Borland StarTeam 14.0 Update 1

Workflow
Extensions User
Guide

Contents

Introduction

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

Documentation

The documentation is your guide to using the product suite. StarTeam documentation is provided in several formats: online help, HTML, and Adobe PDF. Documentation is available from the **Help** menu within the product.

If you are using a Microsoft Windows system, you can locate documentation for the products by clicking **Start** > **Programs** > **Borland StarTeam** > **<Product>** > **Documentation**. The Documentation menu lists all of the available documentation for the selected product.

You can also download documentation directly from: *http://supportline.microfocus.com/productdoc.aspx*.

HTML Documentation

Readme files can be found directly under the root installation directory (or on the root of the installation CD). For documentation available in other languages (Japanese, French, or German), the language-specific versions of the release notes are indicated with an appropriate `_countrycode` in the filename. For example, `readme_ja.html` contains release note information for the Japanese language. PDF manuals are located in the `Documentation` subfolder on the product CDs.

Adobe PDF Manuals

The following documentation is provided in Adobe PDF format. All manuals distributed in Adobe Acrobat (.PDF) format require Adobe Acrobat Reader to display them. The installation program for Adobe Acrobat Reader is available from the Adobe web site at: www.adobe.com.

Release Notes	Contains system requirements and supported platforms for the products.
StarTeam Installation Guide	The StarTeam Installation Guide contains detailed instructions for installing and configuring the core StarTeam products.
StarTeam Server Help	This manual is identical to the online help version.
StarTeam Cross-Platform Client Help	This manual is identical to the online help version.
StarTeam Command Line Tools Help	Explains how to use the command-line tools and provides a reference for the various commands.
StarTeam File Compare/Merge Help	This manual is identical to the online help version of the StarTeam File Compare/Merge help.
StarTeam Workflow Extensions User's Guide	Explains how to design and manage StarTeam Extensions such as alternate property editors (APEs). It also covers the StarTeam Workflow Designer and StarTeam Notification Agent.
StarTeamMPX Administrator's Guide	Explains the basic operation and architecture of a StarTeamMPX system, and presents instructions on installing and configuring the StarTeamMPX components.

Note: Depending upon which products you purchased and installed, not all of the application manuals will be on your system.

Contacting Support

Borland (a Micro Focus company) is committed to providing world-class services in the areas of consulting and technical support. Qualified technical support engineers are prepared to handle your support needs on a case-by-case basis or in an ongoing partnership. Micro Focus provides worldwide support, delivering timely, reliable service to ensure every customer's business success.

For more information about Borland support services, visit the Micro Focus SupportLine Web site at *http:// supportline.microfocus.com* where registered users can find product upgrades as well as previous versions of a product. Additionally, users can find the Knowledge Base, Product Documentation, Community Forums, and support resources.

When contacting support, be prepared to provide complete information about your environment, the product version, and a detailed description of the problem, including steps to reproduce the problem.

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

StarTeam Workflow Extensions Features

This chapter describes workflow, the StarTeam Workflow Extensions and its components, and gives you a roadmap for implementing workflow in your organization.

Workflow

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, and so on. 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, StarTeam Workflow Extensions along with StarTeam give 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 StarTeam, such as 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, see the *StarTeam Help*.)

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 Extensions

StarTeam Extensions include:

- Alternate Property Editors (APEs), forms written in Java, replace the standard properties dialog boxes 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 StarTeam Cross-Platform Client to start applications that add new features to the application.
- Various other utilities and libraries written in Java.

The default installation folder is `C:\Program Files\Borland\ StarFlow Extensions`, but we recommend copying these files to a different folder for each server configuration that runs on the same StarTeam 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.

StarTeam Workflow Designer

You must use Workflow Designer to develop custom workflows. 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 StarTeam Notification Agent read from these files.

Workflow Designer can be installed separately on any workstation.

StarTeam Notification Agent

The StarTeam 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 StarTeam Server, but it can be installed anywhere. When the StarTeam Notification Agent is not on the same computer as the StarTeam Server, the two computers' clocks must be synchronized. The StarTeam Notification Agent's time cannot be ahead of the StarTeam Server's time.

When changing the enumerated values for fields used to control notification (for example, when updating the workflow for a project or view), restart the Notification Agent so that it updates its cache.

Repository Customization

StarTeam 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 StarTeam and, to some degree, by using StarTeam Workflow Designer. See the customization chapter in *StarTeam Help* or the *Administering and Using StarTeam* PDF to learn more about creating custom fields.

StarTeam SDK

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

See the online documentation for APEs (that can be installed with StarTeam Workflow 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 are:

1. Install StarTeam Server, StarTeam Workflow Extensions (StarTeam Extensions, StarTeam Workflow Designer, and StarTeam Notification Agent), and the StarTeam Cross-Platform Client in appropriate places. See the *StarTeam Installation Guide*.

2. Create the necessary projects and folders.

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 may help you develop your workflow model. The flow diagram for the example workflow model explained in this chapter was designed using flowcharting software.

4. Create a custom enumerated property and values on which the workflow is based. (See *Adding and Modifying Custom Properties* and the StarTeam Help 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 the StarTeam Workflow Designer* 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* for more information.

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

   a) 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*, *Understanding Locator*, and *Saving and Exporting Workflows*.

   b) 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.

   c) Activate the workflow in the Test project by setting the appropriate APE settings in that project. See *Configuring StarTeam Projects to Use APEs*.

   d) 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* and *Starting Notification Agent and Notification Setup*.

   e) Use Notification Agent's **Run Now** button to simulate movement from step to step.

   f) Correct the workflow if necessary.

10. Deploy the workflow:

   a) 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*, *Understanding Locator*, and *Saving and Exporting Workflows*.

   b) Activate the workflow for the appropriate projects by setting the appropriate APE settings for those projects. See *Configuring StarTeam Projects to Use APEs*.

   c) 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* and *Starting Notification Agent and Notification Setup*.

   d) When users open items under workflow control, StarTeam Extensions displays the APE form.

   e) 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.

   f) 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.

Using StarTeam Extensions

The chapter explains how to create and configure a StarFlow Extensions project for each server configuration, how to create and deploy APEs, and how to use various utilities included with StarTeam 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, a program used to distribute code and XML updates to client workstations for some clients, will look for a project with the name StarFlow Extensions and use the latest revision it finds for the required work flow elements for the APE. The clients provide 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.

Because of the StarFlow Extensions project:

- You, as the administrator, are guaranteed that the user has connectivity to these files since they reside on StarTeam 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. The clients look for this project when doing anything regarding APEs and workflow. If this project is renamed or missing, StarTeam Extensions functionality will fail.

> ⚠️ **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 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 StarTeam Server with the server configurations that will be using workflow.
3. Start the StarTeam 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) Select the folders to be included in your project.
   b) Drag the folders to an empty spot in the StarTeam Cross-Platform Client window and release the mouse button. The **New Project Wizard** dialog appears. If you drag the folders to an open view window, the StarTeam Cross-Platform Client 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 15, 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** field.

    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 StarTeam Cross-Platform Client. 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** field. 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** field. The address is the computer name or IP address.

4. Enter the correct TCP/IP endpoint in the **Endpoint** field. 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 Help*.

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

Adding Files to the StarFlow Extensions Project

After you create the StarFlow Extensions project, you must add the files that are Not in View. To add files:

1. In StarTeam Cross-Platform Client, 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** button 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 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).

a) Click **File** > **Select** > **Select All** to select all the files simultaneously.

b) Do one of the following:

To quickly add all the files	Click the **Add Files** button on the toolbar to add files without using the **Add Files** dialog.
Click File>Add Files	The **Add Files** dialog appears. 1. Enter a generic description for all files in the **File Description** field. 2. From the **Lock Status** group, make sure that the **Unlocked** option 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. See the *StarTeam Help* for more information. 5. (Optional) If you are using the StarTeam Cross-Platform Client, click **Advanced** to set advanced features. See the *StarTeam Help* for more information. 6. Click **OK**.

As you add new files to StarTeam Cross-Platform Client, 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

To use APEs, you enter `Locator` before the name of the APE in the **Project Properties** dialog box. See "Configuring StarTeam Projects to Use APEs" for more details.

Some clients use a tool called **Locator**; others, such as the StarTeam Cross-Platform Client, use this word as a signal to check the StarTeam Extensions project. The clients open the StarTeam Extensions project, find the correct files automatically, load them into memory and execute the main entry point for the APE .jar file. The resources required by the APE are specified in the `.propertyeditor.xml` file.

With the StarTeam Visual Studio .NET integration, you can user Locator and 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 you upgrade StarTeam Extensions to a new release, you must check in new files to the StarFlow Extensions project.

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 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.

The client uses the folder hierarchy to search for resources in all folders. If it does not locate the files it needs, an error message is displayed.

To add a folder to a view using StarTeam Cross-Platform Client:

1. Right click the **Folder** tree and select **New**.

   The **New Folder Wizard** dialog box 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** field 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** field, 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, an application folder is added 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 box 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 box, 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 the **See file and its properties** right, the **See history** 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 StarTeam Cross-Platform Client 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".
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".

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 with 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 StarTeam Cross-Platform Client, 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, StarTeam Cross-Platform Client 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 StarTeam Cross-Platform Client as a separate tab with the component name.

To edit a particular item's properties, select the item and then click **item_type** > **Properties** from the menu bar. You 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`). See "StarTeam Extensions Samples" for more information.

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 pre-configured 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 LayoutDesigner APE

`LayoutDesignerAPE` has been added as a sample of StarFlow Extensions. `LayoutDesignerAPE` can be used and customized to support rendering a custom Layout Form as an Alternate Property Editor in the StarTeam Cross-Platform Client. `LayoutDesignerAPE` is useful for Enterprise Advantage customers wanting to enable workflow with a form designed in the Layout Designer.

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`, `starteamxx.jar` and `guicomponents.jar` files.

   a) Click **Tools** > **Configure Libraries** from JBuilder's menu bar.
   b) Select the `User Home` folder from the **Configure Libraries** dialog box.
   c) Click **New** to create a `StarTeam Extensions` folder under the `User Home` folder that contains the `starflow-extensions.jar` and `guicomponents.jar` files.

