Solutions Business Manager
SBM Composer Guide (On-Premise Version)

Table of Contents

Part 1: Introduction

This section contains the following information:

Chapter 1: Introduction

This section contains the following information:

About SBM Composer

SBM Composer is the visual design component of Solutions Business Manager (SBM). Use SBM Composer to design the structure of a process app, which can contain applications and orchestrations.

Applications and orchestrations contain workflows that coordinate the flow of information through business processes. Application workflows coordinate the work of people; orchestrations coordinate Web service-enabled systems. In SBM Composer, design tasks include:

- Defining workflows that control the flow of data

- Creating roles that control the actions users can perform and the information they can access

- Designing custom forms that present "view" and "update" pages to your users

- Defining the events and workflows used by orchestrations

After process apps are designed, they are deployed to a runtime environment, where the processes are run by two engines: the Application Engine runs applications workflows and the Orchestration Engine runs orchestration workflows.

SBM Composer provides two mechanisms to deploy a design to a runtime environment:

- You can publish a process app to the SBM Application Repository, from which an administrator can deploy the process app to the target environment.

- When administrative control is not needed, you can use the quick deploy command in SBM Composer to directly deploy a process app through the Application Repository to a specified environment.

Once a process app is deployed, use the SBM Application Administrator to perform configuration tasks, such as creating projects that store process items, assigning users to roles, and creating notifications. Once configuration tasks are complete, the process app is ready for users.

Key Terms

The following terms will help you better understand how to use SBM Composer. Terms are grouped by category.

- **Design Packaging**

  ▪ **Process App**

  Comprises one or more applications and orchestrations.

  ▪ **Application**

  A collection of elements that coordinate the work of people, such as a development team's work tasks or customer support calls. Applications typically contain workflows, fields, forms, and roles.

  ▪ **Orchestration**

  Coordinate Web service-enabled systems. Orchestrations contain workflows, which are an arrangement of control flow structures, Web services, and data elements.

  ▪ **Design Elements**

  Portions of a process app, such as an application workflow or a decision step.

  ▪ **Blueprint File**

  A file that is created by SBM Composer and contains process app design elements, which include roles, scripts, icons, tables, workflows, and orchestrations. This file is deployable to any database.

  ▪ **Snapshot File**

  Contains design elements and configuration data, such as projects, reports, and role assignments.

- **Design Actions**

  ▪ **Check-in**

  Sends design elements to the SBM Application Repository so they can be checked out by other designers.

  ▪ **Deploy**

  Sends a process app defined in SBM Composer to a runtime environment. Depending on your privileges, you can deploy from SBM Composer or Application Repository.

  ▪ **Publish**

  Sends a process app to the Application Repository so it can be deployed or checked out by other users.

  ▪ **Promote**

  Sends a snapshot to a runtime environment. Promotions are done in Application Repository.

- **Applications**

  ▪ **Workflows**

  A collection of states, transitions, and fields that define your process.

- **Transitions**

  Activates an item's movement from state to state in the workflow.

- **States**

  Positions that an item resides in while moving through the workflow process. States can also be considered a stopping point along a workflow's path.

- **Forms**

  The "view" and "update" pages in which users work with process items. You can use quick forms or create custom forms for states and transitions.

- **Decisions**

  Enables conditional routing of items. For example, customer-reported requests may follow a different route through a workflow than those submitted by employees.

- **Roles**

  A collection of application-related privileges. Examples are the ability to read and update certain fields and the ability to transition items. After a process app is deployed, users are assigned to roles in SBM Application Administrator.

- **Orchestrations**

  - **Synchronous Orchestrations**

    Immediately returns data to an application. Requires the application workflow to wait for a reply before it can continue.

  - **Asynchronous Orchestrations**

    Can run independently of the application that contains it. Does not require the application workflow to wait for a reply. Runs independently of the application workflow.

  - **Workflows**

    Enables the use of Web services for coordinating the interaction between an application workflow and one or more external systems.

  - **Steps**

    Control flow structures that enable an orchestration workflow to make calculations, decide between two possible sets of instructions, handle exceptions, and so on.

  - **Data Elements**

    Include data sent with the event, data expected by or returned by a Web service, and working data created for temporary use during the execution of an orchestration workflow.

Using SBM Composer with Other SBM Components

SBM Composer can be used independently of runtime environments to design process apps, but several SBM runtime components play a role in activating process apps for users:

Application Repository

Along with managing deployment to SBM runtime environments, Application Repository provides basic storage, versioning, and patching capabilities that allow for collaborative development. After you create or modify a process app in SBM Composer, you can check it in to the repository so that it is available for other designers.

To open the repository, click the Application Repository button on the **Deployment** tab of the ribbon. If SBM Composer is not connected to a runtime environment, use this URL: `http://`*serverName*`:`*port*`/mashupmgr`.

Application Administrator

Application Administrator is used to configure applications with runtime data, such as role assignments for users and groups, after process apps are deployed. SBM's "quick administrator" feature simplifies this process by creating a project for each workflow and default notifications when a process app is first deployed. Also, the person who deploys a process app for the first time is automatically assigned to all roles.

For a list of tasks performed in Application Administrator compared to SBM Composer, refer to .

To open Application Administrator, use this URL: `http://`*serverName*`/tmtrack/tmtrack.dll?shell=swc&StdPage&Template=appadmin/index.html`.

To open Application Administrator from SBM Work Center, click the user icon in the upper right corner, and then select Application Administrator.

End-user Interfaces

SBM has multiple Web interfaces that enable users to interact with process apps designed in SBM Composer:

- **SBM Work Center**

  A simplified end-user interface. Users can access the Work Center by opening the following URL:

      http://*serverName*/workcenter

- **Service Request Center**

  A Service Support Manager portal that enables users to submit requests and access knowledge management items. Users can access the Request Center by opening the following URL:

```
http://serverName/tmtrack/tmtrack.dll?shell=srp
```

Chapter 2: Navigating SBM Composer

This section contains the following information:

Navigating SBM Composer

Use the legend below the figure to learn about the different parts of SBM Composer.
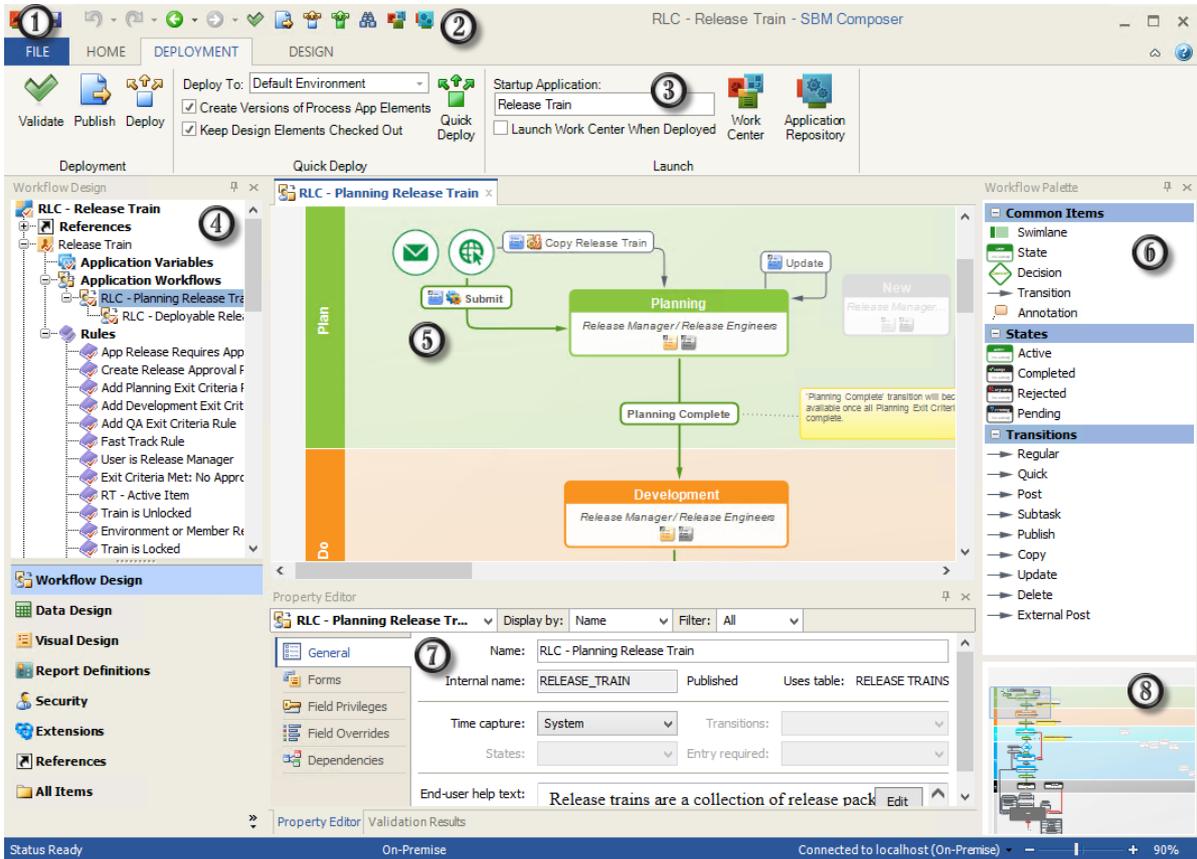
1. **File Options**

   Use **File** options to manage process app files, set options, and exit SBM Composer. Depending on your theme settings, these options are available in the upper left corner from the **File** tab, the Composer Start button, or a drop-down menu. (File Options [page 475])

2. **Quick Access Tool Bar**

   Contains commonly used commands, such as undo, redo, save, and validate. (Quick Access Tool Bar [page 492])

3. **Ribbon**

   Provides a central location for the commands you use to perform design tasks. The Ribbon contains commands appropriate to the design element you are editing and is organized into task-specific tabs:

   - Home [page 493]

     The **Home** tab is selected when SBM Composer is first opened. It contains common commands that are available in all of the editors.

   - Deployment Tab of the Ribbon [page 496]

     The **Deployment** tab is always visible. It contains common commands related to validation, publishing, and deployment.

   - Design [page 497]

The **Design** tab is visible when you open a workflow or form. Its contents vary depending on the design element being edited.

- Appearance [page 502]

    The **Appearance** tab is visible when you customize the appearance of an annotation or a swimlane.

- Script [page 503]

    The **Script** tab appears is visible when you add or edit a script.

    Tip: Use the arrow at the top right corner to collapse and expand the Ribbon if you need more viewable space in the main window.

4. **App Explorer**

    Navigate design elements in an open process app. (App Explorer [page 31])

5. **Design Editor**

    Select elements from the App Explorer to open them in the Editor, which is the main work area. (Editor [page 32])

6. **Palette**

    Contains design artifacts related to the element in the Editor. You can drag and drop design artifacts into the editor. (Palette [page 32])

7. **Property Editor**

    Set properties for the selected design element. (Property Editor [page 32])

8. **Zoom Preview**

    Drag the rectangle to move to a different part of the form or workflow. Use the slider bar below the zoom preview to change the size of the rectangle, so that more or less of the form or workflow is visible in the editor pane.

Using the App Explorer

The App Explorer enables you to navigate an open process app by expanding and collapsing the nodes in the list. The process app itself is at the top of the list, followed by application references, applications, and orchestrations. Each application and orchestration is further divided into forms, tables, roles, and other design elements they include.

Note the following as you navigate the App Explorer:

- Expand and collapse headings by clicking **+** and **–**.

- Use the filter buttons at the bottom of App Explorer—**Workflow Design**, **Data Design**, **Security**, and so on—to view only a few of the headings at a time. Click the **All Items** button to view all the headings again.

- Click the arrows at the bottom right of App Explorer to show more or fewer buttons.

- To change the alphabetical sorting of the design elements within a heading, right-click the header, select **Sort**, and then select **Ascending** or **Descending**.

- To reorder the design elements within a heading, right-click the design element and then click **Move Up** or **Move Down**.

- An asterisk (*) marks each design element that you changed since the element was checked in to the repository. Elements that appear in yellow (or orange) denote warnings that were detected.

- When you click a heading in the App Explorer, its elements are listed in the editor pane to the right. When you click the name of an element in App Explorer, or double-click it in the list to the right, it is displayed in the Using the Editor Pane [page 32].

- Design elements in referenced applications are read-only. If you need to modify them, click the yellow alert bar at the top of the applicable editor pane to open the related process app.

- If the icon for a design element has a red bookmark over it, the design element is mandatory and cannot be modified or deleted.

- For information about moving and hiding App Explorer, see Working with Panels [page 34].

Using the Editor Pane

The editor pane appears in the center of the SBM Composer window. The content of this area changes depending on the element that you are editing. For example, when you select a role in the App Explorer, the role editor shows the privileges for that role.

You can do the following:

- Edit multiple design elements by right-clicking each element in App Explorer and selecting **Open in New Tab**.

- Reorder tabs by dragging them.

- Change the size of the editor pane or the tiled tabs by dragging their borders.

Palette

For many entities in App Explorer, a palette of related objects is displayed to the right of the editor pane. For example, when you edit a workflow, the palette contains states, steps, and the various types of transitions. You can drag these objects onto the entity in the editor pane, and double-click some objects to add them to the editor pane.

> **Tip:** For information about moving and hiding a palette, see Working with Panels [page 34].

Using the Property Editors

The Property Editor appears at the bottom of the SBM Composer window, beneath the design element you are editing. Forms, roles, tables, and the other design elements have different properties, and the corresponding Property Editor groups those properties in tabs that are at the left edge of the Property Editor. By default, the Property Editor is shown, but you can hide it by clearing the **Property Editor** check box on the Home Tab of the Ribbon [page 493].

For example, if you click a transition in a workflow, the Property Editor displays the properties for that transition, grouped in **General**, **Options**, **Form**, **Field Privileges**, **Field Overrides**, **Actions**, and other tabs. If you have the workflow checked out, you can edit the properties displayed on the tabs.

The contents of the Property Editor can include:

- A design element list at the top left that contains the components of the form, workflow, or other design element you are editing. When you select a component from this list, its Property Editor opens and it is selected in the main editor. For example, suppose you have a long form and need to change the name of a tab. Instead of searching for the tab in the form editor, you can select the tab from the list.

- For some design elements, a **Display by** list that appears at the top of the Property Editor. Use this list to specify how the names in the design element list are displayed. For tables, you can choose between the name and the database name, and for workflows, states, and transitions, you can choose between the name and the internal name.

- For some design elements, a **Filter** list that appears at the top of the Property Editor that lets you limit the items in the design element list according to component type. For example, if you need to find a particular state in a large workflow, you could select **States** from the filter, and only states appear in the design element list.

If you select a component in the editor, and that component is not included in the selected filter, the filter selection changes to **All**.

If you change the filter selection, and the component currently being shown in the Property Editor (and selected in the editor) does not match the new filter, the first component that matches the filter becomes selected in the editor. For example, if you select **Decisions** in the filter, the first decision in the workflow is selected.

> **Tip:** If an application workflow, form, or orchestration workflow is too large to fit in the displayed editor space, when you select a component from the design element list, the editor scrolls to the selected component. To prevent scrolling, press the CTRL key while you select the component.

For information about moving and hiding Property Editors, see Working with Panels [page 34].

Viewing Validation Results and Log Information

Validation and logging information appears at the bottom of the Editor.

- **Validation Results**

   Process apps must be validated before they can be published or deployed to the Application Repository or exported to a file. Use the check boxes at the bottom of the list to specify the messages that you want to see based on their logging level and other criteria. For more information about the Validation Results, see Chapter 34: Using the Validation Results [page 533].

- **Common Log**

   The Common Log Viewer displays a list of messages that tell you what went wrong if a process app encountered an error or if it did not work as expected during runtime. The Log Viewer also contains general diagnostic messages, such as notifications of

when a Web service is invoked or when the SBM Orchestration Engine is sending a message. For more information about the Common Log Viewer, see Chapter 35: Using the Common Log Viewer [page 537].

- **Activity Log**

  The Activity Log displays messages about high-level process app operations such as deploy, publish, check in, check out, and export. For more information about the Activity Log, see Chapter 36: Using the Activity Log [page 549].

  Note: Validation results and log information are displayed if the appropriate check boxes are selected on the Home Tab of the Ribbon [page 493].

Working with Panels

You can close, hide, and move panels such as Property Editors, palettes, the Validation Results, the Common Log Viewer, the Activity Log, and App Explorer:

- To close a panel, click the ☒ button at the top right corner.

- To hide a panel, click the ☐ button at the top right corner. The panel is hidden but a tab-like label remains visible.

- To restore a hidden panel, hover over the tab-like label until the panel appears, and then click the ☐ button at the top right corner.

- To move a panel, drag it and drop it on the new location.

- To dock a panel in another location, drag it over one of the arrow icons, such as this one ▲ . A blue area that represents the target location appears. When you find the target location you want, drop the panel on it.

- To restore a panel to its default location, double-click its title bar.

Finding Items

Process apps can become complex over time. SBM Composer provides several mechanisms to help you find design elements in an open process app:

- **Quick Find**

  Click CTRL+F to open the **Find** dialog box and search for specific design elements in the process app or in referenced applications.

  Type the text you want to find, and then press Enter. The first design element is highlighted in App Explorer, and its editor opens. To continue searching with the dialog box open, click **Find** or Enter. To continue searching with the dialog box closed, press F3. For details, refer to Find Dialog Box [page 503].

- **Advanced Find**

  Expand or limit the scope of your search by context (current application or entire process app, for example), by element type (state or field) and other find options.

  Click the **Find Items** button on the Ribbon or quick access toolbar, press Alt and then F, or press Ctrl+Shift+F. For details, refer to Find Items Dialog Box [page 504].

- **Find by Usage**

  With a design element selected, click **Where Used** on the Ribbon, and then specify search criteria. When you click **Find**, a list of items where the design element is used is returned. This is useful for understanding how changes to a design element may impact your process app.

  For details, refer to Find Where Used Dialog Box [page 504].

Working Offline or Connected

You can use SBM Composer without a network connection or without being connected to a repository.

To check design elements in or out, open process apps in the repository, or publish and deploy process apps, however, you must be connected to a repository.

Use the control near the bottom right corner of the SBM Composer window to toggle the connection to the configured repository. Your connection status shows you as either **Offline** or **Connected to *ServerName* (*ServerType*)**.

Before you check process apps into the repository, they are stored in the local cache. This special area on the file system of your computer stores design elements while you work on them. Initially, SBM Composer creates the Local Cache in the AppData directory allocated by Windows for the Windows login ID that was in effect when you installed SBM Composer. You can specify a different location as needed.

If you switch to working offline while a process app is open, SBM Composer gives you the opportunity to download any design elements that are not yet saved to the Local Cache.

To configure your repository connection or modify your local cache location, select **Composer Options** from the **File** menu, and then select the **Repository** tab. For details, refer to Repository Options [page 483].

Working with Multiple Instances of SBM Composer

You can run multiple instances of SBM Composer simultaneously. If a process app is opened multiple times (from the local cache or remote repository, or from import from an *.msd* file), the second and all subsequent instances of the process app are read-only.

Screen Resolution and Sizing

SBM is designed for screen resolutions of 1280x1024 or higher. You may experience problems if you use a lower resolution, such as 800x600.

If SBM is difficult to read at the minimum resolution of 1280x1024, you can increase the browser's Zoom setting. For example, changing the browser's Zoom to 125 percent will increase the size of text, buttons, and other controls throughout the interface. Be aware that increasing the Zoom from 100 percent may require you to scroll through pages as you work, however.

Part 2: Process Apps

This section contains the following information:

Chapter 3: About Process Apps

This section contains the following information:

- Process Apps [page 39]

- Concurrent Development [page 44]

- Compare and Merge [page 45]

Process Apps

A process app includes one or more applications used to track items through workflow processes. A process app can also include one or more orchestrations that provide the logic for the invocation of Web services.

In SBM Composer, you create process apps independently from the environment in which they are intended to run. When process apps are deployed to runtime environments, they interact with and runtime data, such as user and group accounts and projects. Process apps that contain orchestrations are initiated based on SBM events.

This separation of the process app and the "runtime" data enables you to modify and test process apps without disrupting your organization's normal business operations. Some organizations use multiple environments (development, test, and production) to further this "path to production" process.

After a process app is deployed, most design elements, such as workflows, tables, forms, and other elements are viewable (read-only) in SBM Application Administrator. These items can be changed only by editing the process app in SBM Composer and redeploying.

About Process App Distribution

Process apps have three output formats:

- **Blueprints**

  Represent a complete design of one or more applications, orchestrations, or both. Process app blueprints are created in SBM Composer, published to Application Repository, and deployed a runtime environment. When they are exported from SBM Composer, blueprints have an .msd file extension.

  Blueprints can also be created in Application Repository by the "Get Process App from Application Engine" feature in the **Environments** area. You can then save this version of the process app to a file.

  You can distribute process app blueprints through Publishing [page 40], Deployment [page 40], and Export [page 41].

- **Templates**

  A type of process app that can be used a starting point for creating other process apps. For example, if you have some database fields that you want to include in every process app you create, you could create a template that includes those fields. For details, refer to About Templates [page 42].

Templates cannot be published or deployed, but they can be distributed by exporting them to an .mst file.

- **Snapshots**

  Contain both the blueprint information and configuration data, such as projects, notifications, and overrides defined in Application Administrator. Snapshots can be promoted from Application Repository to different environments, such as testing and production. When they are saved from Application Repository, snapshots have an .mss file extension. Snapshots are provided with solutions, such as Release Control and Service Support Manager.

  You can distribute process app snapshots through Promotion [page 41].

Blueprints and templates contain information about the version of SBM Composer they were developed in and exported from. You can import these files into the same or later version of SBM Composer, but not an earlier version.

Publishing

Publishing packages process app created in SBM Composer and makes them available for deployment in Application Repository. You can publish a process app in one of the following ways:

- In SBM Composer, publish the process app to Application Repository.

- In SBM Composer, export the process app to a file. Then, in Application Repository, load the file.

For details, refer to Publishing a Process App [page 51].

Deployment

Deployment makes process apps defined in SBM Composer available on the SBM Server. There are several ways to deploy process apps:

- In Application Repository, you can deploy process apps that have been published or loaded from a file. This method is primarily used by organizations who want to closely monitory and control deployments to runtime environments and therefore do not allow deployments from SBM Composer.

- In SBM Composer, you can deploy to environments where you have been granted permissions and where deployments from SBM Composer have been allowed. Use one of two methods:

  - **Deploy**

    Use the full deployment method when you need to specify an environment, map endpoints, schedule deployments, and other default settings for a particular process app. When the deployment is completed, design elements are checked in. This method is recommended the first time you deploy a process app. For details, refer to Deploying a Process App [page 52].

  - **Quick Deploy**

    Use this method to deploy a process app without being prompted to specify deployment settings. In addition, you can opt to deploy the process app without creating a new revision in the repository and to keep design elements checked

out after a quick deploy so you can continue working in the process app. For more information, see Using Quick Deploy [page 53].

After a process app that contains an application is deployed, use SBM Application Administrator to create projects, add project-level overrides, and assign users and groups to the roles defined in SBM Composer.

Export

Use the Export feature in SBM Composer to create a file version of a blueprint (.msd) that can be distributed outside of the Application Repository. The file can then be imported into SBM Composer or loaded into the Application Repository. For details, refer to Exporting a Process App [page 54].

You can also export template files (.mst).

Promotion

Promotion enables administrators with access to Application Repository to replicate a process app from one environment to another. After deployment and testing, for example, a process app in the test environment can be replicated to the production environment. When you promote, you transfer both the process app design (created in SBM Composer) and the configuration data such as users, groups, notifications, and projects (set up in SBM Application Administrator) to another environment. The combination of design and configuration data is called a process app snapshot.

> **Note:** For more information about promotion and process app snapshots, see the *SBM Application Repository Guide*.

About Endpoints

Endpoints represent location and authentication information for SOAP and REST Web services and are used in applications and orchestrations. The endpoint in the process app itself is called a *process app endpoint* and corresponds to a place in the process app where a Web service is called. The values of the location URL and authentication information that are mapped to the endpoint at deployment or promotion are called *environment endpoints*. Environment endpoint values can be modified when you deploy or promote a process app into an environment, enabling you to use different values for different environments.

> **Note:** Internal system endpoints are created automatically.

Endpoints enable you to test a process app with one environment, perhaps on an isolated network on a test server, and then roll out the process app to a production environment, with a more secure server. By specifying different endpoints in each of these environments, you can change the service used and/or authentication information without having to edit the details of the process app workflows.

Not all authentication types that are configurable in SBM Composer and in Application Repository are usable for all types of Web services. For example, the NTLM authentication type only applies to custom endpoints when calling REST services (from the REST Grid, RESTCaller, or in JavaScript that is embedded on an Application Engine page). Refer to the following table for a summary:

Authentication Type	SOAP Calls	REST Calls
None	✔	✔
SSO	✔	✔
HTTP Basic	✔ (see below)	✔
NTLM	Not applicable	✔
OAuth 2	Not applicable	✔
WS-Addressing (WSA)	✔ (see below)	Not applicable

- You can set HTTP Basic on the endpoint or explicitly in an orchestration. If both are set, the endpoint configuration overrides the orchestration setup.

- You cannot set WSA authentication on an endpoint in Application Repository, but WSA authentication can be set explicitly in orchestrations in a limited way.

On-premise installations only: To restrict access to an endpoint, you can use SBM Application Administrator to remove the permission to edit an environment in general and the permission to edit endpoints in particular. Note that without permission to edit the environment overall, a user can still edit the endpoints associated with that environment if the user has the **Create, Edit, and Delete Endpoints** privilege. The **Global Administration** privilege also enables you to edit endpoints, even if **Create, Edit, and Delete Endpoints** is not selected.

About Templates

You can create a template from an existing process app, then use the template as a starting point for additional process apps. For example, you may want to create custom form styles, then create a template to ensure that forms in all of your process apps use the same styles.

To convert a process app to a template, refer to Creating a Template [page 55].

You can then add templates to your local library so they are always available to you as you create process apps. For details, refer to Adding Templates to Your Library [page 56].

A template has the following characteristics:

- A template cannot be referenced by other process apps. (For information about references, see About References [page 369].)

- A template can reference other process apps (for example, the Global Process App).

- A template cannot be published or deployed.

- A template can be exported.

- A template has an `.mst` file extension.

- A template can be imported, with all settings preserved.

- You can add templates to your template library so they are readily available as you create process apps.

- All process apps created from a template have unique design numbers. Templates do not have design numbers. (For information about design numbers, see Managing Internal Identity and Design Numbers [page 62].)

- A blueprint can be converted into a template, but a template cannot be converted to a blueprint.

Global Process App and Global Application

Every SBM database includes the Global Process App, which is a special process app that contains only the Global Application. The Global Application initially includes system auxiliary tables—`Companies`, `Contacts`, `Problems`, `Resolutions`, and others—and their associated default icons. In a system upgrade from TeamTrack, the Global Application also initially contains any other existing auxiliary tables and scripts that are not associated with a specific application and any existing triggers that are used in transition actions.

To access the Global Application, you first "get" the Global Process App into the repository (as described in the Application Repository documentation). Then, you can create a reference to the Global Application in an existing process app. (See About References [page 369] for details.) You can also check out the Global Process App, modify its contents, check it in, and publish and deploy it like any other process app.

> **Important:** The Global Process App does not have to be deployed unless you modify it.

When you open the Global Process App, the repository could list several Global Process Apps, identifiable by the names of the environment sets from which they were gotten, as in **Global Process App (*environment-set-name*)**. For example, an environment set might include Development, Testing, and Production environments for a single SBM system. Your company might have multiple environment sets.

CAUTION:

> The Global Process App should be deployed *only* to an environment in the same *environment set* as the environment from which the Global Process App was originally gotten. If you deploy the Global Process App to a different environment set, the Global Process App in the deployed-to (target) environment set will contain everything in the deployed Global Process App, and some system-provided elements in the target Global Process App will be overwritten.
>
> If you decide to rename the Global Application for some reason, consider leaving the environment set name unchanged in the new name for the Global Application. This helps you avoid the potential problems associated with deploying the Global Application to a different environment set.
>
> For example, if the Global Process App from one environment set is deployed to an environment in a different environment set, and that second Global Process App is gotten into the repository, SBM Composer could display two triggers with the same name, and you have no way to tell which of the Global Process Apps they came from.

In SBM Composer, you can do the following to the Global Application:

- Add auxiliary tables (and their supporting design elements: table forms, images, JavaScripts, and styles) and scripts that you want to make available for use in multiple applications.

- Create triggers to be used in transition actions.

- Create roles and assign role privileges for auxiliary tables that are defined in the Global Process App. The role privileges are granted to users and groups that you assign to the roles on the base project in Application Administrator.

   Tip: To populate *User* and *Multi-User* fields that you add to auxiliary tables in the Global Process App, edit the user field in SBM Composer, associate one or more roles, deploy the process app, and then assign users and groups to those roles in Application Administrator using the following steps:

   1. In Application Administrator, select **Projects**.

   2. In the **Projects** tree, select **All Projects**.

   3. Select the base project, and then click **Details**.

   4. Open the **Roles** view, select a role, and then click **User Assignment** or **Group Assignment** to assign users.

   The assigned users will now appear as selections in the user fields that you added to the auxiliary table.

Concurrent Development

SBM Composer provides features that enable you and other designers to work on a process app at the same time.

   Note: This topic contains high-level information about concurrent development. For detailed information, see the "Best Practices for Concurrent Development" white paper, which is available from the Composer Start Page.

Version Control

In SBM Composer, process apps contain design elements that represent the smallest unit of design. Design elements include custom forms, workflows, rules, and tables. Each design element is independent and has its own version. You must check out a design element before changing it, so no other designer can work with a design element until you check it in again. However, you and other designers can work on different design elements in the same process app simultaneously.

When you check out a design element, it is stored in the Local Cache on your computer. When you check in a design element, its version is incremented and it is stored in the repository, which is a shared location that you and other designers can access. The repository supports standard version control operations, as described in Repository Menu [page 490].

Before publishing a process app, note the following:

- If you do not use the **Get Latest** command before publishing a process app, the repository may have newer versions of design elements than those in your Local

Cache. By default, when you publish, you are prompted to specify which versions should be used.

- When the **Get Latest** option is selected, the older versions in your Local Cache are overwritten by the newer versions before publishing takes place.

- When the **Ignore** option is selected, the older versions of the design elements are published. The newer versions in the repository are not overwritten. You might use this option when you want to deliver changes that you have made to a process app without having to uptake changes made by someone else that might interfere with your tests or are not necessary for you to uptake at this time.

For information about setting these options permanently, see Repository Options [page 483].

Compare and Merge

SBM Composer provides the ability to visually compare two versions of a process app and manually copy selected additions and changes from one process app to another. The typical use case for this feature is customers who upgrade from one version of a solution to another and want to keep the custom changes they made to the process apps in the earlier version. In this way, customers can combine aspects of their design with aspects of the solution developer's design.

Note: For more information about this feature, see Compare and Merge [page 45].

Patch Context

You can use a "patch context" to patch a production environment while continuing to work on a process app in a development environment. This involves opening the labeled version of the process app that was deployed to production. When you work in a patch context, the repository commands such as **Check In**, **Check Out**, and **Get Latest** are applied against the patch context label.

Note: For more information, see Working in a Patch Context [page 61].

Compare and Merge

You can visually compare two versions of a process app and manually copy selected items from one process app to the other. For example, suppose you receive a new version of a solution that you previously deployed. The solution contains a process app; before deployment, you changed it to meet your specific needs and processes. Now you want to upgrade to the new version, while keeping the changes you made to the earlier version.

This feature lets you view both versions of the process app side-by-side in SBM Composer. You can see the items that differ between them, copy new items from the new version, and modify existing items to match the new version. You can "dismiss" differences you do not want to merge, so there is no need to process them again if you need to finish the comparison in a later session. If you localized a process app, you can specify that the translated strings are not displayed as differences when you compare it with a new, unlocalized version.

> **Note:** For details about using the compare and merge feature, see Compare Process Apps Mode [page 70] and Comparing and Merging Process Apps [page 57].

Examples

- A company has version 2.1 of the Problem Management solution. Before the solution was deployed, Paul made modifications to the process app. The company wants to upgrade to version 3.0 to adopt new features and enhancements, but wants to keep the changes Paul made.

  Paul examines the differences between the version 2.1 and 3.0 process apps. Because the changes to version 3.0 are quite substantial compared to his changes, he copies his changes from version 2.1 into version 3.0.

- Before the company deployed version 3.0, Ramini made additional changes to the process app. The company now wants to upgrade to version 3.1 to adopt bug fixes. The version 3.1 changes are minor, so Ramini copies the modified items from version 3.1 to the version 3.0 process app that she customized.

Key Benefits

- Two versions of a process app can be compared side-by-side, with the differences highlighted in meaningful, customizable colors.

- New items in the comparison process app can be copied into the open process app using the standard editor.

- Changed items in the comparison process app can be manually modified in the open process app using the standard editor.

- Differences you do not want to merge can be dismissed, so you do not have to process them again in subsequent SBM Composer sessions. This is especially useful when you need to perform large or complex comparisons.

- Strings that can be translated can be ignored during the comparison. This is important when you compare a localized process app with a new version, because it eliminates the need to manually dismiss differences based solely on language.

Chapter 4: Working with Process Apps

This section contains the following information:

- Starting with Process Apps [page 47]
- Delivering Process Apps [page 49]
- Advanced Process App Tasks [page 55]

Starting with Process Apps

There are multiple ways to create process apps. Use the following table to determine which method best meets your needs.

Creation Method	Use for:	Refer to:
New	Quickly getting started with default templates provided with SBM Composer or with templates you have imported and added to your library.	Creating a Process App from a Template [page 47]
Import	Implementing a process app that you have saved to your computer or that you have downloaded from an external site, such as the Community website. In this case, you can modify the process app to meet your needs, but the process app keeps its original identity.	Importing Process Apps and Templates [page 48]
Import as new	Creating a process app based on another process app. In this case, you intend to use the process app as a starting point and plan to change its identity. For example, you may want to rename an "IT Help Desk" process app to "Service Desk."	Importing as a New Process App [page 49]

Creating a Process App from a Template

Process apps can be created from existing templates. The following types of templates are available by default:

- **Application Process App + Forms**

  A process app with the basic elements of an application. Also includes custom forms with auto-sections.

- **Application Process App**

  A process app with the basic elements of an application.

- **Orchestration Process App**

  A process app with the basic elements of an orchestration.

- **Empty Process App**

  An empty process app to which you can add applications and orchestrations.

Templates you have added to your library are also shown. For details, refer to Adding Templates to Your Library [page 56].

To create a process app from a template:

1. Select **File** | **New**.

   The **Create New Process App** dialog box opens.

