibility field and changes the responsibility for an item manually to the first user. The first user accepts responsibility using the email message from Notification Agent. Notification Agent notices that the item's responsibility has already changed. It notifies the first user that it did not make him the responsible party because the item has already be updated. At about the same time, the Server notifies the first user that he is responsible for the item. This can be very confusing for the first user and can be avoided if only Notification Agent sets responsibility for such items or if the Server's email notification is turned off.
  - Status tray notification per user:

    The application client notifies you by placing icons in the status tray of the task bar whenever:

    - A change request that is the user's responsibility is created or modified.
    - A requirement that is the user's responsibility is created or modified.

  - A task that is the user's responsibility has been created or modified.
  - A topic (that has the user as a recipient) has been created or modified. If a topic has no recipients listed, no one receives notification.

  This type of notification is turned on and off per component and per user. There are personal options for system tray notification in the Windows client. The Cross-Platform client does not have this functionality.

- Run Notification Agent Setup to identify each server configuration to be monitored by Notification Agent and to set notification properties for that configuration.

  As part of this process, a Notification Agent folder will be added to the server configuration's StarFlow Extensions project. Notification Agent uses this folder to store the setup information for the server configuration in a file named NotificationAgent.conf.xml.

  This process also creates a *serveraddress_port*.nac file (for example, orion_49201.nac) for each server configuration. The .nac file contains the mailbox password but in an encrypted format. The file is stored in Notification Agent's installation folder.

Setting Access Rights for Notification Agent

The user running Notification Agent and Notification Agent Setup must have the following access rights:

- At the server level:
  - The server access right "Change user/operation time". Notification Agent needs this right because it impersonates other users. When a user accepts responsibility for an item, Notification Agent impersonates the accepting user as it sets the workflow responsibility property to the name of that user. The audit log records the action as if it were done by the impersonated user.

    For more information about setting server access rights, see the *StarTeam Administrator's Guide.*
  - The server access right "Administer user accounts"

    Notification Agent needs this right to obtain email addresses for users and other user information.
- Notification Agent cannot set the responsibility property for an item to a specific user unless that user (who is being impersonated by Notification Agent) has the correct application access rights. Make sure that the appropriate items can be accessed by the participating users. For example, Quality Assurance team members would have most or all of the access rights available for change requests. To set the responsibility for a change request to a specific user, that user must have the rights to modify the properties of a change request, and so on.
- The user running Notification Agent Setup must be able to access the StarFlow Extensions project, create a child folder, and add a file.

  At the project level:
  - For the Project node, set the "See object and its properties" right.
  - For the View node, set the "See object and its properties" right.
  - For the Child Folders node, set the following rights:
    - See object and its properties
    - Create folders

- For the File node, set the following rights:
  - See object and its properties
  - Modify properties
  - See History
  - Set exclusive locks
  - Check in file
  - Check out file
  - Add file

Starting Notification Agent and Notification Setup

Both Notification Agent and Notification Setup work only with the Windows-based Server. They can be installed on the same computer as the Server they monitor or another computer. When using two computers, their times must be synchronized.

Locate them on your Start menu at Start > Programs > StarTeam > Notification Agent > Agent and Start > Programs > StarTeam > Notification Agent > Agent Setup.

Identifying a Server Configuration

An instance of Notification Agent can monitor only one server configuration on an application Server. However, you can run several instances on the same computer if you need to.

To fully identify the server configuration, you must know:

- The name or IP address of the computer on which the configuration runs
- The port number for the server configuration
- The user name and password to be used by Notification Agent as it logs on to the server configuration
- The name and password for the mailbox created for Notification Agent on your mail system for this server configuration
- The email address and display name for that mailbox

To identify a server configuration:

1. Start Notification Agent Setup by choosing Start > Programs > StarTeam > Notification Agent > Agent Setup from the Start menu.
2. In the resulting *Notification Agent Setup* window, choose File> New from the menu bar.
3. In the resulting *Select a Configuration* dialog:
   a. Click Add Configuration.
   b. In the resulting *Add Configuration* dialog, identify the server configuration by:
      1. In the Server Address text box, enter the name or IP address of the computer on which the Server resides.
      2. In the Port text box, enter the port number used to access the server configuration.
      3. In the StarTeam User Name and StarTeam Password text boxes, enter the user name and password that Notification Agent will use to log on to this server configuration.

4 In the Mailbox and Mailbox Password text boxes, enter the name of and password for the mailbox in your mail system to be dedicated to Notification Agent for use with this server configuration.

5 In the Display Name and Email Address text boxes, enter the name and email address for that mailbox.

6 Click OK to return to the *Select a Configuration* dialog, which now displays the server address and port number (for example “orion [49201]”) as an entry.

A .nac file (for example orion_49201.nac) is created for this server configuration and stored in Notification Agent’s installation folder.

Tips

You can click the Configuration Properties button to change the settings that identify the server configuration or click the Delete Configuration button to delete a server configuration from the list. Deleting the configuration also deletes the corresponding .nac file. It does not delete information about the entry (computer/port combination) from the NotificationAgent.conf.xml file stored in the server configuration’s StarFlow Extensions project.

Setting Up Notification Agent to Monitor a Server Configuration

Before Notification Agent can monitor any server configuration you identify, you must set options for that configuration using Notification Agent Setup. Before you set the options, ensure that:

- The server configuration is running.
- The StarTeam user name and password provided for the configuration are valid.
- That user name has access rights that will allow Notification Agent Setup to access the StarFlow Extensions project, create a child folder, and add a file. See “Setting Access Rights for Notification Agent” on page 96.
- The server configuration already has a StarFlow Extensions project.

  If the server configuration does not already have this project, Notification Agent will create one using a subfolder of Notification Agent’s installation folder as its working folder. This is not recommended.

To set or edit monitoring options.

1 Start Notification Agent Setup by selecting Start > Programs > StarTeam > Notification Agent > Agent Setup from the Start Menu.

2 In the resulting *Notification Agent Setup* window, choose File > New from the menu bar.

3 In the resulting *Select a Configuration* dialog, select the entry for this server configuration.

4 Click OK.

If the server configuration is not running or the user name and password for accessing the configuration are incorrect, you cannot set monitoring options.

5 If this server configuration already has a StarFlow Extensions project, click Yes when you are asked to create the Notification Agent folder in this project.

6 When the *Notification Agent Setup* window displays the options, click Add to identify the projects and views to be monitored.

a From the resulting *Add Watched Views* dialog, select the project, one or more of its views, and one or more item types for Notification Agent to monitor.

You can multi-select views and types or use the Select All Views or Select All Types check boxes to quickly select everything in the corresponding list box.

b Click OK to return to the *Notification Agent Setup* window with the specified information displayed.

7 Accept the system default polling periods or change them to meet your needs.

Important Your Server may have a setting for Inactivity Timeout. If so, both your Minimum Polling Period and Maximum Polling Period settings should be smaller than the server's Inactivity Timeout setting.

a Enter numbers of minutes in the Minimum Polling Period and Maximum Polling Period text boxes.

These numbers control how long Notification Agent waits before checking the server for new items and changes to the workflow control property and the responsibility property in existing items.

For example, the defaults are 3 and 30 minutes. Notification Agent checks the server every 3 minutes as long as new and changed items are found. When nothing is found, the minimum time is doubled.

Notification Agent checks the server after a 6-minute wait. If new or changed items are found, Notification goes back to checking every 3 minutes. If nothing is found, the waiting period is doubled once more and Notification Agent checks the server after a 12-minute wait. This process continues as long as the waiting period is fewer than 30 minutes, the maximum polling period. In this example, Notification Agent would never wait longer than 24 minutes (12 minutes doubled), because doubling 24 minutes exceeds the maximum polling period.

b Enter a number of minutes in the Task Refresh Period text box. This controls how often Notification Agent checks the list of tasks it has created. This value affects how often Notification Agent checks for exceptions.

8 Enter the http and mail server information for your installation in the Server area. See your system administrator if you need help.

a (Optional) You can fill the Http Server Address text box for only one of the server configurations that run on the same computer—unless you installation Notification Agent multiple times. With each installation, another copy of the TomCat http server (a product of The Apache Software Foundation) is installed. This software is in the conf subfolder beneath installation folder. For example, you might have C:\Program Files\Borland\StarTeam Notification Agent_1\conf and C:\Program Files\Borland\StarTeam Notification Agent_2\conf.

Each copy defaults to port 9000. If you use additional TomCat http servers, each copy must use a different port number.

If this configuration will use the http server, fill the Http Server Address text box by typing the IP address or name of the computer on which Notification Agent is installed, followed a colon (:) and the port number. Localhost is not accepted.