      If you have a `StarFlow` folder under the `User Home` folder, you need to change it.
   d) Enter `StarTeam Extensions` in the **Name** field in the **New Library Wizard** dialog box.
   e) Click **Add**.
   f) On the **Select One or More Directories** dialog, select the `starflowextensions.jar` and `guicomponents.jar` files from the `StarFlowExtensions\Projects` folder at `C:\config_name\StarFlow Extensions\Projects`.
   g) Click **OK** until you return to the **Configure Libraries** dialog box.
   h) Click **Add**.
   i) Select the `User Home` folder from the **Configure Libraries** dialog box.
   j) Click **New** to create a `StarTeam` folder under the `User Home` folder that contains the `starteamxx.jar` file.

      If you have a `StarGate` folder under the `User Home` folder, you need to change it.
   k) Enter `StarTeam` in the **Name** field in the **New Library Wizard** dialog box.
   l) Click **Add**.
   m) On the **Select One or More Directories** dialog box, select the `starteamxx.jar` file (usually located at `C:\Program Files\Borland\ StarTeam SDK x.x\Lib\starteamxx.jar`).
   n) Click **OK** until you return to the **Configure Libraries** dialog box.
   o) Select the `User Home` folder from the **Configure Libraries** dialog box.
   p) Click **New** to create a `GUI SDK` folder under the `User Home` folder that contains the `starteam-gui.jar` file.
   q) Enter `GUI SDK` in the **Name** field on the **New Library Wizard** dialog.
   r) Click **Add**.
   s) On the **Select One or More Directories** dialog, select the `starteamgui.jar` file from the `StarFlowExtensions\Projects` folder (for example, at `C:\config_name\StarFlow Extensions\Projects`).
   t) Click **OK** until you return to the **Configure Libraries** dialog box.

   `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) Click **Add** in the **Palette Properties** dialog box.
   b) On the **Add Page** dialog box, enter `StarTeam Extensions`.
   c) Click **OK**.
   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 **Select a Different Library** dialog box, select the `StarTeam Extensions` library.

i) Click **OK**.

j) From the **Palette Page** list, select the **StarTeam Extensions** 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 box.

l) Click **OK**, and you are ready to use the **StarTeam JavaBean** components.

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 `MakeitemtypeJar.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:\jbuilderx` 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	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	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.

> **Note:** 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. Click **Project** > **Properties**. 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. You enter the word Locator followed by the name of the APE.

Depending on the client, Locator is either a program or a signal to distribute resources, such as .jar and .xml files to client workstations. If specified here, whenever the application requests the APE, the client 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, the client 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

The client 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 client 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
<?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 to the client in the **Project Properties** dialog, for example, "Locator ChangeRequest". The client 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 client 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_typeLauncher.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.

Using the Form Server

The StarTeam form server is a background application that improves the performance and responsiveness of the APE feature in the StarTeam Visual Studio .NET integration. Once loaded, 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 does not need to 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 StarTeam Cross-Platform Client. Buttons on the toolbar start tool applications that are usually written for your organization by Micro Focus consultants. The tools are often time-saving applications that access the StarTeam 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.

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, but 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 but 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.
LinksAndTracesComponent.java	This component shows all of the historical links and new traces and has a similar format to the `LinksComponent` table. Traces are the more advanced link asset created since StarTeam 2009.
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. We recommend 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.

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.

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	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	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	This method returns lists of values that are valid "next" values for a particular property. For instance, this method supplies the values for the 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, isPropertyDisabled	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	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	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	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	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

This section contains known issues for StarTeam Extensions.

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 work-arounds:

- 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 StarTeam 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.

Using the 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.

StarTeam Workflow Designer meets both the simplest and most complex workflow needs. Using StarTeam 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 StarTeam Workflow Designer looks for server descriptions in the `ServerList.st` file. The StarTeam Cross-Platform Clients use the new `starteam-servers.xml` file. If you update a server description in StarTeam Workflow Designer, it will not be recognized by the StarTeam Cross-Platform Clients and vice-versa.

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.

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. Micro Focus 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.

> **Note:** 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 StarTeam 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.

The StarTeam 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 the StarTeam 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 links 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

On a Microsoft Windows platform, click **Start** > **Programs** > **Borland StarTeam** > **Workflow Designer**.

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

Contact Professional Services at *www.microfocus.com/services/*.

Using the Menu

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

File Menu

New	Creates a new workflow on the StarTeam 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	Shuts down the program.

Edit Menu

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 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.

Layout Menu

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.
- 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.

> **Note:** 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.

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" or the customization chapter in the *StarTeam Administrator's Guide*.

Creating a Workflow

When you create a new workflow, you need 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.

1. Click **File** > **New**. The New dialog box appears.
2. Select a server configuration. If the **Log On** dialog box appears, log on using a valid user name and password. Click **New**. The **New Workflow Properties** dialog box displays.
3. 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.
4. 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.
5. 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**).

6. 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 specific name define the constraints for an item as it is initially created.

7. 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 box has its own toolbar allowing you to add/edit server information and select folders

Understanding the "0. Entered" Entry Step

The entry step is a special pass-through step. As a user creates an item, the APE uses the step properties for the "0. Entered" status to control:

- The initial options for form fields (that is required, hidden, or disabled).
- The values that can be selected for the workflow control property at the time that the item is created. (These values are equivalent to steps in the workflow process.)

For example, as a user creates a change request, the only values that can be selected for the workflow control property are those specified for the "O. Entered" step on its "Next Steps" tab.

"0. Entered" is never used as a "Next Step" for any workflow step. That means that users cannot select "0. Entered" as the value for the workflow control property. "0. Entered" can have no links directed to it—only from it. Because no item ever displays this status, there is no reason to set notifications, participants, dynamic defaults, or a duration for this step. They will never be enforced.

While "0. Entered" is never the selected value for the workflow control property in the APE, there always is a selected value. This can cause some initial design confusion. As you create the settings for steps that correspond to values of the workflow control property, be aware that, when a user creates an item, settings assigned to both "0. Entered" and the step that corresponds to the currently selected value of the workflow control property affect the workflow.

For example, suppose that the only "Next Step" for "0. Entered" is "1. New". As the user creates a new change request, the default value (in fact the only possible value) for the workflow control property is "1. New". However, the form field settings assigned to both the "0. Entered" and the "1. New" steps affect the APE. If "0. Entered" requires the Type field and the "1. New" step requires the Platform field, both Type and Platform will be required. Notifications, participants, dynamic defaults, and duration come only from the "1. New" step.

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.

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

1. Click **File** > **Open**.
2. If you have not already logged on to a server configuration, the **Open** dialog box displays a list of servers.

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

If the **Open** dialog box displays a list of server configurations or if you want to log on to a different server configuration, select one. If the **Log On** dialog box appears, log on using a valid user name and password.

The **Open** dialog box 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 box showed these files initially.

3. Locate the file to be worked on. You may have several project and view folders to choose from, many of which have workflow files.
4. Click **Open**.

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

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.

1. Click **File** > **Import**. The **Import Workflow Document** dialog box opens.
2. Locate a `.Workflow.xml` file and click **OK**. The **Select Server** dialog box opens.
3. Select the server configuration that will store this workflow file.
4. If the **Log On** dialog box appears, log on using a valid user name and password.
5. Click **Select**. Workflow Designer displays the workflow steps from the selected file in the workspace.

   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.

1. Click **Edit** > **Workflow Properties**. The **Workflow Properties** dialog box opens.
2. 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.
3. 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.

4. 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).

5. 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.

6. Click **OK**.

You can create a custom enumerated property from the **Workflow Properties** dialog box. 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*.

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 box.
- 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. We recommend 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.

To Create All Steps at Once

Click **Edit** > **Create All Steps** 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. Click **File** > **New Workflow Step**. The mouse pointer changes to a cross-hair.
2. Click and drag in the workspace to define a rectangle. When you release the mouse button, the **New Step** dialog box opens.
3. 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.
4. Click **OK**. A new step with the selected name appears in the workspace. You can resize and align the step at any time.

   > **Note:** You can also use the **Add Value** button to add values to the workflow control property from this dialog box.

To Copy Steps

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

1. Right-click the step.
2. Select **Copy Step**.
3. Position the mouse where you want the new step to be located and right-click and select **Paste Step**.
4. Add links (**Next Steps**) out of this step as appropriate. Be sure to create at least one link into this step from another step as well.

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 box 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, do one of the following in the workspace:

- 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 box. If the connected steps have not already been drawn, this procedure draws them as well.

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

If the steps were already drawn manually, but you defined links using the **Next Step** tab of the **Edit Step** dialog box, 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 box.

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

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

    • Double-click an existing step.
    • Select an existing step and then select the **View** > **Edit Step Properties**.
    • Right-click inside an existing step and select **Step Properties** from the context menu.
    • Select an existing step and then select **Edit** > **Step Properties**.

    The **Edit Step** dialog box displays.
2. 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**. The **Select Next Steps** dialog box displays.
5. Select the steps to transition to from the list of values.
6. Click **OK**.