2. Select one of the available templates, and then click **Create**.

3. Complete the dialog box as described in Configure Process App Dialog Box [page 72].

Importing a Process App

You can import blueprint files (.msd) that were previously exported from SBM Composer, saved from Application Repository, or downloaded from an external source. You can also import process app template (.mst) files.

Importing Process Apps and Templates

The **Import** option is best used to import a process app that has not yet been deployed to an environment. The process app will maintain its identity, meaning that you will need to choose to overwrite existing versions of the process app if it already exists in the repository.

You can also use this method to import a template file. Once you import a template, you can modify it, then export it and add the latest version to your template library.

To import a process app or a template:

1. Select **File** | **Import and Export** | **Import**.

2. Navigate to the .msd or .mst file you want to import.

3. You may be warned that the version in the local cache is overwritten with the imported process app. If you continue with the import, any work you did on that process app is lost.

   If the process app was also checked in to the repository, SBM Composer warns you that, on checking it in, you have to decide whether to check in new versions of its design elements created from the local process app, or overwrite the local process app with the latest versions of its design elements from the repository.

   To prevent loss of any work you did on the local copy:

   a. Cancel the import.

   b. Export the local process app to a file.

   c. Restart the import process.

d. Send the exported blueprint file to another designer or load it directly into Application Repository.

Importing as a New Process App

The **Import as New** option is best used to create a process app based on another process app. The process app will have a new identity that you can modify.

To create a new process app based on another process app:

1. Select **File | Import and Export | Import as New**.

2. Navigate to the .msd file you want to import.

3. You are warned that the process app will have a new identity, with some exceptions. You must change the name of key process app elements before you can publish and deploy the process app to the same repository from which it originated. For details, refer to Resetting Internal Identity [page 63].

Opening an Existing Process App

There are two ways to open an existing process app:

- To open a process app, select **File | Open**. You can open a process app that you saved to the Local Cache or a process app that is in the repository (including earlier labeled versions).

- To open a process app that has been exported from SBM Composer and saved to a file system on a computer, double-click the `.msd` file that was saved.

Checking Out Process Apps and Design Elements

- To check out a single design element, right-click the element in the App Explorer, and then select **Check Out**.

- To check out all design elements of a specific type, such as forms, right-click the element type in the App Explorer, and then select **Check Out**.

- To check out all design elements in a process app, select **File | Repository | Check Out All**.

In the dialog box that opens, select the elements you want to check out. See Check Out Design Elements Dialog Box [page 70] for details.

Delivering Process Apps

- Saving a Process App [page 50]

- Validating a Process App [page 50]

- Checking In a Process App [page 51]

- Publishing a Process App [page 51]

- Deploying a Process App [page 52]

- Exporting a Process App [page 54]

Saving a Process App

You can save an open process app to the Local Cache to preserve your work without checking the process app into the repository.

The Local Cache contains only the design elements that were downloaded from the repository. If you know you will work on the process app while disconnected from the repository, click the **Connected** indicator (in the bottom right corner of the SBM Composer window) and enable the **Work Offline** option on the menu that opens. You are prompted to download any design elements not already saved to the Local Cache.

> **Note:** Unlike the **Export** command, the **Save** command does not create a file that you can import into SBM Composer or load into Application Repository.

To save the open process app:

In the Quick Access Toolbar, click **Save**.

Validating a Process App

Before a process app can be published or deployed to Application Repository or exported to a file, it must pass validation. As you work with your process app, you can validate as you work to catch potential problems.

To validate the open process app:

In the Quick Access Toolbar, click **Validate**.

Details about the validation are displayed in the Validation Results. You can use the filtering options at the bottom to choose which types of message you want to view. (See Limiting Validation Results [page 534].)

By default, the name of any design element that contains errors appears in red in App Explorer. You can disable this feature or change the color on the **General** tab of the SBM Composer Options [page 479].

> **Note:** To optimize performance, the validation checks only those scripts that changed or were added since the process app was last validated. (See Chapter 37: Troubleshooting Scripts [page 551] for information about handling validation errors.)

Here are some typical situations that generate error and warning validation messages:

- You used a process app, application, or table with a name that already exists in a deployment environment.

- One or more references in a process app are unresolved.

- SBM Composer detects that a process app or a feature in a process app is incompatible with SBM On-Demand deployment. You receive these error messages if you are working in SBM On-Demand mode, or if you are working in SBM On-Premise/PaaS mode and the **Include validation for On-Demand compatibility** option is selected on the **General** tab of the SBM Composer Options [page 479] dialog box, and if one or more of the following conditions is true:

  - Scripts or triggers are included an application in the process app.

  - A **Publish** or **External Post** transition is included in an application workflow.

- A **Post** transition with an **Item link type** that includes "triggers" is included in an application workflow.

- A transition that uses a **Trigger** action type is included in an application workflow.

- A state that has a transition action with a **Trigger Received** condition is included in an application workflow.

- The database name for a table name exceeds the maximum number of characters that are allowed.

- A table in the process app includes a *Folder* field, *Company* field, *Contact* field, *Resolution Summary* field, or *Resolution Description* field.

- Various project and table privileges are selected for a role (for example, **View Item if Contact**).

Checking In a Process App

- To check in a single design element, right-click the element in the App Explorer, and then select **Check In**.

- To check in all design elements of a specific type, such as forms, right-click the element type in the App Explorer, and then select **Check In**.

- To check in all design elements in a process app, select **File | Repository | Check In All**.

In the dialog box that opens, select the elements you want to check in. See Concurrent Development [page 44] or Check In Design Elements Dialog Box [page 69] for details.

Publishing a Process App

Prerequisites:

Publish privilege (set in SBM Application Repository)

You publish a process app to make it available in Application Repository for deployment to the SBM Server.

Process apps are automatically published as part of a deployment process, but you may want to publish a process app so that is available to others designers before it is deployed. You can also use publish if you do not have privileges to deploy, but do have privileges to publish.

Once a process app is published, certain design elements are locked. To learn about unlocking these elements, refer to Modifying Locked Elements in a Published Process App [page 64].

When you publish a process app, SBM Composer checks in the design elements you select and then checks them out again. The version number of the process app is incremented accordingly.

If you do not perform the **Get Latest** command frequently to get the most recent versions of the design elements in the process app you are working with, your Local Cache could be out-of-date. For example, suppose Jane changed the owner of a state in a workflow and checked in the workflow, making it version 1.2. When you last performed the **Get Latest** command, the version was 1.1. To prevent obsolete design elements from being published, SBM Composer will by default warn you and give you the opportunity to get the latest versions before you publish. On the Repository Options [page 483] tab of the **Composer Options** dialog box, you can instead choose to always get the latest versions or always ignore them, without being prompted.

To publish a process app:

1. In the Quick Access Toolbar, click **Publish**. The **Publish Process App** dialog box opens.

2. Complete the **Publish Process App** dialog box.

    See Publish Process App Dialog Box [page 79] for details.

A message box and the Activity Log indicate the success or failure of publishing the process app.

Deploying a Process App

Prerequisites:

- **Deploy Process Apps To This Host** privilege (set in SBM Application Administrator)

- **Deploy** privilege (set in Application Repository)

- **Enable Deployment** option selected in the **Composer** list in the **Edit Environment** dialog box in Application Repository for the environment to which you want to deploy

    Note: A process app must be published before it can be deployed. The publish operation is performed automatically when you deploy.

Use the **Deploy** or **Quick Deploy** command to make a process app available to users. See Using Quick Deploy [page 53] for information about the second command.

To deploy a process app using the Deploy command:

1. In the Quick Access Toolbar, click **Deploy**.

    The **Deploy Process App** dialog box opens.

2. Perform the following steps:

    a. Select the environment to which you want to deploy.

    b. Type a version name, if you do not want to use the default value.

    c. Specify whether you want others to deploy the process app.

    See Deploy Process App Dialog Box [page 74] for details.

3. In the Deploy Options Dialog Box [page 74], set deployment options, such as a delayed start time, an e-mail notification, and whether to stop the deployment on warnings or only on errors.

4. Map the process app endpoints to environment endpoints.

    Process app endpoints can be mapped to environment endpoints automatically using the initial configuration, but you can change the environment endpoint configuration as needed to deploy to different environments. Some process apps do not require endpoints.

    See New Environment Endpoint Dialog Box [page 76] for details.

> **Note:** If the process app you are deploying contains a report with the same name as an existing report in the target environment, and the two reports are in the same project, are stored in the same table, and have the same access level (for example, public or private), the deployment generates a warning about duplicate reports.

CAUTION:

> If you deploy a process app that contains a Web service, and the defining `.wsdl` file contains an incorrect service location for the environment to which you deployed, you need to edit the endpoint in SBM Application Repository to point to the correct service location and then redeploy the process app so the changes can take effect. (You may also want to get the latest changes back to SBM Composer.)
>
> If you then clone the environment and deploy the same process app to the cloned environment using default settings, the new deployment will not use the existing endpoints that you edited; it will instead create new endpoints with similar names. These new endpoints will contain the service location from the defining `.wsdl` file, so you need to edit them (as you did in the original environment) to point to the correct service location for the Web service. (Alternatively, you can edit the deployment in the Deploy Options Dialog Box [page 74] in SBM Composer to use the correct endpoints and delete the duplicate ones.) In either case, you need to redeploy the process app to the cloned environment so these changes can take effect.

Using Quick Deploy

The **Quick Deploy** command enables you to deploy a process app without being prompted to provide the information in the **Deploy Process App** dialog box.

> **Note:** If this is the first time you are deploying this process app, it is recommended that you use the **Deploy** command instead. The **Deploy** command enables you to see the default settings and change them, if necessary.

To deploy a process app using the **Quick Deploy** command:

1. In the **Quick Deploy** area of the **Deployment** tab, clear the **Create Versions of Process App Elements** check box if you do not want to create process app versions in Application Repository.

It is a good practice to clear this check box when you want to deploy and test a process app you are designing, but are not ready to commit your changes. This also prevents extraneous versions of the process app in the database.

Select this check box if you are ready to commit your changes.

> **Note:** This check box cannot be cleared if the environment to which you are deploying does not have the **Enable Development Deployment** option selected in the **Composer** list in the **Edit Environment** dialog box in Application Repository.

2. Select the **Keep Design Elements Checked Out** check box if you want design elements that were already checked out to remain checked out after the deployment. This eliminates the step of checking out design elements after you deploy, so you can continue to design.

   Clear this check box if you have finished your design work and want to use the **Quick Deploy** command to deploy the process app.

   > **Note:** This check box is unavailable if the **Create Versions of Process App Elements** check box is cleared.

3. In the **Deploy To** list, select the environment to which you want to deploy. You are prompted to map any unmapped endpoints that cannot be automatically mapped.

4. Click **Quick Deploy**.

To determine the status of the deployment, look at the Message List, or right-click the process app name at the top of App Explorer, and select **Get Deployment Status**.

Using Other Deployment Features

After you perform a deployment, you might want to have SBM Work Center open automatically so you can see how the process app works, or open the SBM Work Center without doing a deployment. You also might want to open Application Repository to give other users permission to deploy the process app.

To use other deployment features:

1. Click the **Deployment** tab on the Ribbon.

2. In the **Launch** area of the tab, clear or select the **Launch Work Center When Deployed** check box to specify whether you want to open the SBM Work Center after the deployment completes.

3. An application is selected in the **Startup Application** list. If you want to change it, select another one.

4. Click **Work Center** to open SBM Work Center without performing a deployment. SBM Work Center opens to the application specified in the **Open Application** list, from the environment to which the process app was last deployed.

5. Click **Application Repository** to open Application Repository.

Exporting a Process App

Use export to create process app blueprint (.msd) or template (.mst) files.

For example, you can export a process app to send the blueprint file to someone who needs to approve it, but who is not authorized to check in the process app or publish the process app. The recipient could import the blueprint, make any needed changes, and export it again to send the blueprint file back to you.

You could also export a process app and send the blueprint file to someone else to load into SBM Application Repository, effectively publishing the process app.

To export a process app:

1. Open the process app or template and select **File | Import and Export | Export to File**.

   SBM Composer validates the process app. (You cannot export a process app that fails validation.)

2. If the process app is valid, in the **Save As** dialog box, navigate to the directory where you want to save the exported file.

Advanced Process App Tasks

- Creating a Template [page 55]
- Comparing and Merging Process Apps [page 57]
- Deleting a Process App [page 60]
- Upgrading a Process App [page 60]
- Upgrading a Snapshot [page 60]
- Working in a Patch Context [page 61]
- Managing Internal Identity and Design Numbers [page 62]
- Modifying Locked Elements in a Published Process App [page 64]

Creating a Template

A template is a type of process app that is a starting point for creating other process apps. Process apps created from templates have unique design numbers. See About Templates [page 42] for more information.

CAUTION:

⚠️ The operation of converting a process app into a template cannot be undone. You cannot convert a template back into a process app. As a precaution, back up your process app before you convert it by selecting **File | Import and Export | Export to File**.

To create a template:

1. Create a process app that you want to use as a template or open an existing process app to use as a template.

2. Click the process app name in the **All Items** filter of App Explorer, and then click the **Convert to template** button in the process app editor.
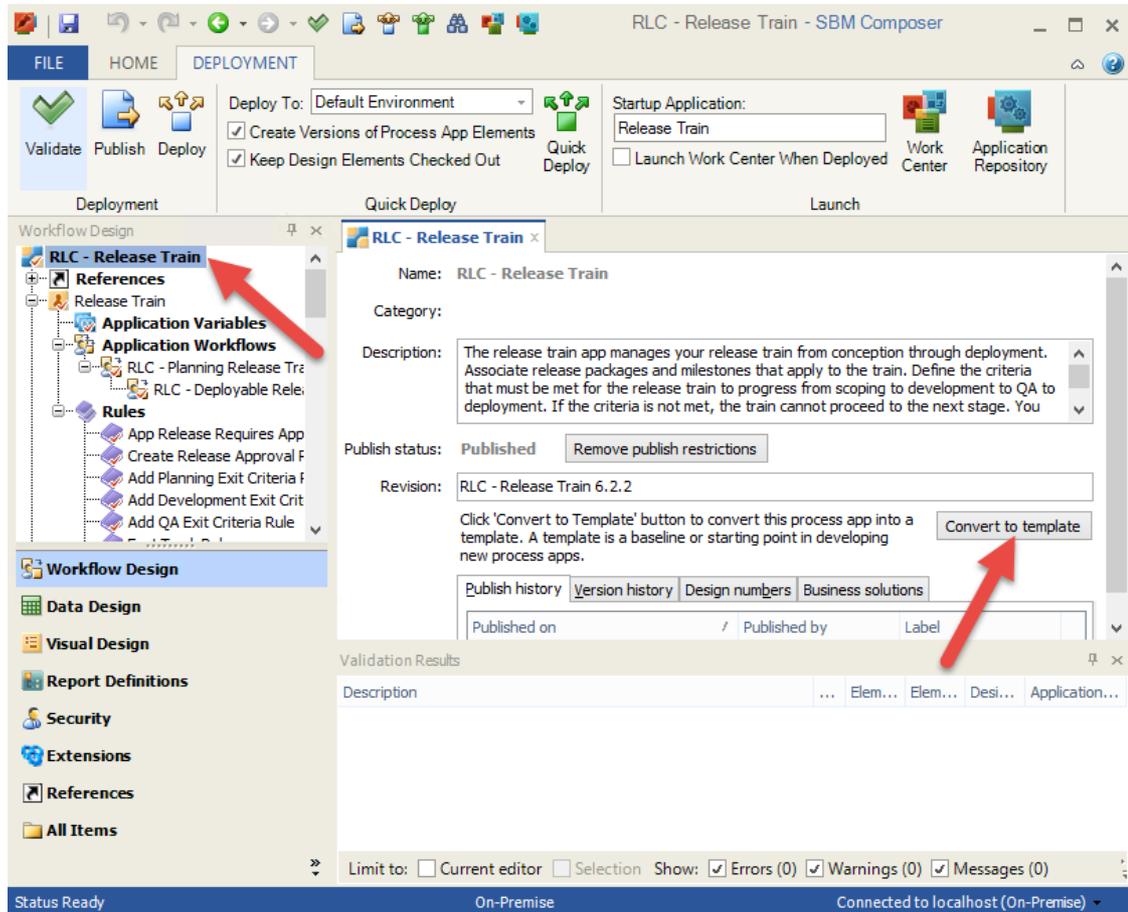
Click **Yes** when you are warned that the applications in the process app you are converting cannot be undone. The color of the process app icon changes from yellow and blue to green.

3. Export the process app to a file by selecting **File** | **Import and Export** | **Export to File**. The **Save As** dialog box opens. Navigate to the location where you want to save the template.

> **Tip:** To save the template directly to your local library, save it to the `My Documents\SBM Composer\Library\Templates` directory.

4. Save the process app with an `.mst` file extension.

Adding Templates to Your Library

Add templates to your library so they are available when you create new process apps.

To add a template to your library, you can either copy them directly to the template location (`My Documents\SBM Composer\Library\Templates`) or follow these steps:

1. Select **File** | **New**.

   The **Create New Process App** dialog box opens.

2. Click **Add to Library**.

3. Navigate to the template (.mst) file, and then click **Open**.

4. Click **Create** to add a new process app based on the template. The template will be available the next time you create a new process app.

Updating a Template

To update an existing template:

1. Select **File | Import and Export | Import**.

2. Navigate to the .mst file you want to update, then click **Open**.

3. Modify the template as needed.

4. Select **File | Import and Export | Export to File**.

5. Optionally, add the template to your library, using the steps in Adding Templates to Your Library [page 56].

Comparing and Merging Process Apps

This topic describes how to use the compare and merge feature. For high-level information and use cases, see Compare and Merge [page 45].

To compare two process apps:

1. Open one of the process apps. This needs to be the process app that you want to merge items into, and is referred to as the "open process app."

2. Select **File | Compare**, and then select the process app that you want to compare to the open process app. This is referred to as the "comparison process app" and will be displayed on the right, while the open process app will be displayed on the left. To open this process app, do one of the following:

    • Choose **With Local File** to compare with a blueprint file stored on your computer. In the standard Windows dialog box that opens, locate the file (typically with an `.msd` extension).

    • Choose **With Published Process App** to compare with a process app stored in the repository. In the Select Published Process App Dialog Box [page 80], select the version of the published process app.

    • Choose **With recently compared Local File** to compare with the last process app you compared that is stored on your computer.

3. The Compare Process Apps Mode [page 70] opens. The comparison process starts, and a progress bar is displayed at the bottom of the window.

4. After the comparison process finishes, the differences are highlighted in the colors shown in the **Legend**. A **Comparison Report** at the top of the editor uses a callout to textually summarize each change in both process apps.

5. If you click an item in App Explorer in either the **Comparison Report**, the open process app, or the comparison process app, the item opens in the other two areas as well. The views are synchronized, so you can see the item side-by-side for easy

comparison. If an item only exists on one side, an explanatory message is displayed on the other side.

To copy items:

1. In the comparison process app, right-click the item you want to copy.

2. Select **Copy to Open Process App**. This command is available only for items that can be copied (that is, items that do not exist in the open process app).

> **Note:** For performance reasons, the comparison is not fully updated automatically. A "comparison is out-of-date" link warns you about this. Click the link or the **Refresh** button at the top right corner to update the comparison.

> **Note:** The following limitations currently apply:
>
> - You cannot merge the properties of an item; you must manually change them. For example, if the **Size** value on the **Options** tab of a *Text* field Property Editor differs, you must type the comparison value into the Property Editor for the field in the open process app. (Alternatively, you can delete the field from the open process app and then copy it from the comparison process app.) However, you can copy objects in an item. For example, if a *Multi-Selection* field has an additional value in the comparison process app, you can copy it from the **Options** tab of the Property Editor for the field.
>
> - Related items are not copied along with an item. For example, when you copy a state to an open process app, the connecting transitions are not copied with it. In some cases, if a related item is essential to an item you want to copy, you must copy the related item first. For example, if a form exists only in the comparison process app, and the form is based on a table that is only in the comparison process app, you must copy the table first. SBM Composer provides a warning in this case.

To dismiss an item:

- Click the callout in the **Comparison Report**. If the **Show dismissed items** check box is selected at the top of the report or in the **Options** menu, the item is crossed out and in a lighter shade; otherwise, it is hidden. Dismissed items are not highlighted as differences in the comparison process app or open process app.

  CAUTION:

  > The state of an item persists in subsequent comparison sessions, if the sessions are consecutive. For example, if you dismiss an item, close the comparison or close SBM Composer, and then run the comparison again in a later session, dismissed items remain dismissed. However, if you compare with a different process app between sessions, the item will no longer be dismissed when you resume the comparison.

To restore a dismissed item:

1. Make sure the **Show dismissed items** check box is selected.

2. Click the callout.

To restore all dismissed items:

- Select **Restore Dismissed Differences** in the **Options** menu.

To hide translated strings:

- Select the **Ignore localizable properties** check box on the **Comparison Report** or in the **Options** menu at the top of the report.

To hide unchanged items:

- Clear the **Show unchanged items in collections** check box on the **Comparison Report** or in the **Options** menu at the top of the report.

To export the comparison report:

1. Click **Export** at the top right corner.

2. Browse to the folder where you want to store the report, and then click **OK**.

Three files are placed in the folder. Open the HTML file named "report" to view the report.

To close the comparison:

- Click the **Close Comparison** button at the top right corner. The normal view of the open process app is restored.

To change the comparison colors:

1. Click **Composer Options** on the **File** menu.

2. Click the **Comparison** tab.

3. Change the colors as described in Comparison Options [page 484].

To store, restore, or reset the layout of the three windows in comparison mode:

1. Click **Layout** at the top right corner.

2. Do one of the following:

   - Click **Store Layout** to store the current layout as a custom layout.

   - Click **Restore Layout** to arrange the windows using the stored custom layout.

   - Click **Reset Layout** to arrange the windows using the default layout.

   - Click **Auto Restore** if you want the windows to be arranged using the stored custom layout every time you enter comparison mode.

For more information about these options, see Compare Process Apps Mode [page 70].

Deleting a Process App

To delete a process app, select **File | Delete Process App** and then select whether you want to delete the process app from the repository as well as the copy in the Local Cache.

> **Note:** Deleting a process app from the repository does not affect any environments to which it has been deployed. For details, see the *SBM Application Repository Guide*.
>
> If the deleted process app is already deployed, an SBM System Administrator user has to eliminate application-name and table-name conflicts by deleting the primary table and associated auxiliary tables for applications in that process app. The new process app cannot be deployed until this happens.

Upgrading a Process App

When you upgrade to a later version of SBM Composer, process apps created in the earlier release might need to be upgraded. To do so, upgrade the blueprint that contains the process app.

To upgrade a process app:

1. Select **File | Import and Export | Import**. The **Import Process App Blueprint** dialog box opens.

2. Select the blueprint file for the process app, and click **Open**. A message tells you if you need to upgrade the blueprint to work with the current version of SBM Composer.

3. Click **OK**. The **Import Process App** box shows the status of the upgrade. After the process app is upgraded, it opens in SBM Composer.

4. Deploy or redeploy the process app.

Upgrading a Snapshot

Snapshots created in an earlier version of SBM Composer might need to be upgraded when a new version is installed. If you attempt to load or promote a snapshot created in an earlier version of SBM into Application Repository, you receive a warning that the snapshot needs to be upgraded.

> **Important:** After you upgrade a snapshot, you cannot edit the process app that includes it in an earlier version of SBM Composer.

To upgrade a snapshot:

1. Select **File | Import and Export | Upgrade Snapshot**. The **Upgrade Process App Snapshot** dialog box opens.

2. Select the snapshot file (either an `.mss` or a `.zip` file) and then click **Open**.

After the snapshot is upgraded and saved to a file, the upgraded snapshot file must be loaded into Application Repository (using the **Load from File** command) and then promoted from Application Repository into the SBM Application Engine Web Server (using the **Promote** command).

Working in a Patch Context

A patch context allows parallel and concurrent development to occur on the same process app, so that patches can be applied to production process apps as ongoing development of the main process app continues. When performing maintenance work, a designer can create a patch context by opening a published process app by its label.

The patch context serves as the baseline for the maintenance work. Multiple process app designers can do concurrent development in a patch context.

Restriction: Only one process app designer can work on a single design element at the same time.

Changes a designer makes when working in a patch context apply only to the patch context. They do not affect ongoing development of the head (tip) version of the process app.

Note: A patch context is not the same as a branch in a version control application.

Some changes can be made directly in a patch context. If you want these changes to also apply to the head version, make them manually in the head version. Other changes cannot be made directly in a patch context, and are described in the following section.

Adding Design Elements to a Patch Context

To protect against data loss, the following design elements cannot be added directly to a patch context:

- Applications, including these design elements:

  - Tables

  - Fields

  - Selections for *Single Selection* and *Multi-Selection* fields

  - Application Workflows

  - States

  - Report Definitions

  - Roles

- Orchestrations and orchestration workflows

You can, however, copy these design elements from the head version to the patch version. If a design element does not exist in the head version, create it there first.

Restriction: Applications and orchestrations are exceptions to this. You cannot add applications or orchestrations to a patch context, and cannot copy them from the head version.

When you right-click the design element heading in App Explorer in the patch context, and then select **Add Existing <Design Element>**, the **Add Existing <Design Element> from the Head (Tip) Version** dialog box opens. In this dialog box, select the design element that you want to add and then click the **Add** button.

> **Note:** Design elements containing fields, states, tables, and so on in the head version must be checked in before you can copy these entities to the patch context.

When adding items from the head version, the parent design element in the patch context must be checked out. For example, to add a workflow from the head version, make sure the application in the patch context is checked out.

After a table, application workflow, report definition, or role is added from the head version, that version of the design element is immediately labeled with the patch label, and the parent design element in the patch context is checked in (and checked out again if it was previously checked out).

After a field or state is added from the head version, the field or state is simply added to the parent design element and the parent design element in the patch context is checked in (and checked out again if it was previously checked out).

Example of Working in the Patch Context

Version 1.0 of a process app is in production. Susan checks out Version 1.0 (the head version) to continue main development on the process app.

At the same time, a patch needs to be applied to Version 1.0 for customers using the process app in production. The owner of a state needs to change. Bill opens Version 1.0 by its label in the **Open Labeled Version** dialog box. The *Version 1.0* label becomes *Version 1.0 PATCH*. Bill changes the owner of the state in the patch context (Version 1.0 PATCH).

Bill deploys Version 1.0 PATCH, and its label becomes *Version 1.1*. He manually changes the owner of the state in the head version so it matches the patch context. When he deploys the head version, its label becomes *Version 1.2*.

The **Open Labeled Version** dialog box resulting from this example is shown in the following illustration.

Open Labeled Version			
Label	Published By	Published On	Comment
Version 1.0	bill	3/2/2009 4:31:...	
Version 1.0 PATCH	-	-	(not published)
Version 1.1	bill	3/2/2009 4:35:...	
Version 1.2	bill	3/2/2009 4:40:...	

Managing Internal Identity and Design Numbers

SBM Composer uses several mechanisms for maintaining process app identity. This ensures process apps are not overwritten as they are worked on by multiple developers in the same environment. There are times, however, when you may need to reset a process app's identity, as discussed in the following sections.

When process apps are created, SBM Composer assigns two types of identities:

- **Internal Identity**

Used to ensure a process app is unique in the repository and in the runtime environment. Internal IDs are sometimes referred to as UUIDs.

- **Design Identity**

  Used primarily for referencing design elements in other applications.

Resetting Internal Identity

The internal identity of a process app cannot be changed once the process app is deployed. You can import a process app and give it a new identity, however. This is useful for creating a process app based on a different process app.

To reset the internal identity for a process app:

1. Select **File | Import and Export | Import as New**.

2. Navigate to a blueprint (.msd) file that was previously exported or downloaded from an external source, such as the Community website.

3. Carefully read the message that appears, then click **Yes** to continue the import process.

4. Rename the following elements in the imported process app:

    - Process app names

    - Primary table database table names

    - Auxiliary table database table names

    - Workflow names

    - Application tab names (not required to be unique, but recommended)

You can now deploy the process app into the same environment as the original process app.

Resetting Design Identity

Design numbers are used to define a process app's design identity.

Typically, you do not need to pay attention to design numbers. However, it becomes important when an application in one process app references an application in another process app. In this case, the reference contains not only the internal identity of the referenced application, but also the design number of the referenced application. This permits the reference to continue to be valid, even though the internal identity may have been changed if the process app was created from a blueprint file. When there is ambiguity about which application was referenced, SBM Composer uses design numbers to present you with a choice of those applications the referencing application was designed to work with.

If process apps with the same design number diverge in purpose and content such that they cannot still be considered related, you can reset the design numbers of their applications.

To reset design numbers:

1. Select the process app in App Explorer.

2. In the process app editor, click the **Design numbers** tab.

3. Click **Reset design numbers**.

Modifying Locked Elements in a Published Process App

After a process app is published, certain design elements are locked and cannot be modified. This is because data that depends on those elements could have been created at runtime. For example, you cannot delete a state because it could still exist in the state history for one or more items, and you cannot change the style of a numeric field because items could already exist in that table with that field set to a number with the original style.

However, there may be cases when you need to modify or even delete locked elements. For example, you may add elements to your process and determine during testing that they need to be removed or modified.

The following list describes the items that are locked after a process app is published:

- The process app name and category cannot be changed.

- The database name for fields and tables cannot be changed.

- Internal names for workflows, states, and transitions cannot be changed.

- Fields, selection values, and states cannot be *permanently* deleted. They are instead *marked* deleted (or disabled, in the case of states).

- The **Backfill to existing items** check box is not available for *Multi-Selection* fields.

- Some styles options for *Numeric* and *Date/Time* fields cannot be changed.

- *Text* fields cannot be changed to the **Fixed length** style from the **Memo** or **Journal** style and cannot be changed from the **Fixed length** style to the **Memo** or **Journal** style.

- The related table for *Relational* fields cannot be changed.

- The prefix for applications cannot be changed.

To reset a process app so you can modify locked items:

1. Select the process app at the top of the App Explorer.

2. Click **Remove Publish Restrictions**.

3. Acknowledge the warning that informs you that you must first undeploy the process app if you plan to deploy it to an environment where it has already been deployed. If you do not first undeploy the process app, your changes made in these steps are ignored. The publish status changes to "published/unlocked."

4. Modify items as needed.

5. Deploy the process app.

   a. In the Quick Access Toolbar, click **Deploy**.

b. In the **Deploy Process App** dialog box that opens, select your environment in the **Deploy to** list, and complete other fields as needed.

> **Note:** To restore original elements, manually change them to their original values.

Chapter 5: Process App Settings

The following sections provide descriptions of settings used to configure process apps. Sections are organized by dialog box.

Create New Process App Dialog Box

Element	Description
Available process app templates	Lists available templates. When you select a template icon, a brief description is displayed in the pane on the right. The file name is included in the description. Select a template icon and click **Create** to open the **Configure Process App** dialog box and create the new process app. For more information, see Configure Process App Dialog Box [page 72]. When you click the **Browse** button, a standard **Open** dialog box opens that lets you bring in a template file or a blueprint file that is stored on your computer.
Library location	Indicates the location for your local template library.
Add to Library	Click to add templates you have created to your library. This ensures the templates are always available when you want to create a new process app based on the template. For details, refer to Creating a Template [page 55]. **Note:** To delete templates, remove them from the library location.

Process App Editor

Element	Description
Name	The process app name. Note that process app names should be unique within the repository.

Element	Description
Category	The category to which the process app was assigned when it was created. **Note:** You cannot change the name or category of a published process app unless you first remove publish restrictions. For details, refer to Modifying Locked Elements in a Published Process App [page 64].
Description	Optional comments or notes about the purpose of the selected process app.
Publish Status	Indicates one of three statuses (not published, published, or published/unlocked). Use this information to determine if certain elements of the process app can be changed. For example, you cannot change the database name for fields and tables after a process app is published. To unlock a published process app, click **Remove publish restrictions**. This enables you to modify locked elements. For details, refer to Modifying Locked Elements in a Published Process App [page 64]. **CAUTION:** If you plan to redeploy the process app to an environment after changing previously locked elements, you must first undeploy the process app from that environment for changes to the unlocked process app take effect. For details on undeploying, refer to the *SBM Application Repository Guide*.
Revision	Optional comments or notes about this revision of the process app.
Convert to template	Converts the process app into a template, which can be a starting point for creating other process apps. For more information, see Creating a Template [page 55].
Publish history	The record of the process app blueprint as published to Application Repository, including (where applicable) the existing process app as "gotten" from a deployment environment.
Version history	The record of the process app design element as checked in to the repository. You can use the labels to match design elements to process app blueprints.

Element	Description
Design numbers	The application name and its associated design number. Applications in some process apps have the same design number, because they serve a similar purpose and can reference each other. If the process apps have diverged enough from their original purpose, and you want them to have separate design numbers, click **Reset design numbers**. **Note:** For more information about design numbers, see Managing Internal Identity and Design Numbers [page 62]. For more information about references, see About References [page 369].
Business Solutions	Shows information about pre-built solutions, such as Release Control.

Add Existing Design Element from the Head (Tip) Version Dialog Box

Use this dialog box to copy a design element from the head (tip) version. This is the only way to add some design elements to a patch context.

For more information, see Working in a Patch Context [page 61].

Important: The design element you want to add must already exist in the head version.

Check In Design Elements Dialog Box

Use this dialog box to indicate which design elements you want to check in to the repository.

For details, refer to Concurrent Development [page 44].

Element	Description
Process app tree	Select the check boxes to check in individual elements, entire types of elements, entire applications or orchestrations, or the entire process app. Click **+** and **−** to expand and collapse levels of the hierarchy.
Comment	Enter text that to appear in the version history for the selected design elements.

Check Out Design Elements Dialog Box

Use this dialog box to indicate which elements you want to check out of the repository. Elements must be checked out before you can edit them.

For details, refer to Concurrent Development [page 44].

Element	Description
Process app tree	Select the check boxes to check out individual elements, entire types of elements, entire applications or orchestrations, or the entire process tree. Click **+** and **−** to expand and collapse levels of the hierarchy.

Compare Process Apps Mode

This mode is used to compare two process apps. For an overview about this functionality and for use cases, see Compare and Merge [page 45].

There are three windows: at the top, a comparison report that textually describes the differences; on the left, the open process app; and on the right, the comparison process app. This is the default layout.

You can move and resize each window and store the arrangement as a custom layout. This is useful when you have multiple monitors, because you can maximize use of the available screen space. For example, you could move the compared process app window to its own screen. The next time you enter comparison mode, you can quickly restore this custom layout. You can also have it automatically applied each time you enter comparison mode, or reset the default layout. For more information, see the "Layout" row in the following table.

Differences are marked by highlight colors. This provides a quick, visual way to determine items that are in one process app but not the other, and items that are in both process apps but that differ. These indicators cover a wide range of detail. For example, if a state in the application workflow in one process app has an additional outgoing transition, the workflow name in App Explorer will be highlighted, as well as the transition in the workflow editor. If a field option is different in one process app, the option will be highlighted in the field Property Editor.