If Notification Agent runs on the orion computer that has the IP address 55.55.55.55, you can enter either of the following:

```
orion:9000
55.55.55.55:9000
```

If the TomCat server for this server configuration uses the default port, which is "9000", you can omit the ":9000".

Notes

- If you need to change the port number for a TomCat http server, see "Changing the Http Server Port Number" on page 104.
- The Http Server Address specified here is the value that will replace the {{{link_base}}} variable in step notification messages and is where shared files, such as logo image files, should be stored.

- See the following URL for more information about TomCat:
- http://jakarta.apache.org/
- Leave both the Http Server Address text box and the Mail Store Address text box empty if:
- You will create an email notification that does not require a response
- You want Notification Agent to send a default email notification message that does not require a response

b (Required) All configurations must have the SMTP server name in the SMTP Server Address text box because Notification Agent sends email messages using SMTP. The name should be followed by a colon (:) and a port number.

For example, you might use:

```
Exchange:25
```

Often the SMTP server is an Exchange server, although it is usually not named Exchange.

If the port number is 25, the default, you can omit the colon and port number.

c (Optional) Enter the name of the mail store followed by a colon (:) and the mail store's port number.

For example, you might use:

```
Exchange:110
```

If the port number is 110, the default for this Notification Agent setting, you can omit the colon and port number.

The default port number for POP3 is 110, and the default port number for IMAP is 143.

d In the Mail Store Type text box, enter either imap or pop3. The default is imap. Notification can receive email messages using either of these two protocols.

This field is ignored if you leave the Mail Store Address text box empty.

e In the Mail Store Address text box, enter the name of the folder used by the mail store for incoming mail. The default is inbox.

This field is ignored if you leave the Mail Store Address text box empty.

9 Indicate the hours during which Notification Agent should send notification email. For example, hours of operation may be 8 A.M. to 5 P.M., Monday through Friday.

The default is 24-7 operation: Notification Agent sends email every day of the week from 12 midnight to 11:59:59 P.M.

a In the Activity Group box, select a day of the week for which the hours of notification should be limited to fewer than 24 hours. By default, Monday is selected.

b To replace the 24-hour time interval with one or more other intervals:

1 Select the 24-hour interval.

2 Click Remove.

3 Click Add.

4 From the resulting *Add an Active Interval* dialog, set the appropriate start and end times.

5 Click OK.

6 Repeat steps 3 through 6 if you need additional intervals for the selected day.

c To copy the time intervals already specified for another day.

1 Click Copy.

2 From the resulting *Copy Activity Times* dialog, select the day to be copied from the drop-down list.

3 Click OK.

10 Indicate any holidays on which Notification Agent should *not* send email notification.

a In the Holidays portion of the Activity group box, click Add.

b In the resulting *Add a Holiday* dialog, enter the name for this holiday in the Holiday Name text box.

c Select a date using the calendar.

1 Click >> and/or << to move the calendar forward or backwards a year at a time.

2 Click> and/or < to move the calendar forward or backwards a month at a time.

3 Click the day of the month.

d Select the Recurring Holiday check box if this calendar date will be a holiday every year.

11 Choose File > Close to close the *Notification Agent Setup* window.

12 Click Yes when asked to save your notification configuration.

13 In the resulting *Revision Comment* dialog, enter a comment about this set of options.

14 Click OK.

Notification Agent Setup creates a file named NotificationAgent.conf.xml and stores it in the Notification Agent folder in the server configuration's StarFlow Extensions project. It uses the revision comment as either the description or comment for the revision, depending on whether this is the first or a later revision of the file.

Tips To remove a row in the Watched Views list, select the row and click the Remove button. To export an exact copy of the NotificationAgent.conf.xml file, choose File > Export. Since every server configuration monitored by Notification Agent has a file with this name, it is a good idea to rename the exported file.

Running Notification Agent

If you choose to, you can run Notification Agent 365 days a year, 24-7. If you stop Notification Agent and restart, it starts processing the notification tasks it missed while it was shut down. Each instance of Notification Agent monitors one server configuration.

You can run Notification Agent manually or as a service.

Running Notification Agent Manually

Notification Agent and Notification Agent Setup are separate, but similar looking applications.

To run Notification Agent manually:

1 Start Notification Agent by selecting Start > Programs > StarTeam > Notification Agent > Agent from the Start Menu.

The *Notification Agent* window opens, displaying a list of server configurations, one for each .nac file you created as you added server configurations using Notification Agent Setup.

2 Select a configuration.

3 Click Start Agent. After the agent starts, the button text changes to Shut Down and the word “Running” appears beside the configuration’s name.

If you use Notification Agent Setup to change the information in a server configurations .nac file, you can update Notification Agent by clicking the Refresh button.

To test notification and workflow processes, you can use the Run Now button. Clicking Run Now moves an item to the next workflow step regardless of notification times. For example, suppose that Notification Agent has just notified the first participant that an item needs attention. Clicking Run Now prompts Notification Agent to notify the next participant without waiting for the step notification interval to expire. See “Testing the Workflow” on page 91.

Running Notification Agent as a Service

Many customers run both the StarTeam Server and Notification Agent as services.

When running the Server and Notification Agent on the same computer, the Server process must start before the Notification Agent attempts to start. The best way to accomplish this is to set up a Service dependency within Windows.

If you are running the Server as a service and Notification Agent as a dependent service, you cannot shut down the Server unless the Notification Agent service is shut down first.

To avoid dependent servers, some customers run the Server and Notification Agent on different computers. When using two computers, you must first synchronize the times on both computers. If Notification Agent returns a View Configuration exception, the two computers are probably not synchronized.

To run an instance of Notification Agent as a service:

1 Do one of the following:

- Modify the command stored in the NotificationAgentService.bat file and then run the batch file.
- Copy the updated command to the DOS command line.

The example initially in this file is:

```
JavaService.exe -install NotificationAgent "C:\Program Files\Borland\Java\
Sun1.4.1_02\bin\client\jvm.dll" -Djava.class.path="C:\Program Files\Borland\
StarTeam Notification Agent\agent.jar";"C:\Program Files\Borland\StarTeam
Notification Agent\starteamxx.jar" -Djava.library.path="C:\Program Files\
Borland\StarTeam Notification Agent";"C:\Program Files\Borland\StarTeam SDK
x.x\lib" -start com.starbase.starteam.workflow.NARuntime.
NotificationAgentFrame -params -service "C:\Program Files\Borland\StarTeam
Notification Agent\localhost_49201.nac" -stop
com.starbase.starteam.workflow.NARuntime.NotificationAgentFrame -method
stopAgent -out "C:\Program Files\Borland\StarTeam Notification Agent\
NAService.log" -err "C:\Program Files\Borland\StarTeam Notification Agent\
NAService.log" -current "C:\Program Files\Borland\StarTeam Notification
Agent"

pause
```

This file contains several complete paths that you may need to replace with appropriate path names from your system. Most of these paths are delimited by double-quotation marks. These paths represent:

- The path to the Java Runtime Environment on your computer. The .bat file uses the following path as an example.
  `C:\Program Files\Borland\Java\Sun1.4.1_02\ bin\client\jvm.dll`
- The class path, which requires two paths
  - The path to the Notification Agent .jar file (agent.jar). If you used the default installation location for Notification Agent, this is "C:\Program Files\Borland\ Notification Agent\agent.jar" and does not need to be changed.
  - The path to the StarTeam Runtime .jar file (starteam*xx*.jar). If you used the default installation location for Notification Agent, this is "C:\Program Files\ Borland\ Notification Agent\starteam*xx*.jar" and does not need to be changed.
- The library path, which requires two paths
  - The first is very similar to the path to the folder that contains agent.jar. If you used the default installation location for Notification Agent, this is "C:\Program Files\ Borland\ Notification Agent\agent" and does not need to be changed.
  - The second is the path to the folder that stores the Java version of the StarTeam Runtime file (starteam-sdk.jar). For example, if you used the default installation folders, this would be "C:\Program Files\Borland\StarTeam SDK *x.x*\lib" as found in the .bat file.
- The service path, which is the complete path to the .nac file (Notification Agent configuration file) for the instance of the agent being run as a service. This is usually similar to the path already in the .bat file ("C:\Program Files\Borland\ Notification Agent\localhost_49201.nac"). The .nac file contains options for a specific instance of the agent and its naming convention identifies both the computer and the port used by Notification Agent.
- The err and out paths, which are the paths to an output log file. If you used the default path, this would be “C:\Program Files\ Borland\StarTeam Notification Agent\ NAService.log” as found in the .bat file following both -out and -err.
- The current path, which is the path to the Notification Agent installation folder. This is usually "C:\Program Files\Borland\ StarTeam Notification Agent", as in the batch file.