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.
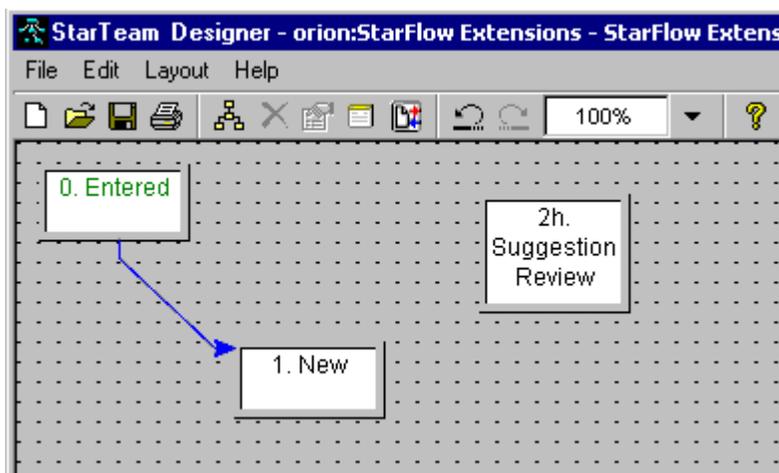
The next figure shows how this conditional transition would be displayed in the **Next Step** tab of the **Edit Step** dialog box. You must use this dialog to create conditional links. They cannot be created manually within the workspace.
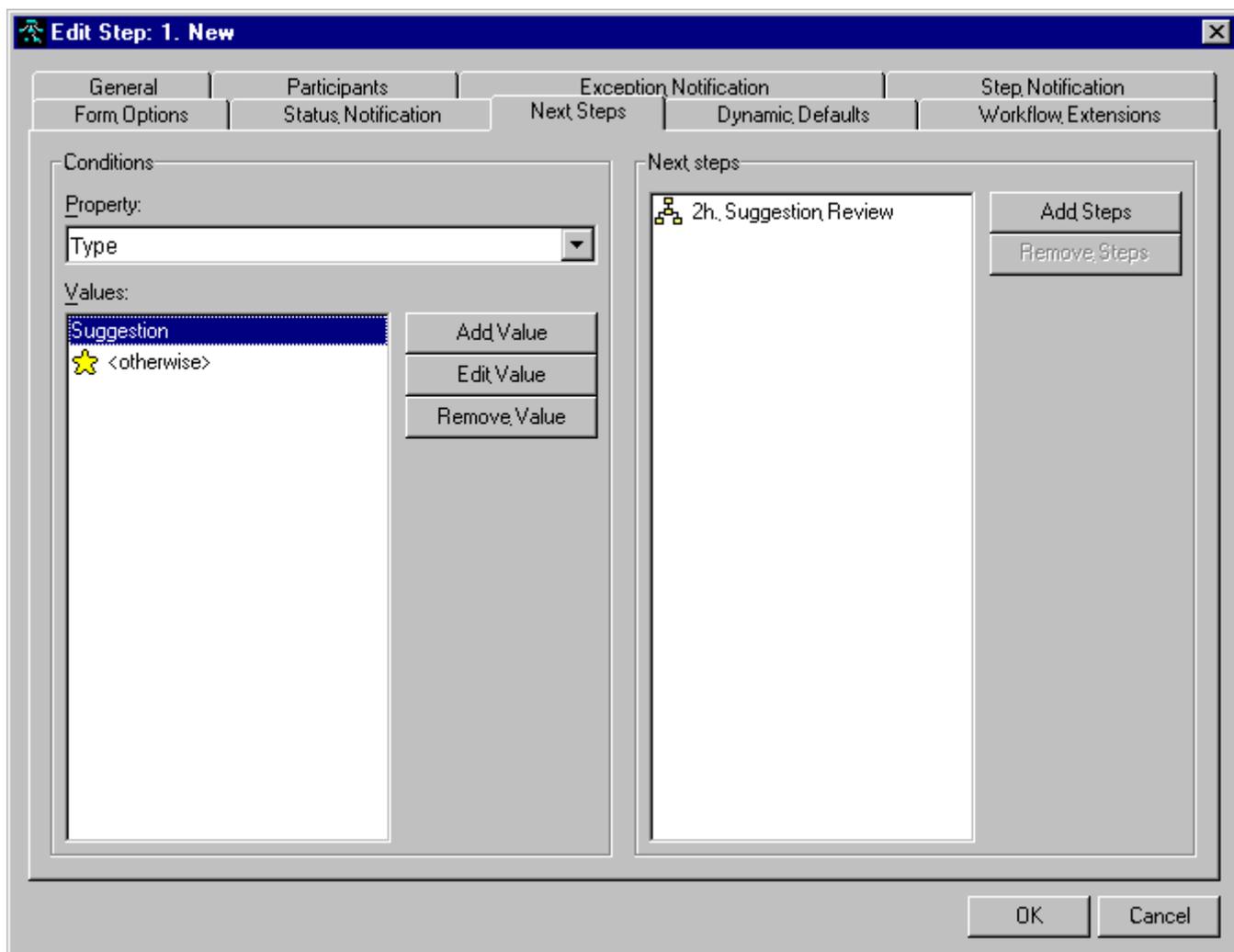
When you specify a condition, you select a property and one or more values. For each value, you select one or more steps that are appropriate transitions for an item whose property is set to that value—an item that meets that condition. In this example, the conditional property is Type. When the value 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.
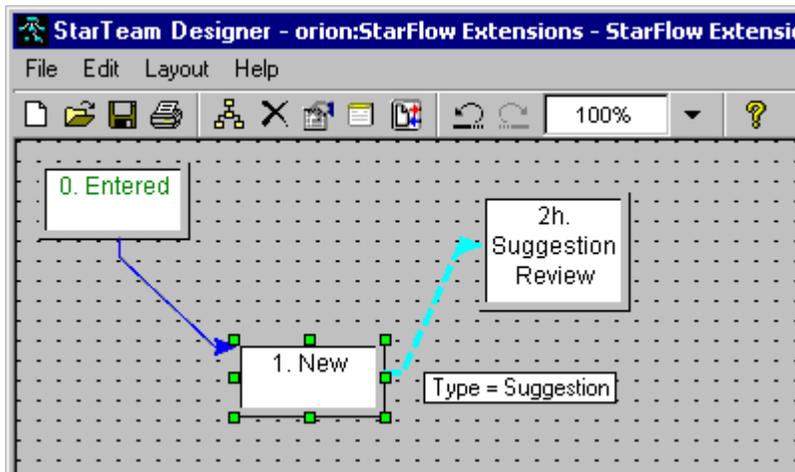
If the steps were already drawn manually, but not the links, Workflow Designer draws the links. In each case, the link starts from the first step's left edge and ends at the second step's right edge, regardless of the locations of these steps in the workspace.

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. We do 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. The **Edit Step** dialog box opens.
2. 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**. The **Select a Value** dialog box opens.
5. Do one of the following:
   a) Select a value for the condition (for example, `Suggestion`), then click **OK**. The value appears in the **Values** list and is selected.
   b) 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."
6. Click **OK**.
7. Click **Add Steps**. The **Select Next Steps** dialog box opens.
8. Select one or more next steps and click **OK**.
9. (Optional.) Repeat steps 4 through 8 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.
10. 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 7 and 8 of this procedure to select those workflow steps.

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

12. When you click **OK** to close the **Edit Step** dialog box, any steps that were specified as next steps and were not yet drawn appear in the upper left corner of the 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 box for the step from which the link originates. The dialog box 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, click **Edit** > **Link Options** > **Display Link Text**.

Aligning Steps

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

1. 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 **LayoutAlignLeft** 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.

1. 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 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

Enter the prerequisites here (optional).

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. Click **Edit** > **Link Options** > **Default Link Colors**. 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. Click **Edit** > **Link Options** > **Default Conditional 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.

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. Point to the link for which you want to change the color and right click to display the context menu.
2. 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**. The color selection dialog box displays. 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.
- 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 box 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.

> **Note:** 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. Double-click an existing step. The **Edit Step** dialog box opens.
2. 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.
   - Click **OK** or another tab.

Controlling Dynamic Changes to Forms

Use the **Dynamic Defaults** tab of the **Edit Step** dialog box 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** 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. Double-click an existing step. The **Edit Step** dialog box opens.
2. 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. The **Select Property** dialog box opens.
   b) 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. The **Select Condition Value** dialog box opens (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).

   b) 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. The **Select Property and Value to Assign** dialog box opens.
   c) 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.

If you use the workflow control property in the **Dynamic Defaults** tab, avoid confusion by never using it as a property to be changed. That is, avoid placing the workflow control property in the columns entitled **Assign values to other properties**. When you use the workflow control property in the leftmost column (the **Property** column), always set it to the value that corresponds to the step. Otherwise it will be ignored. For example, in the step "5. Testing", you can set dynamic defaults for the workflow control property when it has the value "5. Testing". If your dynamic defaults specifies a change to the **Comment** field (for example to "Workflow Status is 5. Testing"), that comment would appear in the history for the item and indicate which revisions had that status.

Limiting the Users Listed in a User Selection Field

To limit the users associated with a project or view that will be listed in a user selection field in a step:

1. On the **Context Limits** tab of the **Edit Step** dialog box, clear the **Use global workflow context limits for this step** check box if it is selected.

2. From the **Context** tree view, select the project or view for which you want to limit the associated users listed in a user selection field. The **Add** button is enabled.

3. Click **Add**. The **Select Property** dialog box opens.

4. From the drop-down list, select the **Property** for which you want to limit the list of available users.

5. Click **OK**.

6. Select the same user selection field under **Property** on the **Context Limits** tab of the **Edit Step** dialog box.

7. Do one or both of the following:

   • Click the Add Users button to select the user(s) to be listed in the user selection field. Use the Ctrl or Shift key to make multiple selections.

   • Click the Add Groups button to select the group(s) to be listed in the user selection field. Use the Ctrl or Shift key to make multiple selections.

   🖉 **Note:** Clicking the **All Users** button will remove all previous selections and list all the users associated with the selected project or view in the user selection field.

8. Optionally, click the **Add References** button to select the user selection field(s) for which you want the same list of available users. Use the `Ctrl` or `Shift` key to make multiple selections.

9. Optionally, select the **Use global workflow context limits for this step** check box to set the same user selections for every step in the workflow.

   🖉 **Note:** User selections are automatically set globally on the **Context Limits** tab of the **Workflow Properties** dialog box.

10. Click **OK**.

11. Save the workflow.

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 StarTeam 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 box 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 StarTeam Notification Agent if no message is provided.

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. Double-click an existing step The **Edit Step** dialog box opens.
2. Click the **Participants** tab.
3. Select one of the following types of notification from the **Notification Type** list:

Blast	Instructs the StarTeam 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 StarTeam 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 StarTeam 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 StarTeam 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. The **Select Notified Users** dialog box opens.
   b) Select one or more users.
   c) Click **OK**.

   To add one or more user groups:

   a) Click the **Add Groups** button next to the **Step Notification** list to add a group to the list. The **Select Notified Groups** dialog box opens.
   b) Select one or more groups.
   c) Click **OK**.

   To add one or more users by reference:

   a) Click the **Add References** button next to the **Step Notification** list to add users to the list. The **Select Referenced Fields** dialog box opens.
   b) Select one or more user ID fields.
   c) Click **OK**.

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.

   > 💡 **Tip:** 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.

> 💡 **Tip:** 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, StarTeam 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. On the **Edit Step** dialog box, 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. The StarTeam 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** field blank to use the default subject line for step notification messages.
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, the StarTeam Notification Agent sends the default body for step notification messages. 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. On the **Select a Reference** dialog box, select one or more property field names.
   c. Click **OK**.

   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**.

Understanding Notification List Operation

The StarTeam 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 and 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 the StarTeam 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, the StarTeam 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 and 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 box) 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, the StarTeam Notification Agent moves on to the next entry in the list.

Step Notification List	Resolution
Tom	A single message is sent to Tom. If Tom declines or does not reply before the notification interval expires, the StarTeam Notification Agent moves on to the next entry in the list.
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 StarTeam Notification 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 StarTeam Notification Agent generates an exception, following the instructions on the **Exceptions** tab.

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

Limiting the Time Allowed to Complete the Task

Enter the prerequisites here (optional).