After you examine the differences between the two process apps, you could merge some or all of the changes. You can copy items from the comparison process app into the open process app, and make manual changes in the open process app. For instructions, see Comparing and Merging Process Apps [page 57].

Element	Description
Comparison Report	This window includes an App Explorer-like navigation pane and a summary pane. When you click an item in the navigation pane, details about it are described in the summary pane. Annotations and colors indicate the differences between the items in the two process apps. Clicking an item in the navigation pane also opens the relevant Property Editor or editor in both the open and comparison process apps.

Element	Description
Options	Shows or hides items that were dismissed or that are identical to items in the other process app. **Ignore localizable properties**: Do not display differences for strings that can be translated into other languages. These strings include labels, names, end-user help text, display text, values for *Binary/Trinary* fields, and so on. If this check box is not selected, all translated strings will show up as differences when you compare with a new, unlocalized version of the process app. **Show dismissed items**: Display items that were dismissed. These items are crossed out and in a lighter shade. This option must be selected if you want to restore individual dismissed items. **Show unchanged items in collections**: Display all items, even those that are the same in the other process app.
Legend	Defines the meaning of the highlight colors used in the editor to denote differences. To change the colors, see Comparison Options [page 484].
Refresh	Refreshes the comparison after you copy an item from the comparison process app to the open process app, or change something else in the open process app. You can also click the link to the left of this button to refresh the comparison.
Export	Exports the comparison report to a folder on the file system of your computer.
Options *(menu)*	**Show Dismissed Items** and **Show Unchanged Items in Collections** shows or hides items that were dismissed or that are identical to items in the other process app, as described above. **Restore Dismissed Differences** restores all dismissed items, even if they are not shown.
Layout	The comparison mode elements can be resized and moved to create a custom layout. The following options are available: **Store Layout**: Store the current layout as a custom layout. **Restore Layout**: Arranges the windows using the stored custom layout. **Reset Layout**: Arrange the windows using the default layout, in which the comparison report is at the top, the open process app is on the left, and the comparison process app is on the right. **Auto Restore**: Arranges the windows using the stored custom layout every time you enter comparison mode. After this option is clicked, a check mark appears next to it. The option remains selected in subsequent sessions until you clear the check box.
Close Comparison	Closes the comparison view and restores the normal view of the open process app.

Element	Description
Open process app	By default, the open process app is displayed on the left side, below the Comparison Report.
Comparison process app	By default, the comparison process app is displayed on the right, next to the open process app. This process app is read-only.

Configure Process App Dialog Box

Use this dialog box to configure a process app that you are creating based on a template under **Available templates** in the **Create New Process App** dialog box. For more information, see Create New Process App Dialog Box [page 67].

> **Note:** The process app name or application name of the template is included in the label for each group of elements. This helps you avoid using the same name in your new process app.

> **Note:** The **Application**, **Table**, and **Workflow** elements are not in this dialog box if you are creating an empty process app or an orchestration process app.

Element	Description
Process app: Process app name	Type a name for the process app. You cannot use the name of the process app template.
Process app: Category	Type or select a process app category. Categories let you sort process apps in your Local Cache or the repository. Categories that you previously typed in this field are not in the list.
Application: Application name	Type a name for the process app application. Applications within a process app must also be uniquely named. Applications created independently in different process apps can have the same name. In this case, changes to one do not affect the other.
Application: Internal name	Type the uppercase name under which the application is stored in the database. > **Note:** The internal name can contain only ASCII alphanumeric characters (A-Z, a-z, 0-9). Any other characters are ignored as you type them.
Application: Tab name	Type the label (up to 16 characters) that appears to users. The name on this label must be unique.

Element	Description
Table: Table name	Type the name for the primary table (such as `Issues`). **Tip:** The name that you type in this field automatically populates the **Database table name** field, which must be unique within the SBM database and is limited to 24 Unicode characters. You should limit the logical name to 24 characters, as additional characters will be left off of the **Database table name.**
Table: Singular item name	Type the name of a single item in the table (such as `Issue`). You cannot use the name that is used in the process app template. **Note:** This field is automatically filled as you type in the **Table name** field, so you probably need to edit this field to use it as intended.
Table: Database table name	Type the uppercase name of the table in the database. This name must be unique. SBM Composer automatically fills this field as you type the **Table name**, ignoring characters after the first 24. **Note:** The database table name can contain only ASCII alphanumeric characters (A-Z, a-z, 0-9). Any other characters are ignored as you type them.
Workflow: Workflow name	Type the name (32 characters or less) of the application workflow to be used by the process app.

Create Patch Context Dialog Box

Use this dialog box to create a new label, called a patch context, for the changed version of a published process app. Appending a unique identifier (such as "Patch 1") to the original label helps relate the new version to the original and helps keep subsequent versions straight.

If you do not accept the option to create a patch context, SBM Composer opens the selected process app in read-only mode. This means that you cannot check out, check in, or undo the checkout of the process app.

Deploy Process App Dialog Box

Use this dialog box to deploy a process app. If the process app has not been validated, checked in, or published, these functions are performed transparently, unless there is an error.

Element	Description
Environment	Select the environment to which you want to deploy the process app. For environments to be seen in this list, in the **Edit Environment** dialog box in Application Repository, **Enable Deployment** must be selected from the **Composer** list.
Label	Type the label or version name for the process app, if you want to modify the text that identifies this version of the process app.
Visibility	Select the **Allow others to deploy this version of the process app** check box or clear it.
Comment	*(Optional)* Type information about the deployment.
Options	Click this button to open the Deploy Options Dialog Box [page 74] if you want to specify other deployment options.

Note: Applications and tables in your deployment environment must be uniquely named. If the deployment fails because an application or table in the environment already has the same name as one of the applications or tables in the process app you are deploying, rename the application or table in the process app you are deploying, and try again.

Deploy Options Dialog Box

Use this dialog box to specify options when you deploy a process app.

Element	Description
Stop deploy	Specify whether the deployment should stop automatically if any warnings are raised during the deployment process. (By default, deployment always stops if there are errors.)
Schedule	Select this check box to delay the deployment. Use the controls to set the time and date.
Verify endpoints	Select this check box to confirm that each endpoint can be accessed before starting the deployment.

Element	Description
Email notification, Email address	Use these controls to specify what deployment related notifications that you want to receive and to what address they should be sent. **Important:** This only works if your administrator enabled e-mail notifications. If this was not done, notifications will not be sent, even if you complete the fields in this dialog box.
Endpoint mappings	This section of the dialog box shows the list of process app endpoints with the type of endpoint, Automatic, Custom, or System, the source applications and orchestrations where the endpoints were created, and the environment endpoint to which each process app endpoint will be mapped. For each **Process App Endpoint**, click to the right of the **Mapped Environment Endpoint**, and then select one of the following: • **New Endpoint** Opens the New Environment Endpoint Dialog Box [page 76]. • **New Endpoint based on** Opens the New Environment Endpoint Dialog Box [page 76] and allows you to use the current endpoint as a template. • **View selected endpoint details** Opens the **Endpoint Details** dialog box, which provides a read-only view of the current process app endpoint settings. • *CurrentEndpointName* Leaves the current mapping in place. • *OtherEndpointName* Select a different endpoint to use as the template, including any custom endpoints.

Delete Item Dialog Box

This dialog box opens when you try to delete a design element, but the design element cannot be deleted because it is being used by the process app.

For example:

- You add a form to an application, use the form on a state or transition, and try to delete the form. The form cannot be deleted because the state or transition uses it.

- You create a report definition for an auxiliary table, and then try to delete the auxiliary table. The auxiliary table cannot be deleted because the report definition uses it.

To go to the part of the process app that contains the design element that is being used, either double-click the element in the dialog box.

> **Note:** If you want to delete a design element that is being used by the process app, you must remove the connections between that design element and all of the design elements that use it.

Get Latest Design Elements Dialog Box

Use this dialog box to get the latest version of the selected process app or design elements from the repository. This command does not check out the process app or design elements.

> **Note:** This command is disabled for design elements you checked out. If you want to overwrite these design elements with the repository version, first perform the **Undo Checkout All** command.

Element	Description
Process app tree	Click the check boxes to get individual design elements, entire types of design elements, entire applications or orchestrations, or the entire process app. Click **+** and **–** to expand and collapse levels of the hierarchy.

New Environment Endpoint Dialog Box

For more information about endpoints, refer to About Endpoints [page 41].

Element	Description
Name *(required)*	Name by which the new environment endpoint is identified in the **Endpoint mappings** section of the **Deploy Process App** dialog box.
Description	An optional description of the environment. This text is displayed below the list of available environments in the **Deploy Process App** dialog box.

URL *(required)*	Enter (or select) the URL for the environment endpoint.
	
	Examples of valid URLs follow:
	
	`http://`*serverName*`:`*port*`/eventmanager/services/`
	`ALFEventManager`
	
	`http://`*serverName*`:`*port*`/orchestrationName-`
	→*orchestrationWorkflowNameInLowercase*`servlet`

Element	Description

Authentication	Select the authentication method to be used for the environment endpoint:
	
	• **None**
	
	  Select **None** if the endpoint does not require authentication.
	
	• **HTTP Basic**
	
	  Select **HTTP Basic** if the endpoint points to a Web service that requires Basic Access Authentication and you want to provide the credentials (user name and password).
	
	  > **Important:** The **HTTP Basic** credentials set in Application Repository for an endpoint override Basic credentials that are set in orchestration workflows. For example, if username `joe` and password `pwd` are specified in the orchestration, but Application Repository specifies **HTTP Basic** with different credentials, the Application Repository credentials are used instead. If you do not want to override the credentials specified in the orchestration workflow, select **None** in Application Repository.
	
	• **NTLM**
	
	  Select **NTLM** if the endpoint points to a Web service that requires Windows domain credentials. Enter the username, password, and domain name.
	
	• **Security Token**
	
	  Select **Security Token** if the endpoint points to an SBM Web service or other internal endpoint.
	
	• **Oauth 2**
	
	  Select OAuth 2 for RESTful calls that use OAuth 2 security. For details on obtaining and specifying the OAuth 2 token that is required, refer to the *SBM Application Repository Guide*.

Open Labeled Version Dialog Box

Use this dialog box to select the version by the label assigned to the process app on publication. If you use labels to denote patches to your process apps (as in MyProcessApp, MyProcessApp 1, MyProcessApp Patch 1.1, and so on), they are easier to find in this dialog box.

You have the option to create a patch context in which to make any changes, if one does not already exist. If you do not accept the option to create a patch context, SBM Composer opens the selected process app in read-only mode.

For detailed information about patch contexts, see Working in a Patch Context [page 61].

Open Process App Dialog Box

Use this dialog box to select the latest version of a process app that was checked in to the repository or a process app that you saved to your computer (the Local Cache).

Note: When you open SBM Composer for the first time, the list of recently opened process apps is empty. Any new process apps are added to the list after you close them.

Element	Description
Look in	Choose whether to select a process app stored on your computer (Local Cache) or a process app that was checked in to the repository.
	If a process app in the repository is listed twice, it means someone loaded the process app directly into the repository (from an exported blueprint file or from the SBM Application Engine) since the last time the process app was checked in or published from SBM Composer. The process app with the red icon is the one that was loaded directly into the repository. The process app with the yellow and blue icon is the one that was checked in or published from SBM Composer.
	A process app with a green icon is a template. For more information about templates, see About Templates [page 42].
	Use the icon, the time stamp, or the description that is displayed below the list to determine which one you want to open.
	CAUTION:
	If you select a process app from the repository that was opened on your computer (that is, a process app that already exists in the Local Cache), SBM Composer warns you that the local process app will be overwritten. You can open the local process app or overwrite it with the process app in the repository, losing any changes you made since the process app was checked in.

Element	Description
Open labeled version	Opens the **Open Labeled Version** dialog box, in which you can open an earlier labeled version of the selected process app. You have the option to create a patch context in which to make any changes, if one does not already exist. For more information, see Open Labeled Version Dialog Box [page 77] and Working in a Patch Context [page 61].

Note: The timestamp in the **Updated on** column reflects the last time the process app as a whole was checked in.

Publish Process App Dialog Box

Use this dialog box to publish the process app so it can be deployed. If the process app was not validated or checked in, those tasks are performed first.

Element	Description
Label	In the **Version name** field, type the label or version name for the process app.
Visibility	If you do not want others to deploy this version of the process app, clear this check box.
Optional Comment	Type an explanation of or comment about the deployment, if needed.

Tip: If other designers have checked in design elements in your process app, by default you will be prompted whether you want to get the latest versions of them from the repository before you publish. You can specify whether the latest versions are always used or ignored without being prompted on the Repository Options [page 483] tab of the **Composer Options** dialog box.

Tip: Applications and tables within the repository must be uniquely named. If the publication fails because an application or table in the repository already has the same name as one of the applications or tables you are checking in, rename the local design elements, and try again. See Check In Design Elements Dialog Box [page 69] for related information.

If the process app was published, a different **Publish Process App** dialog box opens when you right-click the process app name in App Explorer and select **Change Published Scope**.

Element	Description
Name	Displays the name of the process app.
Comment	Type text that will appear in the publication history for the process app.

Element	Description
Scope	**Public**: Anyone with the appropriate privileges can deploy the process app. **Private**: Only you can deploy the process app (typically used for process apps under development).
Baseline Label	Type a baseline label. This label is displayed in Application Repository. **Note:** You cannot change this field when you are changing the scope of publication.

Select Published Process App Dialog Box

Element	Description
Look in (Repository)	Lists the latest published version of the process apps in the repository. **Note:** If the **Look in** element shows Local Cache, SBM Composer is disconnected from the repository. If this is the case, then perform the following steps: 1. Cancel the compare operation. 2. In the lower right corner of the SBM Composer window, click the **Offline** status indicator. 3. Click **Work Offline** to disable that option (that is, to connect to the repository), and make sure that SBM Composer reports its status as **Connected**. 4. Select **File | Compare | With Published Process App** again. The **Look in** element should now show "Repository."

Sort By Dialog Box

Use this dialog box to sort the rows in the displayed table by the values in up to three of the columns in the table.

For each column, select the column name from the drop-down list, and select the sort order (A–Z or Z–A).

Undo Check Out Design Elements Dialog Box

Use this dialog box to discard any changes you made to the selected design elements since you checked them out. This leaves the elements checked in and unchanged in the repository.

Element	Description
Process app tree	Click the check boxes to undo the check out of individual elements, entire types of elements, entire applications or orchestrations, or the entire process app. Click the **+** and **−** boxes to expand and collapse levels of the hierarchy.

Refresh Status of Design Elements Dialog Box

Use this dialog box to refresh the repository status of the selected design elements.

Element	Description
Process app tree	Click the check boxes to update the repository status of individual elements, entire groups of elements, entire applications or orchestrations, or the entire process app. Click the **+** and **−** boxes to expand and collapse levels of the hierarchy.

Where Used Dialog Box

This dialog box enables you to see where any design element is used in a process app and in referenced applications. For example, you could have a large process app that has many transitions and states. It would be useful if you could find out which transitions and states use a particular form. If you have a process app with a referenced application, it would be useful to find out which relational fields use values from a referenced table.

If you searched for items of the same type, select each item in the results to see where it is used. To go to the part of the process app that contains the design element being used, either double-click the design element in the dialog box, or select the design element in the dialog box and then click **Go to location**.

Element	Description
Name	The name of the item you are searching for.
Type	The type of item you are searching for.
Usage Count	The number of times the item is referenced by another design element.

Element	Description
Item	The name of the design element that contains the item being used (for example, **Assign**).
Type	The type of design element (for example, **Regular Transition**).
Design element	The design element that contains the design element being used (for example, **ChangeRequestAppWorkflow**).
Application/ Orchestration	The name of the application or orchestration that contains the design element (for example, **ChangeRequestApp**).

Version History Dialog Box

Use this dialog box to see the history of versions of the process app or selected element that is checked in to the repository. Sort by any of the column headings. Switch to the **As text** tab for a format you can copy and use somewhere else.

To see check-in comments on the **As list** tab, either select the **Show comments** check box, or right-click a version on the tab and then select **Show Comments**.

Chapter 6: Process App Tutorial

The exercises in this tutorial demonstrate how to use SBM Composer to create and deploy a simple process app. The process app will allow you to create an item, assign the item to a user, and then have the user close the item.

The process app that you create will be used as the basis for the orchestration tutorial found in the *SBM Orchestration Guide* or SBM Composer online help.

> **Note:** The complete process app tutorial will take around thirty minutes to complete.

This tutorial covers the following steps:

- Step 1: Create an Application Workflow [page 83]
- Step 2: Add States and Transitions [page 84]
- Step 3: Define Fields [page 85]
- Step 4: Define Security [page 85]
- Step 5: Deploy the Process App [page 86]
- Step 6: Associate Users with Roles [page 87]
- Step 7: Run the Process App [page 87]

Step 1: Create an Application Workflow

In this exercise, you create an application process app. You will use this process app throughout the rest of this tutorial.

To create an application workflow:

1. Click the Composer button, and then select **New**.

2. In the **Create New Process App** dialog box, select **Application Process App** and then click **Create**.

3. In the **Configure Process App** dialog box, in the **Process App Name** box, type `MyProcessApp`.

4. In the **Category** box, type `Examples`. Categories help organize your process apps.

5. In the **Application Name** box, type `MyApp`. This causes the other fields to be populated with variations of MyApp. For information on what each field refers to, press **F1**.

6. Change the **Workflow Name** to `MyAppWorkflow`.

7. Click **OK**. The **MyApp** application and associated design elements appear in App Explorer.

8. Continue to Step 2: Add States and Transitions [page 84].

Step 2: Add States and Transitions

In this exercise, you add an "active" state called *Assigned* and a "completed" state called *Closed* to **MyAppWorkflow**. Then you create transitions between the **New** state and the **Assigned** state and between the **Assigned** state and the **Closed** state.

To add states and transitions in MyAppWorkflow:

1. In App Explorer, under the **Application Workflows** heading, select **MyAppWorkflow**.

2. Set the **Manager** role as the owner of the **New** state. This determines which users will own items as they are submitted.

   a. Select the **New** state in the workflow.

   b. Launch the **Add Owner Field** wizard by selecting **<Add Owner>** in the **Owner** field of the **General** tab of the Property Editor.

   c. On the **Select or Add Roles** panel, enter the following information for the fields then click **Add Role**.

      - **Name**—`Manager`

      - **Template**—`Administrator`

   d. Click **Next**, select **Create new field**, and then click **Finish**. A new role and new field have now been created. The Manager role has all permissions. These permissions will be reduced in Step 4: Define Security [page 85].

3. From the **States** section of the **Workflow Palette**, drag an "active" state onto the application workflow editor and drop it to the right of the **New** state.

4. In the Property Editor, change the **Name** to `Assigned` and for **Owner**, select **<Add Owner>**. This time in the **Select or Add Roles** panel, select the **User** role and click **Next**.

5. On the **Field Selection** panel, choose to create a new field called `Employee`, and then click **Finish**.

6. From the **States** section of the **Workflow Palette**, drag a "completed" state onto the application workflow editor and drop it to the right of the **Assigned** state. Change the **Name** to `Closed`.

7. From the **Transitions** section of the **Workflow Palette**, drag a "regular" transition onto **New**, release the mouse button, and then click the **Assigned** state.

   This creates a transition from the **New** state to the **Assigned** state.

8. Change the **Name** to `Assign`, and then press the Enter key.

9. Add another regular transition from the **Assigned** state to the **Closed** state, and change the transition name to `Close`.

10. Continue to Step 3: Define Fields [page 85].

Step 3: Define Fields

In this exercise, you define the fields in the **MyApp** primary table. You add selection values to the existing *Item Type* field, and add a new custom field to your table. The custom field tracks whether the item was submitted by a customer or not.

To define fields in MyApp primary table:

1. In App Explorer, select **Data Design**. The **MyApp** table is displayed in the table editor. If not, select the **MyApp** node.

2. Add selection values to the *Item Type* field:

   a. In the MyApp (Primary Table), select the *Item Type* field.

   b. In the Property Editor, add a selection by clicking **Add** on the **Options** tab.

   c. Change the selection's **Value** to `Bug` and set the **Item ID Prefix** to `BUG`.

   d. Add another selection with a value of `Enhancement` and set the **Item ID Prefix** to `ENH`.

3. Add a custom field to track customer submitted issues:

   a. In the **Field Types** section of the **Table Palette**, drag a *Binary/Trinary* field onto the table editor.

   b. On the **General** tab of the Property Editor, change the name of the field to `Customer Submitted`.

   c. If it is not already done, on the **Options** tab of the Property Editor, type `Yes` for the **First Value** and `No` for the **Second Value**.

   d. On the **Attributes** tab of the Property Editor, select `No` as the **Default Value**.

4. Continue to Step 4: Define Security [page 85].

Step 4: Define Security

In this exercise, you assign privileges to the **User** role and restrict privileges for the **Manager** role.

To add and define roles in the MyApp process app:

1. In App Explorer, select **Security**.

2. Select the **User** role.

3. Select the following privileges for the **User** role:

   - Submit New Items

   - Own Items

   - View All Items

   - View Attachments if Owner

- Update Item if Owner

- Transition Item if Owner

4. In App Explorer, select **Manager**.

5. In the Property Editor, clear the check boxes for the following privileges for the **Manager** role:

   - Delete Items

   - Delete Notes on any Item

   - Delete Guest-Level reports

   - Delete Manager-Level reports

     > **Tip:** In a production process app, you might restrict privileges for the **Manager** role, such as removing the privileges to delete items and reports. These privileges should be reserved for administrators.

6. Continue to .

Step 5: Deploy the Process App

You must publish and deploy the process app to your SBM Server to use the process app.

To publish and deploy the process app:

1. Click the Composer button, and then select **Publish**. If the process app has not been saved yet, or if any part of the process app has changed since the last time it was saved, a message box opens reminding you to save the changes. Click **OK**.

2. When the **Check In Design Elements** dialog box opens, you can enter an optional comment in the **Comment** box, and then click **OK**.

   The **Publish Process App** dialog box opens.

3. Complete the **Publish Process App** dialog box:

   - Type an optional comment in the **Comment** box.

   - Type text to identify the version of the process app in the **Label** box.

   - Select the **Allow others to deploy this version of the process app** check box under **Visibility**.

4. Click **Publish**.

   A message box opens indicating whether the process app published successfully or the operation failed.

5. On the **Deployment** tab of the Ribbon, click the **Deploy** button.

   The **Deploy Process App** dialog box opens.

6. From the **Environment** list, select a runtime environment, and then click **Deploy**.

In the Activity Log, a message appears notifying you that the deployment process started successfully.

7. Wait for deployment to complete.

   - If the deployment operation is successful, the following message appears in the Activity Log: "Deployment of 'MyProcessApp' has completed."

   - If the deployment operation fails, the following message appears in the Activity Log: "Deployment of 'MyProcessApp' has aborted."

   If the Activity Log is not available, on the **Home** tab of the Ribbon, in the **Common Views** area, select the **Activity Log** check box. If this check box is already selected and the details are still not visible, select the **Activity Log** tab in the area under the editor pane.

8. Continue to .

Step 6: Associate Users with Roles

In this exercise, you associate users with roles using SBM Application Administrator.

To associate users with roles:

1. Add users to the **Manager** and **User** roles that you created. For example, to add Joe Manager to the **Manager** role:

   a. Log on to Application Administrator.

   b. Click the **Users** button.

   c. Select **Joe Manager**.

   d. Click **Details**.

   e. Click the **Roles** tab.

   f. Under **Projects**, select **MyAppProject**.

   g. In the roles area, click **Manager** and then select the **Disabled (Default)** check box to enable it.

   h. Click **Save**.

2. Continue to .

Step 7: Run the Process App

In this exercise, you test MyProcessApp by running it in SBM Work Center.

To run a process app in SBM Work Center:

1. Log on to SBM Work Center.

   > **Tip:** In SBM Composer, you can click the Work Center button on the Launch tab of the Ribbon to do this. You can also navigate directly to `http://serverName/workcenter`.

2. Click the **MyApp** icon.

3. Click **+New**.

4. On the **Browse** tab, click the **MyAppProject** link.

   The submit form opens.

5. Complete the fields, entering text in the **Title** box and selecting values for **Item Type**, **Manager**, **Employee**, and **Customer Submitted**, and then click **OK**.

   The item moves to the **New** state, and the owner becomes the manager selected in the **Managers** field.

6. Click the **Assign** button.

   The **Assign** transition form opens, where you can update the fields, such as selecting a different employee.

7. Click **OK.**

   The item is assigned to the user selected in the **Employee** field. The user becomes the owner of the item as the item moves to the **Assigned** state.

8. Close the item by clicking the **Close** transition on the item and then clicking **OK**.

9. To locate the new item, click the **Search** icon and enter a keyword to search.

10. In the **Search Results** list, find the new item.

11. To access the item, click the item row.

The item has now completed the workflow. It has moved through three states. In the **New** and **Assigned** states, the item was assigned to the selected manager or employee.

Part 3: Applications

This section contains the following information:

Chapter 7: Application Overview

This section contains the following information:

- About Applications [page 91]

- Creating an Application [page 95]

- Application Settings [page 95]

About Applications

Applications comprise the elements needed to support processes used by teams of people. All applications contain a single primary table that stores process items and, optionally, one or more auxiliary tables used for storing auxiliary data that supports the process.

Applications also contain workflows that determine the flow of items through the process, fields that store data, forms for viewing and adding data, roles that control access, and more.

Application Names

Applications within a given deployment environment must be uniquely named, even if they are used in different process apps. Applications within a given process app must be uniquely named.

Applications created independently in different process apps can have the same name. Changes to one do not affect the other.

Comparison: SBM Composer and SBM Application Administrator

SBM has two components, SBM Composer and SBM Application Administrator, that you use to create and manage applications and orchestrations.

The following sections describe the tasks you perform in each component.

Project Management

Task	Tool
Creating and editing projects	SBM Application Administrator

Workflow, State, and Transition Management

Task	Tool
Creating a workflow (application or orchestration)	SBM Composer

Task	Tool
Adding states to a workflow	SBM Composer
Adding transitions to a workflow	SBM Composer
Adding, deleting, or modifying actions for a state or transition (including scripts, triggers, events, Web services, and transitions)	SBM Composer
Restricting transitions by type, role, or rule	SBM Composer
Restricting transitions by group	SBM Application Administrator
Creating forms	SBM Composer
Associating privileges with a form (using roles)	SBM Composer

Table and Field Management

Task	Tool
Adding a primary or auxiliary (non-system) table	SBM Composer
Adding system auxiliary tables (created when you create a database with Create Database wizard)	SBM System Administrator
Adding fields to a table	SBM Composer
Modifying field properties	SBM Composer
Modifying field overrides in a workflow, state, or transition	SBM Composer
Setting general field overrides	SBM Composer
Setting field overrides for specific projects or user fields	SBM Application Administrator
Adding roles to a user-type field	SBM Composer
Adding groups or users to a user-type field in primary table (via workflows)	SBM Application Administrator
Adding groups or users to a user-type field in an auxiliary table	SBM System Administrator

Task	Tool
Enabling data import for primary or auxiliary tables	SBM System Administrator
Importing data into primary or auxiliary tables from a spreadsheet	SBM Application Administrator
Importing data into primary or auxiliary tables using an ODBC connection	SBM System Administrator

User, Group, and Role Management

Task	Tool
Creating roles	SBM Composer
Creating users and groups	SBM Application Administrator
Assigning permissions to roles	SBM Composer
Assigning privileges to users	SBM Application Administrator
Assigning privileges to groups	SBM Application Administrator
Assigning users to groups	SBM Application Administrator
Assigning users and groups to roles	SBM Application Administrator
Modifying privileges for specific project or field section	SBM Application Administrator

Relationships Between Items

This topic describes terms and concepts that help you understand the relationships between items in an application.

Primary Items

An application uses its primary table to define a single type of primary item. The primary table determines the set of data fields the primary item contains. At runtime, when a primary item is created, it occupies a row in the primary table. This means that all items created in an application have the same set of data fields, but the value of any particular field will be specific to the particular item or row in the primary table.

All of the workflows in an application create and process the same type of items. However, each workflow can be designed to process the various data fields in the item in a different way.

Primary items are created by **Submit** transitions. These transitions can be initiated directly by a user, or indirectly from an existing item through a **Post**, **Subtask**, **Copy**, or **Publish** transition. After items are created, they can be connected through a link or a relational field reference.

Links

Links use a built-in attachment mechanism that is available in all workflows. Users can explicitly create links between existing items. Links can be created implicitly through **Post**, **Subtask**, **Copy**, or **Publish** transitions. With implicit link creation, a link can be created between the existing (original) item that is being transitioned and the new item.

Links can connect items in the following ways:

- The original item connects to the new item (one-way link).

- The new item connects to the original item (one-way link).

- The original item and the new item connect to each other (two-way link).

> **Note:** The Subtask transition implicitly creates a two-way link that appears in the **Subtasks** section of the item. You can optionally configure the **Subtask** transition to create item links in addition to the **Subtask** section links. These links appear in the **Attachments** section of the item.

Relational Field References

> **Note:** In this section, "relational field references" means relationships between items established through *Single Relational* and *Multi-Relational* fields. These fields can be in applications in the same process app or in applications in other process apps. (In the latter case, you must first create a reference to the application in the other process app. This is a different type of reference. For more information, see About References [page 369].)

Relational field references are often used to model parent-child relationships between items where the parent item stores a reference to one or more child items in a *Relational* field. This creates a one-way, one-to-one or one-to-many relationship.

You can also use relational field references to create multiple relationships, two-way relationships, and chains of relationships between items. An additional *Relational* field must be added to the primary table of the referencing item for each additional relational field reference. For example, to create a two-way relationship between items from two different applications, each primary table must define a *Relational* field that points to the primary table of the other application.

Transitions

Links, subtasks, and references allow items that have a relationship to be transitioned automatically when one of the related items is transitioned. There are three mechanisms for this:

- **Subtask transitions** work with Subtask attachment links.

- **Transition actions** allow an item to automatically transition another item that it references or that references it. Transition actions work with references.

- **Trigger actions** work across applications that may be aware of each other, but do not have to be. Trigger actions can be configured to work with subtasks, references, and links.

Creating an Application

Applications deployed to a given deployment environment must be uniquely named, even if they are used in different process apps. Applications within a given process app must be uniquely named.

Applications created independently in different process apps can have the same name. Changes to one do not affect the other.

Application Settings

The following sections provide descriptions of settings used to configure applications. Sections are organized by editors and dialog boxes.

Application Editor

Element	Description
Type	Indicates that you are working on an application.
Logical name	The name by which the application is known. Application names must be unique within a process app.
Version	The version of the application design element as checked in to the repository.
Internal name	The name by which the application is identified in the SBM database.
Prefix	A required prefix of up to 3 characters, constrained as described next to **Prefix**.
Tab name	A label of up to 16 characters that is displayed to users.
Options	Indicates that this application is accessible by the mobile app. If this check box is selected, additional validation steps are performed that indicate if parts of the application will function in the mobile app.

Element	Description
Logo	An application image that appears to users. In the area to the right, the image is displayed in three sizes. This lets you determine how to scale your image for the best results. By default, **(No image)** is selected in the list. To add a new image, select **(New image...)** in the list. An **Open** dialog box that contains numerous images opens. You can select one of these images or navigate to another image stored on the file system of your computer. You can use any HTML-compatible image format (for example, `.png`, `.gif`, `.jpg`, and `.bmp`). To use an existing image, select it in the list. **Note:** The new and existing images appear under the **Images** heading in App Explorer.
Description	Add comments that will appear to end users who hover over the Application name in certain areas of the SBM Work Center. HTML tags included in the comments are not rendered at runtime. For details, refer to Chapter 24: Providing Custom End-user Help [page 443].
History	The history of versions of the application from the repository.
Design number	A unique design number assigned to this application. If this application is in a process app that is used as a template, "template" is displayed. **Note:** For more information, see Managing Internal Identity and Design Numbers [page 62] and About Templates [page 42].

New Application Dialog Box

Use this dialog box to add a new application to a process app.

Element	Description
Name	Provide a name for the new application. Applications within a deployment environment must be uniquely named, even if they are used in different process apps. Applications within a process app must also be uniquely named. Applications created independently in different process apps can have the same name. In this case, changes to one application do not affect the other application.

Element	Description
Category	Select an existing category, or type the name of a new category.

Application Configuration Dialog Box

Use this dialog box to provide the required configuration information for a new application.

Element	Description
Application: Application name	The name you entered in the **New Application** dialog box.
Application: Internal name	The uppercase name under which the application is stored in the database.
Application: Tab name	The label (up to 16 characters) that appears to users. The name on this label must be unique.
Table: Table name	The name for the primary table (such as `Issues`). **Tip:** The name that you type in this field automatically populates the **Database table name** field, which must be unique within the SBM database and is limited to 24 Unicode characters. You should limit the logical name to 24 characters, as additional characters will be left off of the **Database table name**.
Table: Singular item name	The name of a single item in the table (such as `Issue`). **Note:** This field is automatically filled as you type the **Table name**, so you probably need to edit this field to use it as intended.
Table: Database table name	The uppercase name of the table in the database. This name must be unique. SBM Composer automatically fills this field as you type the **Table name**, ignoring characters after the first 24. **Note:** The database name can contain only ASCII alphanumeric characters (A–Z, a–z, 0–9). Any other characters are ignored as you type them.

Chapter 8: Managing Workflows

This section contains the following information:

About Application Workflows

This section contains the following topics:

Application Workflow Overview

An application workflow ensures the proper flow of primary items using a defined process that consists of fields, states, transitions, and decisions. Items must follow this application workflow from the time they are submitted to the time they are closed.

A sub-workflow is derived from an existing workflow. You could create a sub-workflow if you want to recreate the basic process defined in the existing workflow, but do things such as removing some states or changing the privileges on some transitions in the sub-workflow. Each workflow can contain multiple sub-workflows, and sub-workflows can contain their own sub-workflows. By default, when you make a change to a parent workflow, you get a warning that its sub-workflows will also be affected. You can disable this warning on the **Workflow Options** tab in SBM Composer SBM Composer Options [page 479].

You design workflows in SBM Composer. In SBM Application Administrator you assign the workflows to projects.

Elements of an Application Workflow

States: A state is a position in an application workflow where a primary item resides. While an item resides in a given state, it has a primary owner who is responsible for performing a specific task with the item before it can be transitioned to the next state. You can also set up an application workflow so that one or more users are secondarily responsible for items while they reside in a particular state.

Transitions: A transition activates the movement of a primary item from one state to another in the application workflow. A user who has ownership of an item transitions the

item to the next state in the workflow when the corresponding task is finished. Transitions can be customized in many ways. For example, you can restrict transitions so that they apply only to certain item types, such as bugs and problem reports, or so that only members of particular groups can execute them.