After running NotificationAgentService.bat, this instance of Notification Agent runs as an automatic service and can be managed as any other automatic service.

Caution Changes to the registry should only be made by users who are experienced with the registry. Before making any changes always make a backup of the registry.

To set up a service dependency for Windows NT or 2000:

1. Select Start > Run from the Start menu.
2. Enter Regedt32.
3. Click OK.
4. Select HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\ Services\ NotificationAgent
5. Select Edit > Add Value
6. Set the Value Name to DependOnService.
7. Set the Data Type to REG_MULTI_SZ.
8. Enter your StarTeam Server Service name into the data window (Example: StarTeamServer-StarDraw). See “To find your service name:” on page 104 for more details.

To set up a service dependency for Windows XP

1 Select Start > Run from the Start menu.

2 Enter Regedt32.

3 Click OK.

4 Select HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NotificationAgent

5 Select Edit > New > Multi-String Value.

6 Set the Value Name to DependOnService.

7 Double-click the DependOnService value to open the Data window.

8 Enter your StarTeam Server Service name into the Data window (Example: StarTeamServer-StarDraw). See “To find your service name:” on page 104 for more details.

To find your service name:

Each application service has a service name and a display name. You are probably familiar with the display names that you have seen in the Services applet in the control panel.

The DependOnService command requires the service name, not the display name.

1 Select Start > Run from the Start menu.

2 Enter Regedt32.

3 Click OK.

4 Select HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services

Search the list for any items prefaced with “StarTeamServer” All application repositories are in a format of “StarTeamServer-*RepositoryName*”.

For the StarDraw Repository, for example, the service name would be "StarTeamServer-StarDraw" Use the service name you find here for step 8 of the setting up a service dependency process.

When the Notification Agent runs, log files are placed into the installation folder "...\StarTeam Notification Agent". Review these logs to determine if the service is working successfully or not. Should changes need to be made to the NotificationAgentService.bat file to resolve problems, the following command will need to be run first to uninstall the service:

```
JavaService.exe -uninstall NotificationAgent
```

You can use a .bat file or enter the command at the DOS prompt. Run the command from the same location as you ran NotificationAgentService.bat file.

If an attempt to re-install the service fails after a successful uninstall, the server should be rebooted to complete the uninstall operation.

Changing the Http Server Port Number

The http server default port number is 9000, but you can change it if necessary. For example, you may want to run more than one http server on one computer and they need different ports.

To change the http server port number:

1 Stop Notification Agent.

2 Each TomCat http server has a server.xml file.

As you install Notification Agent, a copy of TomCat is placed a subfolder of the installation folder. That folder is named conf and, inside that folder, is the file named server.xml.

Any additional installations of Notification Agent result in additional copies of TomCat and reside in conf subfolders of those installation folders. Each of their server.xml files needs a different port number.

3 Open server.xml in Notepad or another text editor.

4 Locate the following <Connector> statement:

```
<Connector className="org.apache.catalina.
connector.http.HttpConnector" port="9000" minProcessors="5"
maxProcessors="75" enableLookups="true" redirectPort="8443"
acceptCount="10" debug="0" connectionTimeout="60000" />
```

5 Change the port number to an appropriate number.

For example, if the <Connector> statement contained:

```
port="9000"
```

you might change it to:

```
port="9999"
```

6 Save the modified file.

7 Start Notification Agent Setup.

8 Open the desired configuration.

9 In the Http Server Address text box, enter the name of the computer, a colon, and the correct port number. For example, if the computer name is orion, you would use orion:9999.

10 Save this configuration.

11 Restart Notification Agent or click its Refresh button. Notification Agent will use the new port number.

Message Sending and Exception Processing

Notification Agent constantly monitors project views for items that need to be processed and for exceptions in normal processing. Exception messages go to those users or groups listed on the Exceptions tab of the *Edit Step* dialog for a given workflow step. Exception messages are generated by the system and cannot be changed. Additionally, exception messages are sent only once and no further notice is provided.

Files Created by Notification Agent

Notification Agent creates the following server files in the local installation folder for server configurations:

serveraddress_port.nac

This is a configuration file discussed in “Getting Ready to Use Notification Agent” on page 94 and elsewhere, for example, orion_49201.nac.

serveraddress_port.*itemtype_viewid*.tsp, for example, orion_49201.ChangeRequesty_13.tsp.

Such files are view timestamps used for shutting down and restarting view monitoring processes.

serveraddress_port.tasks.dat, for example, orion_49201.tasks.dat.

Such files archive the notification tasks managed by this instance of Notification Agent It is used when shutting down and restart view monitoring processes.

serveraddress_port.NA-*date_time*.log, for example, NA-2002-9-25-21-12-10.log

Such files are used by Notification Agent to log events and exceptions.

Another file, NotificationAgent.conf.xml contains configuration information used by Notification Agent per server configuration and is checked in to each server configuration's StarFlow Extensions project into a folder named Notification Agent.

Known Issues for Notification Agent

This section covers issues about which you may need to be aware.

Revision Assumptions

Notification Agent assumes that only Notification Agent and the users set by Notification Agent are editing (and, therefore, creating revisions to) items under workflow control. If, for example, a third party changes the responsibility for an item to someone who is authorized to edit it, Notification Agent assumes that the responsible person has already been notified.

Only Notification Agent Should Change Responsibility Field

Notification Agent assumes that only Notification Agent and the users set by Notification Agent are editing (and, therefore, creating revisions to) items under workflow control. If a third party changes the responsibility for an item to someone who is authorized to edit it, Notification Agent assumes that the responsible person has already been notified.

Out-of-Memory Errors from Java

When you start Notification Agent from the Start menu, options from the notification.stjava file are used. This file is in the Notification Agent installation folder (by default, C:\Program Files\Borland\StarTeam Notification Agent).

The "options=" parameter in this file is now set to "-Xmx256m", a Java option used to control memory usage. If you encounter out-of-memory errors, you may need to tweak this setting.

The documentation that Sun currently provides about this option is as follows:

```
-Xmxn
```

Specify the maximum size, in bytes, of the memory allocation pool. This value must a multiple of 1024 greater than 2MB. Append the letter `k` or `K` to indicate kilobytes, or `m` or `M` to indicate megabytes. The default value is 64MB. Examples:

```
-Xmx83886080
```

```
-Xmx81920k
```

```
-Xmx80m
```

The default content of the notification.stjava file is similar to the following:

```
[StJava]
class=com.starbase.starteam.workflow.NARuntime.NotificationAgentFrame
starteam=release_number
test=0
[Java VM]
```

```
name=java_version
classpath=installation_folder\agent.jar;%CLASSPATH%
librarypath=%LIBRARYPATH%;installation_folder
options=-Xmx256m
```

Appendix

A

Installing StarTeam Workflow Applications

This appendix explains how to install:

- StarTeam Extensions
- StarTeam Workflow Designer
- StarTeam Notification Agent

Installing StarTeam Extensions

StarTeam Extensions is installed in segments. You can install the runtime files, documentation, and samples where they are most needed or easiest to work with.

The default installation folder is C:\Program Files\Borland\StarFlow Extensions, but Borland recommends copying the runtime files to a different folder for each server configuration that runs on the same Server after the installation. For example, you might use C:\config_name\StarFlow Extensions. In this way, you can ensure that each configuration's StarFlow Extensions project has different working folders.

Even if you currently run only one server configuration, it is wise to plan ahead. This also ensures that, when you upgrade or reinstall for some other reason, you are unlikely to overwrite any files in the StarFlow Extensions project that you have modified on your workstation and have not checked in.

Caution In earlier releases, StarTeam Extension files were installed along with the server. If you uninstall those releases, the StarFlow Extensions folder is not deleted. This is to prevent the loss of data.

However, when you install the files on your own, you may not want to overwrite the existing files. If you are using the default installation folder for any server configuration's StarFlow Extensions project, Borland recommends that you check in everything you need that is not current prior to reinstalling or upgrading StarTeam Extension files. For example, you might move your existing StarFlow Extensions folder to a new location C:\config_name\StarFlow Extensions, change the working folder for the StarFlow Extensions project in that server configuration to be the new location, and then do the reinstallation or upgrade.

The menu mentioned in the following section is for Windows installations. For non-Windows platforms, use the .zip files on the CD: StarTeam Extensions.zip, SECaliberRM.zip, StarTeam Extensions Documentation.zip, and StarTeam Extensions Samples.zip.

To install StarTeam Extensions:

1. If the StarTeam master setup program does not start automatically from the CD, click setup.exe.
2. From the StarTeam master setup menu, select StarTeam Extensions.
3. From the resulting Install StarTeam Extensions menu, select StarTeam Extensions.
4. A third menu appears on the screen. From it you can select which parts of StarTeam Extensions to install. The parts are:

    - Install Runtime

      Click Install Runtime to install the files that are critical to each server configuration that uses workflow, APE forms, and/or custom tools.