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 box allows you to specify recipients for exception messages that are generated at this step. Exception messages are generated by the StarTeam 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. Double-click an existing step. The **Edit Step** dialog box opens.
2. 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** field 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**.

Specifying an Initial Value for an Empty Text Field

Enter the prerequisites here (optional).

To specify an initial value for an empty text field in a step:

1. Under **Initialize Values** on the **General** tab of the **Edit Step** dialog box, click the **Add** button. The **Select Property and Value to Assign** dialog box opens.

2. Select the **Property** from the list.

3. Do one of the following:

- Select the **Value** option to type a new value in the associated text field.
- Select the **Value** of property option to select the value from the associated list of currently available values.

4. Click **OK** twice.

5. Save the workflow.

Monitoring an Item's Status

Use the **Status Notification** tab on the **Edit Step** dialog box 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.

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

To create a status notification list and message:

1. Double-click an existing step. The **Edit Step** dialog box opens.

2. 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.

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 list. **3.** The word Everyone appears in the list. There is no need to add individual users, groups, or references.
To add one or more users	**1.** Click the **Add Users** button next to the list to add users to the list. **2.** On the **Select Notified Users** dialog box, select one or more users. **3.** Click **OK**. The selected users appear in the list.
To add one or more user groups	**1.** Click the **Add Groups** button to add groups to the list. **2.** On the **Select Notified Groups** dialog box, select one or more groups. **3.** Click **OK**. The selected groups appear in the list.
To add one or more users by reference	**1.** Click the **Add References** button to add users to the list.

2. On the **Select Referenced Fields** dialog box, select one or more user ID fields.
3. Click **OK**. The referenced user ID fields appear in 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.

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, the StarTeam Notification Agent sends the default body for status notification messages.

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

   1. Click **Insert Reference**.
   2. From the resulting **Select a Reference** dialog box, select one or more property field names.
   3. 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**.

Managing Exceptions

Use the **Exception Notification** tab on the **Edit Step** 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 the StarTeam Notification Agent based on the above and on system errors. These messages vary according to the exception that occurred, and cannot be modified.

The StarTeam Notification Agent sends one message to all the users in the exception notification list simultaneously.

To create an exception notification list:

1. Double-click an existing step. The **Edit Step** dialog box opens.
2. 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.

   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 list.

	3. The word `Everyone` appears in the list. There is no need to add individual users, groups, or references.
To add one or more users	1. Click the **Add Users** button next to the list to add users to the list. 2. On the **Select Notified Users** dialog box, select one or more users. 3. Click **OK**. The selected users appear in the list.
To add one or more user groups	1. Click the **Add Groups** button to add groups to the list. 2. On the **Select Notified Groups** dialog box, select one or more groups. 3. Click **OK**. The selected groups appear in the list.
To add one or more users by reference	1. Click the **Add References** button to add users to the list. 2. On the **Select Referenced Fields** dialog box, select one or more user ID fields. 3. Click **OK**. The referenced user ID fields appear in the list.

4. Click **OK**.

Extending the Workflow's XML

The **Workflow Extensions** tab allows you or Micro Focus 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 box.
- A particular step using the **Workflow Extensions** tab in the **Edit Step** dialog box.

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: click **Edit** > **Workflow Properties**.
2. On the **Workflow Properties** or **Edit Step** dialog boxes, 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.

You can create a custom enumerated property from the **New Workflow Properties** and the **Workflow Properties** dialog boxes. These dialog boxes 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 box (**File** > **New**).
• **Workflow Properties** dialog box (**Edit** > **Workflow Properties**).
• **New Step** dialog box (**File** > **New Workflow Step**).
• **Edit Step** dialog box's General tab (double-click a step in the workspace).

Creating an 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 box. 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 box or the **Workflow Properties** dialog box, click **Add Property**. The **Add Enumerated Property** dialog box opens.
2. 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** field.
4. Click **Add** to enter the first value for this enumerated type. The **Add Value** dialog box 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** field.
6. Type the name for this enumerated value in the **Name** field.
7. Click **OK**.
8. Repeat Steps 4-7 to add additional values.

   ✏️ **Note:** 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.

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.

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 box. The display name and the values can be changed from StarTeam, but not from Workflow Designer.

To add an enumerated value to a custom field:

1. From the **Workflow Properties** dialog box, **New Step** dialog box or **Edit Step** dialog box's **General** tab, click **Add Value**. The **Add Enumerated Value** dialog box opens.
2. Click **Add**. The **Add Value** dialog opens.
3. Use the displayed code or enter another in the **Code** field.
4. Type the name for this enumerated value in the **Name** field.
5. Click **OK**.

Changing the Name of a 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. Double-click an existing step. The **Edit Step** dialog box opens.
2. Click the **General** tab.
3. From the **Step Name** list, select another value as the name of this step.
4. Click **OK**.

Checking for Errors

When you open a workflow that contains errors or when you select **File** > **Check For Errors**, the 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 box.

Specific fields display question marks when they are in error in the Workflow Properties dialog. The **Edit Step** dialog box 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. We recommend 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.

Saving a New Revision of an Existing 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.

1. Click **File** > **Save as**.
2. On the **Revision Comment** dialog box, 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.

Saving a Workflow to a New Location

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.

1. Click **File** > **Save as**.
2. On the **Save As** dialog box,
   a) Type a name for this workflow.
   b) Select an appropriate folder to store it in.
3. Click **OK**. The **Revision Comment** dialog box displays.
4. 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**. 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

Enter the prerequisites here (optional).

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**.
2. Click **Yes** when a message asks you to confirm. The **Save As** dialog box opens.
3. Type a name for this workflow.
4. Select an appropriate folder to store it in.
5. Click **OK**. The **Revision Comment** dialog box displays.
6. 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.

7. Click **OK**. 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. Click **File** > **Export**. The **Export Workflow Document** dialog box opens.
2. 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 StarTeam Notification Agent by writing a StarTeam SDK script or having the Professional Services Organization from Micro Focus 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.

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 the StarTeam Notification Agent. If you are adding the use of these features, see *Changing Workflow Control Property Values Automatically*.

Workflow Roadmap

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

1. Create the necessary projects and folders.
2. Set up appropriate users and groups for the StarTeam 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 StarTeam Notification Agent to check the notifications.

Setting Up Projects and Files

Enter the prerequisites here (optional).

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

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

The Test project uses workflow and APE files in the Test folder in the StarFlow Extensions project.
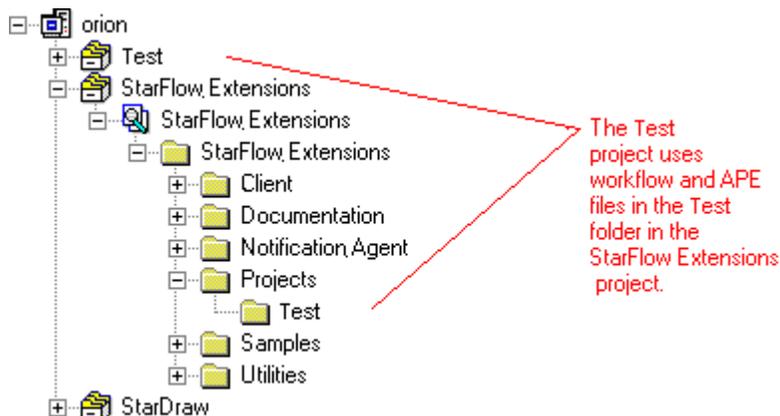
1. Use the new project wizard to create the `Example` project in your server configuration.
2. Open the StarFlow Extensions project.
3. Create an `Example` folder as a child of the `Projects` folder. You will store the custom forms and workflows to be used in the `Example` project here.

   > **Note:** 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 Help*.

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 the **Server Administration** tool, open the **User Manager** dialog box.
2. Create the following groups and users:

Groups	Customer Support group
	Marketing group
	Product Maintenance group
	Product Management group
	Research and Development 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

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.

Understanding the Example Use Case

```
┌─────────────┐
│   Customer  │
│  generates  │
│   feedback  │
└──────┬──────┘
       │
       ▼
┌─────────────┐
│ Customer Liaison │
│ responds & assigns │
└──────┬──────┘
       │
       ▼
     ╱────────╲                    ┌─────────────┐
    ╱  Is it a  ╲ ─── yes ──────▶  │Product manager│
    ╲ suggestion?╱                 │  evaluates  │
     ╲────────╱                    └──────┬──────┘
       │                                  │
       no                                 │
       │                                  ▼
       ▼                                ╱────────╲
┌─────────────┐                        ╱ Should we ╲
│ Engineering │ ◀─── yes ───           ╲ implement? ╱
│   manager   │                         ╲────────╱
│ schedules & │                            │
│   assigns   │                            no
└──────┬──────┘                            │
       │                                   │
       ▼                                   │
┌─────────────┐                            │
│Implement change│◀────────┐              │
└──────┬──────┘            │              │
       │                   no             │
       ▼                   │              │
┌─────────────┐            │              │
│ Test change │            │              │
└──────┬──────┘            │              │
       │                   │              │
       ▼                   │              │
     ╱────────╲            │              │
    ╱ Did it pass ╲────────┘              │
    ╲  the test? ╱                        │
     ╲────────╱                           │
       │                                  │
       yes                                │
       ▼                                  │
┌─────────────┐                           │
│ Close the change│◀─────────────────────┘
│   request   │
└─────────────┘
```
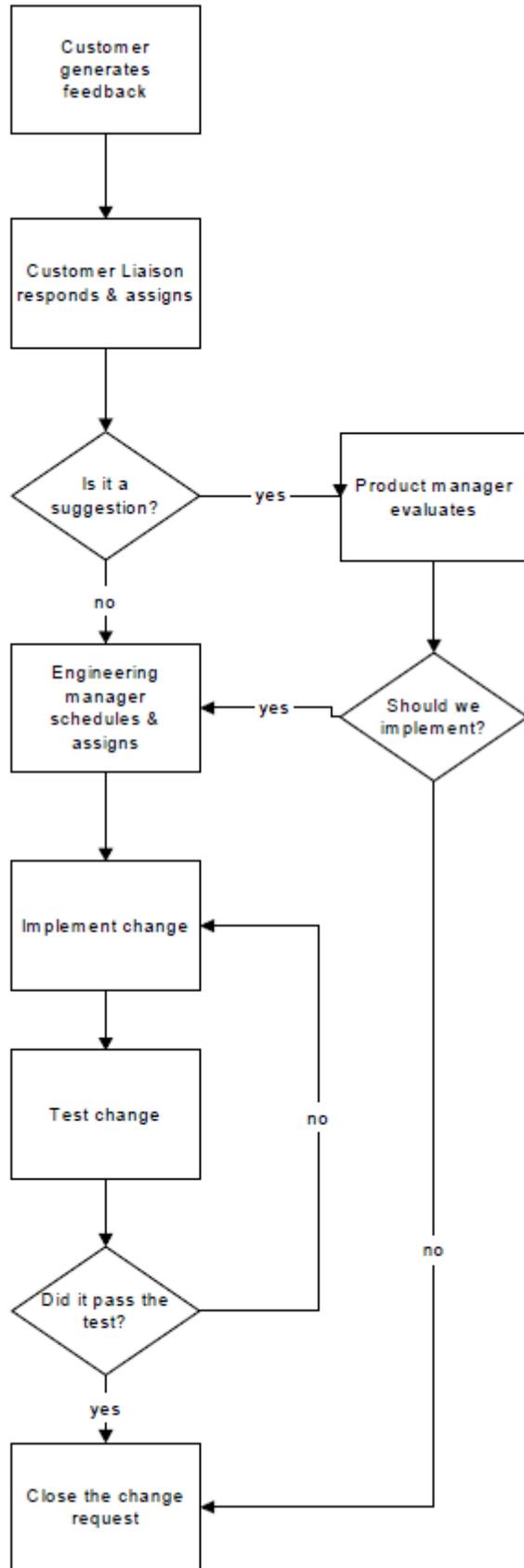
The example workflow process controls a change request (CR) item and has several possible steps:

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.