Fields: Fields let users provide information about the primary items they are submitting or transitioning from state to state. The fields store the information in a primary table, located in a central database. They can be tailored to prompt users for pertinent information as a primary item moves through the application workflow.

Forms: Forms determine the information that is presented to users as they work with items. View forms are used with states, while update forms are used with transitions.

The following figure shows an application workflow that defines the states and transitions an item must go through in order to reach a "closed" (completed) state.
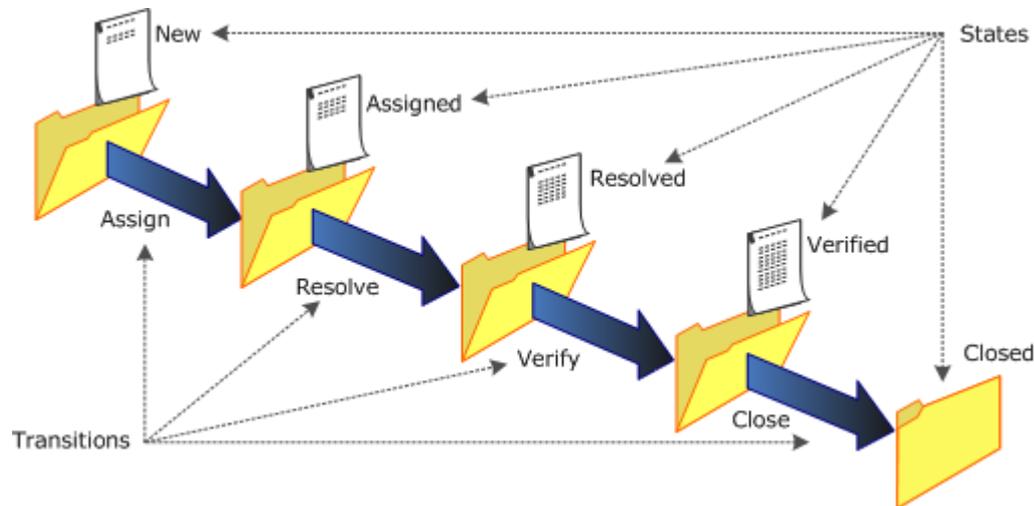
Other Elements: You can add the following other elements to an application workflow.

- Decisions are required for conditional routing. A decision has one incoming transition and two or more outgoing transitions. When an application workflow reaches a decision, the rule for each outgoing transition is evaluated, and the outgoing transition with the first rule that evaluates to "true" is executed.

- Swimlanes are a way to visually group states in an application workflow, so others can quickly understand the workflow design. A swimlane typically represents either the activities performed by the same role or participant, or a distinct phase of the process defined by the workflow.

- Annotations are notes that you can add to a workflow or sub-workflow.

- The Relationships bar provides visibility into design elements, such as forms, that an application workflow uses. This lets you easily understand and explore the relationship between the states and transitions in the workflow and the related design elements.

Internal Names

Workflows, states, and transitions have internal names that are automatically assigned when they are created or added. Because internal names are unique, they provide the ability to unambiguously identify a state or transition in scripts and Web service calls.

Because internal names cannot be changed after publication, you can safely use them to identify a state or transition without concern that a change to the name would break scripts or orchestrations.

The internal name is displayed on the **General** tab of the Property Editor for the workflow, state, or transition. For states and transitions, the internal name has two parts that are separated by a period:

1. The internal name of the workflow in which the state or transition is defined. This part can never be changed.

2. A local part that can be changed until the process app that contains the state or transition is published.

The internal name of a workflow must be unique within an application. For states and transitions:

- The local part of the internal name must be unique among the other states and transitions defined in the workflow. For example, if there are two transitions in a workflow called "Assign," their internal names could be "MyWorkflow.Approve" and "MyWorkflow.Approve1."

- If the two transitions were defined in different workflows, the local part can be the same because the internal workflow names are different, making the internal name as a whole unique. For example, "MyWorkflow.Approve" and "MySubWorkflow.Approve."

> **Note:** The workflow name is not prepended to the default **Update** and **Delete** transitions. Internal names are not used for system states (**Any**, **Deleted**, **Email**, **Submit**).

Application Workflows Heading

The **Application Workflows** heading in App Explorer displays all the application workflows and sub-workflows that you defined for the selected application. You can click the **Workflow Design** filter at the bottom of App Explorer to see only the **Application Workflows** and **Orchestration Workflows** headings.

About the Relationships Bar

The Relationships bar provides visibility into design elements, such as forms, that an application workflow uses. This lets you easily understand and explore the relationship between the states and transitions in the workflow and the related design elements.

When you select a design element in the Relationships bar, the transitions or states that use the design element are highlighted in the application workflow. Likewise, when you select a transition or state in the application workflow, the referenced design elements are highlighted in the Relationships bar.

For example, suppose you want to see which forms are used for states and transitions in the workflow. You can click the **Forms** accordion in the Relationships bar, and an icon for each form appears. If you select a form in the Relationships bar, the states and transitions that use the form are highlighted in the application workflow.

You can double-click an icon in the Relationships bar to open the editor and Property Editor for the design element and make changes. Quick forms are an exception to this, because they cannot be modified.

Key Benefits

- Visual cues instantly demonstrate the relationship between an application workflow and the design elements it references.

- You can open the editor and Property Editor for a design element directly from the Relationships bar, so you do not have to navigate through App Explorer.

About Swimlanes

Swimlanes are a way to visually group states in an application workflow, so others can quickly understand the workflow design. A swimlane typically represents either the activities performed by the same role or participant, or a distinct phase of the process defined by the workflow.

For example, you could put all states that are associated with software development tasks into one swimlane. Another swimlane could include states that are associated with testing tasks. The **Any**, **Deleted**, and **Email** states that have no owner could be moved into an "unassigned" swimlane below or to the right of the other swimlanes.

Note the following points about swimlanes:

- Swimlanes are displayed in the workflow as horizontal or vertical bands. You can toggle between a horizontal or vertical orientation.

- The left and right edges of horizontal swimlanes, and the top and bottom edges of vertical swimlanes automatically cover the entire workflow.

- You can tailor the appearance of swimlanes by modifying their colors, labels, boundary lines, and so on.

- A swimlane encloses the states that are in its boundaries when you add the swimlane to the workflow. You then drag states among swimlanes to depict the process in the best way.

- A sub-workflow inherits swimlanes from its parent workflow.

- Swimlanes are included in the workflow diagrams that users available to users.

Key Benefits

- Swimlanes are an efficient way to organize the activities or phases encompassed by an application workflow.

- Other designers and users can understand the workflow design at a glance.

- Numerous formatting options let you tailor the appearance of swimlanes to your needs.

About Annotations

Annotations are notes that you can add to a workflow or sub-workflow. For more information, see Working With Annotations [page 117] and General Tab of the Annotation Property Editor [page 128].

About Time Capture

Overview

The Time Capture feature enables users to record the amount of time they spend working on primary items. Time can be captured on state and transition forms. The Time Capture feature can be enabled or disabled at various levels (system, workflow, project, or for specific states and transitions).

For transition forms, time entries always apply to the current state of an item. For example, if an item is in an "Assigned" state, all time entries applied during a transition out of the state or during an update are attributed to the "Assigned" state. In other words, work is considered to have occurred while the item was assigned. You can also require users to record time spent for all transitions or for specific transitions.

For state forms, all states an item has resided in are available for users to capture time. This is so users can capture time for work they completed while the item resided in a particular state.

Entries can be in quarter-hour increments in digit format (4.25 or 4,25, for example, to represent 4 hours and 15 minutes), cannot exceed 30 days, and cannot be in the future. Dates are shown in the date/time format selected in each user's profile.

When users are viewing a state form, a time summary shows the total time captured for a particular item for all users. When users are working with a transition form, the summary shows entries for the item's current state.

Summary reports can be used to sum time capture entries for particular projects based on specific report criteria.

Time Capture changes are noted in the Change History section.

Enabling Time Capture Options

By default, Time Capture options are disabled, but they can be enabled for your entire system or at various levels. You can override the settings at each of these levels. This

allows users to record time spent on items only when the information is required or is meaningful for your process.

Time Capture options can be enabled or disabled at these levels:

- **System**

  In SBM Application Administrator, use Time Capture options in the Base Workflow to enable or disable this feature for transitions and states in all workflows and projects.

- **Application Workflows**

  In SBM Composer, enable or disable Time Capture options for all states and transitions in specific workflows.

- **Projects**

  In SBM Application Administrator, set or override Time Caption options for all states and transitions in some projects assigned to a workflow, but not others.

- **States and Transitions**

  Explicitly show or hide Time Capture options for individual states and transitions in workflows (SBM Composer) or projects (SBM Application Administrator).

You can also require users to enter the amount of time spent on an item for all transitions in your system, in specific application workflows, for all transitions in a project, or for individual transitions.

Time Capture Form Placement

If enabled for a workflow or project, Time Capture options are available on forms as follows:

- **Quick Forms**

  - For states, options appear after the Change History section.

  - For transitions, options appear at the top of the form.

- **Custom Forms Without the Time Capture Widget**

  If you do not place the Time Capture widget on custom forms in SBM Composer, Time Capture options are placed in the same locations as they are placed for quick forms.

- **Custom Forms With the Time Capture Widget**

  Use the Time Capture widget in SBM Composer to determine placement of Time Capture options on state and transition forms.

Working with Application Workflows

This section contains the following topics:

Creating a Workflow

You can create a new workflow in an application.

When you create a new application workflow, the only elements that exist in the new workflow are system fields; the **Submit**, **New**, **Email**, and **Deleted** states; the **Any** item; and the **Submit**, **Update**, and **Delete** transitions. You add states, transitions, decisions, and other objects from the **Workflow Palette** at the right side of the workflow editor. Any sub-workflows derived from the new workflow inherit the states, transitions, decisions, swimlanes, and annotations you added.

To create a workflow:

1. In App Explorer, right-click the **Application Workflows** heading, and select **Add New Workflow**.

2. Right-click the workflow you just added, select **Rename**, type the new name (32 characters or less), and press the Enter key.

Creating a Sub-workflow

You can add a new sub-workflow to an existing workflow. The new sub-workflow appears as a child of its parent workflow in App Explorer.

To create a sub-workflow:

1. In App Explorer, right-click the workflow where you want to add the sub-workflow and select **Add New Sub-workflow**.

2. Right-click the sub-workflow you just added, select **Rename**, type the new name, and press Enter.

In sub-workflows, transition lines and state borders are doubled when the transition or state is inherited from the parent workflow.

Opening a Workflow

You can open an individual workflow or view the list of existing workflows. The workflow list shows version number, creation date, and update date.

To view a list of all workflows:

- In App Explorer, do one of the following:

  ▪ Expand the **Application Workflows** heading.

  ▪ To open the list in a new tab, right-click the **Applications Workflows** heading, and select **Open in New Tab**.

To open a workflow or a sub-workflow:

- In App Explorer, do one of the following:

  ▪ Click the workflow or sub-workflow.

  ▪ Right-click the workflow or sub-workflow, and select **Open in New Tab**.

  ▪ Click the **Application Workflows** heading.

    In the workflow editor, double-click the workflow or sub-workflow that you want to open.

To open the parent workflow of the open sub-workflow:

- With the sub-workflow open in the workflow editor, right-click on the background and select **Open Parent Workflow**.

Duplicating a Workflow

You can duplicate any workflow or sub-workflow. The duplicate is added to the **Application Workflows** heading. A duplicate sub-workflow is added as a child of the same parent workflow.

Duplicated workflows inherit the states, transitions, and decisions in the originating workflow.

> **Note:**
>
> - When you duplicate a workflow that contains sub-workflows, only the parent workflow is duplicated.
>
> - The default name for a duplicated workflow is *workflow name* 2. The default name for a duplicated sub-workflow is *sub-workflow name* 2.

To duplicate a workflow:

- In App Explorer, right-click the workflow or sub-workflow, and select **Duplicate**.

  The duplicated workflow is added to App Explorer.

Deleting a Workflow

You can delete any workflow or sub-workflow from an application.

> **Note:** You cannot delete a workflow that contains sub-workflows. You must first delete its sub-workflows.

To delete a workflow:

- In App Explorer, right-click the workflow or sub-workflow, and select **Delete**.

Changing the Order of Workflows in App Explorer

In App Explorer, you can change the order of workflows and sub-workflows by moving them up and down the tree.

> **Note:** You cannot move a sub-workflow from one workflow to another.

To change the order of workflows in App Explorer:

1. In App Explorer, right-click the workflow or sub-workflow that you want to move.

2. Select **Move Up** or **Move Down**.

Working in the Workflow Editor

This section contains the following topics:

Using the Workflow Editor

The workflow editor is where you design and modify application workflows. In the workflow editor you can:

- Add and modify objects such as states, transitions, decisions, swimlanes, and annotations.

- Add and modify design elements, such as forms and actions.

- Override inherited field properties.

Use the following features in the workflow editor to help you work when you are designing your workflows:

- Zoom in to get a close-up view of your workflow, or zoom out to see more of it. The zoom range is between 10% and 500%.

- Resize the workflow to fit the workflow editor.

- Use the blue rectangle in the zoom preview (in the bottom right corner) to move around in the workflow.

- Use the About the Relationships Bar [page 101] to explore the relationship between design elements and the workflow.

To select a preset zoom level:

- In the **Zoom** area on the **Home** tab of the Ribbon, do one of the following:

  ▪ Click **Zoom**, and select a zoom level.

  ▪ Click **100%**.

To zoom in and out:

- Do one of the following:

  ▪ Below the zoom preview (in the bottom right corner), click the **+** and **−** buttons on the slider, or move the slider to the right and left.

    The current zoom level is displayed to the left of the slider.

  ▪ With the pointer over the workflow editor, press the Ctrl key while you move the mouse wheel backward and forward.

To fit the workflow to the window:

- In the **Zoom** area on the **Home** tab of the Ribbon, click **Zoom** and then select **Fit to Window**.

To move around in the workflow editor:

- Do one of the following:

    ▪ In the zoom preview (in the bottom right corner), drag the blue rectangle to the section of the workflow that you want to view in the workflow editor window.
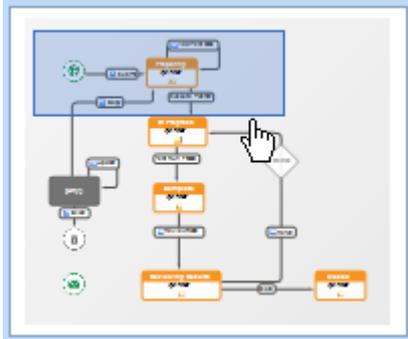
    As you drag the rectangle around in the thumbnail view, the section of the workflow that is visible in the editor changes. This is useful when you have zoomed in to a large workflow and want to move to another section without having to zoom out again. The blue rectangle indicates the portion of the workflow that is currently visible in the workflow editor—when you change the zoom setting, the rectangle resizes.

    ▪ In the workflow editor, with the pointer over the background (white space), press and hold the right mouse button while you drag the workflow graphic within the editor pane.

> **Note:** By default, SBM Composer remembers the zoom level and position of the workflow within the editor pane after you close the process app. To override this, see the "Workflow Tab" section in Application Options [page 486].

> **Tip:** If you click **Properties** in the **View Mode** area on the **Design** tab of the Ribbon, icons are displayed in the workflow editor that indicate certain properties (such as field overrides) that are set. If you click an icon, the applicable tab opens in the Property Editor.

Displaying Workflow Design Elements

You can toggle the display of certain workflow components. See Design Tab of the Ribbon [page 497] for details.

Using the Workflow Palette

In the workflow editor, you drag items from the **Workflow Palette** and arrange them in a process flow. You then use Property Editors to configure the items. This topic describes each section of the **Workflow Palette**.

Common Items

The following items are included in the **Common Items** section.

Item	Notes
Swimlane	See About Swimlanes [page 102] for information.
State	This item is identical to **Active** in the **States** section. See About States [page 131] for information.
Decision	Two outgoing transitions are automatically added to the decision: **Otherwise** and **Branch**. Both transitions are required. See About Decisions [page 342] for more information.
Transition	This item is identical to **Regular** in the **Transitions** section. See About Transitions [page 143] for information.
Annotation	See Working With Annotations [page 117] for information.

States

The following items are included in the **States** section. After you drag a state to the workflow editor, you can rename the state and change its properties. For more information about states, see About States [page 131].

Item	Notes
Active	A generic active state.
Completed	A preconfigured inactive state.
Rejected	A preconfigured inactive state.
Pending	A preconfigured inactive state.

Transitions

The following items, which represent transition types, are included in the **Transitions** section. For more information, see Transition Types [page 527].

- Regular
- Quick
- Post
- Subtask
- Publish
- Copy
- Update
- Delete

- External Post

After you drag a transition to the workflow editor, you can rename the transition and change its properties. For details, refer to Configuring Transitions [page 144].

Arranging States and Transitions

In the workflow editor, you can choose the following layout options for states and transitions:

- **Auto Arrange**: Automatically arranges the states and transitions.

- **Re-Inherit From Parent**: Update the layout to that of the parent workflow.

To arrange states and transitions:

Do one of the following:

- In the **Layout** area on the **Design** tab of the Ribbon, click **Auto Arrange** or **Re-inherit From Parent**.

- Right-click the workflow background, and select **Auto Arrange**.

- Right-click the sub-workflow background, and select **Auto Arrange** or **Re-Inherit From Parent**.

Adjusting a Transition

You can adjust a transition in the following ways:

- Move it (or part of it) in almost any direction.

- Add corners to route it around states and transitions.

- Move its starting or ending point to a different state.

    Note: A **Copy** or **Update** transition that starts on any state always ends on that same state.

- Use waypoints (small white circles on a selected transition) to bend a transition into an angle.

Moving and Resizing a State

In the workflow editor, you can move a state anywhere in the same workflow or sub-workflow. You can also resize a state manually or adjust it so that it automatically fits the title inside the box.

To move a state:

- Drag the state to its new location in the workflow.

To resize a state:

1. Select the state.

2. Select a point on the border of the state, and drag it to resize the state.

To resize a state to fit the title:

- Right-click the state, and then select **Size to Text**.

Transition Styles

You can use transition styles to provide a visual reminder of the purpose, use, or importance of a transition. Transition styles have no effect on transition behavior or form layout.
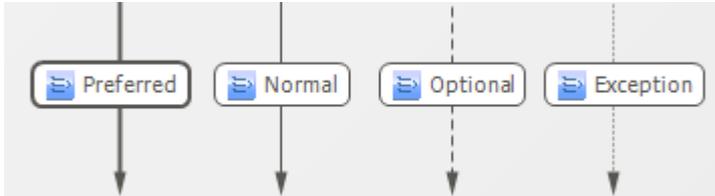
To modify the style of a transition:

- Do one of the following:

    ▪ Right-click the transition, select **Transition Style**, and then select the style.

    ▪ Select the transition, and then click the style in the **Transition Style** area of the Ribbon.

    Note:

    - You can change the default transition styles. To do so, click the **Styles** heading in App Explorer, and then make changes in the styles editor.

    - In sub-workflows, transition lines are doubled when the transition is inherited from the parent workflow.

Displaying Transition Labels

You can toggle the display of labels for all transitions. When labels are displayed, you can toggle the display of labels for individual transitions.

To toggle the display of all transition labels:

- In the **Show/Hide** area of the **Design** tab of the Ribbon, select or clear the **Labels** option.

To toggle the display of the label for an individual transition:

1. Right-click the transition.

2. On the menu that opens, select or clear **Show Label**.

Using the Relationships Bar

This topic contains procedures for using the About the Relationships Bar [page 101].

To show the Relationships bar:

1. Select the application workflow in App Explorer.

2. Click **Relationships** in the **View Mode** area of the **Design** tab of the Ribbon.

3. Click the **Relationships** tab at the bottom of the SBM Composer window.

To work with the Relationships bar:

1. Click the accordion for the relationships you want to see. The related design elements appear. For example, if you want to see the forms used by the workflow, click the **Forms** accordion.

2. To view the relationship between a design element and the workflow, do one of the following:

    - Click a design element in the Relationships bar. The states and transitions that use the design element are highlighted in the workflow.

    - Click a state or transition in the workflow. A green check box is displayed on the accordion or accordions with the related design elements. If one of these accordions is selected, the associated design elements are highlighted in the Relationships bar.

3. To open the editor and Property Editor so you can edit a design element, do one of the following:

    - Double-click the element in the Relationships bar.

    - Right-click the element in the Relationships bar and then select **Open *design element*** or **Open *design element* in New Tab**.

        Restriction: You cannot open or edit quick forms.

To preview a custom form or orchestration workflow:

- Do one of the following:

    ▪ Click **Relationships** in the Ribbon, click an accordion (such as **Forms** or **Orchestrations**), and then hover over the design element in the Relationships bar.

    ▪ Click **Presentation** in the Ribbon, and hover over the relationship hint on the state or transition in the workflow editor.

    A thumbnail view of the form or orchestration workflow opens.

    Note: In Presentation view, if multiple orchestration workflows are associated with a single state or transition, the thumbnail view includes arrows that let you navigate through the workflows.

To select the state or transition that uses a design element:

- Right-click the design element in the Relationships bar and then select **Select *design element* in Workflow**.

The state or transition that uses the design element is selected in the workflow.

To hide the Relationships bar:

- Click **Presentation** or **Properties** in the **View Mode** area on the **Design** tab of the Ribbon.

Using Swimlanes

This section contains the following topics:

Adding a Swimlane

You can add any number of swimlanes to an application workflow. The workflow editor expands as new swimlanes are added.

> **Important:** Decide whether you want horizonal or vertical swimlanes before you arrange states and transitions within them. States and transitions do not automatically adjust to the new orientation, so you will otherwise need to rearrange them manually. For information about changing swimlane orientation, see Changing Swimlane Orientation [page 116].

To add a swimlane:

1. Drag a **Swimlane** item from the **Common Items** section of the **Workflow Palette**.

   - If no other swimlanes exist in the workflow, a horizontal swimlane is placed at the top of the workflow or a vertical swimlane is placed at the left side of the workflow.

   - If you drop the new swimlane on the label of an existing swimlane, the new swimlane is placed before the existing swimlane. Otherwise, it is placed at the end.

2. Alternatively, do one of the following:

   - Right-click an empty area of the workflow, and then select **Add New Swimlane**.

   - Right-click the label of an existing swimlane, and then select **Add New Swimlane**.

Selecting a Swimlane

You need to select a swimlane before you can change its appearance, resize it, rename it, delete it, and so on. There are four ways to select a swimlane:

- Select the swimlane label.

- Select the swimlane in the drop-down list in the workflow or swimlane Property Editor.

- Select a swimlane, and then use the Tab key to select other swimlanes.

- Drag a rectangle around the swimlane label.

Renaming a Swimlane

You can rename swimlanes.

To rename a swimlane:

1. Click the swimlane label.

2. Right-click and select **Rename.**

3. Type the new name in the box that opens.

Deleting a Swimlane

When you delete a swimlane, the steps and transitions within it are moved to the swimlane that is above or to the left of it.

To delete a swimlane:

1. Click the label of a swimlane.

2. Right-click and select **Delete**.

Moving States Between Swimlanes

You can drag a state from one swimlane to another. States and transitions can also be placed on the line between two swimlanes, or in an area of the workflow not enclosed by a swimlane.

Changing Swimlane Styles

There is a default swimlane style in the styles editor. To view the style settings, click the **Styles** heading in App Explorer, and then select **Swimlane Style**. You can modify the default style in the styles editor, and add additional styles. For more information, see Customizing Styles [page 438].

The styles in the styles editor are included as items in the **Style** area on the **Appearance** tab of the Ribbon. When you select a style, its settings are applied to all swimlanes in the workflow.

> **Important:** When you add a new swimlane, the background color and line settings you specified in the style editor are overridden. To apply these settings to new swimlanes, select the swimlane, and then select the desired style on the **Style** area on the **Appearance** tab.

The styles you modify or create in the styles editor initially affect all swimlanes in a workflow. You change the style of one or more swimlanes in a workflow while maintaining the default style for the other swimlanes.

To change the default style of a swimlane:

1. Click the swimlane label to select it.

2. Click the **Appearance** tab in the Ribbon.

3. To change the swimlane border, in the **Line** area of the Ribbon, select a style and width.

4. To change the text formatting in the swimlane label, in the **Font** area of the Ribbon, select the options you want to change.

5. If you want add or change the image displayed in the background of the image, do the following in the **Background** area of the Ribbon:

    a. Select or add a new image.

    b. Specify whether you want the image to repeat vertically, horizontally, or both.

    > **Note:** If you want to remove an image, select **(No image)**.

6. To change the alignment of the label text, select an icon in the **Alignment** area of the Ribbon.

Changing Swimlane Orientation

Swimlanes can run horizontally or vertically in an application workflow.

> **Important:** Decide whether you want horizonal or vertical swimlanes before you arrange states and transitions within them. States and transitions do not automatically adjust to the new orientation, so you will otherwise need to rearrange them manually.

To change swimlane orientation:

1. Click the **Design** tab of the Ribbon.

2. In the **Swimlanes** area, click **Horizontal** or **Vertical**.

Changing Swimlane Order

You can change the order of swimlanes in an application workflow.

To change swimlane order:

1. Click the label of a swimlane.

2. Do one of the following:

    -   Right-click and then select **Move Up** or **Move Down**.

    -   On your keyboard, press the Ctrl key and then click the up arrow or down arrow key.

    -   Drag the swimlane. If you drop it on the label of a horizontal swimlane, it is placed above it. If you drop it on the label of a vertical swimlane, it is placed to the left of it. Otherwise, it is moved to the bottom or right side of the workflow.

The swimlanes and the states and transitions within them are moved as you reorder them.

Resizing Swimlanes

Swimlanes can be resized. The workflow editor expands or contracts as needed to accommodate the new sizes.

The way you resize a swimlane determines whether the states and transitions in the swimlane move to an adjacent swimlane if necessary, or remain in place. The following table describes ways to resize a swimlane to achieve either result.

Result/Methods
Content of swimlane can move to an adjacent swimlane after the swimlane is resized. **Method 1:** 1. Select the swimlane you want to resize. 2. Press Shift+Up to decrease the size, or Shift+Down to increase the size. **Method 2:** 1. Position the mouse pointer over the line separating the swimlanes. 2. Drag the line to adjust the size of the swimlane.
Content of swimlane does not move to an adjacent swimlane after the swimlane is resized. **Method 1:** 1. Right-click and position the mouse pointer over the line separating the swimlanes. 2. Drag the line to adjust the size of the swimlane. **Method 2:** 1. Press Ctrl and position the mouse pointer over the line separating the swimlanes. 2. Drag the line to adjust the size of the swimlane.

Working With Annotations

- Adding Annotations [page 117]

- Customizing Annotations [page 118]

- Finding Annotations [page 119]

- Deleting Annotations [page 119]

Adding Annotations

You can add as many annotations as you need and place them anywhere in an application workflow or sub-workflow.

To add an annotation to a state:

- Do one of the following:

  - Drag **Annotation** from the **Common Items** section of the Using the Workflow Palette [page 109], and drop it on the target state, which becomes highlighted.

  - Right-click the state, and select **Add Annotation**.

To add an annotation to a transition:

- Do one of the following:

  - Drag **Annotation** from the **Common Items** section of the Workflow Palette, and drop it on the target transition, which becomes highlighted.

  - Right-click the transition, and select **Add Annotation**.

To add an annotation to a workflow:

- Do one of the following:

  - Drag **Annotation** from the **Common Items** section of the **Workflow Palette**, and drop it on the workflow.

  - Right-click the workflow background, and select **Add Annotation**.

    Note: After you add one annotation to a workflow, additional annotations are placed directly on top of the first annotation. Drag the additional annotations to separate them from the first one.

Customizing Annotations

To move an annotation:

- Drag the annotation to another location.

To resize an annotation:

- Drag any of the points on its border.

To add a connector line:

1. Select the annotation.

2. In the General Tab of the Annotation Property Editor [page 128], select **Draw connector line**.

To apply a system annotation style:

1. Select the **Styles** heading in App Explorer.

2. In the styles editor, under **Annotation Styles**, select a defined style.

To modify the background color:

1. Select the annotation.

2. In the **Background** area of the styles editor, click the paint bucket icon.

3. Do one of the following:

- Select one of the "theme colors" or "standard colors."

- Select the default color.

- Select **Transparent**.

- Click **More Colors**, select a color from the **Standard** or **Custom** tab, and click **OK**.

To add or remove a background image:

1. Select the annotation.

2. In the **Background** area of the styles editor, select one of the following options:
   - **(No image)**: Remove the current background image from the annotation.

   - **New image**: Locate an image, and click **Open** to make it available for selection.

   - *Image filename*: Select a background image you have already "imported."

To modify the border style and width:

1. Select the annotation.

2. In the **Line** area of the styles editor, select the border style and width.

To edit the text:

1. Do one of the following:
   - Double-click the annotation.

   - Right-click the annotation, and select **Edit Text**.

2. Edit the text, and then press the Enter key to record your changes.

To modify the text styling:

1. Select the annotation.

2. In the **Font** area of styles editor, select the font name, size, and styling.

Finding Annotations

To search for an annotation:

1. Click inside the workflow editor.

2. Press **CTRL** + **F**.

3. Search for the annotation.

Deleting Annotations

You can delete any annotation from a workflow or sub-workflow.

To delete an annotation:

- Do one of the following:
  - Right-click the annotation, and select **Delete**.

▪ Select the annotation, and press the Delete key.

Workflow Settings

The following sections provide descriptions of settings used to configure applications. Sections are organized by property editors and dialog boxes.

General Tab of the Application Workflow Property Editor

Use this tab to modify the general properties of the selected application workflow or sub-workflow.

Element	Description
Name	Modify the name of the workflow or sub-workflow.
Internal name	View or modify the internal name of the workflow or sub-workflow. This field is read-only if the process app has been published. The default value of this field is the workflow name, in upper case. The internal name cannot contain spaces or periods. The maximum number of characters is the same as the number for the **Name** field.
Time Capture	Specify whether the Time Capture feature is enabled or disabled for states and transitions in the workflow. For details about the Time Capture feature, see About Time Capture [page 103]. Options for root application workflows are: • **System** Select to use the setting applied for the Base Workflow in SBM Application Administrator after deployment. • **On** Select to enable the Time Capture feature for all states and transitions in the workflow you are editing. You can then hide or show Time Capture options for states and transitions as needed. • **Off** Select to disable the Time Capture feature for all states and transitions in the workflow. For sub-workflows, settings are inherited but you can override them by explicitly turning the feature on or off for the workflow. If you turn the feature on for a workflow, you can then set other Time Capture options for states and transitions in the workflow. If you turn the feature off for the workflow, it is also turned off for states and transitions in the workflow.

Element	Description
Entry Required	With Time Capture options set to "on" for the workflow and "visible" for transitions, you can choose to require users to enter time spent for all transitions in the workflow. This requirement is ignored for automated processes, such as Web services and scripts. Options are: • **System** For the root application workflow, uses the setting applied for the Base Workflow in SBM Application Administrator after deployment. • **Inherited** For sub-workflows, uses the setting inherited from the parent workflow. • **Yes** Requires users to enter Time Capture entries for all transitions in the workflow. • **No** Time Capture options are enabled, but not required for all transitions.
States/ Transitions	• **System** For root application workflows, uses the setting applied for the Base Workflow in SBM Application Administrator after deployment. • **Inherited** For sub-workflows, uses the setting inherited from the parent workflow. • **Visible** Shows Time Capture options for all states or transitions in the workflow. • **Hidden** Hides Time Capture options for all states or transitions in the workflow.
Uses table	Identifies the primary table for the application.

Element	Description
End-user help text	Click **Edit** to open an HTML editor and add optional comments or notes about the purpose of the workflow. The text you enter and a graphical depiction of the workflow is displayed to users who click the workflow icon (⊞) on forms. For details, refer to Chapter 24: Providing Custom End-user Help [page 443].

Forms Tab of the Application Workflow Property Editor

When an application workflow or sub-workflow is selected in the editor, you use this tab to manage its default state and transition forms.

Element	Description
Default state form, Default transition form	Select the form to be associated with new states or transitions added to this workflow. You can select one of the following items: • [inherit from primary table] The default form that was specified for the primary table on the Forms tab of the table Property Editor. This option is available only for application workflows. • [quick form] A form that was generated by the system. • A custom form that was created. You can change the form associated with an existing state or transition at any time.
Default print form	Select the form to be used when a user prints a state form. You can select one of the following items: • [inherit from primary table] The default form that was specified for the primary table on the Forms tab of the table Property Editor. This option is available only for application workflows. • [quick form] A form that was generated by the system. • A custom form that was created.

Element	Description
Default anonymous submit form	Select the form to be used when a non-SBM user submits an item. You can select one of the following items: • [inherit from primary table] The default form that was specified for the primary table on the Forms tab of the table Property Editor. This option is available only for application workflows. • [quick form] A form that was generated by the system. • A custom form that was created. For both quick forms and custom forms, anonymous users can view fields only in the Standard privilege section. They can add file and URL attachments and notes to items while they are submitting them. After they submit an item, however, they can no longer view it or any other part of SBM. Use Application Administrator to enable anonymous submit for a project.
New	You can base a new custom form on the quick form for the workflow or on an existing custom form, or you can start with an empty form. (You can also create a state or transition form by right-clicking the **Forms** heading in App Explorer.)
Preview	View a mockup of the corresponding form as it will appear to a user. You can use the controls at the top to view the form as it will be seen in a different workflow, state, or transition; by users assigned to different roles; and by style (classic or Work Center, for example). The controls in the preview area have no effect.
Edit	If a custom form is selected for the corresponding default form, click to open the form in the form editor.

Field Privileges Tab of the Application Workflow, State, and Transition Property Editor

Use this tab to modify the field privileges of an application workflow, state, or transition.

Element	Description
Override field privileges	Select this option if you want to make changes to field privileges.

Element	Description
List of fields	The fields in this list are divided into sections, such as **Standard** and **Advanced**. The initial field placement is based on the privilege section specified in the field Property Editor.
	To change the field privileges in each section, do one of the following:
	• Drag and drop fields between sections. You can only drop a field into a section that is allowed for the field. For example, you cannot move a required system field into the Not Used section.
	• Select one or more fields, right-click, select **Move To**, and then select the target section. Only those sections that are allowed for all of the selected fields are available.
	To change the field ordering within a section, do one of the following:
	• Drag and drop the fields.
	• Right-click any of the fields, point to **Sort**, and then select **Ascending** or **Descending**.
	• Right-click on any of the fields, and then select **Move Up** or **Move Down**.
	• Select a field and then press CTRL+Up or CTRL+Down.
	To find a field in the list, right-click in the list and then select **Find**.
Fields in the *section-name* Section	When you select a field in the **Override field privileges** list, this section displays the privileges for that field. For example, this section could indicate that fields in the **Advanced** section can be changed by the Administrator and Manager roles, can be read by the Developer role, and are not visible to the User role.
	The Sections privileges specified in the Roles editor determines what is displayed. For example, suppose **View Fields in the 'User' Section (View User Fields)** is not selected for the User role in the Roles editor. A user with the User role will not be able to see any fields in the User section on the form.