      The Runtime should be installed on the same computer as the Server.

      After installing the Runtime:

        - Copy the StarFlow Extensions folder and its subfolders (Client, Projects, and Utilities) to a unique location for each server configuration.
        - Use the StarFlow Extensions folder (in its new location) to create a project named StarFlow Extensions in each server configuration.
        - Add the files in these folders to the StarFlow Extensions project—except for the user.pdf and the readme.htm files in the StarFlow Extensions folder. These files are this guide and the latest readme.

          If you are already using StarTeam Extensions, check in these files as new revisions of existing files and add any new files.

      The Runtime consists of the StarFlow Extensions folder and three subfolders: Client, Projects, and Utilities.

        The Client folder contains the files that are normally installed with one or more of the clients as well as with StarTeam Extensions. These files have changed very little since initially released. However, if a user is having problems with an APE, give that user the latest files from the Client folder in the StarFlow Extensions project. It may solve the problem.

        starteam-extensions.jar provides the Locator functionality (the retrieving of workflow and APE files) in conjunction with starteam-extensions.jar. The starteam-extensions.jar file is installed with both the Windows client and the Cross-Platform client. The Locator files are installed only with the Windows client.

        - The Projects folder contains a variety of .jar, .xml, and .dll files used by StarTeam Extensions.
        - The Utilities folder contains:
            - GroupConverter.bat, which is obsolete. It converts users and groups for use with the APEs.
            - StarFlowIniConverter.bat, which is obsolete. It is installed for customers who may have old files created with the old StarFlow Designer and need to use it.

        Although the Runtime includes the custom toolbar that can be used with the Cross-Platform client, if you will be writing applications for the toolbar or adding buttons to the toolbar, you also need the StarTeam SDK and the StarTeam Custom Toolbar Framework SDK (which are not part of StarTeam

Extensions). You may need assistance from Borland consultants before using the toolbar.

- Install Documentation

  Click Install Documentation to add the Documentation subfolder to the StarFlow Extensions folder.

  The Documentation can be installed on any computer for your convenience.

  You do not need to copy this folder to each server configuration’s StarFlow Extensions folder, nor do you need to check its files in.

  These online documents (.html files) explain the APIs for the APEs and the custom toolbar.

  This folder does not include the readme file or the *StarTeam Extensions User’s Guide*. They are installed in the StarFlow Extensions folder.

- Install Samples

  Click Install Samples to add the Samples subfolder to the StarFlow Extensions folder.

  The Samples can be installed on any computer for your convenience.

  You do not need to copy these to each server configuration’s StarFlow Extensions folder, nor do you need to check them in.

  If you decide to use some of the sample APE forms, as they are or after some modification, you would copy the *item_type*.Workflow.xml, *name*.propertyeditor.xml, and *name*.jar files to a server configuration’s StarFlow Extensions\Projects folder or an appropriate subfolder of the Projects folder. Then you would check these files in to the StarFlow Extensions project.

  If you decide to use some of the sample tools, as they are or after some modification, you would copy the *name*.tool.xml and the *name*.jar files to the Projects folder as well.

More About the Samples

The Samples folder has samples of:

- APE forms
- Workflows
- Tools for the custom toolbar

Expect to make modifications to all of these to suit your organization’s needs.

Sample APEs

The sample APE forms consist of:

- *APE_name*.html

  The .html file is used with the JBuilder project file. It is set up to be used as a “To Do” list.

- *APE_name*.jar

  The APE’s application file.

- *APE_name*.jpx

  The .jpx file is the JBuilder project file for the form.

- *APE_name*.propertyeditor.xml

  The .property.editor.xml file contains details about the application (.jar file).

- Make*APE_name*Jar.bat

  This batch file may be used to package the compiled APE as a JAR file if your development environment does not support the creation of JAR files.

- ReadMe.txt

  The readme file contains information that you will need as you modify the APE file.

The following APE forms are included among the samples. All of them have accompanying source code.

- ChangeRequest
- LinkTypeChangeRequest
- LinkTypeFile
- LinkTypeRequirement
- LinkTypeTask
- LinkTypeTopic
- Requirement
- Task
- Topic

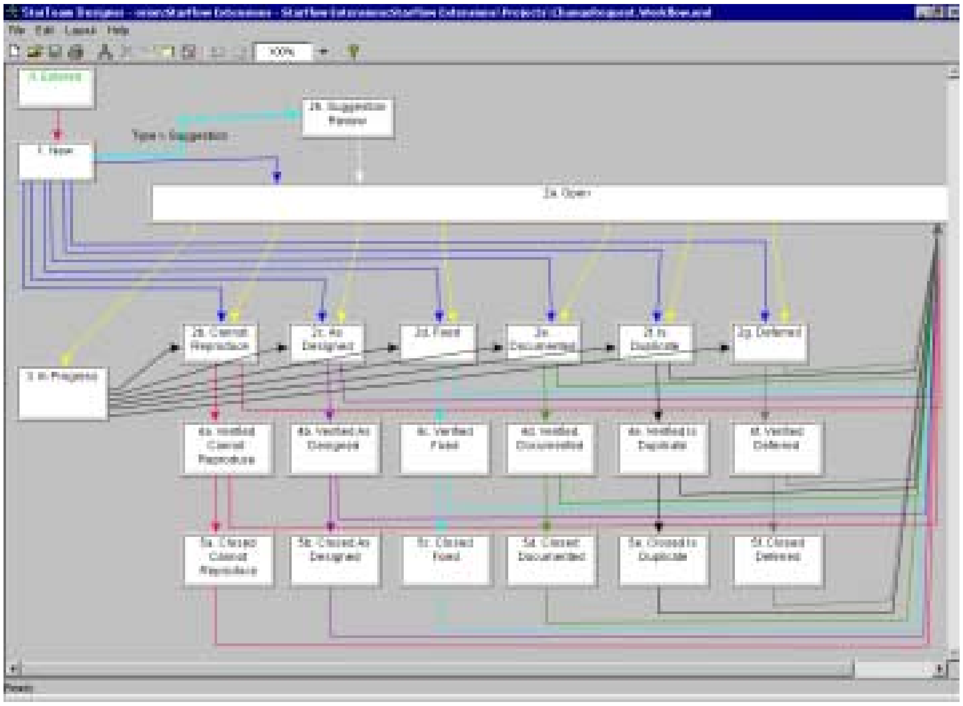
Sample Workflows

The Workflows folder (installed as a subfolder of the Samples folder) includes a sample workflow for each item type, all of them are disabled except the change request workflow (ChangeRequest.Workflow.xml). Each item type that uses a workflow must have a workflow definition file named *item_type*.Workflow.xml.

The disabled workflows are:

- File.Workflow.xml
- Requirement.Workflow.xml
- Task.Workflow.xml
- Topic.Workflow.xml

The following figure shows a graphical representation of ChangeRequest.Workflow.xml, as displayed in the Workflow Designer.



It is similar to the built-in workflow that comes with StarTeam change requests. With Workflow Designer, you can completely redesign this workflow by adding additional steps, changing the behavior of links, modifying the authorization, and adding required, hidden, and disabled fields to each workflow step. However, if you plan to change much, it is easier to create your own workflow “from scratch”, perhaps using this workflow as a model.

It uses the custom enumerated property shown in the following dialog. Although this property exists in the StarDraw sample server configuration that ships with the Server, you would have to create it in your server configuration if you wanted to use it in a project of your own.

Sample Utility

Among the samples is the CRStatusConverter utility. It allows you to change the statuses of change requests that existed before your organization started using customized workflows. The version shipped with this release works with new workflows, not with the old workflows from releases prior to 5.2.

The utility is offered for your convenience, but it may need to be modified to fit your situation and is not a supported utility.

Essentially, any change request that has the entry step’s name as its value for the workflow control property can have that value updated based on the value in its existing change request Status field. This works for any change request with a value in the Status field specifically defined as a dynamic default for any step in the workflow. For details, see “Changing Workflow Control Property Values Automatically” on page 65.

Sample Tools

The Tools folder has several subfolders containing tools that you might want to place on the custom toolbar that comes with StarTeam Extensions and can be used with the Cross-Platform client.

These tools are offered for your convenience, but they may need to be modified to fit your situation. They are not supported tools.

For more information about the toolbar, you need the StarTeam Custom Toolbar Framework SDK used by Borland consultants.