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.

EM Processing

All CRs that are to be acted upon must go to the EM for scheduling and assignment.

Implementation

This step has two possible solutions.

1. Product enhancements proceed to the research and development (R and D) group for implementation.
2. Defect reports go to Product Maintenance to be fixed.

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.

Closed. There are three possible states for closed CRs.

1. 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.
2. 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.
3. 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.

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

> 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.Completed

6c.Reopened

7.Customer Letter

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.

> Note: 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*.

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 box lists the fields that can be customized.
4. Click **Add**. The **Add Field** dialog box 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** field.

7. Click Add to enter the first value for this enumerated type. The **Add Value** dialog box 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** field.

10. Click **OK**.

11. Repeat steps 7–10 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. Completed

   190 6c. Reopened

   200 7. Customer Letter

12. Select "0. Entered" as a default from the **Default Value** list. 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* for more information on using JBuilder to customize application forms.

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 `Makeproject_namejar.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 **ChangeRequest** workflow using the default values..

1. Click **Start** > **Programs** > **StarTeam** > **Workflow Designer**. The Workflow Designer opens.

2. Click **File** > **New**. The New dialog box appears.

3. Select a server configuration. If the **Log On** dialog box appears, log on using a valid user name and password. Click **New**. The **New Workflow Properties** dialog box displays.

4. 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.

5. 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.

6. 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**).

7. From the Entry Step list, select `"0. Entered"`.

8. Click **OK**.

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.

1. Click **File** > **New Workflow Step**. The mouse pointer changes to a cross-hair.

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

3. 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`".

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

Note: 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 box 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, do one of the following in the workspace:

* 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.

Editing the 0. Entered Step

The entry step is a special pass-through step. As a user creates an item, the APE uses the step properties for the "0. Entered" status to control:

* The initial options for form fields (that is required, hidden, or disabled).
* The values that can be selected for the workflow control property at the time that the item is created. (These values are equivalent to steps in the workflow process.)

For example, as a user creates a change request, the only values that can be selected for the workflow control property are those specified for the "O. Entered" step on its "Next Steps" tab.

"0. Entered" is never used as a "Next Step" for any workflow step. That means that users cannot select "0. Entered" as the value for the workflow control property. "0. Entered" can have no links directed to it—only from it. Because no item ever displays this status, there is no reason to set notifications, participants, dynamic defaults, or a duration for this step. They will never be enforced.

While "0. Entered" is never the selected value for the workflow control property in the APE, there always is a selected value. This can cause some initial design confusion. As you create the settings for steps that correspond to values of the workflow control property, be aware that, when a user creates an item, settings assigned to both "0. Entered" and the step that corresponds to the currently selected value of the workflow control property affect the workflow.

For example, suppose that the only "Next Step" for "0. Entered" is "1. New". As the user creates a new change request, the default value (in fact the only possible value) for the workflow control property is "1. New". However, the form field settings assigned to both the "0. Entered" and the "1. New" steps affect the APE. If "0. Entered" requires the Type field and the "1. New" step requires the Platform field, both Type and Platform will be required. Notifications, participants, dynamic defaults, and duration come only from the "1. New" step.

Tab	Field	Value/Note
General	Step Name	`0. Entered`
General	Limit Step Duration	`cleared`
Participants	To	`<Leave blank>`
Exception Notification	To	`<Leave blank>`
Step Notification	To	`<Leave blank>`
Form Options	Field Name	`Severity`
Form Options	Setting	`Required`
Form Options	Field Name	`Synopsis`
Form Options	Setting	`Required`
Form Options	Field Name	`Type`
Form Options	Setting	`Required`
Status Notification	To	`<Leave blank>`
Next Steps	Condition	`<otherwise>`
Next Steps	Next Steps	`1.New`
Dynamic Defaults		`<leave blank>` (not available with initial step; the **Workflow Status** field is never set to `0. Entered`. **Dynamic Defaults** apply only to the currently selected status in the **Workflow Status** field).
Workflow Extensions	Workflow Extensions	`<Leave blank>`

To Edit the Step

1. Double-click an existing step. The **Edit Step** dialog box opens.
2. Click the **Form Options** tab.
3. Set the **Severity** and **Type** fields to `Required` by activating the check box in the **Required** column for each field.
4. Click the **Next Steps** tab.
5. Set the **<otherwise>** condition to step `1.New`.
6. Select **<otherwise>** in the **Values** list.
7. Click **Add Steps**. The **Select Next Steps** dialog box displays.
8. Select `1. New` and click **OK**.
9. In the **Property** list, click **Add**. The **Select Property** dialog opens.
10. Select `Severity` from the list of properties and click **OK**.
11. Select `Severity` in the **Property** column.
12. In the **When Equals Value** column, click **Add**. The **Select Condition Value** dialog opens.
13. Select `High` as the value and click **OK**.
14. Select `High` in the **When Equals Value** list.
15. In the **Assign Values To Other Properties** column, click **Add**. The **Select Property and Value to Assign** dialog box opens.

16. Select `Priority` as the property and **Yes** as the value, then click **OK**.

17. Click **OK**.

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 box.

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.

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

Tab	Field	Value
General	**Step Name**	`1. New`
General	**Limit Step Duration**	`Selected`
General	**Duration**	`4 days`
General	**Relative To**	`<Leave blank>`
Participants	**Notification Type**	`Random`
To	**To**	`Product Management group`
Participants	**Authorized**	`Product Management group`
		`PM`
		`Administrators group`
		Reference to current value of Responsibility property.
Exception Notification	**To**	`PM`
Step Notification	**Importance**	`Normal`
Step Notification	**Notification Interval**	`12 hours`
Step Notification	**Subject**	`<Leave blank>` to use the default step notification message.
Step Notification	**Body**	`<Leave blank>` to use the default step notification message.
Form Options	**Field Name**	`Platform`
Form Options	**Setting**	`Required`