Field Overrides Tab of the Application Workflow and Transition Property Editor

Use this tab to override the properties of the fields for a workflow or transition (except a **Delete** transition).

> **Note:** For information about overrides, see Chapter 18: Understanding Inheritance and Overrides [page 377].

Element	Description
List of fields	Select a field whose properties you want to override. (Fields with overrides are displayed in boldface type.)
	Note: You can sort the fields by clicking the **Arranged by** menu at the top of the field list and selecting a sort option. You can find a field in the list by right-clicking in the list and then selecting **Find**.
	Tip: To sort the list into two sections, showing fields that are overridden and those that are not, select the **Overridden** sort option.
Override field properties for *field-name*	Select this option for each field type whose properties you want to modify.
	For *User*, *Multi-User*, and *Multi-Group* fields, select this check box to allow default values to be set for the workflow in SBM Application Administrator.
	After this check box is selected, the field name is displayed in bold type.

Attributes

Element	Description
Required	Makes the field a required attribute. This option is disabled for *Binary/Trinary* fields.
Read only	Prevents the field from being modified.

Element	Description
Allow mass update	Lets users simultaneously transition, update, or delete multiple primary items and to simultaneously update or delete multiple auxiliary items. For example, if several primary items have been "Deferred" from the system, you can activate all the items at the same time by performing a mass "Restart" transition. This activates all the items at the same time and places them in the same state. This option provides the flexibility of determining the specific fields that can be modified when users perform a mass update. For example, if the **Additional Notes** field is selected for mass update, users can enter a value in the field before completing a mass update. This value overwrites any previous values stored in the field for items that are mass updated. If a value is not provided for the field during the mass update, the original values remain unchanged.
Require appended text (*Journal* fields)	Requires users to append text to the *Journal* field during every transition (if set at the workflow level) or a specific transition (if set at the transition level). This option is available when the check box is selected.
Append only (*Journal* fields)	Forces the text in any new journal entries to be appended to the text in an existing journal entry, preventing users from modifying existing journal entries. Clear this option to let users edit existing journal entries.

Style options

Element	Description
Allow searching	Select this option to enable "Value Find" on submit, transition, and update forms for the field. Value Find lets users search for values.
Single drop-down list	Select this option to let the user select a value from a populated drop-down list.

Value options *(Transitions)*

Element	Description
Leave unchanged	Select this option to leave the value in the field unchanged during a transition.
Clear	Select this option to clear the value in the field during a transition.

Element	Description
Set to default	Select this option to set a default value for a transition field. **Note:** For *Text* fields with the **Journal** display option enabled, this setting applies only to appended text.
Default value	Type or select a default value. You cannot change the default value for fields that are automatically populated by the system or that need to be set in SBM Application Administrator. For guidance on setting default values, refer to Setting Default Values for Fields [page 218].
Set to calculation	This option is only available for *Date/Time* and *Numeric* fields. This option allows you to calculate a value for these field types. For example, you can calculate an amount owed based on a rate and the actual time to resolve. The calculation can be made read-only and placed in the **Hidden Fields** section. Selecting this option displays several additional controls that you use to set up date calculations. See Calculating Values for Date/Time and Numeric Fields [page 153] for a description of your options for populating this field.

Value options *(Workflows)*

Default value	Type or select a default value. You cannot change the default value for fields that are automatically populated by the system or that need to be set in SBM Application Administrator. For guidance on setting default values, refer to Setting Default Values for Fields [page 218].
Selection Value Overrides	This section is available for *Single Selection* and *Multi-Selection* fields, and can be changed even if the **Override field values for *field-name*** check box is not selected. To change the status for a selection value, click the **Status** column in the applicable row and then select **Enabled**, **Disabled**, **Inherited**, or **None**. After a status is changed, the field name is displayed in italic type.

Dependencies Tab of the Application Workflow Property Editor

Use this tab to define dependencies for the workflow or sub-workflow in the editor.

Element	Description
Select independent field	Select the field for which you want to define a dependency.
Restrict dependent field…	When a user selects a value in the specified independent field, the dependent field you specify here is automatically limited to the values you specify here. In the **To these values** column, you can use Ctrl+click, Shift+click, and Ctrl+A to highlight multiple values, then click any one of the highlighted check boxes to select all the highlighted values. Click **Check All** or **Uncheck All** to select or clear all values.
Change value to	Define a specific dependent field default value for each independent field value, based on the following table.

The following table describes how the different options affect the dependent field when the independent field is changed.

Option	Behavior when the independent field value is changed
<Leave entry unchanged>	The dependent field selection will not change (assuming the previously selected value of the dependent field is also valid for the new value of the independent field).
<First valid entry>	The dependent field will be preset to the item listed immediately after **<None>**.
<None>	The dependent field will be preset to **<None>**.
list of values	The dependent field will be preset to the *value* you select here. This option only applies to *Single Selection* fields that are set as dependent fields.

General Tab of the Annotation Property Editor

Use this tab to toggle display of the arrow pointing to the state that was selected when the annotation was added.

If no state was selected when the annotation was added, the arrow points to the nearest corner of the workflow background.

> **Note:** If you move the annotation, the arrow shifts so that it always points to the nearest corner of the state or workflow background.

Element	Description
Draw connector line	Toggle display of the arrow pointing to the associated state, if any, or to the nearest corner of the workflow background.

Chapter 9: Managing States

This section contains the following information:

About States

A state is a position in an application workflow where a primary item resides. A state provides accountability as items are transitioned through the workflow process. While an item is in a given state, it has a primary owner who is responsible for performing a specific task with the item before it can be transitioned to the next state. You can also set up an application workflow so that one or more users have secondary responsibility for items while they are in a particular state.

States can be *active* or *inactive*. In an active state, the owner of the state is to some degree engaged in a task. In an inactive state, activity on the item is suspended. The Using the Workflow Palette [page 109] includes one active state and three preconfigured inactive states. All states can be renamed and reconfigured.

> **Note:** There are four system-provided states included in every workflow that serve special purposes. For more information, see System-Provided States [page 133].

Ownership of Items

SBM provides two types of ownership: primary ownership and secondary ownership. Primary ownership means that a single user is responsible for an item while it resides in a particular state. Secondary ownership lets multiple users own an item in a state. With appropriate privileges, secondary owners can view, update, and transition items as needed.

Each state may have a primary owner, a primary owner and secondary owners, or just secondary owners.

Considerations For Setting Primary Ownership Of Items

You define primary ownership for each state by selecting a single *User* field, such as *Manager*, *Engineer*, or *Tester*, in the Owner drop-down list on the General tab for each state. The selections for the *User* field determine who is available as a primary owner. You can specify role selections for the field in SBM Composer. User and group selections for *User* fields are made in SBM Application Administrator.

The selections from that *User* field are available to users, but they can only select a single user as a primary owner for each state in which the item resides.

Consider the following when you set primary ownership of items:

- When you configure a workflow, define single *User* fields before assigning the Owner property for each state. Then, in each transition that moves items into the state, move the *User* field that populates the Owner property into the Standard section so that it is available to all users who can transition the item.

- In some workflows and projects, you may want to pre-select a specific owner for items in each state and hide the *Owner* field so that the system alone manages ownership. For example, the manager of a specific project needs to be the owner of items in the "Submit" and "Assigned" states. You can modify the project and select a default value for the *Owner* field, and then move the field to the Hidden section in the "Submit" and "Assigned" states. The *Owner* field automatically contains the value. Therefore, ownership does not need to be selected by the user.

- You can set the *Owner* field as read-only if you want to keep the field visible. This lets you explicitly see who the owner of the item is and prevents users from changing the owner.

- You can select "None" as the ownership value for a state. While this is an appropriate value in some cases, such as closed items, keep in mind that an item without an owner might be forgotten.

Considerations for Secondary Ownership

You can set up a workflow so that one or more users are secondarily responsible for items in a particular state. For example, you can set up a queue into which all new incidents are submitted and let users who are specified as secondary owners for new incidents transition items from the queue into their inbox. Or, you can designate a single user as a secondary owner for a particular state, letting this user transition items from that state instead of the primary owner.

A *User*, *Multi-User*, or *Multi-Group* field can be used to populate the secondary ownership property. The selections for the *User*, *Multi-User*, or *Multi-Group* field determine which users and groups are available as secondary owners. You can also specify whether users can select individual users or groups as secondary owners.

Consider the following when you set secondary ownership of items:

- The *Secondary Owner* field is an optional system field that must be added to a primary table before you can configure secondary ownership.

- When you configure a workflow, define *User*, *Multi-User*, and *Multi-Group* fields before assigning the "Secondary Owner" property for each state. Then, in each transition that moves items into the state, make sure the appropriate *User*, *Multi-User*, and *Multi-Group* field is moved so that it appears in the Standard section and is available to all users.

- You can set up your workflow so that items in a particular state have a primary owner and secondary owners, or you can specify that items in a state have only secondary owners. For example, a manager may want to assign items to a group of several users, who can then decide who should take ownership of the items. Each member of this group can be specified as secondary owners since the group as a whole has responsibility for items in that state. When items are assigned to the group, each user in the group may see the item on his or her home page and may receive e-mail notifications about the items.

- You can also use a *Multi-Group* field to define secondary ownership, letting members of multiple groups set selections for the field to transition items as needed.

- Use the **Selection Mode** option to control the list of possible user and group values. Groups are available in the *Secondary Owner* field if the source field that determines secondary ownership is a *Multi-User* field and you select either **Groups & Users** or **Group Members & Users**.

   For best performance, select the **Groups & users** option. Also, be aware that if you change the selection mode for a *Multi-User* or *Secondary Owner* field back to individual users or group members after populating groups in either field, performance may be impacted when the groups are unrolled to display large lists of individual users.

- You can select None as a secondary owner value for a state.

If you configured your workflow so that only secondary ownership is enabled for a state, the primary owner value displays as "None" in the state change history, change history, and the *Owner* field.

System-Provided States

All workflows contain four unique "states" that are not available from the **Workflow Palette**, but that can be viewed in the workflow editor. You cannot add these states to or delete them from the workflow, and their properties cannot be changed.
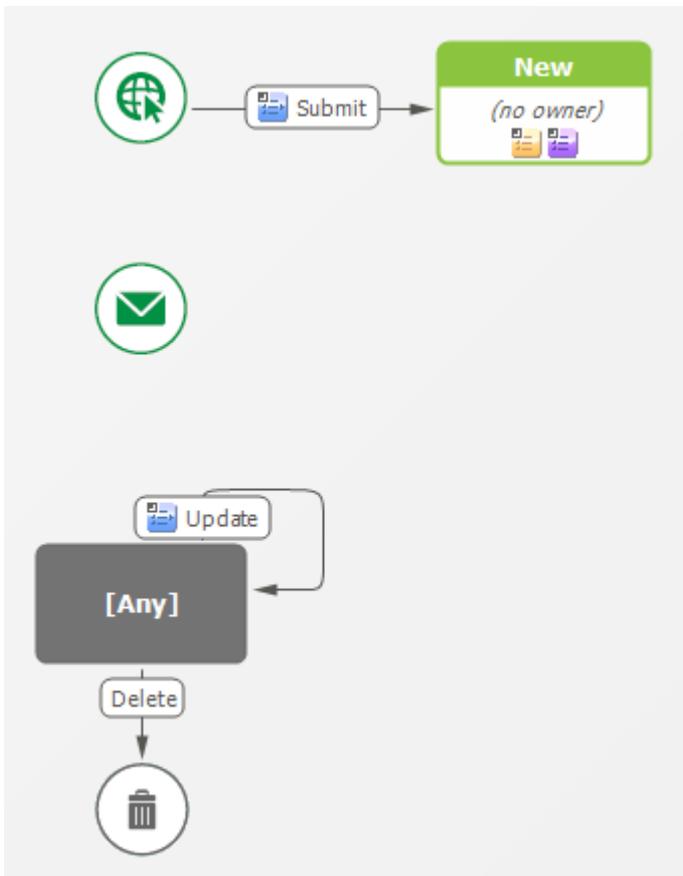
- **Submit** is represented by a globe with a mouse pointer. This is one of the starting points in the workflow and corresponds to the user submitting an item.

- **Email** is represented by a circle with an envelope. This is the other starting point in the workflow and corresponds to the user submitting items through e-mail.

- **Any** represents every other state in the workflow in that it allows you to create one or more transitions that apply to every state in the workflow. By default, every state in the workflow has an **Update** transition button, which allows users with appropriate privileges to edit fields in the item without leaving the current state; and a **Delete** transition button, which allows users with appropriate privileges to delete an item from any state. You can add other transitions from the **Any** state; for example, a **Defer** transition that moves an item in any state to a **Deferred** state.

  You cannot have an outgoing transition from a regular state to the **Any** state. You can, however, have an outgoing transition from a decision to the **Any** state, if the only incoming transition to the decision is from the **Any** state.

  The **Any** state is provided so you do not have to create separate transitions from every state in the workflow. For example, if the **Any** state did not exist, and you wanted to have a **Delete** transition button in every state form, you would have to add outgoing **Delete** transitions from every state to the **Deleted** state.

  Normally, transitions on the **Any** state appear on the state form for all states. However, you can exclude these transitions for particular states. For example, suppose you have a **Delay** transition on the **Any** state. It makes sense to show this transition button on all states but the **Delayed** and **Closed** states. For information about how to do this, see Transitions Tab of the State Property Editor [page 140].

- **Deleted** is represented by a trash can. It serves as the target state for **Delete** transition types. Items sent to this state are permanently deleted from the database.

Working with States

This section contains the following topics:

Adding a State

You can add states to an application workflow or sub-workflow. Primary items in your system are either active or inactive when they reside in each state.

To add a state:

1. To add an active state, do one of the following:

   - Drag **State** from the **Common Items** section of the **Workflow Palette**.

- Drag **Active** from the **States** section of the **Workflow Palette**.

> **Note:** These two objects are identical.

2. To add an inactive state, drag **Completed**, **Rejected**, or **Pending** from the **States** section of the **Workflow Palette**.

> **Note:** These three objects are identical, except for their names and properties, which can be changed.

3. Type a name for the state, and press the Enter key.

4. To reposition the state, drag it to a new location in the editor.

> **Note:** You can also double-click the state in the **Workflow Palette** to add the state to the workflow. The state is added to the right of a selected state, on the right side of a selected transition, or to the center of the workflow editor if no state or transition is selected.

> **Tip:** A state that is added to a swimlane takes on the swimlane color. A state that is not added to a swimlane takes on the color that corresponds to the owner of the state.

Adding an Owner Field to a State

Use the **Add Owner Field to State** wizard to quickly establish ownership for items in a state by assigning a user field to the **Owner** property of the state. The role or roles associated with this field identify which users can own items in this state. There is one *Owner* system field. The value of this field changes as the item moves through the workflow, and its value is taken from the field you specify in the **Owner** property of the state.

You first identify the roles, and then either select or create a user type field that is constrained by those roles to use as the owner for the state. For example, the owner of a **Researching** state could be a *Research* field with an associated **Developer** role. Only users with the **Developer** role can be assigned to the field and can become owners of items in this state.

> **Note:** For information about states, see About States [page 131]. For information about adding a secondary owner to a state, see Adding a Secondary Owner to a State [page 136]. For information about user type fields, see User Field Types [page 178].

To add an owner to a state:

1. Select a regular state in an application workflow.

2. On the **General** tab of the state Property Editor, click the drop-down list next to the **Owner** field, and select **<Add Owner>**. Alternately, you can right-click the state in the application workflow and then select **Add Owner**.

> **Note:** The drop-down list next to the **Owner** field in the state Property Editor contains all valid potential owners for the state. If you want to use an existing field, you can select it from this drop-down list.

3. Follow the instructions on the wizard screens.

Select or Add Roles Screen

On this screen, specify the role or roles that define the users who can own items in the selected state. You can select existing roles or add a new role. If you are adding a new role, you can select an existing role to use as a template. The new role has the same privilege settings as the template role.

Field Selection Screen

On this screen, create a new user field or select an existing user field.

If you are creating a new field, specify its name and database name.

If you are selecting an existing field that corresponds to the selected roles, note that the number of available fields in the list is shown in the title bar of the screen. For example, **2 matching field matching field(s)** is shown if there are two fields in the drop-down list.

Adding a Secondary Owner to a State

Use the **Add Secondary Owner to State** wizard to quickly establish secondary ownership for items in a state by assigning a *User*, *Multi-User*, or *Multi-Group* field to the **Secondary Owner** property of the state. The role or roles associated with this field identify which users can be secondary owners of items in this state. There is one Secondary Owner system field. The value of this field changes as the item moves through the workflow, and its value is taken from the field you specify in the **Secondary Owner** property of the state.

You first identify the roles, and then either select or create a user type field that is constrained by those roles to use as the secondary owner for the state. For example, you can set up a queue into which all new incidents are submitted. Users who are specified as secondary owners for new incidents can transition items from this queue into their inboxes. Alternately, you could designate a single user as a secondary owner for a state, allowing this user to transition items from that state instead of the primary owner.

> **Note:** For information about states, see About States [page 131]. For information about adding an owner to a state, see Adding an Owner Field to a State [page 135]. For information about user type fields, see User Field Types [page 178].

To add an secondary owner to a state:

1. Make sure the *Secondary Owner* field has been added to the primary table. The *Secondary Owner* field is an optional system field that must be added to a primary table before you can configure secondary ownership.

2. Select a regular state in an application workflow.

3. On the **General** tab in the state Property Editor, click the drop-down list next to the **Secondary owner** field, and select **<Add Secondary Owner>**.

    > **Note:** The drop-down list next to the **Secondary Owner** field in the state Property Editor contains all valid potential owners for the state. To use an existing field, you can select it from this drop-down list.

4. Follow the instructions on the wizard screens.

Select or Add Roles Screen

On this screen, specify the role or roles that define the users who can be secondary owners of items in the selected state. You can select existing roles or add a new role. If you are adding a new role, you can select an existing role to use as a template. The new role has the same privilege settings as the template role.

Field Selection Screen

On this screen, create a new user field or select an existing user field.

If you are creating a new field, specify its name, database name, and field type. The field type can be *User*, *Multi-User*, or *Multi-Group*.

If you are selecting an existing field that corresponds to the selected roles, note that the number of available fields in the list is shown in the title bar of the screen. For example, **2 matching field(s)** is shown if there are two fields in the drop-down list.

Renaming a State

To rename a state:

- Right-click the state, select **Rename**, type the new name, and press the Enter key.

Duplicating a State

You can duplicate any state in an application workflow or sub-workflow. The default name for a duplicated state is *state name* 2.

To duplicate a state:

- In the workflow editor, right-click the state, and select **Duplicate**.

  A duplicate (copy) of the state is created in the workflow.

  Note: The duplicate is unreachable until you connect it with a transition *from* another state. Use the **Unreachable Paths** option in the **Show/Hide** area of the **Design** tab of the Ribbon to change the appearance of unreachable states and transitions.

Deleting a State

You can delete most states from a workflow or sub-workflow.

Note: The **Submit**, **Email**, **Any**, and **Deleted** states cannot be deleted, even if they were renamed.

To delete a state:

- Do one of the following:
  - Right-click one or more states, and then select **Delete**.
  - Select one or more states, and then press the Delete key.

  Note: Transitions that originate and end in the deleted states are also deleted.

State Settings

The following sections provide descriptions of settings used to configure states. Sections are organized by property editors.

General Tab of the State Property Editor

Element	Description
Type	*(Display only)* The state type.
Name	The name of the state. For clarity, state names should be unique within a workflow as it appears to a particular role; but your design could include multiple states with the same name that appear for users in different roles. **Note:** You cannot rename system states: (**Any**, **Deleted**, **Email**, **Submit**).
Internal name	View or modify the internal name of the state. This field is read-only if the process app has been published. The default value of this field is the name of the workflow, a period, and the state name. The internal name cannot contain spaces or additional periods. The maximum number of characters is the same as the number for the **Name** field. **Note:** Internal names are not used for system states. **Tip:** You can copy the internal name and paste it into a script or a Web service to reference the state.
Status	Select the state status (active or inactive). **Note:** This field is not displayed for system states.
Owner	Select the owner of the state. **Note:** This field is not displayed for system states.
Subtask	Select a binary/trinary field to determine the status of all subtasks. The value of this field supplies the subtask status ("In Progress," "Completed," or "Rejected") associated with each state. **Note:** This field is not displayed for system states.

Element	Description
Time capture	*(Optional)* This option is visible only if the Time Capture is turned on for the workflow. Select to show or hide Time Capture options on the state form. By default, the state inherits the setting applied for states at the workflow level. For more information, see About Time Capture [page 103].
End-user help text	Click **Edit** to open an HTML editor and add optional comments or notes about the purpose of the state. The text you enter is displayed to users. For details, refer to Chapter 24: Providing Custom End-user Help [page 443].

Form Tab of the State Property Editor

Element	Description
State form, Print form	Select the state form or print form to be associated with the selected state. You can select one of the following items: • **[Inherit from workflow]** The default form that was specified for the workflow on the **Forms** tab of the workflow Property Editor is used. • **[quick form]** A form that was generated by the system is used. • A custom form that was created is used. **Note:** For details about the differences between quick forms and custom forms, see Quick Forms or Custom Forms? [page 235]. For information about custom form types, see About Forms [page 233].
New	You can base a new custom form on the quick form for the state or on an existing custom form, or you can start with an empty form. (You can also create a state or transition form by right-clicking the **Forms** heading in App Explorer.)
Preview	View a mockup of the state form as it will appear to users. Use the controls at the top to view the form as it will be seen in a different state, by users assigned to different roles, and by style (classic or Work Center, for example). Click the transition buttons at the top of the form to experience the user flow without deploying the process app. **Note:** You can preview a state form from the Form Preview Dialog Box [page 317] if the selected role has any "view item" privilege granted to it in the Roles Editor [page 409]. You can preview a transition form if the selected role has any "transition item" privilege. (An exception to this is the "Update" transition type, which requires any "update item" privilege.)

Element	Description
Edit	If a custom form is selected for the state, click to open the form in the form editor.

Actions Tab of the State and Transition Property Editor

Element	Description
Actions	Lists actions defined for the selected state or transition. **Tip:** To the extent possible, actions are performed in the order in which they are listed here. Select actions and use **Move up** and **Move down** to change that order.
New	Opens the **Action Wizard** to create a new action.
Edit, View	Opens the selected action for modification in the **Action Wizard**. **Note:** If the workflow containing this state or transition is not checked out, this button is labeled **View**, and you cannot change the action definition.
Delete	Removes the selected action.
Move up, Move down	Cause the selected action to be performed earlier or later. **Note:** • Some action types (such as pre- and post-state Web service calls) are order-dependent. For those types, these controls are unavailable. • When there are multiple asynchronous orchestration actions on the same transition, only one event of each event type is raised. Events start orchestration workflows that will run simultaneously depending on available resources, so the ordering of event actions on the **Actions** tab has no effect on the execution order of the orchestration workflows. See Considerations for Using Actions [page 412] for details.
Show action summaries	Displays textual summaries of the actions.

Transitions Tab of the State Property Editor

You can define multiple transitions for a state. When you select a state in an application workflow or sub-workflow, you can do the following:

- Modify the order of its transitions. This order determines the sequence in which standard transition buttons will be displayed in the button bar on the form for the state.

> **Note:** Transitions are listed in descending order. That is, transitions that appear first on the form appear higher in the list.

- For certain states, exclude the transition button for a transition on the **Any** system state. (For information about this system state, see System-Provided States [page 133].)

Element	Description
Transitions	Lists the transitions defined for the selected state. Transition details are displayed. Such details include the target state, the status of the transition, and for sub-workflows, the parent workflow.
	> **Note:** On quick forms, transitions are listed in descending order. That is, transitions that are displayed first on the form appear higher in the list.
	- To change the transition order, use **Move up** and **Move down**.
	- To change the status of a transition in a workflow, click the status and then select **Enabled** or **Disabled**. For transition in a sub-workflow, select **Enabled**, **Disabled**, **Inherited (Enabled)**, or **Inherited (Disabled)**.
	- To exclude a transition button for an enabled transition on the **Any** system state, click the status and then select **Excluded for this state**.
	> **Note:** You cannot change the **Excluded for this state** status in an inherited state in a sub-workflow. However, if the status in the parent workflow is **Inherited (Enabled)**, you can change it to **Excluded for this state** in a sub-workflow.
Move up	Raises the order of the selected transition.
Move down	Lowers the order of the selected transition.
Hide disabled transitions	Hides the transitions whose status is **Disabled**.
Show transition summaries	Describes what the transition does when it is executed. For example, "show transition quick form, and transition to Assigned state."

Element	Description
Limit button bar to *n* buttons	Enables you to show a specified number of transitions as buttons and hide the rest in a drop-down list.

Chapter 10: Managing Transitions

This section contains the following information:

- About Transitions [page 143]
- Working with Transitions [page 143]
- Transition Settings [page 156]

About Transitions

Transitions are used to:

- Submit new primary items.
- Move primary items to a different state.
- Update existing items.
- Delete items.

Most transitions are available as buttons on state forms. Users click a transition button to initiate the transition. In most cases, users can enter data about an item on a transition form before completing the transition.

You can replace a transition button with a button, hyperlink, or image control that can be placed anywhere on the form. For button and hypertext controls, you either use the transition name as the control label, or type a different name. For examples, see Custom Transition Control Tutorials [page 282].

Transitions can also be executed without user intervention. For example, you can:

- Create actions that execute a transition based on the value of a specific field. For details, refer to About Actions [page 411].
- Create a notification that transitions items based on a rule you define. This task is performed in Application Administrator. For details, refer to the *SBM Application Administrator Guide*.
- Use the SBM Application Engine Web Services API, along with transition actions. For details, refer to the *SBM Web Services Developer's Guide*.

Transition Types

A variety of transition types are available. Regular transitions are the most commonly used transition type because they move items from one state to another.

Other transition types post items into different workflows or applications or copy items. For details, refer to Chapter 33: Transition Reference [page 527].

Working with Transitions

This section contains the following topics:

- Configuring Transitions [page 144]

- Creating Additional Submit Transitions [page 145]

- Restricting Transitions [page 145]

- Enabling Submit on Behalf of Another User [page 148]

- Creating a Post Transition Between Two Primary Tables [page 149]

- Mapping Fields for Post, Publish, Copy, or Subtask Transitions [page 150]

- Calculating Values for Date/Time and Numeric Fields [page 153]

Configuring Transitions

Adding a Transition [page 144]

Renaming a Transition [page 144]

Duplicating a Transition [page 144]

Deleting a Transition [page 145]

Adding a Transition

You can add transitions between two states in an application workflow or sub-workflow.

To add a transition between two states:

1. From the **Workflow Palette**, drag a transition type onto the state in which you want the new transition to begin.

   > **Note:** Update and Copy transitions use only a single state. For these transition types, drag the transition on the state where it should begin and end.

2. Release the mouse button.

3. Click the state in which you want the new transition to end.

4. Type a name for the transition, and then press the Enter key.

5. Apply transition settings on each tab as they apply. Refer to the online help for details on each setting.

Renaming a Transition

You can rename a transition in an application workflow or sub-workflow.

To rename a transition:

- Right-click the transition, select **Rename**, type the new name, and press the Enter key.

Duplicating a Transition

You can duplicate transitions in any workflow or sub-workflow. This enables you to quickly copy transition attributes, such as form selection, form actions, and field overrides. For

actions that execute asynchronous orchestrations, new mapping definitions are automatically created for the duplicated transition.

By default, duplicated transitions use the same "to" and "from" states as the original transition. You can change the originating and destination states as needed.

To duplicate a transition, right-click the transition, and then select Duplicate.

Deleting a Transition

You can delete any transition from a workflow or a sub-workflow.

> **Note:** You can choose to disable a transition rather than delete it. Disabling a transition allows you to enable it later; deleting a transition removes it permanently from the workflow and any sub-workflows.

To delete a transition:

- Right-click the transition, and select **Delete**.

Creating Additional Submit Transitions

Submit transitions serve as entry points into your process app.

You can define multiple Submit transitions, but only one transition can be set as the "default" transition. This default transition is used when users manually submit new items.

Additional Submit transitions can be used for submitting items using Post or Create Subtask transitions or through an automated process. This enables you to set different field properties, such as required fields, for items submitted through these methods.

You can also use additional Submit transitions for e-mail submissions. For setup steps, refer to the "Preparing Your System for E-mail Submission" section in the *SBM Application Administrator Guide*.

To create an additional Submit transition:

1. In the Workflow editor, drag a Regular transition on to the Submit state.

2. Release the mouse button, and then click the state where you want newly submitted items to reside.

3. On the **Options** tab, clear the **Default submit transition** check box. (By default, the first transition from a Submit state has this option selected, and other transitions have this option cleared.)

4. Review required fields for the transition and either remove the requirement or set default values for these fields.

5. Deploy your process app.

Restricting Transitions

SBM offers several mechanisms for restricting the transitions that are available to users.

Restricting Transitions for All Items

Privileges enable you to restrict transitions for all items in a project. You can grant or remove transition privileges for roles in SBM Composer or for individual users and groups in Application Administrator.

The following privileges are available for controlling transitions for all items in a project, depending on a user's product-access type:

- Transition All Items

- Transition Item if Owner

- Transition Item if Secondary Owner

- Transition Item if Submitter

Restricting Individual Transitions

In some cases, you may need to limit individual transitions that are available to users. For example, you may want to restrict an "Approved" transition to users with a Manager role. In this case, users with the "Transition All Items" privilege would not see the "Approved" transition unless they are assigned to the Manager role.

The following restriction types are defined in SBM Composer and are the best method for restricting individual transitions:

- **Role Restrictions**

  Transitions are not available to users assigned to restricted roles. Refer to Restrict By Role Tab of the Transition Property Editor [page 165].

- **Rule Restrictions**

  Transitions are not available based on business rules, such as removing an "Escalate to Management" transition for low-priority items. Refer to Restrict By Rule Tab of the Transition Property Editor [page 166].

- **Item Type Restrictions**

  Transitions are not available based on item types. For example, you can restrict a "Send to Software Team" transition for items with a "Hardware Requests" item type. Refer to Restrict By Type Tab of the Transition Property Editor [page 165].

In SBM Application Administrator, on-premise customers can restrict transitions so they are unavailable for members of specific groups. For guidance, refer to the *SBM Application Administrator Guide* located on the Documentation Center.

Role Privileges and Restrictions for Transitions

The transition privileges associated with roles and the restrictions specified for transitions are processed separately, in the following three steps:

1. Check privileges to make sure the user has permission to perform any transitions on the item. This is based on the item itself (for example, whether the user is the owner of the item); and the role, group, and user privileges granted to the user.

CAUTION:

⚠️ Privileges are additive and can be indifferent to role-based transition restrictions. For more information, see Associating Multiple Roles with a User [page 405].

2. If the user has permission to transition the item, determine which transitions are available. This is primarily based on the state the item is in.

3. Determine whether each available transition has any restrictions on it. The restrictions can be based on role, item type, rule, and group (set in SBM Application Administrator).

This means that both step 1 and step 3 must pass before a user can see the transition. For example, consider the following scenario:

- An application workflow has five states: **New**, **Assigned**, **In Progress**, **Tested**, and **Closed**; and five transitions: **Submit**, **Assign**, **Start Work**, **Test**, and **Close**.

- All roles have privileges to view, update, and transition the **Assigned**, **In Progress**, and **Closed** states.

- The **Assign**, **Start Work**, and **Test** transitions have no restrictions on them.

- Only the "Tester" role can execute the **Close** transition, which moves the workflow from the **Tested** state to the **Closed** state. To implement this, the **Closed** state has "Tester" has the owner. This is specified on the **Restrict by Role** tab of the transition Property Editor.

- The "Manager" role can only see the states it owns. This is specified in the roles editor, where "Transition Item if Owner" is the only transition privilege selected in the **Items** category. The "Developer" and "Tester" roles have the "Transition All Items" privilege.

- Amy is associated with the "Manager" role. Emily is associated with the "Developer" role. John is associated with the "Tester" role. Eric is associated with all three roles.

The following use cases describe how access to transitions is granted:

- Amy submits an item. No owner is specified for the **New** state. Because she is not the explicit owner of the state, instead of seeing the **New** state form with the **Assign** transition button, she sees a message saying "The item was successfully submitted."

- Emily submits an item. Because the "Developer" role can transition all items, and because there is no **New** state owner, she sees the **New** state form with the **Assign** transition button. The same is true for the **Assigned**, **In Progress**, and **Tested** states and their associated transition buttons. However, the **Closed** state has "Tester" as the owner, so Emily does not see the **Closed** state form with the **Close** transition button. Instead, she sees a message saying "The item was successfully transitioned, or, if she has "update" privileges, she sees the state form with only an **Update** button on it.

- John submits an item. Because the "Tester" role can transition all items, and because this role is the owner of the **Closed** state, he can see all state forms and execute all transitions.

- Eric submits an item. Because he is associated with all roles and has the "Transition All Items" privilege, he can see all state forms and execute all transitions. The "Tester" role and its associated privileges and transition restrictions takes precedence over the more limited "Manager" and "Developer" roles.

Enabling Submit on Behalf of Another User

The **Allow submit on behalf of another user** option enables privileged users to submit new items on behalf of other users in your system. For example, you might enable this feature for service desk technicians to quickly submit tickets on behalf of business users that call the Help Desk.

The option appears in the **Options** tab of the transition Property Editor in the parent workflow's **Submit** transition. The option is not editable in the **Submit** transition in sub-workflows; however; administrators can override this setting at the project level in SBM Application Administrator.

Key Benefits

- Automatically moves the system *Submitter* field to the top of the **Standard Field** section on the **Submit** transition quick form, which allows users with the **Submit on Behalf of Another User** privilege to select a submitter that has the privilege to submit items in the project.

- Creates a specific change history entry on the new item, indicating that the item was submitted by the technician on behalf of the business user.

- Enables administrators to create notifications and reports for items that were submitted on behalf of another user.

To allow users to submit on behalf of other users:

1. Enable the **Allow submit on behalf of another user** option in the parent workflow's **Submit** transition.

2. Optionally, select the **Allow update of submitter** check box on non-**Submit** transitions (except the system **Update** transition). This enables users to change the *Submitter* value after the item is submitted in the event of a mistake or required change. For example, if the original submitter leaves the company and tickets need to be associated with a different submitter, select this option to make the *Submitter* field accessible for users with the **Update Submitter Field** privilege.

3. Add the *Submitting Agent* field to your primary table. This system field is automatically populated with the name of the user who physically executes the submit transition on behalf of another user.