Installing StarTeam Workflow Designer

Workflow Designer outputs .xml files that formalize the steps in a workflow, specifies who will be notified about each step and about exceptions, and so on. Each file is named *item_type*.Workflow.xml and can be used for an entire project or individual views within that project. The StarTeam Extensions workflow engine and Notification Agent read from these files.

You must use Workflow Designer to develop custom workflows. The resulting .Workflow.xml file can be deployed in StarTeam 5.1 or 5.2 installations—as long as the most recent StarTeam Extensions is also installed. Workflows written using the releases of the Workflow Designer prior to 5.2 (which were tightly coupled to Microsoft Outlook) must be re-saved using the new Workflow Designer.

Workflow Designer can be installed separately on any workstation.

With the help of Professional Services, it can be added to the custom toolbar that can be used with the Cross-Platform client.

To install Workflow Designer on a Windows computer:

1. If the StarTeam master setup program does not start automatically, click setup.exe.
2. From the master setup menu, select StarTeam Extensions.
3. From the resulting Install StarTeam Extensions menu, select StarTeam Workflow Designer and follow the directions on the screen.

Installing StarTeam Notification Agent

Notification Agent monitors server configurations to determine who needs to be notified about work to be performed and about exceptions that occur in the workflow process.

It must be installed and set up separately. It is usually installed on the computer that runs the Server (the Windows version), but can be installed anywhere. When Notification Agent is not on the same computer as the Server, the two computers clocks must be synchronized. Notification Agent’s time cannot be ahead of the Server’s time.

When both are installed on the same computer and run as services, Notification Agent must be run as a dependent service and be shut down first. See “Running Notification Agent as a Service” on page 102.

To install Notification Agent:

1. If the master setup program does not start automatically, click setup.exe.
2. From the master setup menu, select StarTeam Extensions.
3. From the resulting Install StarTeam Extensions menu, select StarTeam Notification Agent and follow the directions on the screen.

Using the GroupConverter Utility

Starting with the 5.3 release, the GroupConverter utility is no longer needed. However, it is still included in the Utilities folder. The following few paragraphs explain it former purpose and how to use it.

The GroupConverter utility reads the user groups defined for your server configuration and saves the names of the users in each group in the Groups.xml file. This file is then automatically checked in to the server configuration's StarFlow Extensions\Projects folder.

The syntax to execute this utility can be seen in the GroupConverter.bat file provided in the Utilities folder under C:\Program Files\Borland\ StarFlow Extensions, but is also shown here:

```
%java%-cp"%_CLASSPATH%"com.starbase.starteam.starflow.GroupConverter
Administrator:Administrator@localhost:49201
```

You must edit the file to identify the server configuration, port, and so on.

Caution As this file contains a user name and password, you want to be careful who has access to it both in the application and on the computer, should the file be checked out.

Appendix

B

Notification Tags and Messages

This appendix describes the tags that can be embedded in the notification messages served by the notification agent via email. The tags are replaced at run-time by actual values.

Tag Syntax Overview

The general structure of a tag is as follows:

```
{{tag-name [attrib="attrib-value"]}}
```

In particular:

- The tag must be enclosed in double curly braces ({{ and }}).
- The tag name may contain the following characters: A-Z, a-z, 0-9, _ (underscore), and - (dash). The tag name may not contain spaces or any other special characters.
- The tag may contain zero, one, or more attributes. Attributes are separated by one or more spaces.
- Certain attributes may be required while others are optional.
- The attribute has a name and value separated by the = (equal) sign.
- The attribute name may contain the following characters: A-Z, a-z, 0-9, _ (underscore), and - (dash). The attrib name may not contain spaces or any other special characters.
- The attribute value is quoted. It may contain any characters; however, including quotation mark and backslash characters must be done as follows:
  - The quotation mark (") must be entered as \" (backslash-quote), which will be converted to a quotation mark at run-time.
  - The backslash (\) must be entered as \\ (two backslashes), which will be converted to a single backslash character at runtime.

Tag Reference

The following sections provide a quick reference to the tags, each of which is explained in one of the following sections. The tags are used in the default notification messages and can be used in your own messages.

{{link-base}}

The {{link-base}} tag creates a URL to a resource in the built-in http server. This allows for the placement of additional resources, which can be referenced in the email messages (formatted as html), on the built-in server. For example, you might want to embed a company logo in each email notification.

Attributes

None.

Examples

The following excerpt

```
<IMGheight="73"src="{{link-base}}/logo.gif"width="287"align="center">
```

is converted to:

```
<IMG height="73" src="http://agent_server:8080/logo.gif" width="287"
align="center">
```

which results in the logo.gif file to be pulled from the virtual root directory of the http server built into Notification Agent.

{{part-boundary}}

The {{part-boundary}} tag indicates a boundary in a multi-part mime message. The format is assumed to be html or plain. Anything that is not html will be considered plain text. The html part is detected by searching for the <html> and </html> tags.

Attributes

None.

Examples

```
The plain part.
  {{part-boundary}}
  <HTML>
    <HEAD>
    </HEAD>
    <BODY>
      <FONT STYLE="color:blue;font-weight:bold"> The fancy part.</FONT>
    </BODY>
  </HTML>
```

{{project-name}}

The {{project-name}} tag is used to output the name of the application project.

Attributes

None.

Examples

The following template excerpt from a subject line:

```
{{type-name}} {{property name=\"@primary-descriptor\"}} was created in
{{project-name}}.
```

produces for CRs:

```
Change Request 22563 was created in project Cygnus.
```

{{property}}

The {{property}} tag is dereferenced by Notification Agent using the property value from the corresponding item object. For example,

```
{{property name="Status"}}
```

is converted to the current value of the Status property. The name is the internal identifier for the property, not the display name.

The tag has a number of style attributes that allow for emphasizing the value text if the value has changed from the previous revision of the item. For example, if the Status property changed from New to Open, and causes a notification to occur, the tag can produce the following output in the notification message:

```
Status: <FONT STYLE="color:blue;font-weight:bold"> Open </FONT>
```

Note that style and modified-style must contain valid html style attributes, or they will not render properly. Notification Agent does not perform any syntax checks on the values of the style attributes.

Attributes

Name: name

Optional: no

Permitted Values: Item-specific property identifiers, or @primary-descriptor, @workflow-property, @responsibility-property

Comment: This is the property name as recognized by the application. @primary-descriptor is a symbolic reference to a primary descriptor property of a given application type. For a file, the primary descriptor is its name. For a change request, requirement, task, or topic, the primary descriptor is its number.

Name: style

Optional: yes

Permitted Values: text

Comment: This is the default style. This attribute is only applicable in the html part of the message. In the plain text part, it is ignored.

Name: modified-style

Optional: yes

Permitted Values: text

Comment: This style is used if the property value has changed from the previous revision of the item. This attribute is only applicable in the html part of the message. In the plain text part, it is ignored.

Name: modified-prefix

Optional: yes

Permitted Values: text

Comment: This is the text that is placed in front of the dereferenced property value. This attribute impacts output in both the plain text and html formats of the message.

The property name is not the same as the display name shown by the application to the user. It is an internal identifier that is frequently the same as the display name, but is not guaranteed to be. The list of property identifiers are provided in the StarTeam SDK documentation and in the *StarTeam User's Guide*.

Custom (user-defined) properties have the prefix “Usr_” in front of their internal identifier, for example, Usr_WorkflowStatus.

The Workflow Designer provides an easy way to insert property references (the entire property tag) into the message subject or body.

Examples

The following template excerpt:

```
Status: {{property name="Status" style= "color:black" modified-style=
"color:blue;font-weight:bold"}} <B>
```

will change at run-time to:

```
Status: <FONT STYLE="color:black"> Open </FONT>
```

if the Status value did not change. If it did change, the tag will produce:

```
Status: <FONT STYLE="color:blue;font-weight:bold"> Open </FONT>
```

Similarly, if you use:

```
Status: {{property name="Status" modified-prefix= "***Modified*** "}}
```

the out put will be:

```
Status: Open
```

for an unchanged Status property and:

```
Status: ***Modified*** Open
```

for a changed Status property.

{{property-style}}

The {{property-style}} tag is used to avoid defining style attributes on every {{property}} tag. The scope of the {{property-style}} tag is the mime message part in which it is placed (if the mime message is defined). The tag can be placed anywhere in the message, but Borland recommends placing it at the beginning of the mime part.

Attributes

Name: style

Optional: yes

Permitted Values: text

Comment: This is the default style. This attribute is only applicable in the html part of the message. In the plain text part, it is ignored.

Name: modified-style

Optional: yes

Permitted Values: text

Comment: This the style used if the property value has changed from the previous revision of the item. This attribute is only applicable in the html part of the message. In the plain text part, it is ignored.