Tab	Field	Value
Status Notification	To	`<Leave blank>`
Next Steps	Condition	`Type`
Next Steps	Value:	`Suggestion`
Next Steps	Next Steps	`2. PM Evaluation`
Next Steps	Condition	`<otherwise>`
Next Steps	Next Steps	`3. EM Processing`
Dynamic Defaults	Property	`Severity`
Dynamic Defaults	When equals value	`High`
Dynamic Defaults	Assign Property:	`Priority`
Dynamic Defaults	Value:	`Yes`
Dynamic Defaults	Property	`Severity`
Dynamic Defaults	When equals value	`<otherwise>`
Dynamic Defaults	Assign Property:	`Priority`
Dynamic Defaults	Value:	`No`
Dynamic Defaults	Property	`Workflow Status`
Dynamic Defaults	When equals value	`1. New`
Dynamic Defaults	Assign Property:	`Comment`
Dynamic Defaults	Value:	`Status is 1. New`
Workflow Extensions	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 box 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
```

This yields 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.

We recommend that you specify a next step for the `<otherwise>` condition. If you do not specify an `<otherwise>` condition, we recommend 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.
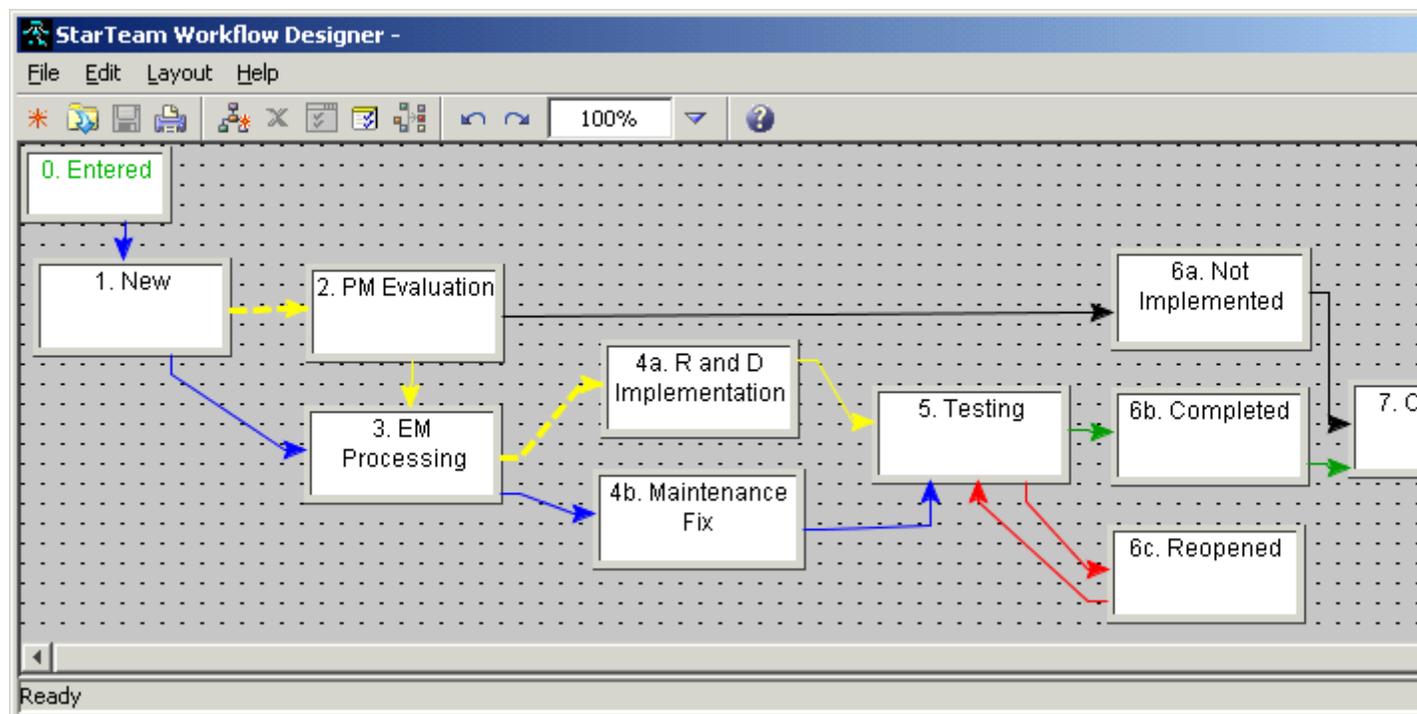
Summary of Other Step Properties

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.

Tab	Field	Value
General	**Step Name**	2. PM Evaluation
General	**Limit Step Duration**	`Selected`
General	**Duration**	7 days
General	**Relative To**	`<Leave blank>`
Participants	**Notification Type**	Ordered (does not matter because there is only one participant)
Participants	**To**	PM
Participants	**Authorized**	PM
		Administrators group
		Reference to current value of Responsibility property value
Exception Notification	**To**	PM
		General Manager
Step Notification	**Importance**	`Normal`
Step Notification	**Notification Interval**	4 days
Step Notification	**Subject**	`<Leave blank>` to use the default step notification message.
Step Notification	**Body**	`<Leave blank>` to use the default step notification message.
Form Options		`<Leave blank>`
Status Notification	**To**	`<Leave blank>`
Next Steps	**Condition**	`<otherwise>`
Next Steps	**Next Steps**	3. EM Processing
		6a. Not Implemented
Dynamic Defaults	**Property**	`Workflow Status`
Dynamic Defaults	**When equals value**	2. PM Evaluation
Dynamic Defaults	**Assign Property:**	`Comment`
Dynamic Defaults	**Value:**	Status is 2. PM Evaluation
Workflow Extensions	**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.

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

4a. R and D Implementation

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

Tab	Field	Value
General	Step Name	4a. R and D Implementation
General	Limit Step Duration	Selected

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

4b. Maintenance Fix

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

Tab	Field	Value
General	Step Name	4b. Maintenance Fix
General	Limit Step Duration	`Selected`
General	Duration	5 day
General	Relative To	`<Leave blank>`
Participants	Notification Type	Ordered
Participants	To	Reference to current value of Responsibility property Product Maintenance group
Participants	Authorized	Reference to current value of Responsibility property Product Maintenance group Administrators group

Tab	Field	Value
Exception Notification	To	EM
Step Notification	Importance	Normal
Step Notification	Notification Interval	24 hours
Step Notification	Subject	`<Leave blank>` to use the default step notification message.
Step Notification	Body	`<Leave blank>` to use the default step notification message.
Form Options		`<Leave blank>`
Status Notification	To	`<Leave blank>`
Next Steps	Condition	`<otherwise>`
Next Steps	Next Steps	5. Testing
Dynamic Defaults	Property	`Workflow Status`
Dynamic Defaults	When equals value	4b. Maintenance Fix
Dynamic Defaults	Assign Property:	`Comment`
Dynamic Defaults	Value:	Status is 4b. Maintenance Fix
Workflow Extensions	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.

Tab	Field	Value
General	Step Name	5. Testing
General	Limit Step Duration	`Selected`
General	Duration	2 days
General	Relative To	`<Leave blank>`
Participants	Notification Type	Random
Participants	To	Testing group
Participants	Authorized	Testing group
		Administrators group
		Reference to current value of Responsibility property
Exception Notification	To	QA Manager
		EM
Step Notification	Importance	Normal
Step Notification	Notification Interval	12 hours

Tab	Field	Value
Step Notification	Subject	`<Leave blank>` to use the default step notification message.
Step Notification	Body	`<Leave blank>` to use the default step notification message.
Form Options		`<Leave blank>`
Status Notification	To	QA Manager
Next Steps	Condition	`<otherwise>`
Next Steps	Next Steps	6b. Completed 6c. Reopened
Dynamic Defaults	Property	`Workflow Status`
Dynamic Defaults	When equals value	5. Testing
Dynamic Defaults	Assign Property:	`Comment`
Dynamic Defaults	Value:	Status returns to 5. Testing
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.

Tab	Field	Value
General	Step Name	6a. Not Implemented
General	Duration	2 days
General	Relative To	`<Leave blank>`
Participants	Notification Type	`<Leave blank>`
Participants	To	`<Leave blank>`
Participants	Authorized	Customer Liaison Administrators group Reference to current value of Responsibility property
Exception Notification	To	PM General Manager
Step Notification	To	`<Leave blank>`
Form Options		`<Leave blank>`
Status Notification	To	Research and Development group EM Product Management group PM Customer Liaison

Tab	Field	Value
Status Notification	**Subject**	`<Leave blank> to use the default step notification message.`
Status Notification	**Body**	`<Leave blank> to use the default step notification message.`
Next Steps	**Condition**	`<otherwise>`
Next Steps	**Value:**	7. Customer Letter
Dynamic Defaults	**Property**	`Workflow Status`
Dynamic Defaults	**When equals value**	6a. Not Implemented
Dynamic Defaults	**Assign Property:**	`Comment`
Dynamic Defaults	**Value:**	Status is 6a. Not Implemented
Dynamic Defaults	**Assign Property:**	Responsibility
Dynamic Defaults	**Value:**	Customer Liaison
Workflow Extensions	**Workflow Extensions**	`<Leave blank>`

6b. Completed

If a change request has been successfully fixed or implemented, the appropriate users and groups are notified that the change request 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.

Tab	Field	Value
General	**Step Name**	6b. Completed
General	**Duration**	2 days
General	**Relative To**	`<Leave blank>`
Participants	**Notification Type**	`<Leave blank>`
Participants	**To**	`<Leave blank>`
Participants	**Authorized**	Customer Liaison
		Administrators group
		Reference to current value of Responsibility property
		value
Exception Notification	**To**	PM
		EM
		General Manager
Step Notification	**To**	`<Leave blank>`
Form Options		`<Leave blank>`
Status Notification	**To**	Research and Development group
		EM
		Product Management group

Tab	Field	Value
		PM
		Customer Liaison
		Marketing group
Status Notification	Subject	`<Leave blank>` to use the default step notification message.
Status Notification	Body	`<Leave blank>` to use the default step notification message.
Next Steps	Condition	`<otherwise>`
Next Steps	Value:	7. Customer Letter
Dynamic Defaults	Property	`Workflow Status`
Dynamic Defaults	When equals value	6b. Completed
Dynamic Defaults	Assign Property:	`Comment`
Dynamic Defaults	Value:	Status is 6b. Completed
Dynamic Defaults	Assign Property:	Responsibility
Dynamic Defaults	Value:	Customer Liaison
Workflow Extensions	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.

Tab	Field	Value
General	Step Name	7. Customer Letter
Participants	To	`<Leave blank>`
Exception Notification	To	`<Leave blank>`
Step Notification	To	`<Leave blank>`
Form Options		`<Leave blank>`
Status Notification	To	General Manager
Next Steps		`<Leave blank>`
Dynamic Defaults	Property	`Workflow Status`
Dynamic Defaults	When equals value	7. Customer Letter
Dynamic Defaults	Assign Property:	`Comment`
Dynamic Defaults	Value:	Status is 7. Customer Letter
Workflow Extensions	Workflow Extensions	`<Leave blank>`

Testing the Workflow

To test a workflow, you need to:

1. 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.

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. Open the `Test` project and click **Project** > **Properties**. The **Project Properties** dialog box opens.

6. Click the **Editors** tab.

7. Select the **Use Alternate Property Editor For Change Requests** check box.

8. Type `Locator ChangeRequest` in the field under the check box.

9. Click **OK**.

10. Setup and start the StarTeam Notification Agent to monitor the Test project for the correct item type.

11. Use the StarTeam Notification Agent's **Run Now** button to simulate movement from step to step.

12. 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. 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 box.

4. Setup and start the StarTeam Notification Agent to monitor the `Example` project for the correct item type. The StarTeam Notification Agent notifies appropriate users about pending work, item statuses, and exceptions that occur in the process.

Using StarTeam Notification Agent

This chapter provides an overview of StarTeam Notification Agent and explains how to:

- Ensure that StarTeam Notification Agent will run successfully by creating mailboxes and user names and setting access rights in advance.
- Set up and run StarTeam Notification Agent.
- Change the http server port 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.

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

StarTeam 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 StarTeam Notification Agent can process the responses. You can provide both addresses. If you provide only an **Http Server Address**, StarTeam 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 StarTeam Notification Agent with a mailbox for your email system. StarTeam 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, StarTeam Notification Agent will send the responsibility message that includes email links.

Those that require no response

StarTeam Notification Agent sends mail messages using the SMTP protocol. If StarTeam 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.

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, StarTeam 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 StarTeam 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 StarTeam 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 StarTeam Notification Agent).

Getting Ready to Use StarTeam Notification Agent

Each instance of StarTeam Notification Agent can monitor only one server configuration. That means that you must run StarTeam Notification Agent once for each server configuration to be monitored.

Before you can run an instance of StarTeam Notification Agent, you must:

- Create a mailbox in your mail system for the sole use of the StarTeam Notification Agent instance if you will be sending email notification messages that require a response. You will use a different email address for each server configuration, as one instance of StarTeam Notification Agent can monitor only one server configuration. You may need to request that this mailbox be created by an MIS representative. If you have more than one server configuration, each server configuration will need its own mailbox.
- Decide what StarTeam user name and password will be used by StarTeam Notification Agent. We recommend dedicating a user license to StarTeam 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**. StarTeam 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 StarTeam Notification Agent is that of a team member and you plan to send email notification messages that require a response, StarTeam Notification Agent must use a different email address. You will not want to fill the team member's mailbox with email from the StarTeam Notification Agent.

    For details about adding users, see the *StarTeam Administrator's Guide*.

    🖉 **Note:** Be sure that the user who runs StarTeam Notification Agent has adequate access rights.

- 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 StarTeam Notification Agent email messages.

Redundant example	Suppose StarTeam 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 StarTeam Notification Agent and notification email from the StarTeam Server.
Conflict example	Suppose StarTeam 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 StarTeam 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 StarTeam Notification Agent. StarTeam 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 StarTeam 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 StarTeam Notification Agent sets responsibility for such items or if the StarTeam 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 Microsoft Windows client. The StarTeam Cross-Platform Client does not have this functionality.

- Run StarTeam Notification Agent setup to identify each server configuration to be monitored by StarTeam Notification Agent and to set notification properties for that configuration.

As part of this process, a StarTeam Notification Agent folder will be added to the server configuration's StarFlow Extensions project. StarTeam 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 StarTeam Notification Agent's installation folder.

Setting Access Rights for StarTeam Notification Agent

The user running StarTeam Notification Agent and StarTeam Notification Agent Setup must have the following access rights:

- At the server level:

- The server access right "Change user/operation time". StarTeam 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.
- The server access right "Administer user accounts" - StarTeam Notification Agent needs this right to obtain email addresses for users and other user information.

- StarTeam Notification Agent cannot set the responsibility property for an item to a specific user unless that user (who is being impersonated by StarTeam 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 StarTeam Notification Agent and Notification Setup

Both StarTeam Notification Agent and Notification Setup work only with the Microsoft Windows-based StarTeam Server. They can be installed on the same computer as the StarTeam Server they monitor or another computer. When using two computers, their times must be synchronized.

To access them:

Click **Start** > **Programs** > **Borland StarTeam** > **Notification Agent** > **Agent**.

Click **Start** > **Programs** > **Borland StarTeam** > **Notification Agent** > **Agent Setup**.

Identifying a Server Configuration

An instance of StarTeam 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 StarTeam Notification Agent as it logs on to the server configuration.
- The name and password for the mailbox created for StarTeam 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. Click **Start** > **Programs** > **Borland StarTeam** > **Notification Agent** > **Agent Setup**. The **Notification Agent Setup** dialog box opens.
2. Click **File** > **New**. The **Select a Configuration** dialog box opens.
3. Click **Add Configuration**. The **Add Configuration** dialog box opens.
4. Enter the name or IP address of the computer on which the StarTeam Server resides in the **Server Address** field.
5. Enter the port number used to access the server configuration in the **Port** field.
6. In the **StarTeam User Name** and **StarTeam Password** fields, enter the user name and password that StarTeam Notification Agent will use to log on to this server configuration.
7. Enter the name of and password for the mailbox in your mail system to be dedicated to StarTeam Notification Agent for use with this server configuration in the **Mailbox** and **Mailbox Password** fields.
8. Enter the name and email address for the mailbox in the **Display Name** and **Email Address** fields.
9. Click **OK** to return to the **Select a Configuration** dialog box, 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 StarTeam Notification Agent's installation folder.

💡 **Tip:** You can click **Configuration Properties** to change the settings that identify the server configuration or click **Delete Configuration** 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 StarTeam Notification Agent to Monitor a Server Configuration

Before StarTeam Notification Agent can monitor any server configuration you identify, you must set options for that configuration using StarTeam 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 StarTeam Notification Agent Setup to access the StarFlow Extensions project, create a child folder, and add a file.
- The server configuration already has a StarFlow Extensions project. If the server configuration does not already have this project, StarTeam Notification Agent will create one using a subfolder of StarTeam Notification Agent's installation folder as its working folder. This is not recommended.

1. Click **Start** > **Programs** > **Borland StarTeam** > **Notification Agent** > **Agent Setup**. The **Notification Agent Setup** dialog box opens.
2. Click **File** > **New**. The **Select a Configuration** dialog box opens.
3. Select the entry for this server configuration and 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.
4. 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.

5. 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 StarTeam 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.

6. 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** fields.

   These numbers control how long StarTeam 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. StarTeam 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.

   StarTeam 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 StarTeam 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, StarTeam 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** field. This controls how often StarTeam Notification Agent checks the list of tasks it has created. This value affects how often StarTeam Notification Agent checks for exceptions.

7. 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 install StarTeam 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 by a colon (:) and the port number. Localhost is not accepted. If StarTeam 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
```