   > **Important:** You must add the *Submitting Agent* field to your primary table in order to deploy the process app without validation errors after you select the **Allow submit on behalf of another user** option.

4. Grant the **Submit on Behalf of Another** and **Update Submitter Field** privileges to the necessary roles. By default, these privileges are granted to the Administrator role for new applications. You must explicitly grant these privileges to other applicable roles.

> **Tip:** These privileges are only applicable to users with Regular User or Managed Administrator product-access. External Users and Occasional Users cannot submit items on behalf of another user.

5. Add the *Submitter* field to any custom forms as necessary. If you use the quick form on your **Submit** transition, you do not need to make any changes; the *Submitter* field automatically appears the top of the **Standard Field** section for users that have the **Submit on Behalf of Another** privilege.

> **Tip:** If you enable the **Allow submit on behalf of another user** setting, and your submit form contains the system *Contact* and *Company* fields, the values for these fields are automatically populated with the contact and company that are associated with the user that is manually set in the *Submitter* field. The *Contact* and *Company* fields are set to **(Auto)** when **Allow submit on behalf of another user** is enabled.

6. Administrators can log in to SBM Application Administrator and override the **Submit** transition settings at the project level as necessary. For example, you can disable the feature for a certain project by editing the project's **Submit** transition and turning **Submit on Behalf of Another User** off. Similarly, for non-**Submit** transitions, edit the project, select the desired transition, and turn **Update Submitter** off if you do not want users to edit the *Submitter* field.

7. In SBM Application Administrator, administrators can grant the **Submit on Behalf of Another** and **Update Submitter Field** privileges to individual users and groups as needed.

8. In SBM Application Administrator, create and subscribe users to notifications that notify the submitter when a new item is created on his or her behalf. For example, use a rule like **Any issue is Submitted on Behalf of Another and Submitter is Equal to (Current User)**, and then subscribe one or more users to the notification. These users will receive a notification when an item is submitted on their behalf.

Creating a Post Transition Between Two Primary Tables

This example describes how to create a **Post** transition that posts from one primary table to another, without changing the state in the active workflow. It uses primary tables named Issues and Incidents in an example process app containing Issue Management and Incident Management applications.

To create a Post transition between two primary tables:

1. In the App Explorer, select the workflow in which you want to create the **Post** transition.

2. In the workflow editor, drag a **Post** transition from the **Workflow Palette** onto the workflow, releasing the mouse button over the state from which you want the post to occur.

3. Click on the same state again. This lets users post an item and leave the original item in the same state.

4. Type a name (such as `Post to Engineering`) for the transition. This name typically appears to the user as a button on the state form, so it is helpful to provide a name that conveys the type and purpose of the transition.

5. On the **Options** tab of the transition Property Editor, select **Quick Transition** to automatically transition the original item. This lets users quickly post the new item. This option should be selected only if there are no required fields for this transition or if you do not want users to provide data for the original item before posting it.

6. On the **Post Options** tab of the transition Property Editor:

   a. From the **Post application** drop-down list, select the application that contains the table to which you want to post items.

      Note: The target application must be referenced in the source process app. For more information, see About References [page 369] and Defining a Reference [page 371].

   b. From the **Post table** drop-down list, select the table (such as Issues) associated with the application to which you want to post items.

   c. If your workflow contains *Single Relational* or *Multi-Relational* fields to the table and in the table selected in the **Post table** drop-down list, you can select these fields from the **Set new item in this item's field** and **Set this item in new item's field** drop-down lists. This lets you create a field relationship between the two tables.

      Tip: You can create a **Submit** transition in the receiving workflow that can be used by **Post** transitions. You can configure field properties as needed for this transition.

   d. For **When finished, show**, select **Original item** to return users to the item from which they execute the **Post** transition, after the transition is complete. Select **New item** to display the new item instead.

   e. From the **Item link type** drop-down list, select the type of link (such as "2-Way, no triggers") you want to add between the original and posted item. By default, no link is created.

7. Click **Map Matching Fields** to quickly map matching fields of the same type. Fields are matched first by database name, then display name.

8. Modify field mappings as needed. For details, refer to Mapping Fields for Post, Publish, Copy, or Subtask Transitions [page 150].

9. From the **File** menu, click **Save** (or click the **Save locally** icon in the Quick Access toolbar) to save your workflow changes.

Mapping Fields for Post, Publish, Copy, or Subtask Transitions

You can map fields in the originating item to fields in the newly created item for Post, Subtask, Publish, and Copy transitions.

Each transition type has default mappings that are used if you have not applied additional field mappings. Once you do this, the default mapping no longer applies and you must manually map to fields, such as *Submitter* or *Title*.

> **Note:** Prior to SBM 11.5, when copying field values to another item, the system used default mappings for each field, but you could override those mappings by explicitly mapping fields. For example, when posting an item into the same primary table, all field values were copied to the new item and any manual field mappings that you made would override the default mapping. To revert to this behavior, follow the steps in solution S142752.

The following table describes mapping rules for the different transition types.

Transition Type	Default Mapping	Notes
Post Subtask Publish	Items posted within the same primary table contain all the fields values of the original items, except for the *Submitter* and *Submit Date/Time* fields. Items posted across tables contain the *Title* and *Description* of the original item.	You can map to compatible fields types.
Copy	All field values are copied from the source item to the new item, except for the *Submit Date/Time* and *Item ID* fields.	You can map to compatible fields types.

The following table lists compatible field mappings.

Source field type	Allowed receiving field type
Binary/Trinary	Binary/Trinary or Text
Date/Time	Date/Time or Text
File	File
Folder	Folder or Text
Multi-Group	Multi-Group or Text
Multi-Relational	Multi-Relational or Text
Multi-Selection	Multi-Selection or Text
Multi-User	Multi-User or Text
Numeric	Numeric or Text
Single Relational	Single Relational or Text

Source field type	Allowed receiving field type
Single Selection	Single Selection or Text
Sub-Relational	Text
Summation	N/A
Text	Text
URL	URL
User	User or Text

Consider the following information when you apply field mapping:

- Selections for *Single Selection*, *Multi-Selection*, *Multi-User*, and *Multi-Group* fields should have identical names when mapped. If a selection is not available for the field, **None** is selected as the value in the posted item. If a selection has been disabled or deleted, **Disabled** is displayed, and users must select a valid value before completing the transition.

- Default values for *Single Selection*, *Multi-Selection*, *Multi-User*, and *Multi-Group* fields in the receiving project or table are overwritten by values from the source item. For example, a **Post** transition is used to post items from the Issues table to the Incidents table, and the *Issue Type* field is mapped to the *Incident Type* field, which has a default value of **Problem Report**. If a user selects **Bug Report** for the **Issue Type** field before posting an item from the Issues table to the Incidents table, the **Problem Report** default value is ignored and is replaced by **Bug Report**.

- When you map *Text* fields, verify that the **Enable Rich Text** option is consistently set for both fields. For example, if rich text editing is enabled on the originating *Text* field, it must be enabled on the destination *Text* field. If not, posted text will not be rendered.

- When mapping fields for **Publish** transitions, the mapping applies only to the Problems table, not to the Resolutions table.

To map fields when you edit a workflow in which a **Post**, **Publish**, **Copy**, or **Subtask** transition exists:

1. In App Explorer, select the workflow that contains the transition for which you want to map fields.

2. In the workflow editor, select the transition.

3. On the **Post Options** tab of the transition Property Editor, scroll down to the **Mapping** area.

4. Under **This item's field**, select the field to be mapped to a field in another project or table, or to a field in a new item that uses the same table.

5. Under **Mapped to** in the same row, use the drop-down list to select the field you want to map to (from the fields in the table you are posting to).

> **Tip:** To map all fields with the same name and type, click **Map Matching Fields**. You can then change the mappings for individual fields as needed. If you change your mind, click **Clear Mappings**.

For field mapping details, refer to Post Options Tab of the Transition Property Editor [page 161].

Calculating Values for Date/Time and Numeric Fields

Field calculations for transitions provide a way to collect metrics using *Date/Time* and *Numeric* fields. For example, you could calculate how long Help Desk calls wait before a Support representative begins working the issue, or calculate interest values based on the current date.

The **Set to calculation** option appears when you select a *Date/Time* or *Numeric* field on the **Field Overrides** tab of the transition Property Editor.

To calculate values for *Date/Time* and *Numeric* fields:

1. In App Explorer, select the workflow containing the transition.

2. In the workflow editor, select the transition for which you want a *Date/Time* or *Numeric* field to perform a calculation.

3. In the transition Property Editor, select the **Field Overrides** tab, and select the **Override field properties for '*field name*'** option.

4. Optionally, click **Arranged by**, and select a different sort order for the list of fields.

5. In the field list, select the *Date/Time* or *Numeric* field for which you want to define the calculation to be performed.

6. In the **Value options** area, select the **Set to calculation** option.

7. Do one of the following:

   - Select **Field** from the **Operand 1** menu, and then select a field from the drop-down list or a keyword from the date/time tool. You cannot select the field being edited or a field that could cause a recursive calculation. For a *Date/Time* field, see Date/Time Fields [page 218] for a description of your options for populating this field.

     CAUTION:

     > The "(Auto)" string is not supported as a default value for *Date/Time* fields.

   - Select **Value** from the **Operand 1** menu, and then enter a constant value in the adjacent text box.

     > **Tip:** Calculations do not work for system date fields (Submit Date/Time, Close Date/Time, Last State Change Date, Last Modified Date) in a transition calculation for which the date will be calculated. Instead, use the "Now" keyword, which gives you the same calculation.

8. Select an operator from the **Operator** drop-down list.

   The list contains only valid operators for this field and the current operand. If you change the operand, the operator list is updated with the appropriate choices. If you select an operator that is not valid with the operand, the operator changes, usually to **None**. (The **None** operator is useful if you want to transfer the value of another field to this field during the transition.)

9. If you select an operator other than **None**, you can specify a second operand from the **Operand 2** menu. This list offers the same options and constraints as the first.

10. From the **Perform calculation** list, specify whether the calculation should be performed before the transition form opens, after it is submitted, or both.

11. From the **Empty operand fields** list, specify how the transition should treat empty operands:

    - **Are Invalid**: Select this option to require users to provide values for fields used as operands. The transition cannot be completed if values are not provided.

    - **Skip Calculation**: Select this option to let users complete the transition without providing values for fields used as operands. The calculation is simply skipped if values are not provided.

    - **Treat as Zero**: Select this option to perform the calculation using zeros in place of any unprovided values for fields used as operands.

12. For *Numeric* fields and *Date/Time* fields with the **Elapsed time** option enabled, select the **Add calculation to current value** check box to add the result of the calculation to the current value of the field. Use this option to increment the current value or to calculate the total of the calculation and the current value.

About Operand Fields and Operator Selection Lists

The following table lists the valid choices for the operand fields and operator selection lists for a *Numeric* or *Date/Time* field using the calculation feature. The valid choices are based on the type of transition field being edited.

Field Type	1st Operand Constant	1st Operand Field Types	Operators	2nd Operand Constant	2nd Operand Field Types
Elapsed Time	Elapsed Time	Elapsed Time	None, +, −	Elapsed Time	Elapsed Time
Elapsed Time	Elapsed Time	Elapsed Time	*, Truncating /, Rounding /	Float	Numeric Int, Single Select, Summation
Elapsed Time	Elapsed Time	Elapsed Time	Truncating *, Rounding *, Truncating /, Rounding /	Float	Numeric Float

Field Type	1st Operand Constant	1st Operand Field Types	Operators	2nd Operand Constant	2nd Operand Field Types
Elapsed Time	Date/Time	Date/Time	–	Date/Time	Date/Time, Date
Elapsed Time	*(not applicable)*	Date	–	Date/Time	Date/Time, Date
Date/ Time	Date/Time	Date/Time	None, +, –	Elapsed Time	Elapsed Time
Date/ Time	*(not applicable)*	Date	None, +	Elapsed Time	Elapsed Time, Time
Date/ Time	*(not applicable)*	Date	–	Elapsed Time	Elapsed Time
Date	Date	Date/Time, Date	None, Truncating +, Rounding +, Truncating –, Rounding –	Elapsed Time	Elapsed Time
Time	Time	Time	None, +, –	Elapsed Time	Elapsed Time
Time	*(not applicable)*	Date/Time	None (assigns time portion of date/time)	*(not applicable)*	*(not applicable)*
Numeric Int	Int	Numeric Int, Single Select, Summation	None, +, –, *, Truncating /, (integer math), Rounding /	Int	Numeric Int, Single Select, Summation
Numeric Int	Int	Numeric Int, Single Select, Summation	Truncating +, Truncating –, Truncating *, Truncating /, Rounding +, Rounding –, Rounding *, Rounding /	Float	Numeric Float
Numeric Int	Float	Numeric Float	Truncating +, Truncating –, Truncating *, Truncating /, Rounding +, Rounding –, Rounding *, Rounding /	Int or Float	Numeric Int/ Float, Single Select, Summation
Numeric Float	Int or Float	Numeric Int/ Float, Single Select, Summation	None, +, –, *, /	Int or Float	Numeric Int/ Float, Single Select, Summation

Field Type	1st Operand Constant	1st Operand Field Types	Operators	2nd Operand Constant	2nd Operand Field Types
Numeric Float	Int or Float	Numeric Int/ Float, Single Select, Summation	*	*(not applicable)*	Elapsed Time (converted to hours)

Transition Settings

The following sections provide descriptions of settings used to configure transitions. Sections are organized by property editors.

General Tab of the Transition Property Editor

Element	Description
Name	For most transitions, the name appears as a button on the item details page that users can click to transition items to another state. Certain transition names do not appear as button labels. For example, names for **Submit** transitions are not used in SBM Work Center. However, the state change history and change history reflect the name you provide here. You can replace a transition button with a button, hyperlink, or image control that can be placed anywhere on the form. For button and hypertext controls, you either use the transition name as the control label, or type a different name. For examples, see Custom Transition Control Tutorials [page 282].
Internal name	View or modify the internal name of the transition. This field is read-only if the process app has been published. The default value of this field is the name of the workflow, a period, and the transition name. The internal name cannot contain spaces or additional periods. The maximum number of characters is the same as the number for the **Name** field. **Note:** The workflow name is not included for the default **Update** and **Delete** transitions. **Tip:** You can copy the internal name and paste it into a script or a Web service to reference the transition.
Type	Select a transition type from the list. You set the transition type by dragging the correct type from the **Workflow Palette**, but you can change it here. See Transition Types [page 527].

Element	Description
From	The state where the transition begins. You set the **From** state when you drag the transition from the **Workflow Palette**, and you can change it by dragging the starting end of the transition to a different state. Every workflow includes three special-purpose states of interest: • **Submit** Used to define submit transitions. For details, refer to Creating Additional Submit Transitions [page 145]. • **E-Mail:** Used to define a **Submit** transition for e-mail submissions. For setup details, refer to the "Preparing Your System for E-mail Submission" section in the *SBM Application Administrator Guide*. • **Any:** Used to define a transition that is available from every state in a workflow. For example, you could define a **Post** transition that lets users post an item to another project, no matter which state the original item is in.
To	The state where the transition ends. You set the **To** state when you drag the transition from the **Workflow Palette**, but (except for **Copy** and **Update** transitions) you can change it by dragging the ending (arrowhead) end of the transition to a different state.
Status	Specifies whether the transition is enabled or disabled. You can tailor your application by enabling or disabling transitions for child workflows. • **Enabled:** Enables the transition for the selected workflow. If you are setting the status for a transition for a child workflow, select this option to use transition properties set for this transition rather than inherited properties. • **Disabled:** Disables the transition for the selected workflow. **Note:** You cannot disable the **Otherwise** transition from a decision. • **Inherited:** Assigns the status in a child workflow to be the same as in the parent workflow. This option is selected by default for transitions in a child workflow.
Time capture	*(Optional)* This option is only visible if the Time Capture option is turned on for the workflow. Select to show or hide Time Capture options on transition forms, except for Submit forms. By default, the transition inherits the setting applied for transitions at the workflow level. For more information, see About Time Capture [page 103].

Element	Description
Require entry	*(Optional)* When Time Capture options are visible to users for a transition, you can require them to enter data when they execute the transition. This requirement is ignored for automated processes, such as Web services and scripts. By default, the transition inherits the setting applied for transitions at the workflow level. For more information, see About Time Capture [page 103].
End-user help text	Click **Edit** to open an HTML editor and add optional comments or notes about the purpose of the transition. The text you enter is displayed to users. For details, refer to Chapter 24: Providing Custom End-user Help [page 443].

Options Tab of the Transition Property Editor

Options

Element	Description
Quick transition (do not show a form)	Bypasses the transition form. This option lets users quickly transition items out of a particular state if fields do not need to be modified. You can also use this option to bypass the transition form for the original item for **Post** or **Subtask** transitions. The form opens if there are required fields specified for the transition. If you select this option, you cannot select the **Required attachment**, **Show "New Note" field**, and **Require "New Note" entry** options.
Show "New Note" field	Inserts a **New Note** field during the selected transition. The field appears after the **Standard Field** section and lets users add a note to the item, automatically recording their name and a date/time stamp when the note was created.
Require "New Note" entry	Specifies the **New Note** field as requiring an entry. Users must enter a note to transition an item successfully.
Default submit transition	Specifies a transition from a **Submit** state for items that a user submits manually. By default, the first transition from a **Submit** state has this option selected, and other transitions have this option cleared. For details, refer to Creating Additional Submit Transitions [page 145].

Element	Description
Hide transition button on state form	Hides the transition button on state forms. This option is typically used to hide transition buttons for transitions that are invoked indirectly, such as Post or Subtask transitions. This option can be set for any transition, except those that start from the **Submit** state.
Transition button on mobile state form	If the application is enabled for mobile access, you can show or hide the transition button on mobile state forms. The options are: • **Always show:** Show on all mobile state forms (both simple and advanced). • **Always hide:** Hide on all mobile state forms. • **Hide in simple view:** Show on advanced mobile state forms only. These options can be set for any transition, except those that start from the **Submit** state.
Allow update of submitter field	Enables users to change the *Submitter* value after an item is submitted in the event of a mistake or required change. For example, if the original submitter leaves the company and tickets need to be associated with a different submitter, select this option to make the *Submitter* field accessible for users with the **Update Submitter Field** privilege.
Reset submitter field to current user	During a copy transition, this replaces the *Submitter* field value from the original item with the user who performs the copy. If this option is not selected, the original *Submitter* value is copied to the new item.

Attachment

Element	Description
Required attachment	If attachments are not required for the transition, select **<None>** . Otherwise, select the type of attachment to be required.

Authentication

Element	Description
Required, Not required, Inherited	**Important:** If transition authentication is used, it is recommended that the SBM Server use HTTPS protocol. Select **Required** if users should supply their login IDs for the transition. The transition fails if a user does not provide the correct login ID and password or attempts to provide the login ID and password of another user. On successful authentication, the login ID and password are recorded as an electronic signature. If you select **Required**, you can also select an authentication option from the *DateTime* field to update list to record the time the transition was performed. The transition is stored in the transitions section of the **Change History** section for each item.

When selecting a *Date/Time* field for transition authentication:

- Only *Date/Time* fields that are set to display the date and time or only the date are available for this option.

- System fields cannot be used for transition authentication.

- Deleted fields cannot be used for transition authentication.

- Consider naming the *Date/Time* field so that it is clear that is used for transition authentication. Examples include "Authentication Time" and "Electronic Signature Recorded At."

- The *Date/Time* field specified for transition authentication is always populated when the transition is executed. To prevent users from changing the date and time, consider moving the field to the **Hidden Fields** section or another section controlled by privileges. You can also set the field as read-only.

- Change history for transition authentication is not recorded on **Submit**, **Copy**, or **Delete** transitions.

When setting authentication options for transitions:

- If your system uses automatic log in (via Windows authentication or a third-party authentication system) for browser users, electronic signature passwords are checked against the same authentication source that Web services use for authentication.

  This means if browser authentication is performed via Windows authentication and Web services are set to authenticate against the internal database, users should synchronize their SBM passwords and their network passwords, unless they choose to specify a unique password for authenticating transitions.

- If your system uses LDAP authentication, LDAP handles password verification.

- Authentication settings apply only to transitions that are executed manually by users. Automatic transitions that require authentication will fail. Use care when

setting authentication options for transitions that are executed as part of actions, by e-mail submission, or by API programs.

- You can override authentication options for child workflows and for projects. For best results, set authentication options at a parent workflow, and then enable or disable them for child workflows and projects as needed.

Post Options Tab of the Transition Property Editor

Use this tab to modify the post options of a **Post**, **External Post**, **Publish**, **Subtask**, or **Copy** transition. You cannot set post options for an **Update**, **Delete**, or **Regular** transition.

Post Settings

Element	Description
Post Application	Select the application that contains the table you want to post to. You may need to create a reference to an application if it is in a different process app. For details, refer to Defining a Reference [page 371].
Post table	Select the table to which you want items to be posted. For **Post** transitions, you can select a primary table or an auxiliary table. For **Copy** and **Subtask** transitions, you must select a primary table.
Project	*(Display only)* By default, users with appropriate privileges are presented with a list of projects to submit the post item or subtask into. Limit the list of projects in SBM Application Administrator or by granting privileges to specific projects to which users can post items.
Set new item in original item's field	This drop-down list contains *Single Relational* and *Multi-Relational* fields in the table for the current item that are related to the specified post table. Select a field that will contain the Item ID and Title of the posted item in the original item. For example, if you create a **Post** transition that lets users post from the Issues table to the Incidents table, you can create a *Multi-Relational* field to the Incidents table and select that field from the **Set new item in this item's field** drop-down list. When users post items to the Incidents table, the Incidents field in the original item contains information about the posted item. If you select a *Multi-Relational* field for this option, all items posted from the original item are selected as values for the field. If you select a *Single Relational* field, the last posted item is selected as the value.

Element	Description
Set original item in new item's field	This drop-down list contains *Single Relational* and *Multi-Relational* fields for the current primary table that are available in the receiving table. For example, if the transition lets users post items from the Issues table into the Incidents table, an Issues *Single-Relational* or *Multi-Relational* field contained in the Incidents table is listed here. Select a field that will contain the Item ID and Title of the original item in the posted item. For best results, select a *Single Relational* field for this option, because the only valid value is the originating item that created the posted item. **Tip:** A relational field icon appears next to fields populated using the **Post**, **Copy**, **Publish**, or **Subtask** transition type. Users with privileges to view items in the relational field table can click the icon to open the item in a pop-up window.
Use submit transition	Each non-default item in this drop-down list consists of a workflow name and a submit transition. From this list, you can select a specific submit transition that can be used for **Post** and **Subtask** transitions. This lets you set different properties for this transition. For example, you could require users to populate certain fields when they submit directly into a project, but require them to populate a different set of fields when they use a Subtask transition. This field also lets you specify a single project into which posted items and subtask items can be submitted, when more than one project is available. For example, if an application contains a principal workflow and two subtask workflows, the submit transition from each of the subtask workflows appears in this list. Select the transition for the project into which this item should be submitted. (A project is created for each workflow when the process app is deployed.)
When finished, show...	Select **New item** to display the new item created by the transition after the transition is complete. Select **Original item** to return users to the item from which they execute the transition after the transition is complete.

Element	Description
Item link type *(Post, Subtask, Publish, and Copy transitions)*	Select one of the following: • **<None>**: Creates the transition without an item link type. • **1-Way, original to other, no trigger**: Creates a link from the original item to the new item without triggering a transition on the linked item. • **1-Way, original to other, with trigger**: Creates a link from the original item to the other item, and fires the trigger specified on the **Actions** tab on the original item. • **1-Way, other to original, no trigger**: Creates a link from the new item to the original item without triggering a transition on the linked item. • **1-Way, other to original, with trigger**: Creates a link from the new item to the original item, and fires the trigger specified on the **Actions** tab on the new item. • **2-Way, both trigger each other**: Creates links in both the original and new items, and fires triggers on both items. • **2-Way, no triggers**: Creates links in both the original and new items without triggering a transition on the linked item. • **2-Way, original triggers other**: Creates links in both the original and the new item, and fires the trigger specified on the **Actions** tab on the new item. • **2-Way, other triggers original**: Creates links in both the original and new items, and fires the trigger specified on the **Actions** tab on the original item.
Database name *(External Post transitions only)*	Select an external database into which the posted item will be added.

Field Mapping

Element	Description
This item's field	Lists fields in the primary table.

Element	Description
Mapped to	For each field in the primary table, select the name of the field to which the field for this item should be mapped. (For an **External Post** transition, the field is in the specified external database.) For **Post** and **Subtask** transitions, you can select the field from a drop-down list.
Map Matching Fields	Click this icon to map all fields with the same name and type. You can then change the mapping for individual fields as needed. (*Sub-Relational* fields are not automatically mapped.)
Clear Mappings	Click this icon to clear all mappings.

For field mapping details, refer to Mapping Fields for Post, Publish, Copy, or Subtask Transitions [page 150].

Form Tab of the Transition Property Editor

Use this tab to select, create, edit, and preview the form for the transition selected in the workflow editor.

Element	Description
Transition form	Select the form to be associated with the selected transition. You can select one of the following items: • **[Inherit from workflow]** The default form that was specified for the workflow on the **Forms** tab of the workflow Property Editor is used. • **[quick form]** A form that was generated by the system is used. • A custom form that was created is used. **Note:** For details about the differences between quick forms and custom forms, see Quick Forms or Custom Forms? [page 235]. For information about custom form types, see About Forms [page 233].
Quick transition (do not show a form)	Bypasses the transition form. This option lets users quickly transition items out of a particular state if fields do not need to be modified. You can also use this option to bypass the transition form for the original item for **Post** or **Subtask** transitions. The form opens if there are required fields specified for the transition. If you select this option, you cannot select the **Required attachment**, **Show "New Note" field**, and **Require "New Note" entry** options.

Element	Description
New	You can base a new custom form on the quick form for the transition or on an existing custom form, or you can start with an empty form. (You can also create a state or transition form by right-clicking the **Forms** heading in App Explorer.)
Preview	View a mockup of the transition form as it will appear to users. Use the controls at the top to view the form as it will be seen for different transitions, to users assigned to different roles, and by style (classic or Work Center, for example). Click the transition buttons or controls to experience the user flow without deploying the process app. **Note:** You can preview a state form from the Form Preview Dialog Box [page 317] if the selected role has any "view item" privilege granted to it in the Roles Editor [page 409]. You can preview a transition form if the selected role has any "transition item" privilege. (An exception to this is the "Update" transition type, which requires any "update item" privilege.)
Edit	If a custom form is selected for the transition, click to open the form in the form editor.

Restrict By Type Tab of the Transition Property Editor

Use this tab to restrict the transition to the selected the item types.

Note: You cannot restrict **Update** or **Delete** transitions by item type, and cannot restrict the **Otherwise** transition from a decision.

Element	Description
Item Types	Select the item types that are allowed to use the transition. The item types listed are the values defined (on the **Options** tab of the Property Editor) for the *Item Type* field in the primary table for the application. **Note:** When you restrict a transition to specific item types, the "None" value may be available to users. If users select "None," the restricted transition is not available.
Edit Item Types	For convenience, you can click this link to add, remove, or modify the values in the *Item Type* field for the primary table. Click ⬅ in the quick access tool bar to return to the current view of the **Restrict By Type** tab.

Restrict By Role Tab of the Transition Property Editor

On the **Restrict by Role** tab for a transition, do the following:

1. Select the check box at the top of the tab.

2. Select any roles for which you want to restrict access to the transition.

> **Note:** If you add a new role and the check box is selected, the new role will not be selected, so it will be restricted. If you add a new role and the check box is not selected, the new role will be selected, so it will not be restricted.

> **Restriction:** You cannot restrict **Submit**, **Update**, or **Delete** transitions by role, and cannot restrict the **Otherwise** transition from a decision.

> **Important:** In SBM Application Administrator, you can also specify groups who will be able to access the transition. In Application Administrator, any role-based restrictions that you set here in the transition Property Editor appear in a read-only state, and if any role-based restrictions exist, the administrator cannot clear the **Restricted** check box.

Restrict By Rule Tab of the Transition Property Editor

Use this tab to define or edit a rule that determines whether the transition is available, or to associate another rule with the transition.

Element	Description
Restrict the availability of this transition based on a rule	Select this check box if you want to make this transition visible on a state form only if the rule associated with the transition evaluates to true.
Rule	Select the rule you want to associate with this transition.
New	Opens the Rule Editor [page 352] so you can define a new rule.
Edit	Opens the rule editor so you can edit the selected rule.
Rule Summary	Contains a read-only string that represents the rule as you define it.

Chapter 11: Managing Tables

This section contains the following information:

- About Tables [page 167]
- Working with Tables [page 170]
- Table Settings [page 171]

About Tables

This section contains the following topics:

- Tables Overview [page 167]
- Considerations for Creating Tables [page 168]
- Relationships Between Tables [page 168]
- About the Value Display Format [page 168]

Tables Overview

The core of the SBM infrastructure is the SBM database. The database consists of two types of tables: primary tables and auxiliary tables.

Primary Tables

The primary table is the foundation of a SBM application. Each application has one primary table in the SBM database. The primary table is created automatically when you add an application to a process app.

A primary table stores a record for each item that follows the corresponding workflow. Primary tables contain a combination of required system fields, optional system fields, and custom fields. These fields are used to collect data as the primary item progresses through the workflow. Workflows depend on the primary table to hold the fields and actual records created by progress through the workflow.

When users with the proper privileges submit, transition, update, and perform other operations on primary items, they update the corresponding records in the primary table.

Auxiliary Tables

Auxiliary tables store items that support, but do not follow, a workflow process. Auxiliary items typically contain information that is collected once and used repeatedly. Auxiliary tables allow this information to be stored separately from primary items, making it easily shared and reusable across multiple applications.

Examples of auxiliary items include company and contact records. The SBM database includes system auxiliary tables (`Companies`, `Contacts`, `Problems`, `Resolutions`, and others).

Because auxiliary tables typically store static information, they are often used to create relationships between tables. *Single Relational* and *Multi-Relational* fields can be added to any auxiliary table or primary table to reference information contained in auxiliary tables. This is the basis for field dependencies, which let you populate selection lists in a selection or relational field type, based on selections in another field. See Configuring Field Dependencies [page 190] for details.

Considerations for Creating Tables

SBM Composer provides an extensive set of system and custom field types for the purpose of collecting information for primary and auxiliary items.

Before configuring a primary table, carefully consider its purpose, what types of information you want to display in state forms, and what data you want to collect in transition forms. Consider who can see the information in state forms and who can provide the data in transition forms.

Before configuring an auxiliary table, decide what type of auxiliary information is important to include in the application to support the workflow.

Relationships Between Tables

You can establish relationships between tables in SBM Composer using *Single-Relational* and *Multi-Relational* fields. SBM Composer also provides *Sub-Relational* fields to retrieve supplementary information from a related table. All primary and auxiliary tables in your process app can be used to define these relationships.

To use tables from applications from another process app (such as the system auxiliary tables in the Global Process App), define a dependency on that application in the current process app. See About References [page 369] for details.

About the Value Display Format

The value display format setting controls many features:

- Item Appearance [page 169]

- Relational Field Value Display and Sorting [page 169]

- Relational Field Value Searches [page 169]

- Auxiliary Data Sorting [page 169]

- Default Report Columns [page 169]

Guidelines for the value display format follow:

- The default value display format is <Title>.

- The value display format you specify is limited to 255 characters.

- HTML tags can be used to format item headers, but tags are not always rendered. For example, HTML tags will appear in the selection lists presented to users and in Manage Data sort columns.

- Privileges control the data returned by the value display format. If users do not have privileges to view a field that is included, asterisks are returned.

- *File* and *URL* field types are not supported in the value display format.

Item Appearance

Use the value display format to control:

- How items appear in lists in SBM Work Center. This includes search results and default columns for reports that return lists of items.

- The format of item headers on forms.

You can set the order in which the fields appear, add text to the format, and use HTML tags to format the content. For example, specify this value display format:

```
<Title> - <b><Owner></b>
```

Items from the table will appear in search results as follows:

ITEM10098 This is the title of the item - **Item's owner**

> **Note:** For primary items, the Item ID is always included.

Relational Field Value Display and Sorting

The value display format determines how values appear in *Single Relational* and *Multi-Relational* selection lists on forms. For example, the value display format for a *Versions* table might be <Product Name> <Version Name>.

Relational field values based on auxiliary tables are sorted by the table's value display format. To change the order in which values sort for the *Relational* field, modify the value display format for the auxiliary table.

Relational Field Value Searches

The value display format determines the fields that are searched and the order in which values are searched for the Relational Field Value Lookup form.

For example, if you add the *Item ID* and *Description* fields as the value display format for a primary table, users can search for values by item ID or keyword for relational fields that reference the table.

Auxiliary Data Sorting

If you specify multiple fields for the value display format for auxiliary tables, the fields are automatically shown in the first column of search results for the **Manage Auxiliary Data** feature.

This enables users to sort results by the value display format. For example, if the column contains a *Title* and *First Name* field in that order, results may appear as:

Manager, Adam

Manager, Bruce

Manager, Carol

Default Report Columns

For reports that return lists of items, fields specified in the value display format are displayed if users have not selected fields to display in the report.

Working with Tables

This section contains the following topics:

Creating a Primary Table

Each primary table is associated with an application. When you add an application to your process app in SBM Composer, the associated primary table is created automatically.

Use the table editor to modify the table, defining its properties and adding fields as needed. To modify fields, select them in the table editor and then make changes in the Property Editor.

Creating an Auxiliary Table

Create auxiliary tables as you need them. You typically add auxiliary tables to existing applications, which already contain one primary table.

In the table editor you can modify, delete, and create system fields and custom fields for auxiliary tables.

To create an auxiliary table:

1. In App Explorer, in the application to which you want to add the auxiliary table, right-click the **Tables** heading and select **Add New Table**. A new auxiliary table is added to the **Tables** heading, and it opens in the table editor. The auxiliary table automatically includes the required *Title* field, which you can rename but cannot delete.

2. Click in the heading area (above the list of fields) to select the entire table, or select a field in the list to display the appropriate Property Editor. See Modifying Table Properties [page 170] for details.

   Note: If the Property Editor is not visible, you can open it by selecting **Property Editor** in the **Common Views** area on the **Home** tab of the Ribbon.

3. Drag fields from the **Table Palette** to the table editor, or select and delete fields as needed.

Modifying Table Properties

To modify table properties:

1. Select the table in App Explorer.

The Property Editor below the table editor displays information about the selected table.

> **Tip:** If the Property Editor is not visible, select **Property Editor** in the **Common Views** area on the **Home** tab of the Ribbon.

2. View and modify information about the selected table using the tabs of the table Property Editor.

> **Note:** If you select a field in the table, the field Property Editor replaces the table Property Editor.