Name: modified-prefix

Optional: yes

Permitted Values: text

Comment: This is the text that is placed in front of the dereferenced property value. This attribute impacts output in both the plain text and html formats of the message.

Examples

The following template excerpt:

```
Status: {{property name="Status" style= "color:black" modified-style=
"color:blue;font-weight:bold"}} <B>

Synopsis:{{propertyname="Synopsis"style="color:black"modified-style=
"color:blue;font-weight:bold"}} <B>

Responsibility: {{property name="Responsibility" style= "color:black"
modified-style= "color:blue;font-weight:bold"}} <B>
```

is equivalent to:

```
{{property-stylestyle="color:black"modified-style="color:blue;font-
weight:bold"}}

Status: {{property name="Status"}} <B>

Synopsis: {{property name="Synopsis"}} <B>

Responsibility: {{property name="Responsibility"}} <B>
```

{{property-table}}

The {{property-table}} tag can be used to output a list of application properties with their corresponding labels in a two-column table.

This tag provides a number of ways to define which properties are to be included and how they are to be rendered.

Attributes

Name: contents

Optional: no

Permitted Values: all, standard, descriptors

Comment: This defines the list of properties to be included in the table.

- all means that all properties are included
- standard means that all but advanced properties are included

- descriptors means that only descriptor properties are included, with the primary descriptor always the first to be listed

The property name is not the same as the display name shown by the application to the user. It is an internal identifier that is frequently the same as the display name, but is not guaranteed to be. The list of property identifiers are provided in the StarTeam SDK documentation and in the *StarTeam User's Guide*.

Custom (user-defined) properties have the prefix "Usr_" in front of their internal identifier, for example, Usr_WorkflowStatus.

Examples

The following template excerpt defined in the message template for a Change Request:

```
{{property-table contents="descriptors" label-style="color:blue;font-
weight:bold" modified-style="color:red;font-weight:bold"}}
```

will change at runtime either to the following html format:

```
<TABLE>

<TR><TD STYLE="color:blue;font-weight:bold">CR Number</TD><TD>242</TD></TR>

<TR><TD STYLE="color:blue;font-weight:bold">Synopsis</TD><TD>This is a test
CR.</TD></TR>

</TABLE>
```

or to the following plain-text format:

CR Number: 242

Synopsis: This is a test CR.

The plain-text versus html behavior of this tag is determined by the type of the mime part it is placed in. If the tag is found inside an html part, it will produce html output. If the tag is found in a plain-text part, it will produce plain text output.

{{request-timing}}

The {{request-timing}} tag is used to provide information about the timing of a single cycle of the responsibility assignment request.

Attributes

Name: value

Optional: no

Permitted Values: duration, deadline

Comment: This attribute defines what this tag produces. See examples.

Examples

The following template excerpt:

```
This request expires in {{request-timing value="duration"}}, that is, on:
{{request-timing value="deadline"}}.
```

produces:

```
This request expires in 10 minutes, that is, on: Wednesday, September 10,
2002 6:12:35 PM PDT.
```

{{responsibility-assignment}}

The {{responsibility-assignment}} tag is used to insert a response link in the body of the responsibility assignment request message. The user receiving this email message can either accept or decline responsibility for the item indicated in the message. Notification Agent dereferences this tag at runtime to a set of instructions for the mail client to execute. The result is an automated response that is back to Notification Agent.

The response instructions can take a form of a URL or a "mailto:" command. In the first case, the mail client would invoke an Internet browser and the browser would send an http request to the http server built into Notification agent. The http request is processed by Notification Agent, and a page is sent back to the browser with information about the result of the operation (error or success).

In the latter case, the mail client reacts by creating a new message window, with a pre-formatted email message that can be mailed by the user to the Notification Agent mailbox. Notification Agent retrieves the email response from the mailbox and processes it accordingly. If the confirm attribute is set, Notification Agent notifies the user about the success of the operation. Errors during processing are always reported via email.

In either case, the {{responsibility-assignment}} tag is converted into a command with a set of parameters that uniquely identify the user, item and request. The user is not required to enter any new data while executing the response.

The {{responsibility-assignment}} tag should be placed in a hypertext link for convenient execution.

Attributes

Name: response

Optional: no

Permitted Values: accept, decline

Comment: Properly formed responsibility assignment request messages should contain this tag twice; once with response=accept and once with response=decline, to give the user a choice.

Name: method

Optional: no

Permitted Values: http, mail

Comment: The mode of responding to the responsibility assignment request: either by using the built-in http server or via email.

Name: confirm

Optional: yes

Permitted Values: yes, no, error

Comment: The default value is yes. Applicable to mail responses. Http responses always trigger confirmation.

Examples

The following template excerpt:

```
<A href="{{responsibility-assignment response="accept" method="mail"}}">Click
here if you accept</A>
```

```
<A href="{{responsibility-assignment response="decline"
method="mail"}}">Click here if you decline</A>
```

would be converted to something like:

```
<A href="mailto:RAResponse:TaskID=5103320a-3723-fc51-3237-
51fcef000000:RequestID=5103320a-3742-fc51-5137-
51fcef000000:Response=accept">Click here if you accept</A>
```

```
<A href="mailto:RAResponse:TaskID=5103320a-3723-fc51-3237-
51fcef000000:RequestID=5103320a-3742-fc51-5137-
51fcef000000:Response=decline">Click here if you decline</A>
```

Similarly,

```
<A href="{{responsibility-assignment response="accept" method="http"}}">Click
here if you accept</A>
```

```
<A href="{{responsibility-assignment response="decline"
method="http"}}">Click here if you decline</A>
```

would be converted to something like:

```
<A href="http://agent_server:8080/NA/AssignTask?TaskID=5f05320a-7918-fc9d-
2779-9dfcef000000&RequestID=5f05320a-7937-fc9d-4779-
9dfcef000000&Response=accept">Click here if you accept</A>
```

```
<A href="http://agent_server:8080/NA/AssignTask?TaskID=5f05320a-7918-fc9d-
2779-9dfcef000000&RequestID=5f05320a-7937-fc9d-4779-
9dfcef000000&Response=decline">Click here if you decline</A>
```

The actual email message might look like the one on the next page. Note the “Click here if…” links. Clicking on the link will invoke the browser if method="http" was used, or it will create a new email message if method="mail" was used.

{{step-timing}}

The {{step-timing}} tag is used to provide information about the timing of a single workflow step.

Attributes

Name: value

Optional: no

Permitted Values: duration, deadline

Comment: This attribute defines what this tag produces. See examples.

Examples

The following template excerpt:

```
This step expires in {{step-timing value="duration"}}, that is, on: {{step-
timing value="deadline"}}.
```

produces:

```
This step expires in 10 minutes, that is, on: Wednesday, September 10, 2002
6:12:35 PM PDT.
```

{{type-name}}

The {{type-name}} tag is used to output the application item type name, for example, File, Change Request, Requirement, Task, or Topic.

Attributes

None.

Examples

The following template excerpt from a subject line:

```
{{type-name}} {{property name=\"@primary-descriptor\"}} was created.
```

produces for CRs:

```
Change Request 22563 was created.
```

{{view-name}}

The {{view-name}} tag is used to output the application view name.

Attributes

None.

Examples

The following template excerpt from a subject line:

```
{{type-name}}{{propertyname=\"@primary-descriptor\"}}wascreatedinview
{{view-name}}.
```

produces for CRs:

```
Change Request 22563 was created in view Cygnus.
```

Default Notification Messages

The default notification message templates are expressed using message tags. The tags are resolved by the Notification Agent at run-time.

The body portion of the default templates shows both the plain text for email and the html tags for an http server. This explains the redundancy.

In some cases, the subject and body text is broken into multiple lines. This was done while formatting this document. The actual templates are correct.

The messages include:

- Status notification messages indicating that items have reached certain steps in the workflow process
- Responsibility request messages that ask users to accept responsibility for items at specific workflow steps
- Confirmation messages indicating that Notification Agent has received users' acceptances or refusals
- Request exception messages that notify specific individuals when no one accepts responsibility for items at specific steps
- Step exception messages that notify specific individuals when steps take longer than expected

Status Notification Message

A status notification message is sent when an item’s workflow control property changes its value. The message is sent to the users defined in the status notification list. If this list is empty, no message is sent.

You can create your own status notification message to replace this message.