   📝 **Note:**

   - 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** and the **Mail Store Address** fields empty if:

     - You will create an email notification that does not require a response.

- You want StarTeam Notification Agent to send a default email notification message that does not require a response.

b) All configurations must have the SMTP server name in the **SMTP Server Address** field because StarTeam 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 StarTeam 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** field, enter either `imap` or `pop3`. The default is `imap`. StarTeam Notification Agent can receive email messages using either of these two protocols. This field is ignored if you leave the **Mail Store Address** field empty.

e) In the **Mail Store Address** field, enter the name of the folder used by the mail store for incoming mail. The default is inbox.

8. Indicate the hours during which StarTeam 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. StarTeam Notification Agent sends email every day of the week from 12 midnight to 11:59:59 P.M. 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.

To replace the 24-hour time interval with one or more other intervals:	Select the 24-hour interval, click **Remove**, click **Add**, from the **Add an Active Interval** dialog box, set the appropriate start and end times. Click **OK** and repeat if necessary.
To copy the time intervals already specified for another day	Click **Copy**. From the **Copy Activity Times** dialog box, select the day to be copied from the list and then click **OK**.

9. Indicate any holidays on which StarTeam Notification Agent should not send email notification. In the **Holidays** portion of the **Activity** group box, click **Add**. The **Add a Holiday** dialog box opens.

10. Enter the name for this holiday in the **Holiday Name** field.

11. Select a date using the calendar.

- Click >>or << to move the calendar forward or backwards a year at a time.
- Click > or < to move the calendar forward or backwards a month at a time.

12. Select the **Recurring Holiday** check box if this calendar date will be a holiday every year.

13. Click **File** > **Close**.

14. Click **Yes** to save your notification configuration.

The **Revision Comment** dialog box opens.

15. Enter a comment about this set of options.

16. Click **OK**. StarTeam 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.

> **Note:** To remove a row in the **Watched Views** list, select the row and click **Remove**. To export an exact copy of the `NotificationAgent.conf.xml` file, click **File** > **Export**. Since every server configuration monitored by StarTeam Notification Agent has a file with this name, it is a good idea to rename the exported file.

Running StarTeam Notification Agent

If you choose to, you can run StarTeam Notification Agent 365 days a year, 24-7. If you stop StarTeam Notification Agent and restart, it starts processing the notification tasks it missed while it was shut down. Each instance of StarTeam Notification Agent monitors one server configuration.

You can run StarTeam Notification Agent manually or as a service.

Running StarTeam Notification Agent Manually

Notification Agent and **Notification Agent Setup** are separate, but similar looking applications. To run StarTeam Notification Agent manually:

1. Click **Start** > **Programs** > **Borland StarTeam** > **Notification Agent** > **Agent**. 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 StarTeam Notification Agent has just notified the first participant that an item needs attention. Clicking **Run Now** prompts StarTeam Notification Agent to notify the next participant without waiting for the step notification interval to expire.

Running StarTeam Notification Agent as a Service

Many customers run both the StarTeam Server and StarTeam Notification Agent as services.

When running the StarTeam Server and StarTeam Notification Agent on the same computer, the StarTeam Server process must start before the StarTeam Notification Agent attempts to start. The best way to accomplish this is to set up a service dependency within Microsoft Windows.

If you are running the StarTeam Server as a service and StarTeam Notification Agent as a dependent service, you cannot shut down the StarTeam Server unless the StarTeam Notification Agent service is shut down first.

To avoid dependent servers, some customers run the StarTeam Server and StarTeam Notification Agent on different computers. When using two computers, you must first synchronize the times on both computers. If StarTeam Notification Agent returns a `View Configuration` exception, the two computers are probably not synchronized.

To run an instance of StarTeam Notification Agent as a service, 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 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 StarTeam Notification Agent .jar file (`agent.jar`). If you used the default installation location for StarTeam 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 (`starteamxx.jar`). If you used the default installation location for StarTeam Notification Agent, this is `"C:\Program Files\ Borland\ Notification Agent\starteamxx.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 StarTeam 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 (StarTeam 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 StarTeam 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 StarTeam 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 StarTeam 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. Click **Start** > **Run**.
2. Enter `Regedt32`.
3. Click **OK**.
4. Select `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\ Services\NotificationAgent`
5. Click **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.

To set up a service dependency for WindowsXP

1. Click **Start** > **Run**.
2. Enter `Regedt32`.
3. Click **OK**.
4. Select `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\ Services\NotificationAgent`
5. Click **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.

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. Click **Start** > **Run**.
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`".

When the StarTeam 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 re-booted 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 StarTeam Notification Agent.
2. Each TomCat http server has a `server.xml` file. As you install StarTeam Notification Agent, a copy of TomCat is placed in 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 StarTeam 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. 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" />
```

4. Change the port number to an appropriate number. For example, if the `<Connector>` statement contained: `port="9000"` you might change it to: `port="9999"`.

5. Save the modified file.

6. Start Notification Agent Setup.

7. Open the desired configuration.

8. In the **Http Server Address** field, 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`.

9. Save this configuration.

10. Restart StarTeam Notification Agent or click its **Refresh** button. StarTeam Notification Agent will use the new port number.

Message Sending and Exception Processing

StarTeam 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 box 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 StarTeam Notification Agent

StarTeam Notification Agent creates the following server files in the local installation folder for server configurations:

serveraddress_port.nac	This is a configuration file. Example, `orion_49201.nac`.
serveraddress_port.itemtype_viewid.tsp	These files are view timestamps used for shutting down and restarting view monitoring processes. Example: `orion_49201.ChangeRequest_13.tsp`.
serveraddress_port.tasks.dat	These files archive the notification tasks managed by this instance of StarTeam Notification Agent It is used when shutting down and restart view monitoring processes. Example: `orion_49201.tasks.dat`.
serveraddress_port.NA-date_time.log	These files are used by StarTeam Notification Agent to log events and exceptions. Example: `NA-2002-9-25-21-12-10.log`.
NotificationAgent.conf.xml	Contains configuration information used by StarTeam 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

This section covers issues about which you may need to be aware.

Revision Assumptions

StarTeam Notification Agent assumes that only StarTeam Notification Agent and the users set by StarTeam 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, StarTeam Notification Agent assumes that the responsible person has already been notified.

Only Notification Agent Should Change Responsibility Field

StarTeam Notification Agent assumes that only StarTeam Notification Agent and the users set by StarTeam 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, StarTeam Notification Agent assumes that the responsible person has already been notified.

Out-of-Memory Errors from Java

When you start StarTeam Notification Agent from the **Start** menu, options from the `notification.stjava` file are used. This file is in the StarTeam 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
```