Copying a Table

You can base a new table on the structure of an existing auxiliary table. The new table will also retain certain table settings, such as record locking.

The following considerations apply:

- The new table does not include items (records) from the original table.

- Fields and field attributes cannot be modified during a copy.

- Relational fields that refer to the original table will refer to the new table. For example, if you copy the `Issues` system auxiliary table, and it has an *Issues Single-Relational* field, the field in the new table will point to itself, not to the original table.

- Fields that are deleted in the original table are not created in the new table. All non-deleted original fields are available for editing in the new table.

- You cannot copy a table that is checked out to another designer.

To copy a table, right-click the table in App Explorer and select **Duplicate**.

Table Settings

The following sections provide descriptions of settings used to configure tables. Sections are organized by property editors.

Table Editor

Use the table editor to design primary and auxiliary tables. It is displayed when you select a table in App Explorer. To hide columns in the table editor or show additional columns, right-click the table header row, select **Columns**, and then select or clear check boxes for the columns in the menu.

Use the tabs in the Property Editor to view and modify the various aspects of the selected table.

- General Tab of the Table Property Editor [page 172]

- Options Tab of the Table Property Editor [page 173]

- Labels Tab of the Table Property Editor [page 174]

- Icons Tab of the Table Property Editor [page 174]

- Forms Tab of the Table Property Editor [page 175]

- Dependencies Tab of the Table Property Editor [page 175]

Palette

A palette of field types and system fields is displayed to the right of the table editor. Add fields to the table by dragging them from the **Table Palette** to the table editor. Each of the system fields (*Active/Inactive*, *Description*, and so on) can be used only once in a table. The other (custom) field types can be used as often as needed.

> **Tip:** You can also add a field by right-clicking the table editor, selecting **Add New**, and selecting the field type.

Delete a field by right-clicking it in the table editor and selecting **Delete** or **Cut**. When you delete or cut a system field type, it is returned to the list to make it available for use. Cutting a field also copies it to the Windows Clipboard, so you can paste it into the same or other tables as needed.

> **Note:** You cannot copy a system field, or delete a required system field (such as the *Title* field in an auxiliary table). The **System** column (if displayed in the table editor) indicates whether the field is a system field.

General Tab of the Table Property Editor

Element	Description
Table name	The name that describes the contents of the table. This name is typically derived from the plural form of the items in the table. For example, if each item in the table represents a customer, the logical name for the table should be `Customers`. You can modify the logical name of an existing table. **Tip:** The name that you type in this field automatically populates the **Database table name**, which must be unique within the SBM database and is limited to 24 Unicode characters. You should limit the logical name to 24 characters, as additional characters will be left off of the **Database table name**.
Singular item name	The name that describes a single item in the table. For example, if each item in the table represents a customer, the singular item name should be `Customer`. SBM Composer automatically fills this field as you type the logical table name. You can modify the singular item name. However, for best results, it should be unique with the SBM database.
Edit tab name link	Shows the application editor, where you can change the application label that is visible to users. When you finish, click ⊙ (in the Quick Access tool bar) to return to the current view of the table editor and the **General** tab of the Property Editor.

Element	Description
Database table name	The unique internal database name for this table. SBM Composer automatically fills this field as you type the logical table name, ignoring characters after the first 24. **Note:** The database name can contain only ASCII alphanumeric characters (A–Z, a–z, 0–9) and underscores. Any other characters are ignored as you type them. You cannot change the database table name after the table is created.
Prefix	A read-only value assigned by the system.
Value display format	The format that controls how items in the table appear as values to users. For details, refer to About the Value Display Format [page 168].
Include 'Item Id'	Shows or hides the Item ID in the value display format of *Single Relational* and *Multi-Relational* fields. This option applies to primary tables and any auxiliary tables that include the *Item ID* field. **Note:** If the *Item ID* field was renamed, that name will appear instead of 'Item Id' in the option.
Description	Optional comment or note about the purpose of this table.

Options Tab of the Table Property Editor

Use this tab to organize search results for the Relational Field Value Lookup form in SBM Work Center.

For example, in an Incidents table, you could select the *Company* and *Contact* fields. When a user performs a search, the results are organized according to relationships among incidents, companies, and contacts.

If you do not need these settings for a specific purpose, simply select **(None)**.

If a form contains the **Social Widget** detail control, you must select the **Enable searching for Social Widget** check box for the table. When this option is selected, indexes are created in the database for the *Title* and *Description* system fields. This enables full-text searching in those fields, so users can do an "expert" lookup. (See Detail Controls [page 312] for information about this control.)

> **Note:** Additional configuration may be required to enable full-text searching for the Social Widget. For details, see the Knowledgebase.

> **Note:** Full-text searching can also be enabled in SBM System Administrator for all *Text* fields.

Labels Tab of the Table Property Editor

Use this tab to rename the field section titles, item details labels, and report privilege category labels for the selected table. The changes made to these labels are displayed to your users in the areas related to the table.

For example, if you rename the report privilege categories on this tab for an auxiliary table, the new names are displayed in the Privilege Category drop-down list on the Save Report forms for this table.

The **Default** column shows the text that will be displayed for each item. You can type a replacement in the **Override** column next to any label you want to change.

Element	Description
Sections	The labels to apply to field sections for the table.
Item Details	The labels to apply to the additional sections for the table.
Report Levels	The labels to apply to the listed report privilege categories for the table.

Icons Tab of the Table Property Editor

Use this tab to customize icons for the various field types.

For each field type listed, you can override the default by selecting an icon already loaded into SBM Composer, or by selecting **Add** to load in and select a new icon. The new icon is also added to the **Images** heading in App Explorer, so it is available for selection as needed.

Element	Description
Relational and Multi-Relational Fields	These icons are displayed when the table is used as a *Single Relational* or *Multi-Relational* field in another table. For example, if you create a *Single Relational* field in an `Issues` table that is associated to an `Incidents` table, the icon specified here for the `Incidents` table is displayed when users work with issues.
Active Items and Inactive Items	Not currently used.

Forms Tab of the Table Property Editor

For each form type, click **New** to create a custom form for the selected primary or auxiliary table. Click **Preview** to see a representative rendering of the form as it will appear to users. Click **Edit** to modify the existing form for the table.

> **Note:** For primary tables, you can override the default form at the workflow, state, and transition level.

Element	Description
State form, View form	A read-only view of primary items. For auxiliary items, these are referred to as view forms.
Transition form, Edit form	An editable form used to submit, transition, and update primary items. For auxiliary items, these are referred to as edit forms.
Print form	A printable version of a state or view form.

Field Privileges Tab of the Table Property Editor

Use this tab to modify the order and privileges of fields in an auxiliary table.

Element	Description
List of fields	The fields in this list are divided into sections, such as **Standard** and **Advanced**. • To change the order of fields within a section, drag and drop the fields, or right-click the field and then select **Move up** or **Move down**. • To change the privileges of a field, drag and drop the field between sections, or right-click the field and then select **Move up** or **Move down**.
Fields in the *Section Name* Section	When you select a field in the list of fields, this section displays the privileges for that field. For example, it could indicate that fields in the Advanced section can be changed by the **Administrator** and **Manager** roles, can be read by the **Developer** role, and are not visible to the **User** role.

Dependencies Tab of the Table Property Editor

This tab is available for auxiliary tables only.

Element	Description
Select independent field	Select the field for which you want to define a dependency.
Restrict dependent field... To these values	When a user selects a value in the specified independent field, the dependent field you specify here is automatically limited to the values you specify here.
Change value to	Define a specific dependent field default value for each independent field value, based on the following table.

The following table describes how the different options affect the dependent field when the independent field is changed.

Option	Behavior when the independent field value is changed
<Leave entry unchanged>	The dependent field selection will not change (assuming the previously selected value of the dependent field is also valid for the new value of the independent field).
<First valid entry>	The dependent field will be preset to the item listed immediately after **<None>**.
<None>	The dependent field will be preset to **<None>**.
list of values	The dependent field will be preset to the *value* you select here. This option only applies to *Single Selection* fields that are set as dependent fields.

Chapter 12: Managing Fields

This section contains the following information:

- About Fields [page 177]
- Working with Fields [page 185]
- Field Settings [page 192]
- Field Dependency Tutorials [page 226]

About Fields

SBM uses field data to provide:

- Real-time visibility into your processes through a variety of report types.
- Historical information through a set of powerful Trend reports.
- Audit capabilities through change history records and reports.
- Rule expressions to control item flow, form behavior, and notifications.

Fields are stored in primary and auxiliary tables, but are presented to users on forms. Users enter data into fields when they submit, update, and transition primary items, also referred to as work items.

Fields are also used to gather data for items in auxiliary tables, which typically store related data, such as customer information. To help support your workflow process, you can use *Single Relational* and *Multi-Relational* fields to pull this static data into primary items.

SBM Composer provides many types of fields, most of which have sets of properties that define their behavior. These fields can be customized to meet your data collection requirements.

- Field Types [page 177]
- About Selection Field Values [page 182]
- Field Organization [page 184]

Field Types

Two categories of fields are provided:

- **System Fields**

  All primary and auxiliary tables contain system fields that help gather data needed by the tracking system. Some system fields are required, but others are optional. For a list of system fields for each table type, refer to Chapter 32: Application Field Reference [page 515].

- **Custom Fields**

    SBM provides a diverse set of field types that you can add to the primary and auxiliary tables in your applications.

    The following sections describe the type of custom fields available.

Simple Field Types

The following fields represent simple values stored within database tables. These field types are the most efficient and provide the best performance when used in reports.

Icon	Field Type	Description
	Binary/ Trinary	Stores either two or three values that you specify. Use two values for situations where you need to give users "yes/no" or "on/off" options. Use three values when you need an additional option, such as a value that helps determine subtask status.
	Date/Time	Stores dates and time values. *Date/Time* fields can be displayed as the date only, date and time, time only, or elapsed time. Date/Time values are shown to users in the format specified in their user profile.
	Numeric	Stores signed integers, floating point values, or fixed precision values. *Numeric* fields can be used as weights in Trend reports. When you use weights, be sure to set a default value, or set the field as required. (See Attributes Tab of the Field Property Editor [page 217] for details.)
	Summation	Sums the values of weights of *Single Selection* fields or values for *Numeric* fields set as integers.
	Text	Can be of fixed length up to 255 Unicode characters, Memo type (whose length varies by DBMS), or a Journal type. SBM supports non-ASCII characters in *Text* field values, allowing runtime data to be displayed in a wide variety of international languages. You can enable Rich Text editing for Memo and Journal fields.

User Field Types

The following field types represent users or groups in your systems. Values for these field types can come from roles defined in SBM Composer and from users and groups defined in SBM Application Administrator. For details, refer to Values for User, Multi-User, and Multi-Group Fields [page 182].

User field types are also used to establish ownership for primary items. For details, refer to Ownership of Items [page 131].

Icon	Field Type	Description
	Multi-Group	Enables users to select one or more groups to associate with an item.
	Multi-User	Enables users to select users, groups, or group members as values for the fields. The type of selections available depend on the settings you apply to the field in SBM Composer.
	User	Enables users to select a single user or group member to associate with an item. System *User* fields, such as *Owner* and *Submitter*, are indicated by a red circle in the bottom left corner of the *User* field icon.

Selection Field Types

The following field types enable users to select values you define for the field in SBM Composer. Use this field type for static values that will likely not change over time. For example, use a *Single Selection* field to create a Priority field that may have values of High, Medium, or Low.

If values will change on a regular basis, consider using *Single Relational* or *Multi-Relational* fields instead.

Icon	Field Type	Description
	Multi-Selection	Enables users to select one or more predefined values.
	Single Selection	Enables users to select one predefined value. You can assign weights to each value to assist with reporting, backlog views, and *Summation* field values.

Relational Fields

Single-Relational and *Multi-Relational* fields are used to establish relationships between data in primary and auxiliary tables. These field types are ideal for managing values that will change over time.

For example, you may have a *Products* auxiliary table that stores information about the products your company sells. Each product has its own record in the *Products* table. You can create a *Multi-Relational* field in your process app that points to the *Products* table, enabling users to select the products that relate to a particular work item. As your company adds products, you can easily add records to the *Products* table.

Relational fields are created in SBM Composer, but Relational field values are typically added using the Auxiliary Data feature in SBM Application Administrator.

The following information applies to Relational fields:

- Consider user privileges carefully when you use Relational fields across tables. Users who do not have privileges to view items in a Relational field table may be able to view those items as values in Relational fields in other tables. If you do not want user to see Relational field values, move Relational fields to a section where users do not have "view" privileges, such as a Manager section.

- The value display format controls how *Single Relational* and *Multi-Relational* fields appear as values to users. For details, refer to About the Value Display Format [page 168] and General Tab of the Table Property Editor [page 172].

- These rules apply to value sorting:

  - Relational field values based on primary tables cannot be sorted manually. Values are sorted by project and then Item ID.

  - Relational field values based on auxiliary tables are sorted by the table's value display format. To change the order in which values sort for the Relational field, modify the value display format for the auxiliary table.

- *Sub-Relational* field values are not added to the database, but define a relationship between fields. *Sub-Relational* fields can be useful in certain situations, but extensive use of them may slow the performance of your process app.

- You can create relationships to fields in the Global Process App tables, but you cannot create relationships from tables in the Global Process App.

Icon	Field Type	Description
	Multi-Relational	Enables users to select one or more items from a primary or auxiliary table as values for the field.
	Single Relational	Enables users to select one value to associate with a particular primary or auxiliary table.
	Sub-Relational	Allows access to additional fields in a related primary or auxiliary table. *Sub-Relational* fields are tied to a *Single-Relational* field already in an application. For example, if an *Incidents* table contains a *Single-Relational* field to an *Issues* table, you can include a *Sub-Relational* field of *Owner*. This lets users see the owner of a related issue within the incident. *Sub-Relational* fields are only available if *Single Relational* fields already exist for a particular table.

Folder Fields

Folder fields are used to add links to primary or auxiliary items to a specified folder.

> **Note:** Folders must be created and configured correctly in SBM Work Center before they can be used as selections for *Folder* fields. Specifically:
>
> - The **Allow new items to be added to this folder** check box must be selected.
>
> - For a shared folder or Knowledge Base folder, a role must be assigned to users or groups to allow them to add links to the folder.

Icon	Field Type	Description
	Folder	Values can be private folders, shared folders, or Knowledge Base folders created in SBM Work Center.

File Fields

File fields enable users to add files to an item during a transition. *File* fields provide greater functionality than simple file attachments, including:

- Adherence to field privilege sets instead of attachment privileges

- Flexibility in form placement – they can appear anywhere on forms, not just on the **Attachments** tab

- The ability to specify which file extensions are allowed or disallowed

- The ability to require a file to be added on a transition

- The ability for users to drag and drop files onto the form

Icon	Field Type	Description
	File	Enables users to add one or more files to an item, depending on the configuration. Users may also be able to edit and delete existing files.

URL Fields

URL fields enable users to add URLs to an item during a transition. *URL* fields provide greater functionality than simple URL attachments, including:

- Adherence to field privilege sets instead of attachment privileges

- Flexibility in form placement – they can appear anywhere on forms, not just on the **Attachments** tab

- The ability to require a URL to be added on a transition

Icon	Field Type	Description
🌐	*URL*	Enables users to add one or more URLs to an item, depending on the configuration. Users may also be able to edit and delete existing URLs.

About Selection Field Values

SBM offers several "selection" field types with each type offering flexibility to users as they work with items. *User*, *Single Selection*, *Multi-Selection*, and *Single Relational* fields are examples of selection-type fields.

Values for selection fields are added in different interfaces, depending on the field type. Refer to the online help in each interface for detail information on setting selection and default values for fields.

Add values for the following field types in SBM Composer:

- Single Selection

- Multi-Selection

- Binary/Trinary

Add values for the following field types for primary items in SBM Application Administrator. For auxiliary items, add values for these field types in SBM System Administrator.

- User

- Multi-User

- Multi-Group

- Folder

> **Note:** You can add roles as values for *User*, *Multi-User*, and *Multi-Group* fields in SBM Composer. User and group selections are added in Application Administrator. For guidance, refer to Values for User, Multi-User, and Multi-Group Fields [page 182].

Add values for the following field types in auxiliary tables using the Manage Data feature in SBM Application Administrator:

- Single Relational

- Multi-Relational

Values for User, Multi-User, and Multi-Group Fields

SBM offers several options for adding values to user-type fields (*User*, *Multi-User*, and *Multi-Group* fields). Depending on the field type, roles, groups, and users can be used to populate value lists.

The following options are available for adding values to *User*, *Multi-User*, and *Multi-Group* fields:

- **Roles**

  Associate roles with the fields in SBM Composer, and then assign users or groups to the role in Application Administrator.

- **Groups and Users**

  Add users and groups as values for default fields in workflows in Application Administrator.

The following sections explain the various value options for user-type fields:

Values Derived From User Accounts

You can assign individual users to *User* and *Multi-User* fields in Application Administrator, but this should only be done in cases where one or two individuals will be needed as values for the field. The best practice is to assign roles or groups to these fields. This eases maintenance over time, since the field values will be based on role assignments or group membership rather than by manually editing fields as users change in your system.

Values Derived From Groups

Groups are assigned to user-type fields in Application Administrator. Group values appear as [Members of: *Group Name*] in the **Values** list on the **Attributes** page.

When you add groups as selections, they become available for you to set as default values in Application Administrator, and for users to select as values, as follows:

- **User Fields**

  A list of users assigned to the groups is available.

- **Multi-User**

  The list of values will contain users or users and groups, depending on the selection mode specified in SBM Composer.

- **Multi-Group Fields**

  A list of groups is available.

Current User Value

Current User is available as a default value for *User* and *Multi-User* fields. This automatically sets the value for a field as the user performing an update or transition. For *Multi-User* fields, the user performing the update or transition can be one of the values.

The following considerations apply to the Current User default value:

- Users who are valid Current User selections must be added to the field's selection list. If the user updating or transitioning an item is not included in the selection list, the field is set as required and the user's name is labeled as disabled. The user performing the update or transition must select a valid user selection.

- Take special care when you use the Current User setting for fields that determine ownership. If you select Current User as the default value for a *Multi-User* field used for the secondary ownership property for a state, the user transitioning items to this state must have ownership privileges. If not, the field is set as required and even though the selection looks valid, users are warned that they must pick a user with correct privileges.

Values Derived From Roles

Roles are assigned to user-type fields in SBM Composer. When you edit the field in Application Administrator, these values appear as [Acting As: *Role Name*] in the **Values** list on the **Attributes** page.

> **Note:** You cannot disable role values for user-type fields in Application Administrator. You must remove the role association from the field in SBM Composer.

When you assign users or groups to roles, they become available for you to set as default values in Application Administrator, and for users to select as values, as follows:

- **User and Multi-User Fields**

  A list of user names is available. The list includes individual users assigned to the role in Application Administrator, as well as users who are members of groups assigned to the role.

- **Multi-Group Fields**

  A list of group names is available. Users can select one or more groups.

Field Organization

Each field is assigned to a privilege section. Privilege sections determine which users can view and modify fields on forms. For example, you could set up the *Owner*, *Project*, and *State* fields to be accessible only to users assigned to the Manager role.

On a quick form, fields are gathered automatically according to their privilege sections. On a custom form, privilege sections control accessibility of fields, but you can arrange the fields any way you like. See Quick Forms or Custom Forms? [page 235] for details.

You can rename field sections on the **Labels** tab and **Icons** tab of the table Property Editor. See Labels Tab of the Table Property Editor [page 174] and Icons Tab of the Table Property Editor [page 174] for details.

Each field type is associated with a default privilege section, which is assigned when you drag it onto the table editor. You can change the privilege section using the **Privilege section** drop-down list on the **General** tab of the field Property Editor. Only those privilege sections that are allowed for the field are available in the list. For example, a required system field cannot be associated with the Not Used privilege section.

By default, field definitions (including the privilege section) are inherited by the associated workflows and passed on to sub-workflows. You can override field definitions in these workflows and in any state or transition associated with them. For details, see Field Privileges Tab of the Application Workflow, State, and Transition Property Editor [page 123].

Provided Field Sections

SBM provides several privilege sections. The following list includes system fields for primary and auxiliary tables as examples of how the various privilege sections are typically used.

- **Standard**: This privilege section is the default assignment for the *Item Type*, *Item ID*, and *Title* system fields. All users can see the fields in this section; there are no role privileges for it.

    Tip: On a quick form, the Standard section is at the top of the form, so you could place required fields in this section to help ensure that users provide a value.

- **User**: This privilege section is empty by default. It does not appear on the form until fields are associated with it.

- **Advanced**: This privilege section is the default assignment for the *Close Date*, *Last State Change Date*, and *Submitting Agent* system fields.

- **Manager**: This privilege section is the default assignment for the *Active/Inactive*, *Owner*, *Project*, *Secondary Owner*, and *State* system fields.

    Tip: It is generally advisable to prevent access to the *Owner*, *Secondary Owner*, *Project*, and *State* fields by most users. These fields are generally handled automatically by the system. (There could be situations in which a manager needs to change the values in these fields manually.)

- **System**: This privilege section is the default assignment for the *Company*, *Contact*, *Last Modified Date*, *Last Modifier*, *Last State Changer*, *Submit Date*, and *Submitter* system fields.

- **Not Used**: This privilege section provides a place to store fields that are not used in a particular workflow or auxiliary table. Fields contained in this section are not available to users on any state or transition form.

- **Hidden**: This privilege section provides a place to store fields that you do not want users to view in a particular workflow, state, or transition.

Working with Fields

This section contains the following topics:

Configuring Fields

Adding Fields

To add fields:

1. Select the **Data Design** filter.

2. Select the primary or auxiliary table you want to add fields to.

3. Drag fields from the Table Palette to the Table Editor. You can choose from unused system fields or custom fields.

4. Provide information on the **General** tab of the Field Property Editor.

5. On the **Options** and **Attributes** tab, configure settings, which in many cases are specific to the field type.

6. Use the **Forms** tab to add the field to forms in your process app.

For guidance on individual settings and options, click inside a tab, and then press F1 to open the online help.

Modifying Fields

All system and custom fields can be modified at these levels:

- Table

- Workflow

- State

- Transition

The level at which you edit the field determines which field properties you can modify.

At the table level, you can modify general field information, options, and attributes. For some fields, you can enter information about field dependencies and enter selection values. You can also set field permissions and field order. These fields become the default fields you use in your workflows and forms.

Changes made at the workflow, state, and transition level are considered overrides. For details, refer to Chapter 18: Understanding Inheritance and Overrides [page 377].

Deleting and Restoring Fields

SBM Composer enables you to delete all custom fields from the primary table and auxiliary tables of published process apps.

To delete a field, right-click the field in the table editor and then select **Delete**. When you delete a field, it appears dimmed in the table editor, but is not removed.

> **Important:** Required system fields cannot be deleted.

To hide deleted fields in the table editor, right-click any field and then deselect the **Deleted Fields** menu option. This setting is applied globally. To revert back to showing deleted fields, reselect **Deleted Fields**.

You can restore a deleted field. When you do so, its properties are also restored. For example, when you restore a *Multi-Selection* field, all of it selection values are restored. To restore a field, right-click the field and then select **Restore**.

Configuring Mass Updates

Mass update enables users to make the same change to multiple items at the same time. For example, users can reopen multiple closed items at the same time.

Depending on their privileges, users can use mass update to simultaneously:

- Transition, update, or delete multiple primary items.

- Update or delete multiple auxiliary items.

- Modify values for specific fields.

In SBM Work Center, users can perform mass updates from Listing reports, activity views, and backlog views.

> **Note:** The Editable Grid also allows users to update multiple items at the same time, but if differs from mass update because users can make a variety of changes to information in multiple items at the same time.

Considerations for Using Mass Update

Consider this information as you enable Mass Update:

- Post Item or Create Subtask transitions are not available for mass updates.

- Transitions that are set to require attachments are not available for mass updates.

- When you select the **Allow mass updates** check box for dependent fields, be sure to do the same for all independent fields that drive the dependent field's values. If you don't do this, users will not have any options to select when mass updating items.

- Custom forms and any features they include are not available for mass updates. For example, AE Plug-In JavaScript API functions on a custom form cannot be used for mass updates. This includes field validation, auto-filled fields, information messages, and filters on *Multi-Relational*, *Single Relational*, *Sub-Relational*, *Multi-Selection*, and *Single Selection* fields.

- If your mass transition executes a script or Web transition in the pre-transition context, the script and Web service appears to execute twice against each item in the list; however, the first execution is performed against a temporary record and the actual database record for each item is only affected once. This behavior is by design to prevent a potential over-consumption of memory.

  Note that if the script or Web service affects other items that are not in the mass transition list (or if they affect external systems or objects that are not in the SBM database), the script or Web service could affect those objects twice. Therefore, it is best to avoid calling pre-transition scripts or Web services in a mass transition, unless you use the MassTransPreForm Shell property to prevent this from happening.

Enabling Mass Updates

To enable mass updates for fields:

1. Select the **Data Design** filter, and then select the table that contains the fields that you want users to change during a mass update.

2. Select the **Attributes** tab.

3. Select the **Allow mass update** check box.

> **Note:** You can override this setting in SBM Application Administrator for default fields in projects and for transition fields.

4. Grant the *Mass Update Items* privilege through roles in SBM Composer or to users and groups in SBM Application Administrator.

Importing Selection Field Values

Single Selection and *Multi-Selection* fields present values defined in SBM Composer to end users as they work with items. To ease the process of managing multiple selection values, use the import and export features to quickly add new selections and update existing selections at one time.

The import and export options are located on the **More** list in the **Values** area for *Single Selection* and *Multi-Selection* fields. For import steps, refer to .

Import/Export Format

Selection values are imported from and exported as comma-separate values (.csv). You can create a .csv file that uses the format described in this section, or you can export values from an existing selection field that has at least one selection value, and then modify the content to meet your needs.

Once you have a .csv file, you can open it in Microsoft Excel for simpler editing of values.

The following example shows comma-separate columns for values exported from a *Single Selection* field.

```
1   {53e41e63-255d-410e-9fee-2781ce043ae1},"1: Required",True,100
2   {2b15e0cd-1e0e-4e37-91a1-fc954b1cd086},"2: Desirable",True,80
3   {369e10bc-6668-46f3-bd20-8d3bfe75a6cc},"3: Voluntary",True,50
4   {25c159f7-6640-4634-861e-d9ea9234916d},"4: Incidental",True,40
5   {ce31f75f-6dad-4ba7-b4d7-c2d756ad47b2},"5: Hold",True,10
```

Each line represents a selection value. Each column in a line is described in the table that follows.

CSV Column	Required?	Export Information	Import Information
UUID (Unique Identifier)	No, but review description.	Provided on export so that you can update values in the file and then reimport them into the same field.	If a selection value with the specified UUID exists, the value is updated with the name, status and weight in the .csv file. If a UUID is not provided or a match is not found, a new selection is added. If included during an import, UUID information must be listed first in each line.
Value Name	Yes	Exported values are surrounded by double quotation marks.	If duplicate name values are found, a number is added to the end of the name and incremented for each duplicate value found. For example, multiple "Priority - High" values would be imported as "Priority - High 1," Priority High - 2," etc. Values do not need quotation marks to be imported unless the value contains embedded quotation marks, which must be surrounded by double quotations marks. For example, Priority - "High" needs to be imported as "Priority - ""High""". Commas included inside of quotation marks are not treated as separators but are included as part of the value.
Enabled Status	No	"True" indicates enabled status; "False" indicates disable status.	If a line does not contain a Status boolean value, it is set to True and the selection is enabled.
Weight (Single Selection fields only)	No		If a weight is not specified, the default weight of 100 is used.

For details on options for selection fields, refer to:

- Single Selection Field Options [page 209]

- Multi-Selection Field Options [page 201]

Import Steps

You can use the import and export features to easily manage values for *Single Selection* and *Multi-Selection* fields.

Use these steps to quickly add new selection values or to edit multiple existing values at the same time.

To import selection field values:

1. In the App Explorer, select **Data Design**, and then select the table that contains a *Single Selection* and *Multi-Selection* field.

2. Select the field, select the **Options** tab, and then do one of the following:

   - If the field already has at least one value, select **Export** from the **More** list, and then export the existing values to a file. Edit the file as needed, using the information in Importing Selection Field Values [page 188] for guidance.

   - If the field does not have any values, create a spreadsheet following the criteria in Import/Export Format [page 188].

3. Select **Import** from the **More** list, and then navigate to your data file.

   Tip: If you are not satisfied with the results, use the Undo feature to return the value list to its original state.

Configuring Field Dependencies

Field dependencies enable you to filter the selection values available for one field based on the value selected in another field.

For example, you can create a *Single Selection* field called "Product" that has a selection for each product your company develops and a *Single Selection* field called Version that has a selection for each product version. You can set up dependencies for the *Products* field that tailor the selections available in the *Versions* field. When a user selects a product from the *Product* field, only applicable versions are available in the *Versions* field. In this case, *Products* is the independent field, and *Versions* is the dependent field. Learn how to configure this dependency in Single Selection Field Dependency Tutorial [page 227].

You can set up a similar dependency using a *Single Relational* field. For example, create two auxiliary tables: Products and Versions. You can then establish a relationship between the two tables that allows specific version records in the Versions table to be available for each product in the Products table. For example, if Version 1 and Version 2 apply only to Product A, those are the only available selections when Product A is selected in the *Product* field. Learn how to configure this dependency in Relational Field Dependencies Tutorial [page 230].

You can also define a dependency that controls which users are available in one field depending on the user selected in a different field. For example, you can limit the selections available in an *Employee* field depending on the selection made in the *Manager* field. For example, if a user selects Kathy Manager from the Manager field, Chris Tester and Hans Tester are available selections for the Employee field. Learn how to configure this dependency in User Field Dependency Tutorial [page 228].

The following table lists the available independent field types and the field types that allow dependencies.

Independent field type	Allowable dependent field types
Single Selection	Single Selection, Multi-Selection, Multi-Group, Multi-User, User
Single Relational	Single Relational, Multi-Relational
User	Single Selection, Multi-Selection, Multi-Group, Multi-User, User

Note: *Multi-Selection*, *Multi-Group*, *Multi-User*, and *Multi-Relational* fields can be used as dependent fields, but not as independent fields. Also, field dependencies do not work for *Multi-Selection*, *Multi-Group*, *Multi-User*, and *Multi-Relational* fields that are displayed as check boxes on forms.

Configuring Advanced Search

Advanced Search enables you to customize the fields that are available on the Advanced Search form for primary and auxiliary tables, including archived primary tables. Keyword searching is provided for all search types; project searching is available for primary item and archived searches.

Users can perform advanced searches for the following items, depending on their privileges:

- Primary items in the selected application and items in auxiliary tables.

- Archived primary items for the selected application.

- Notes and attachments in primary and auxiliary tables.

Considerations for Advanced Search

Consider this information as you specify fields for Advanced Search:

- Fields you select for Advanced Search are visible by all users who can use this feature.

- The **Keyword(s)** and **Project(s)** options are always available on the Advanced Search page.

- The field order for the Advanced Search page is determined by the default field order of the first project in the table in the project hierarchy. You can change the field order in SBM Application Administrator.

- The project selected by your users determines which values are available for selection field types, such as *User* fields, but project selection has no effect on Relational fields.

- To improve performance, you should limit the number of selection fields you include on the Advanced Search page.

- To improve performance, limit the number of *Text* fields you allow for keyword searching. You can add individual *Text* fields to the Advanced Search page, but they are treated differently than keyword searching. Items containing the search criteria are returned, but users must search for exact phrases.

  Instead, consider including *Text* fields in keyword searches. See Text Field Options [page 212] for information about the **Include field in keyword searches** option.

Adding Fields to Advanced Search

To add fields to the Advanced Search page:

1. Select the **Data Design** filter, and then select the table that contains the fields you want to add to the Advanced Search form.

2. Select the **Options** tab.

3. Select the **Appears on searches for this table** check box.

> **Note:** When this option is selected on *User*, *Multi-User*, *Single Selection*, *Multi-Selection*, *Single Relational*, or *Multi-Relational* fields, the fields appear as facets on the Work Center search results page for a single application. This option also adds the field to the Work Center search index, which enables users to search for work items using values in this field.

Field Settings

The following sections provide descriptions of settings used to configure fields. Sections are organized by property editors.

Field Property Editor

The field Property Editor enables you to enter information and select options for the field selected in the table editor. You can use the field Property Editor to specify information for fields in both primary and auxiliary tables.

> **Tip:** When you select a field in the table editor, the field type you are working on is shown in the **Fields** drop-down list in the field Property Editor. Selecting a different field from the drop-down list selects it in the table editor and displays field properties in the field Property Editor.

Controls used for creating field settings are grouped in tabs located on the left side of the field Property Editor.

General Tab of the Field Property Editor

Element	Description
Name	This is the name as it will appear to users.

Element	Description
Database field name	The unique name by which the field is stored in the database.
	The database field name can contain only ASCII alphanumeric characters (A–Z, a–z, 0–9) and underscores. Any other characters are ignored as you type them.
	You cannot change the database field name after the application containing the table is published to the repository. You also cannot change the automatically specified database name of a system field.
Type	The field type, such as *Numeric* and *Text*. The field type cannot be changed. If the field is a system field, **[System]** is appended to the field type.
End-user help text	Click **Edit** to open an HTML editor and add optional comments or notes about the purpose of the field. The text you enter is displayed to users.
	For details, refer to Chapter 24: Providing Custom End-user Help [page 443].
Privilege section	See Field Organization [page 184] for information about privilege sections in quick forms and custom forms.

Options Tab of the Field Property Editor

Use this tab to specify the appearance and behavior of fields. Each field type has different options, which are described in the topics in this section. For a description of each field type, see Field Types [page 177].

Note: For system fields, some options are unavailable, and others cannot be changed.

- Binary/Trinary Field Options [page 194]

- Date/Time Field Options [page 195]

- File Field Options [page 197]

- Folder Field Options [page 197]

- Multi-Group Field Options [page 198]

- Multi-Relational Field Options [page 200]

- Multi-Selection Field Options [page 201]

- Multi-User Field Options [page 203]

- Numeric Field Options [page 205]

- Single Relational Field Options [page 207]

Binary/Trinary Field Options

Style

- **Dropdown list**

  Select this option to let users select a single value from a drop-down list populated with the values from the **First Value**, **Second Value**, and (for trinary fields) **Third Value** properties.

- **Radio buttons**

  Users select one of radio buttons labeled with the values from the **First Value**, **Second Value**, and **Third Value** properties.

- **Check box**
  Users select or clear a check box labeled with the field name specified on the **General** tab. This option is available only for binary fields.

  > **Important:** Internally, SBM treats the first value as 0, which corresponds to "selected," and treats the second value as 1, which corresponds to "not selected."

- **Third value**

  Select this option to create a trinary field, adding a third value to the drop-down list or defining a third radio button. This option is not available when the **Checkbox** option is selected.