Subject Line

```
Status Notification: {{type-name}} {{property name="@primary-
descriptor"}} entered workflow state: {{property name="@workflow-
property"}}.
```

Body

```
{{property-style modified-prefix="***"}}

This item is located in view {{view-name}} in project {{project-
name}}.

{{property-table contents="standard"}}

{{part-boundary}}

{{property-style modified-style="color:red;font-weight:bold"}}

<HTML><HEAD></HEAD><BODY>

This item is located in view {{view-name}} in project {{project-
name}}. <P>

{{property-table contents="standard"}}

</BODY></HTML>
```

Responsibility Request Message

The responsibility request message, also known as the step notification message, is sent to the users defined on the Participants tab for the step. There are two default messages. One asks users to accept or decline responsibility for an item when the item enters that workflow step. It is sent when either the Http Server Address text box or the Mail Store Address text box has a value.

The second is sent when both the Http Server Address text box and the Mail Store Address text box are empty. It does not require a response.

You can create your own responsibility request (step notification) message to replace these messages.

Subject Line for Default Message that All Messages

```
Request: Responsibility assignment request for {{type-name}}
{{property name="@primary-descriptor"}} at workflow step {{property
name="@workflow-property"}}.
```

Body of Default Message that Expects an Email Response

```
This is a request to handle the item indicated below. If you
accept, responsibility for the item will be assigned to you.

Use this link to accept: {{responsibility-assignment method="mail"
response="accept"}}
```

```
Use this link to decline: {{responsibility-assignment method="mail"
response="decline"}}

{{property-table contents="standard"}}

{{part-boundary}}

<HTML><HEAD></HEAD><BODY>

This is a request to handle the item indicated below. If you accept,
responsibility for the item will be assigned to you. <P>

<A href="{{responsibility-assignment method="mail" response="accept"}}">Click
here if you accept</A> <BR>

<A href="{{responsibility-assignment method="mail"
response="decline"}}">Click here if you decline</A> <P>

{{property-table contents="standard"}}

</BODY></HTML>
```

The email message includes “Click here if…” links. Clicking the link invokes the browser if method="http" was used, or creates a new email message if method="mail" was used.

Body of Default Message that Expects an HTTP Server Response

```
This is a request to handle the item indicated below. If you accept,
responsibility for the item will be assigned to you.

Use this link to accept: {{responsibility-assignment method="http"
response="accept"}}

Use this link to decline: {{responsibility-assignment method="http"
response="decline"}}

{{property-table contents="standard"}}

{{part-boundary}}

<HTML><HEAD></HEAD><BODY>

This is a request to handle the item indicated below. If you accept,
responsibility for the item will be assigned to you. <P>

<A href="{{responsibility-assignment method="http" response="accept"}}">Click
here if you accept</A> <BR>

<A href="{{responsibility-assignment method="http"
response="decline"}}">Click here if you decline</A> <P>

{{property-table contents="standard"}}

</BODY></HTML>
```

Body of Default Message that Expects No Response

```
This item is located in view {{view-name}} in project {{project-name}}.

Please use StarTeam to assign the responsibility for this item to yourself.
You may not have been the only person notified that this item needs
attention. If someone else has already accepted responsibility for this item
at this step, you may not need to.

{{property-table contents="standard"}}

{{part-boundary}}

<HTML><HEAD></HEAD><BODY>

This item is located in view {{view-name}} in project {{project-name}}.<P>
```

```
Please use StarTeam to assign the responsibility for this item to yourself.
You may not have been the only person notified that this item needs
attention. If someone else has already accepted responsibility for this item
at this step, you may not need to.<P>
{{property-table contents="standard"}}
</BODY></HTML>
```

Confirmation Messages

Confirmation messages are sent to the user who responded to the responsibility assignment request. They are always sent through email so users have a record of what they have accepted and declined. The confirmation messages fall in three categories:

- Confirmation of acceptance
- Confirmation of the decision to decline
- Processing error message
- Parameter error message

The confirmation messages for acceptance, the decision to decline, and processing errors are generated in a dynamic fashion. They follow the same format except for the specific piece of text inserted into the message at the runtime, which indicates acceptance, decline, or error conditions.

In contrast, the confirmation messages for the parameter errors may contain only limited information because it is impossible to locate tasks or requests because of invalid input data. Also, suppose a user responds to a request that another user already accepted, or after all the requests expired or the step itself has expired. In this latter set of cases, the parameters are correct, but there is no longer an active task to be found.

The general template, shown below, contains a placeholder tag {{responsibility-assignment-result}} which is replaced by the appropriate text for the confirmation messages for acceptance, the decision to decline, and processing errors.

The {{responsibility-assignment- result}} tag resolves to:

- Confirmation of acceptance messages:
  - Your response was processed successfully. Thank you for accepting responsibility for the item indicated below.
  - Thank you for accepting responsibility for the item indicated below. However, no action was necessary since you already had the responsibility for this item.
- Confirmation of decline messages:
  - Your response was processed successfully. Your decision to decline responsibility for the item indicated below has been acknowledged.
- Processing error messages:
  - Your response was past due. Another user was asked to take responsibility for this item.
  - Your acceptance of responsibility could not be processed because of the following error received from the server while attempting to set the responsibility property: *error_message*

    You have *x* minutes to try to remedy the problem and set the responsibility manually. Otherwise the next user on the list will be asked to accept responsibility for this item.

- Your acceptance of responsibility could not be processed because of the following error received from the server while attempting to set the responsibility property: *error_message*

  You have *x* minutes to try to remedy the problem and set the responsibility manually. Otherwise, since you are the last person in the participant list, a request expiration message will be sent to alert others that this item needs their attention.

Subject Line

```
Confirmation:Yourresponsetoresponsibilityrequestforthe{{type-name}}
{{property name="@primary-descriptor"}} at workflow step {{property
name="@workflow-property"}}.
```

Body

```
{{responsibility-assignment-result }}

{{property-table contents="standard"}}

{{part-boundary}}

<HTML><HEAD></HEAD><BODY>

{{responsibility-assignment-result }} <P>

{{property-table contents="standard"}}

</BODY></HTML>
```

Note From the http server, users see “Your response was processed successfully.”; “Your response was processed successfully. A detailed confirmation will be mailed to you.”; and “An error occurred while processing your response. A detailed error report will be mailed to you.”

For parameter error messages, the general format of the message consists of the description of the error and information returned from the server. These are the possible error messages:

- TaskID is invalid or missing.
- RequestID is invalid or missing.
- Acceptance flag is invalid or missing.
- The item moved to another workflow step.
- The responsibility for this item was already assigned to someone else.
- There is no matching Responsibility Assignment task active. The response parameters may be incorrect, another user may have already accepted responsibility, the request or step may have expired, or the item may have progressed to another workflow step.
- There is no matching request for the indicated Responsibility Assignment task.

Request Exception Message

A request exception message is sent when no one accepts responsibility for an item. This message is sent to the users defined on the Exception Notification tab.

You cannot create your own request exception message.

Subject Line

```
Request Exception: No one accepted responsibility for {{type-name}}
{{property name="@primary-descriptor"}} at workflow step {{property
name="@workflow-property"}}.
```

Body

```
None of the workflow participants agreed to accept responsibility
for the item below.

{{property-table contents="standard"}}

{{part-boundary}}

<HTML><HEAD></HEAD><BODY>

None of the workflow participants agreed to accept responsibility
for the item below. <P>

{{property-table contents="standard"}}

</BODY></HTML>
```

Step Exception Message

A step exception message is sent when a workflow step becomes overdue (the item's workflow control property has not been modified). This message is sent to the users defined on the Exception Notification tab.

You cannot create your own step exception message.

Subject Line

```
Step Exception: Workflow step {{property name="@workflow-
property"}} for {{type-name}} {{property name="@primary-
descriptor"}} expired.
```

Body

```
  The current workflow step for the item below has not been completed
  in the time allowed.

  {{property-table contents="standard"}}

  {{part-boundary}}

  <HTML><HEAD></HEAD><BODY>

  The current workflow step for the item below has not been completed
  in the time allowed. <P>


  {{property-table contents="standard"}}
</BODY></HTML>
```

Appendix

C

Upgrading StarTeam Extensions

If you are using an older version of StarTeam, you may need to make some changes to your APEs and other StarTeam Extension features.

Caution Upgrading StarTeam Server or installing an interim release of StarTeam Extensions overwrites some files in the StarFlow Extensions folder (usually located at C:\Program Files\Borland\StarFlow Extensions). If you are using the default StarFlow Extensions folder as the working folder for any server configuration's StarFlow Extensions project, Borland Software Corporation recommends that you check in everything you need that is not current prior to reinstalling or upgrading the server. Move your existing StarFlow Extensions folder to a new location (for example, C:\config_name\StarFlow Extensions), change the working folder for the StarFlow Extensions project in that server configuration to be the new location, and then install the new release.

Upgrading Existing APEs

If you have existing APE forms implemented in Java that were written to operate with StarTeam 4.2 SP3 (before the introduction of the Form Server), a small amount of work must be done to upgrade the existing forms.