StarTeam Extensions Samples

The `Samples` folder in the default `StarTeam Extensions` installation 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).
MakeAPE_nameJar.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
- LayoutDesignerAPE
- 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.
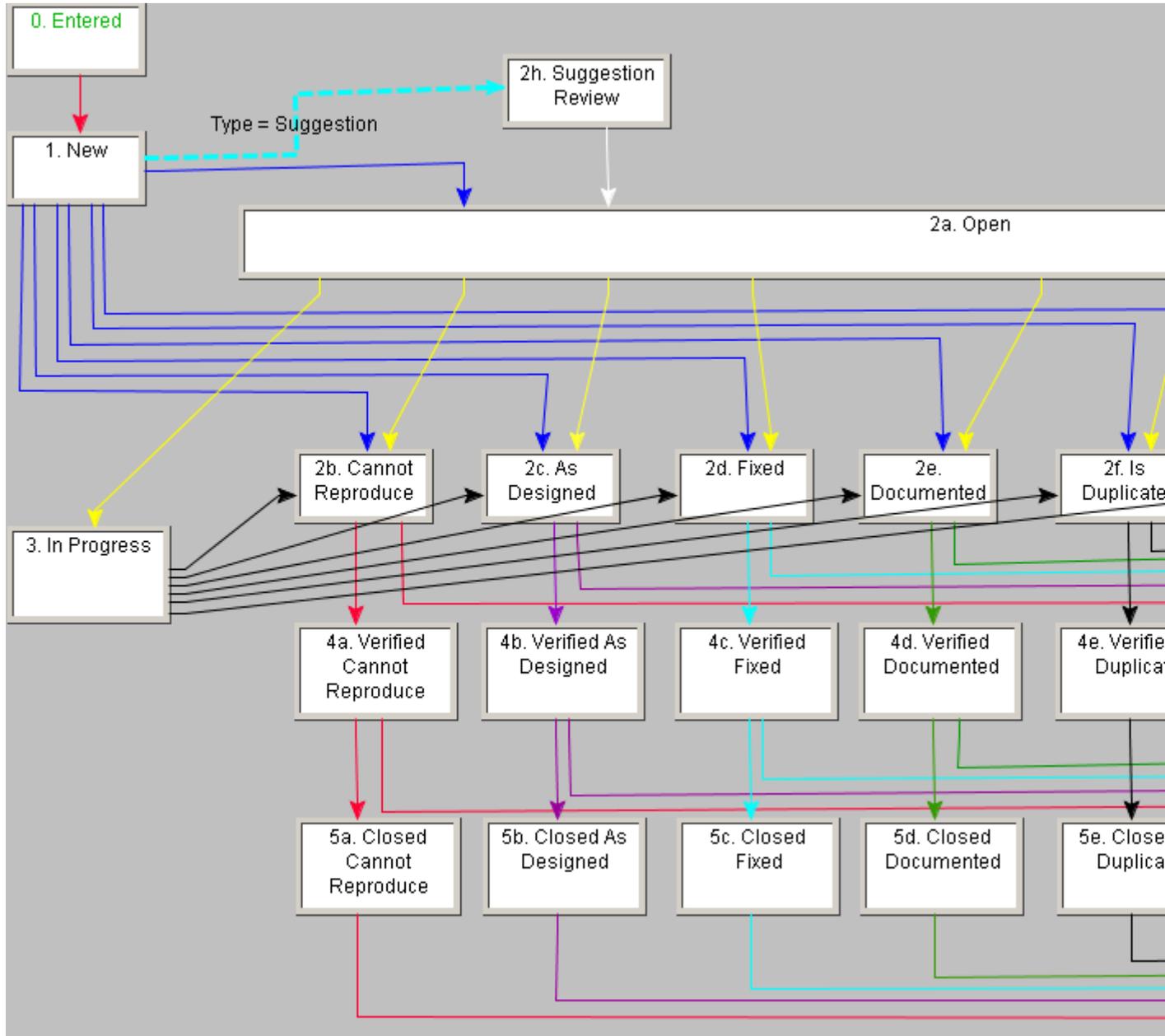
It is similar to the built-in workflow that comes with change requests. With Workflow Designer, you can completely redesign this workflow by adding additional steps, changing the behavior of links, modifying the authorization, and adding required, hidden and disabled fields to each workflow step. However if you plan to change much, it is easier to create your own workflow from scratch, perhaps using this workflow as a model.

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.

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 StarTeam 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.

Notification Tags and Messages

This chapter 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 attribute name may not contain spaces or any other special characters.
- The attribute value is quoted. The value may contain any characters; however, including quotation mark and backslash characters included in the value 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:

```
<IMG height="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">
```

This results in the `logo.gif` file to be pulled from the virtual root directory of the http server built into StarTeam 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 de-referenced by StarTeam 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. StarTeam Notification Agent does not perform any syntax checks on the values of the style attributes.

Attributes

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.
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.
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.
modified-prefix	**Optional**: yes
	Permitted Values: text
	Comment: This is the text that is placed in front of the de-referenced 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*. The Workflow Designer provides an easy way to insert property references (the entire property tag) into the message subject or body. Custom (user-defined) properties have the prefix "`Usr_`" in front of their internal identifier, for example, `Usr_WorkflowStatus`.

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 output 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 we recommend placing it at the beginning of the mime part.

Attributes

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.
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.
modified-prefix	**Optional**: yes
	Permitted Values: text
	Comment: This is the text that is placed in front of the de-referenced 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: {{property name="Synopsis" style= "color:black" modified-style=
"color:blue;font-weight:bold"}} <B>
Responsibility: {{property name="Responsibility" style= "color:black"
modifiedstyle= "color:blue;font-weight:bold"}} <B>
```

is equivalent to:

```
{{property-style style= "color:black" modified-style=
"color:blue;fontweight: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

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*. The Workflow Designer provides an easy way to insert property references (the entire property tag) into the message subject or body. 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

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:
{{requesttiming 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. StarTeam Notification Agent de-references 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 StarTeam 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 StarTeam 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 StarTeam Notification Agent mailbox. StarTeam Notification Agent retrieves the email response from the mailbox and processes it accordingly. If the confirm attribute is set, StarTeam 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

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.

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.

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

value	**Optional**: no
	Permitted Values: `duration`, `deadline`
	Comment: This attribute defines what this tag produces.

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.
```

{{url-folder}}

The `{{url-folder}}` tag is used to output a URL to the folder that contains the item. For example, the URL for the NewCRs folder might be:

```
starteam://10.50.1.60%20:2000/55;ns=Project;scheme=id/293;ns=View;scheme=id/
1900564;ns=Folder;scheme=id;scope=full/
```

Enter the actual information in this section (optional).

Attributes

None.

Examples

The following template excerpt from the message body:

```
Display the folder than contains {{type-name}} {{property
name="@primarydescriptor"}}
by pasting the following URL into Project>Open StarTeam URL:{{url-folder}}
```

produces for CRs:

```
Display the folder than contains Change Request 22563 by pasting the
following
URL into Project>Open StarTeam URL: starteam://
10.50.1.60%20:2000/55;ns=Project;scheme=id/293;ns=View;scheme=id/
1900564;ns=Folder;scheme=id;scope=full/
```

The following template excerpt from the message body assumes other html tags are also in use, but shows the tag for the folder URL:

```
Display the folder than contains {{type-name}} {{property
name="@primarydescriptor"}}
by clicking <a href="{{url-folder}}">{{property name="Folder"}}</a>.
```

produces for CRs:

```
Display the folder than contains Change Request 22563 by clicking NewCRs.
```

{{url-item}}

The {{url-item}} tag is used to output a URL to the item. For example, the URL for Change Request 22,563 might be:

```
starteam://starteam8.borland.com:2000/34;ns=Project;scheme=id/
535;ns=View;scheme=id/4162900;ns=ChangeRequest;scheme=id;scope=full
```

Attribues

None.

Examples

The following template excerpt from the message body:

```
Display {{type-name}} {{property name="@primary-descriptor"}} by pasting the
following URL into Project>Open StarTeam URL: {{url-item}}
```

produces for CRs:

```
Display Change Request 22,563 by pasting the following URL into Project>Open
StarTeam URL: starteam://starteam8.borland.com:2000/34;ns=Project;scheme=id/
535;ns=View;scheme=id/4162900;ns=ChangeRequest;scheme=id;scope=full.
```

The following template excerpt from the message body assumes other html tags are also in use, but shows the tag for the item URL:

```
Display {{type-name}} {{property name="@primary-descriptor"}} by clicking <a
href="{{url-item}}">{{property name="@primary-descriptor"}}</a>.
```

produces for CRs:

```
Display Change Request 22,563 by clicking Change Request #22,563.
```

{{url-view}}

The {{url-view}} tag is used to output a URL to the view that contains the item. For example, the StarTeam URL for the Cygnus view might be:

```
starteam://10.50.1.60%20:2000/55;ns=Project;scheme=id/293;ns=View;scheme=id
```

Attributes

None.

Examples

The following template excerpt from the message body:

```
Display the {{view-name}} view by pasting the following URL into Project>Open
StarTeam URL: {{url-view}}
```

produces:

```
Display the Cygnus view by pasting the following URL into Project>Open
StarTeam URL: starteam://10.50.1.60%20:2000/55;ns=Project;scheme=id/
293;ns=View;scheme=id
```

The following template excerpt from the message body assumes other html tags are also in use, but shows the tag for the folder URL:

```
Display the {{view-name}} view by clicking <a href="{{url-
view}}">{{viewname}}</a>.
```

produces:

```
Display the Cygnus view by clicking Cygnus.
```

{{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}} {{property name="@primary-descriptor"}} was created in view
{{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	Indicate that items have reached certain steps in the workflow process.
Responsibility request messages	Ask users to accept responsibility for items at specific workflow steps.
Confirmation messages	Indicate that StarTeam Notification Agent has received users' acceptances or refusals.
Request exception messages	Notify specific individuals when no one accepts responsibility for items at specific steps.
Step exception messages	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"}}
{{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** field or the **Mail Store Address** field has a value.

The second is sent when both the **Http Server Address** field and the **Mail Store Address** field 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="@workflowproperty"}}.
```

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.

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: Your response to responsibility request for the {{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."

"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="@workflowproperty"}}.
```

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>
```

Index

T

W

workflow information 28