You must remove or replace a few lines of code, recompile, and check the new APE form into the StarFlow Extensions project. Forms in StarTeam 4.2 SP3 have a short-lived, process-centric life-cycle. Now forms have a thread-centric life-cycle within a long-lived process. The form's life-cycle no longer corresponds to the termination of the Java VM in which the form is running.

Upgrade Procedure

This procedure must be performed for each form that was designed for the initial StarTeam 4.2 architecture which will now be used with the StarTeam Form Server. The instructions below are for the ChangeRequestForm sample shipped with StarTeam 4.2 and should be repeated on the corresponding files in the Requirement sample.

1 Open the ChangeRequestLauncher.java file.

2 At the beginning of the class definition, you'll find a member variable declaration and a constructor as follows:

```
private ChangeRequestForm m_dummy;

  public ChangeRequestLauncher() {
     final Thread t = new Thread(new Runnable() {
        public void run() {
              m_dummy = new ChangeRequestForm("dummy");
               }
        });
     t.start();
  }
```

Delete these lines. The only function they served originally was to improve initialization time. This is no longer necessary and will cause excessive resource usage if left.

3 Further down in the class definition, you will find a method named show. Near the end of the show method body, you will find a call to form.addWindowListener as follows:

```
form.addWindowListener(new WindowAdapter() {
    public void windowClosed(WindowEvent ev) {
         System.exit();
    }
  });
```

Delete these lines. This was previously necessary because it was the only way to terminate the form’s process. This requirement no longer exists with the introduction of the Form Server, and will cause the Form Server to continually stop and restart if left.

4 Just below the above lines, in the same method, you will find a try/catch block with a call to System.exit in the catch block as follows:

```
try {
    form.show();
} catch (Exception e) {
    e.printStackTrace();
    System.exit(1);
}
```

Replace the call to System.exit with a call to form.dispose() as follows:

```
try {
    form.show();
} catch (Exception e) {
    e.printStackTrace();
    form.dispose();
}
```

5 Now recompile the form, create the jar file, and check it into the StarFlow Extensions project using the normal procedure and you are done.

Upgrading to Support the Cross-Platform Client

If your existing APE forms are implemented in Java and were written to operate with StarTeam 4.2 SP3, you must make some changes to support the Cross-Platform client.

The changes to upgrade forms so that they work with the StarTeam Form Server are also necessary in order to work with the Cross-Platform client. See "Upgrading Existing APEs" on page 131 for details. Please apply those changes to your forms in addition to the changes described below.

Existing application forms can be made to work with the Cross-Platform client by adding a single adaptor method and adding a few lines of code to the show method.

Upgrade Procedure, Part 1

Because the Cross-Platform client is written in Java, the Java forms shipped with the application can be invoked directly (as opposed to the out-of-process invocation performed by the Windows client). The form must use a slightly different entry point format to gain the accompanying efficiencies.

The procedure is simply a matter of copying a method found in the currently shipping sample forms to the corresponding class in your form.

1 If you have a form based on the ChangeRequest sample form from the initial release of the application, find the ChangeRequestLauncher class in the ChangeRequestLauncher.java file in the currently shipping sample.

2 Copy the method with the following signature (note that there is another method with the same name but a different signature; ignore that one):

```
public static void main(Item[] items, int firstItemIndex,
           boolean isHistoryItem)
      throws IOException {
```

3 Paste this method into the ChangeRequestLauncher class (or corresponding derivative class) in your form.

4 Now recompile the form, create the .jar file, and check it into the StarFlow Extensions project using the normal procedure and you are done.

5 Repeat this procedure for forms based on the Requirement sample.

Upgrade Procedure, Part 2

To improve performance, data is refreshed on an as-needed basis. Some categories of data, like user and label lists, are only refreshed when they are displayed. To see the latest data, you must refresh these categories of data upon form invocation.

1 If you have a form based on the ChangeRequest sample form from the initial release of the application, find the ChangeRequestLauncher class in the ChangeRequestLauncher.java file in the currently shipping sample.

2 Find the method with the following signature:

```
public void show(ItemWrapper[] items, int currentItemIndex,
    boolean refreshItems, boolean editable) {
```

3 Paste the following lines in the beginning of the method:

```
// User lists and label lists don't get refreshed when the user hits F5, so
// we must do so here. Should you ever add a field to the form that was of type
// OBJECT_ID with a subtype other than USER or LABEL, then you would be required
// to call Catalog.refresh on the appropriate Catalog instance here.
final ItemWrapper sample = items[0];
sample.getView().getServer().refreshUsers();
sample.getView().refreshLabels();
```

4 Now recompile the form, create the .jar file, and check it into the StarFlow Extensions project using the normal procedure and you are done.

5 Repeat this procedure for all forms.

Upgrading Requirements from StarTeam 4.2 SP3

In StarTeam 4.2, the Requirement component has the internal name of CStRequirement, while the name in StarTeam 5.x is Requirement, following the standard for the other application components. StarTeam Extensions work with either StarTeam 4.2 or StarTeam 5.x, but you must change the name Requirement in the getFriendlyTypeName method found in the RequirementLauncher source code to CStRequirement, to work with the 4.2 version of the application Server.

Upgrading Workflows from Pre-5.2 Releases

Workflows created in relesases of StarTeam Extensions prior to 5.2 used Microsoft Outlook as their foundation. This version of StarTeam Extensions provides full workflow functionality that is fully integrated with and wholly contained within StarTeam.

Appendix

D

Differences in the New and Old Designers

Since the 5.2 release, Workflow Designer has not been tied to Outlook. This makes it much more flexible. However, some features used with the old StarFlow Designer that are either missing or implemented in a different way. This appendix addresses those differences.

Some of the tabs in the old *Step Properties* dialog are missing. These tabs apply to the old StarFlow product, which allowed users to process change requests using email in Outlook. The StarFlow tab had functionality that was used by both the old StarFlow product and the old StarFlow Extensions product.

All of the features on the tab can be implemented in different ways.

Mapping Steps

Because StarTeam has a built-in workflow based on the Status property, all the sample change request workflows use a different property named Usr_StarFlowStatus. Similarly, any user creating an APE for change requests would have to base it on a property other than the Status property.

The Map This Step To StarTeam Status list allowed you to keep the two status properties synchronized. For example, the “1. New” step would be mapped to the “New” status in the application.

This can still be done using dynamic defaults. If the user selects a possible status from the Usr_StarFlowStatus property (or some other status property), the Status property can be set to a corresponding value. However, unlike the earlier versions of StarTeam Extensions, the user can change this setting after it appears on the APE form. If you do not want the user to be able to change this setting, hide or disable the form field for this property.

Since you are unlikely to display both of the Status properties, mapping is probably not important.

For more information about dynamic defaults, see “Controlling What Properties the Form Displays” on page 50 and “Controlling Dynamic Changes to Forms” on page 50.

Assigning Responsibility

The Assign Responsibility For This Step To list allowed you to make a particular person responsible for an item at this step. This, too, can be done with dynamic defaults. Remember that if you do not want the user to be able to change this setting, you will need to disable or hide the form field for this property at this step.

For more information about dynamic defaults, see “Controlling What Properties the Form Displays” on page 50 and “Controlling Dynamic Changes to Forms” on page 50.

Assigning Revision Comment

The When This Step Is Complete, Add This Revision Comment text box allowed you to use a specific revision comment for an item at this step. This, too, can be done with dynamic defaults. If you do not want the user to be able to change this setting, you will need to disable or hide the form field for this property at this step. If you want users to be able to add to this comment, they would need to be trained never to delete the initial comment, because the form field would have to be enabled and displayed.

For more information about dynamic defaults, see “Controlling What Properties the Form Displays” on page 50 and “Controlling Dynamic Changes to Forms” on page 50.

Using the ‘Next Build’ Variable

For change requests, the old StarFlow Designer allowed you to use the When This Step Is Complete, Set ‘Addressed In Build’ To ‘Next Build’ check box to make the Next Build variable the value of the Addressed In Build property. The new Workflow Designer does not have this check box. However, you can use dynamic defaults to set the Status property to Fixed or Documented, either of which causes the built-in workflow for application change requests to set the Addressed In property to ‘Next Build’. Since you usually do not display the StarTeam Status property on a change request APE form, setting it to a status value that is meaningless or misleading in the context of your workflow is of little importance.

When the next build label is created, ‘Next Build’ is replaced by the name of that label.

Missing Functionality

In the old StarFlow Designer, you were able to specify days of the week and dates in the year as time periods to be ignored by Notification Agent. For example, you might not want users to be notified about items or to have timing exceptions occur on weekends or holidays. The new StarFlow Designer does not include this feature.

Index