- **First**, **Second**, and **Third value**

  If the field is used to determine subtask status, the values you specify must be related to the three applicable subtask statuses: **In Review/Progress**, **Accepted/Complete**, and **Rejected/Reverted**. See Tutorial: Defining Subtask-Driven Actions [page 423] for related information.

  These properties are not available when the **Checkbox** option is selected.

Display

- **Span entire row on forms**

  Select this option to make this field the only field to appear on a single row.

Search & Query

- **Appears in report field lists**

Select this option to include the field in lists on report forms. If you clear this option after the field is used in reports, the changed setting is ignored for those reports and the field is still shown until it is removed from the report definition.

- **Appears on lookup form and relational field value lookup**

  Select this option to add the field to the Auxiliary Data search form and Relational Field Value Lookup form.

  The field order for the Relational Field Value Lookup form for primary items is determined by the default field order of the table's first project in the project hierarchy. Projects are defined in Application Administrator.

- **Appears on searches for this table**

  Select this option to specify that the field will be available for selection on the Advanced Search page. This setting applies to primary and auxiliary tables. See Configuring Advanced Search [page 191] for additional information.

Date/Time Field Options

Style

Select one of the display options for the field. Note that date/time values are always stored in native date/time format in the database. Values for *Date/Time* fields set to record time only or elapsed time are stored as integers. Assuming a default value of "Now", the value stored in the database is equal to the number of seconds that have passed since 12:00:00 a.m. in the submitter's or modifier's time zone.

- **Date and time**

  Used for display of date-and-time values.

- **Date only**

  Used for display of date values.

- **Time of day**

  Used for display of time values.

- **Elapsed time**

  Stores a value representing an elapsed time in hours, minutes, and seconds. You can choose whether to include seconds in the display value, however. Users can provide an elapsed time in the format specified on the form (`d hh:mm:ss`, `d hh:mm`, `hh:mm:ss`, or `hh:mm`) or type an integer to represent the number of hours that should appear in the field. When you select this option, the **Stopwatch**, **Calculate days**, and **Show seconds** options become available.

  - The **Stopwatch** option lets users record elapsed time. The stopwatch starts recording time when a transition form opens for submitting, transitioning, or updating an item. Users can edit the elapsed time by typing a new value in the field when the time is paused on the transition form.

    The timer does not record elapsed time if it is moved to the **Hidden Fields** section. The stopwatch records time only when a form is open, so hiding the *Stopwatch* field in a state does not affect elapsed time. You must hide the *Stopwatch* field in the specific transition if you do not want it to record time. If the *Stopwatch* field is read-only or is placed in a field section the user does not

have privileges or preferences to view, the timer still records elapsed time while the form is open. If the user cancels the transition or resets the form, the elapsed time is not saved.

- Select the **Calculate days** option to calculate and display elapsed time in the format `d hh:mm:ss`. For example, if a user enters the integer `50` in the field, it is interpreted as hours and converted to days, and the resulting value is displayed as 2 2:00:00 (if the **Show seconds** option is also enabled). Note that the elapsed days do not appear until at least one full day has elapsed.

- Select the **Show seconds** option to display the elapsed time values in `00:00:00` format. (Disable the **Show seconds** option to display elapsed times in `00:00` format.) Note that seconds are retained in the database, even if they are not displayed.

  If the **Stopwatch** option is also enabled, seconds are always displayed on transition forms. This behavior indicates to users that the stopwatch has started. Otherwise, they would have to wait for the minute display to change to know that the stopwatch is running.

  Important: You cannot change the style of a *Date/Time* field after a process app is published. For example, you cannot change a date only field to a time of day field. However, there is a way to reset the field and change it if you intend to deploy the process app to a different environment. For more information, see Modifying Locked Elements in a Published Process App [page 64].

Display

- **Span entire row on forms**

  Select this option to make this field the only field to appear on a single row.

Search & Query

- **Appears in report field lists**

  Select this option to include the field in lists on report forms. If you clear this option after the field is used in reports, the changed setting is ignored for those reports and the field is still shown until it is removed from the report definition.

- **Appears on lookup form and relational field value lookup**

  Select this option to add the field to the Auxiliary Data search form in SBM Application Administrator and the Relational Field Value Lookup form. For a *Date/Time* field, this option adds Start and End boxes next to the field on the lookup form, letting users specify a start and end date as part of their search criteria.

  The field order for the Relational Field Value Lookup form for primary items is determined by the default field order of the table's first project in the project hierarchy. Projects are defined in SBM Application Administrator.

- **Appears on searches for this table**

  Select this option to specify that the field will be available for selection on the Advanced Search page. This setting applies to primary and auxiliary tables. See Configuring Advanced Search [page 191] for additional information.

File Field Options

Style

- **Allow multiple files**

  Select this check box to enable users to add up to 100 files to a field.

Restriction

- **Allow these extensions** or **Disallow these extensions**
  Select **Allow these extensions** to limit the files that can be added to the specified extensions. Select **Disallow these extensions** to prevent files from being added with the specified extensions. Use a comma-separated list. For example:

  - **Allow these extensions:** `png, jpg`

  - **Disallow these extensions:** `exe, dll, js`

Display

- **Span entire row on forms**

  Select this option to make this field the only field to appear on a single row.

Search & Query

- **Appears in report field lists**

  Select this option to include the field in lists on report forms. If you clear this option after the field is used in reports, the changed setting is ignored for those reports and the field is still shown until it is removed from the report definition.

- **Include field in keyword searches**

  Select this option to let users search for particular text from keyword searches. The keyword search will search the file label, filename, and file content.

  > **Note:** You must select this option and deploy your process app in order for Work Center users to find items by searching the *File* field.

- **Appears on searches for this table**

  Select this option to specify that the field will be available for selection on the Advanced Search page. This setting applies to primary and auxiliary tables. See Configuring Advanced Search [page 191] for additional information.

Folder Field Options

Style

- **Allow searching**

  Select this option to enable the value find feature for the field on submit, transition, and update forms. This enables users to search for values.

  > **Tip:** If the field will contain 200 or more selections, this option is recommended to ensure best performance. If you do not allow searching, fields that contain over 250 selection values are automatically set as searchable in the Editable Grid.

- **Single drop-down list**

Select this option to let users select a value from a populated drop-down list.

Display

- **Span entire row on forms**

Select this option to make this field the only field to appear on a single row.

Search & Query

- **Appears in report field lists**

Select this option to include the field in lists on report forms. If you clear this option after the field is used in reports, the changed setting is ignored for those reports and the field is still shown until it is removed from the report definition.

- **Appears on lookup form and relational field value lookup**

Select this option to add the field to the Auxiliary Data search form and Relational Field Value Lookup form.

The field order for the Relational Field Value Lookup form for primary items is determined by the default field order of the table's first project in the project hierarchy. Projects are defined in Application Administrator.

- **Appears on searches for this table**

Select this option to specify that the field will be available for selection on the Advanced Search page. This setting applies to primary and auxiliary tables. See Configuring Advanced Search [page 191] for additional information.

- **On lookup or query-at-runtime forms**

  - **Allow searching**

  This option enables the value find and relational field value lookup for the field. (See *Allow Searching* under *Style* above for a description of the *value find* feature.) The relational field value lookup feature provides an advanced searching mechanism that lets users find values for *Single Relational* and *Multi-Relational* fields. If this option is enabled, an additional search icon is available for the field.

  - **Show full list**

  This option lets users select a value from a drop-down list on the Relational Field Value Lookup form and Query-at-Runtime form for reports.

Multi-Group Field Options

Style

- **Allow searching**

Select this option to enable the value find feature for the field on submit, transition, and update forms. This enables users to search for values.

> **Tip:** If the field will contain 200 or more selections, this option is recommended to ensure best performance. If you do not allow searching, fields that contain over 250 selection values are automatically set as searchable in the Editable Grid.

- **List display size**

  Specify the number of items that should be visible, without scrolling, in the list box for this field. This should typically be the number of expected values. If the number is too small, scroll bars are added to the list box, and with the **List box** option, dual list boxes are also presented. This option is unavailable when the **Checkboxes** option is selected.

  > **Note:** This option is not used in SBM Work Center, where all list boxes are displayed side by side on submit, transition, and update forms.

- **List box**

  Select this option to let users select one or more values from a populated drop-down list.

- **Checkboxes**

  Select this option to let users select one or more check boxes.

Associated Roles

Select the roles that determine which values users can select. You must assign at least one role to a *User* or *Multi-User* field before you can set a default value for the field in SBM Application Administrator. For *User* fields, users can select a single value. For *Multi-User* and *Multi-Groups* fields, multiple values can be selected.

Display

- **Span entire row on forms**

  Select this option to make this field the only field to appear on a single row.

Search & Query

- **Appears in report field lists**

  Select this option to include the field in lists on report forms. If you clear this option after the field is used in reports, the changed setting is ignored for those reports and the field is still shown until it is removed from the report definition.

- **Appears on lookup form and relational field value lookup**

  Select this option to add the field to the Auxiliary Data search form and Relational Field Value Lookup form.

  The field order for the Relational Field Value Lookup form for primary items is determined by the default field order of the table's first project in the project hierarchy. Projects are defined in Application Administrator.

- **Appears on searches for this table**

  Select this option to specify that the field will be available for selection on the Advanced Search page. This setting applies to primary and auxiliary tables. See Configuring Advanced Search [page 191] for additional information.

- **On lookup or query-at-runtime forms**

  - **Allow searching**

This option enables the value find and relational field value lookup for the field. (See *Allow Searching* under *Style* above for a description of the *value find* feature.) The relational field value lookup feature provides an advanced searching mechanism that lets users find values for *Single Relational* and *Multi-Relational* fields. If this option is enabled, an additional search icon is available for the field.

▪ **Show full list**

This option lets users select a value from a drop-down list on the Relational Field Value Lookup form and Query-at-Runtime form for reports.

Multi-Relational Field Options

Style

- **Allow searching**

Select this option to enable the value find feature for the field on submit, transition, and update forms. This enables users to search for values.

> **Tip:** If the field will contain 200 or more selections, this option is recommended to ensure best performance. If you do not allow searching, fields that contain over 250 selection values are automatically set as searchable in the Editable Grid.

- **List display size**

Specify the number of items that should be visible, without scrolling, in the list box for this field. This should typically be the number of expected values. If the number is too small, scroll bars are added to the list box, and with the **List box** option, dual list boxes are also presented. This option is unavailable when the **Checkboxes** option is selected.

> **Note:** This option is not used in SBM Work Center, where all list boxes are displayed side by side on submit, transition, and update forms.

- **List box**

Select this option to let users select one or more values from a drop-down list populated with selections added to the *Relational* field table.

- **Checkboxes**

Select this option to let users select one or more check boxes.

Values

> **Note:** If these controls show "(External Application)" and "(External Table)," changing them could break relationships among applications. SBM Composer will prompt you to confirm the change.

- **Application**

Select the application containing the table of values with which you want to populate this field.

- **Table**

Select the table containing the values with which you want to populate this field.

Display

- **Span entire row on forms**

  Select this option to make this field the only field to appear on a single row.

Search & Query

- **Appears in report field lists**

  Select this option to include the field in lists on report forms. If you clear this option after the field is used in reports, the changed setting is ignored for those reports and the field is still shown until it is removed from the report definition.

- **Appears on lookup form and relational field value lookup**

  Select this option to add the field to the Auxiliary Data search form and Relational Field Value Lookup form.

  The field order for the Relational Field Value Lookup form for primary items is determined by the default field order of the table's first project in the project hierarchy. Projects are defined in Application Administrator.

- **Appears on searches for this table**
  Select this option to specify that the field will be available for selection on the Advanced Search page. This setting applies to primary and auxiliary tables. See Configuring Advanced Search [page 191] for additional information.

  > **Note:** When this option is selected, the field appears as a facet on the Work Center search results page for a single application. This option also adds the field to the Work Center search index, which enables users to search for work items using values in this field.

- **Inactive items appear in selection lists**

  Select this option to specify that inactive items will be included in lists on transition forms and report forms.

- **On lookup or query-at-runtime forms**

  - **Allow searching**

    This option enables the value find and relational field value lookup for the field. (See *Allow Searching* under *Style* above for a description of the *value find* feature.) The relational field value lookup feature provides an advanced searching mechanism that lets users find values for *Single Relational* and *Multi-Relational* fields. If this option is enabled, an additional search icon is available for the field.

  - **Show full list**

    This option lets users select a value from a drop-down list on the Relational Field Value Lookup form and Query-at-Runtime form for reports.

Multi-Selection Field Options

Style

- **Allow searching**

Select this option to enable the value find feature for the field on submit, transition, and update forms. This enables users to search for values.

> **Tip:** If the field will contain 200 or more selections, this option is recommended to ensure best performance. If you do not allow searching, fields that contain over 250 selection values are automatically set as searchable in the Editable Grid.

- **List display size**

Specify the number of items that should be visible, without scrolling, in the list box for this field. This should typically be the number of expected values. If the number is too small, scroll bars are added to the list box, and with the **List box** option, dual list boxes are also presented. This option is unavailable when the **Checkboxes** option is selected.

> **Note:** This option is not used in SBM Work Center, where all list boxes are displayed side by side on submit, transition, and update forms.

- **List box**

Select this option to let users select one or more values from a populated drop-down list.

- **Checkboxes**

Select this option to let users select one or more check boxes.

Values

- To modify existing values, click within each row.

> **Tip:** You can use shortcut keys to edit and navigate values. For more information, see Selection Field Shortcut Keys [page 509].

- To manually reorder selections in the list, use the **Move up** and **Move down** controls or click the **Value** column heading and then verify that "User defined" is selected in the **End-user display order** list. You can also choose one of the following display orders:

  ▪ Value ascending

  ▪ Value descending

- For fields in primary tables, you can include or exclude a value by selecting the **Enabled** check box, or you can select **Enable all** or **Disable all** from the **More** list to include or exclude all values.

- To manage a large number of selections, use the **Import** and **Export** options in the **More** list. For details, refer to Importing Selection Field Values [page 188].

- To delete a value, select the value and click **Delete**. In the dialog box that appears, if you want to delete the value even if it is in use, uncheck the **Prevent the deletion if the value is in use** option. All references to the value are deleted as well.

  If you delete this field and later restore it, the values you specify are also restored. For more information, see Deleting and Restoring Fields [page 186].

Display

- **Span entire row on forms**

Select this option to make this field the only field to appear on a single row.

Search & Query

- **Appears in report field lists**

Select this option to include the field in lists on report forms. If you clear this option after the field is used in reports, the changed setting is ignored for those reports and the field is still shown until it is removed from the report definition.

- **Appears on lookup form and relational field value lookup**

Select this option to add the field to the Auxiliary Data search form and Relational Field Value Lookup form.

The field order for the Relational Field Value Lookup form for primary items is determined by the default field order of the table's first project in the project hierarchy. Projects are defined in Application Administrator.

- **Appears on searches for this table**
Select this option to specify that the field will be available for selection on the Advanced Search page. This setting applies to primary and auxiliary tables. See Configuring Advanced Search [page 191] for additional information.

> **Note:** When this option is selected, the field appears as a facet on the Work Center search results page for a single application. This option also adds the field to the Work Center search index, which enables users to search for work items using values in this field.

- **On lookup or query-at-runtime forms**

  - **Allow searching**

    This option enables the value find and relational field value lookup for the field. (See *Allow Searching* under *Style* above for a description of the *value find* feature.) The relational field value lookup feature provides an advanced searching mechanism that lets users find values for *Single Relational* and *Multi-Relational* fields. If this option is enabled, an additional search icon is available for the field.

  - **Show full list**

    This option lets users select a value from a drop-down list on the Relational Field Value Lookup form and Query-at-Runtime form for reports.

Multi-User Field Options

Style

- **Allow searching**

Select this option to enable the value find feature for the field on submit, transition, and update forms. This enables users to search for values.

> **Tip:** If the field will contain 200 or more selections, this option is recommended to ensure best performance. If you do not allow searching, fields that contain over 250 selection values are automatically set as searchable in the Editable Grid.

- **List display size**

  Specify the number of items that should be visible, without scrolling, in the list box for this field. This should typically be the number of expected values. If the number is too small, scroll bars are added to the list box, and with the **List box** option, dual list boxes are also presented. This option is unavailable when the **Checkboxes** option is selected.

  > **Note:** This option is not used in SBM Work Center, where all list boxes are displayed side by side on submit, transition, and update forms.

- **List box**

  Select this option to let users select one or more values from a populated drop-down list.

- **Checkboxes**

  Select this option to let users select one or more check boxes.

Associated Roles

Select the roles that determine which values users can select. You must assign at least one role to a *User* or *Multi-User* field before you can set a default value for the field in SBM Application Administrator. For *User* fields, users can select a single value. For *Multi-User* and *Multi-Groups* fields, multiple values can be selected.

- **Selection mode**

  Use these options to control the list the user and group values that are available to users. Along with this setting, possible values are determined by the roles assigned to the field in SBM Composer and the users and groups assigned to the field in Application Administrator.

  - **Individual users**

    Select this option to provide users as values.

  - **Groups & users**

    Select this option to provide groups and individual users as values. Group selections encompass all members of the group.

  - **Group members & users**

    Select this option to provide users, groups, and members of each group as values.

    If you have HTML5 features enabled, users can click group names on State forms and see a list of group members and user profile cards for each member. This feature is not available for users with External or Occasional User product access.

  For best performance, select the **Groups & users** option. Also, be aware that if you change the selection mode for a *Multi-User* or *Secondary Owner* field back to individual users or group members after populating groups in either field, performance may be impacted when the groups are unrolled to display large lists of individual users.

Display

- **Span entire row on forms**

  Select this option to make this field the only field to appear on a single row.

Search & Query

- **Appears in report field lists**

  Select this option to include the field in lists on report forms. If you clear this option after the field is used in reports, the changed setting is ignored for those reports and the field is still shown until it is removed from the report definition.

- **Appears on lookup form and relational field value lookup**

  Select this option to add the field to the Auxiliary Data search form and Relational Field Value Lookup form.

  The field order for the Relational Field Value Lookup form for primary items is determined by the default field order of the table's first project in the project hierarchy. Projects are defined in Application Administrator.

- **Appears on searches for this table**
  Select this option to specify that the field will be available for selection on the Advanced Search page. This setting applies to primary and auxiliary tables. See Configuring Advanced Search [page 191] for additional information.

  > **Note:** When this option is selected, the field appears as a facet on the Work Center search results page for a single application. This option also adds the field to the Work Center search index, which enables users to search for work items using values in this field.

- **On lookup or query-at-runtime forms**

  ▪ **Allow searching**

    This option enables the value find and relational field value lookup for the field. (See *Allow Searching* under *Style* above for a description of the *value find* feature.) The relational field value lookup feature provides an advanced searching mechanism that lets users find values for *Single Relational* and *Multi-Relational* fields. If this option is enabled, an additional search icon is available for the field.

  ▪ **Show full list**

    This option lets users select a value from a drop-down list on the Relational Field Value Lookup form and Query-at-Runtime form for reports.

Numeric Field Options

Style

Select the format you want displayed as values. Note the following:

- Some large integer and floating point values are reserved for internal use. Integer values larger than 2147483643 and floating point numbers between 4294967292.000 and 4294967295.99999 cannot be used in *Numeric* fields for this reason.

- You cannot change the style of a *Numeric* field after a process app is published. For example, you cannot change an integer field to a floating point field. However, there is a way to reset the field and change it if you intend to deploy the process app to a different environment. For more information, see Modifying Locked Elements in a Published Process App [page 64].

- SBM respects the regional option set for your computer for floating point and fixed precision values. For example, if your regional option is "Czech," a comma is used instead of a decimal point.

- **Integer**

  The maximum positive integer accepted is 2147483643, and the minimum negative integer accepted is −2147483648.

- **Floating point**

  The maximum and minimum accepted range of values are determined by your server hardware.

- **Fixed precision**

  The maximum and minimum accepted range of values are determined by your server hardware.

- **Digits displayed after the decimal point**

  Specify the number of digits (with a minimum of 0 and a maximum of 15) you want to appear after the decimal point. This option is available only when the **Fixed precision** option is enabled.

Display

- **Prefix** and **Suffix**
  Type any plain text and HTML tags you want to display before and after the field's values. Note the following:

  ▪ The rendered HTML is visible in preview mode and at runtime, but not in the form editor.

  ▪ HTML formatting applies to the prefix and suffix for the field label, but not to the value of the field.

  ▪ Use the `<` and `>` character entities to display the less than (<) or greater than (>) signs, respectively. Otherwise, these signs will be interpreted as HTML code. Use the ` ` character entity for each extra space you want to display.

  ▪ By default, there is no space between the field value and its prefix or suffix. If you want a space to appear between the prefix or suffix and the field value, then you must add spaces.

    Note: For best results, formatting changes should be applied using styles on custom forms rather than by adding HTML tags to the field's prefix and suffix. For more advanced HTML formatting, create a custom form for your workflow, and then add an HTML/JavaScript widget to the form.

- **Show thousands separator**

Select this option to display values with commas (such as "1,000" for "1000").

> **Note:** SBM Composer respects the regional option set for your computer. For example, if your regional option is "Czech," a space is used instead of a comma.

- **Span entire row on forms**

Select this option to make this field the only field to appear on a single row.

Search & Query

- **Appears in report field lists**

Select this option to include the field in lists on report forms. If you clear this option after the field is used in reports, the changed setting is ignored for those reports and the field is still shown until it is removed from the report definition.

- **Appears on lookup form and relational field value lookup**

Select this option to add the field to the Auxiliary Data search form and Relational Field Value Lookup form.

The field order for the Relational Field Value Lookup form for primary items is determined by the default field order of the table's first project in the project hierarchy. Projects are defined in Application Administrator.

- **Appears on searches for this table**

Select this option to specify that the field will be available for selection on the Advanced Search page. This setting applies to primary and auxiliary tables. See Configuring Advanced Search [page 191] for additional information.

Single Relational Field Options

Style

- **Allow searching**

Select this option to enable the value find feature for the field on submit, transition, and update forms. This enables users to search for values.

> **Tip:** If the field will contain 200 or more selections, this option is recommended to ensure best performance. If you do not allow searching, fields that contain over 250 selection values are automatically set as searchable in the Editable Grid.

- **Single drop-down list**

Select this option to let users select a value from a populated drop-down list.

Values

> **Note:** If these controls show "(External Application)" and "(External Table)," changing them could break relationships among applications. SBM Composer will prompt you to confirm the change.

- **Application**

Select the application containing the table of values with which you want to populate this field.

- **Table**

  Select the table containing the values with which you want to populate this field.

Display

- **Span entire row on forms**

  Select this option to make this field the only field to appear on a single row.

Search & Query

- **Appears in report field lists**

  Select this option to include the field in lists on report forms. If you clear this option after the field is used in reports, the changed setting is ignored for those reports and the field is still shown until it is removed from the report definition.

- **Appears on lookup form and relational field value lookup**

  Select this option to add the field to the Auxiliary Data search form and Relational Field Value Lookup form.

  The field order for the Relational Field Value Lookup form for primary items is determined by the default field order of the table's first project in the project hierarchy. Projects are defined in Application Administrator.

- **Appears on searches for this table**
  Select this option to specify that the field will be available for selection on the Advanced Search page. This setting applies to primary and auxiliary tables. See Configuring Advanced Search [page 191] for additional information.

  > **Note:** When this option is selected, the field appears as a facet on the Work Center search results page for a single application. This option also adds the field to the Work Center search index, which enables users to search for work items using values in this field.

- **Inactive items appear in selection lists**

  Select this option to specify that inactive items will be included in lists on transition forms and report forms.

- **On lookup or query-at-runtime forms**
  - **Allow searching**

    This option enables the value find and relational field value lookup for the field. (See *Allow Searching* under *Style* above for a description of the *value find* feature.) The relational field value lookup feature provides an advanced searching mechanism that lets users find values for *Single Relational* and *Multi-Relational* fields. If this option is enabled, an additional search icon is available for the field.

  - **Show full list**

    This option lets users select a value from a drop-down list on the Relational Field Value Lookup form and Query-at-Runtime form for reports.

Single Selection Field Options

Style

- **Allow searching**

  Select this option to enable the value find feature for the field on submit, transition, and update forms. This enables users to search for values.

  > **Tip:** If the field will contain 200 or more selections, this option is recommended to ensure best performance. If you do not allow searching, fields that contain over 250 selection values are automatically set as searchable in the Editable Grid.

- **Single drop-down list**

  Select this option to let users select a value from a populated drop-down list.

Values

- To modify existing values, click within each row.

  > **Tip:** You can use shortcut keys to edit and navigate values. For more information, see Selection Field Shortcut Keys [page 509].

- To manually reorder selections in the list, use the **Move up** and **Move down** controls or click the **Value** or **Weight** column heading and then verify that "User defined" is selected in the **End-user display order** list. You can also choose one of the following display orders:

  - Value ascending

  - Value ascending

  - Weight ascending

  - Weight descending

- For fields in primary tables, you can include or exclude a value by selecting the **Enabled** check box, or you can select **Enable all** or **Disable all** from the **More** list to include or exclude all values.

- Weights give a numeric value to each selection that can be useful for *Trend* reports, backlog views, and *Summation* fields.

  You can use the default weight of 100 for each value or specify different weights for each value as needed. By default, "none" values are treated as 0. If you want "none" values to be treated as a value other than 0, change the value in the **Weight for "(none)" value** setting.

- To manage a large number of selections, use the **Import** and **Export** options in the **More** list. For details, refer to Importing Selection Field Values [page 188].

- To delete a value, select the value and click **Delete**. In the dialog box that appears, if you want to delete the value even if it is in use, uncheck the **Prevent the deletion if the value is in use** option. All references to the value are deleted as well.

If you delete this field and later restore it, the values you specify are also restored. For more information, see Deleting and Restoring Fields [page 186].

Display

- **Span entire row on forms**

Select this option to make this field the only field to appear on a single row.

Search & Query

- **Appears in report field lists**

Select this option to include the field in lists on report forms. If you clear this option after the field is used in reports, the changed setting is ignored for those reports and the field is still shown until it is removed from the report definition.

- **Appears on lookup form and relational field value lookup**

Select this option to add the field to the Auxiliary Data search form and Relational Field Value Lookup form.

The field order for the Relational Field Value Lookup form for primary items is determined by the default field order of the table's first project in the project hierarchy. Projects are defined in Application Administrator.

- **Appears on searches for this table**
Select this option to specify that the field will be available for selection on the Advanced Search page. This setting applies to primary and auxiliary tables. See Configuring Advanced Search [page 191] for additional information.

   Note: When this option is selected, the field appears as a facet on the Work Center search results page for a single application. This option also adds the field to the Work Center search index, which enables users to search for work items using values in this field.

- **On lookup or query-at-runtime forms**

   - **Allow searching**

   This option enables the value find and relational field value lookup for the field. (See *Allow Searching* under *Style* above for a description of the *value find* feature.) The relational field value lookup feature provides an advanced searching mechanism that lets users find values for *Single Relational* and *Multi-Relational* fields. If this option is enabled, an additional search icon is available for the field.

   - **Show full list**

   This option lets users select a value from a drop-down list on the Relational Field Value Lookup form and Query-at-Runtime form for reports.

   Note: If a *Single Selection* field has 1,000 items or more, the Advanced Search ignores the **On lookup or query-At-runtime forms** setting, and instead displays it as a searchable field.

Sub-Relational Field Options

Values

- **Relational field**

Select the *Single Relational* field that you are associating with the *Sub-Relational* field.

> **Note:** Selecting a relational field enables the Sub-Field control.

- **Sub-field**

Select the field from the Relational Field table whose value you want to display to users. For example, you could create a *Single Relational* field from a *Companies* table, and then create two new *Sub-Relational* fields: *Company Address 1* and *Company Phone Number*.

> **Note:** This control is unavailable until a relational field has been selected above.

Display

- **Span entire row on forms**

Select this option to make this field the only field to appear on a single row.

Search & Query

- **Appears in report field lists**

Select this option to include the field in lists on report forms. If you clear this option after the field is used in reports, the changed setting is ignored for those reports and the field is still shown until it is removed from the report definition.

- **Appears on lookup form and relational field value lookup**

Select this option to add the field to the Auxiliary Data search form and Relational Field Value Lookup form.

The field order for the Relational Field Value Lookup form for primary items is determined by the default field order of the table's first project in the project hierarchy. Projects are defined in Application Administrator.

- **On lookup or query-at-runtime forms**

  - **Allow searching**

    This option enables the value find and relational field value lookup for the field. (See *Allow Searching* under *Style* above for a description of the *value find* feature.) The relational field value lookup feature provides an advanced searching mechanism that lets users find values for *Single Relational* and *Multi-Relational* fields. If this option is enabled, an additional search icon is available for the field.

  - **Show full list**

    This option lets users select a value from a drop-down list on the Relational Field Value Lookup form and Query-at-Runtime form for reports.

Summation Field Options

Values

- **Fields to sum**

Lists *Single Selection* fields that have an assigned weight or *Numeric* fields set as integers. Select the fields to be summed.

Display

- **Span entire row on forms**

Select this option to make this field the only field to appear on a single row.

Search & Query

- **Appears in report field lists**

Select this option to include the field in lists on report forms. If you clear this option after the field is used in reports, the changed setting is ignored for those reports and the field is still shown until it is removed from the report definition.

- **Appears on lookup form and relational field value lookup**

Select this option to add the field to the Auxiliary Data search form and Relational Field Value Lookup form.

The field order for the Relational Field Value Lookup form for primary items is determined by the default field order of the table's first project in the project hierarchy. Projects are defined in Application Administrator.

- **Appears on searches for this table**

Select this option to specify that the field will be available for selection on the Advanced Search page. This setting applies to primary and auxiliary tables. See Configuring Advanced Search [page 191] for additional information.

Text Field Options

Note: System fields could offer a subset of the options described here.

Style

Important: You cannot change the style of a *Text* field after a process app is deployed. For example, you cannot change a fixed-length field to a memo field. However, there is a way to reset the field and change it if you intend to deploy the process app to a different environment. For more information, see Modifying Locked Elements in a Published Process App [page 64].

- **Memo**

Select this option to let a user enter up to 65,535 Unicode characters in the field. (See **Span entire row on forms**, below.) *Text* fields with the **Memo** option enabled cannot be sorted in reports.

- **Journal**

Select this option to automatically insert a date, time, and user ID whenever a user enters text in the field. A journal field can accept up to 65,535 Unicode characters. *Text* fields with the **Journal** option enabled cannot be sorted in reports.

Tip: For information about expanding this size, refer to S139895.

Selecting this option enables these additional options:

- **Append only**

  Forces the text in any new journal entries to be appended to the text in an existing journal entry, preventing users from modifying existing journal entries. Clear this option to let users edit existing journal entries.

- **Insert newest first**

  Select this check box to insert *new* journal entries at the top of the list, so the journal entries are ordered from newest to oldest. Clear this check box to insert new journal entries at the bottom of the list, so the journal entries are ordered from oldest to newest. (Changing this option does not affect the order in which existing journal entries are listed.)

- **Fixed length**

  Select this option and specify maximum number (up to 255 Unicode characters) allowed in the field. (See **Span entire row on forms**, below.)

  Selecting this option enables the following additional options *(For custom text fields only)*:

  - **Password**

    Forces the characters in this field to be replaced with asterisks (*) on state and transition forms and in the form preview.

    > **Note:** This option does not appear on the **Options** tab for system text fields, like *Title*, or for fields for which the **Memo** or **Journal** option is selected.

  - **Allow translation for value display in relational fields**

    Enables administrators to use the **Localization** feature in Application Administrator to translate fixed length text field values that are displayed in relational fields. The translation only applies to fixed length text values when they are displayed in relational fields—in all other instances the original text is displayed.

    For example, if you use the system Title field in the value display format for this table, you can select this check box, deploy the process app, and then your administrator can translate the Title field values in Application Administrator. Users with the locale that matches the translated content will see the translated values in relational fields that points to this table.

    > **Note:** There is no need to select this check box if the field is not used in the value display format or if you are not translating strings into other languages.

Display

- **Prefix** and **Suffix**

  Type any plain text and HTML tags you want to display before and after the field's values. Note the following:

  - The rendered HTML is visible in preview mode and at runtime, but not in the form editor.

▪ HTML formatting applies to the prefix and suffix for the field label, but not to the value of the field.

▪ Use the `<` and `>` character entities to display the less than (<) or greater than (>) signs, respectively. Otherwise, these signs will be interpreted as HTML code. Use the ` ` character entity for each extra space you want to display.

▪ By default, there is no space between the field value and its prefix or suffix. If you want a space to appear between the prefix or suffix and the field value, then you must add spaces.

▪ Make sure that any HTML placed in the prefix or suffix is complete in itself; do not split up an HTML tag between the prefix and suffix.

▪ Instead of adding a hyperlink to a prefix or suffix, it is recommended that you use a HyperLink control and bind the URL to the text field via refresh.

> **Note:** For best results, formatting changes should be applied using styles on custom forms rather than by adding HTML tags to the field's prefix and suffix. For more advanced HTML formatting, create a custom form for your workflow, and then add an HTML/JavaScript widget to the form.

- **Enable Rich Text**

  Select this check box to enable a Rich Text Editor that lets users apply basic formatting to text in the field. This also enables the Rich Text Editor for the **Default Value** setting located on the **Attributes** tab. For details, refer to Setting Default Values for Fields [page 218].

  Clear the check box if you do not want the Rich Text Editor available for this field or you do not want HTML tags rendered at runtime. You may want to disable the editor if you want HTML tags and code snippets to display as plain text, for example.

  The Rich Text Editor is enabled or disabled at a system level in SBM Application Administrator. If you select the **Enable Rich Text** check box for a field, but HTML5 features are disabled for your system, HTML tags manually added to the field are rendered on State forms.

  The **Enable Rich Text** option applies only to Memo and Journal fields and is selected by default when a new field of this type is added to a table.

  CAUTION:

  > If you disable Rich Text capabilities after formatted data has been added to fields, notes, and e-mail messages, the data may be garbled or unreadable. In this case, you must manually modify data to remove formatting tags.

- **Span entire row on forms**

  Select this option to make this field the only field to appear on a single row.

  > **Note:** This setting does not affect the display of *Text* fields with the **Memo** option or the **Fixed length** option if the specified length is greater than the Max Text Field Size (set in SBM System Administrator).

Search & Query

- **Appears in report field lists**

  Select this option to include the field in lists on report forms. If you clear this option after the field is used in reports, the changed setting is ignored for those reports and the field is still shown until it is removed from the report definition.

- **Appears on lookup form and relational field value lookup**

  Select this option to add the field to the Auxiliary Data search form and Relational Field Value Lookup form.

  The field order for the Relational Field Value Lookup form for primary items is determined by the default field order of the table's first project in the project hierarchy. Projects are defined in Application Administrator.

- **Appears on searches for this table**

  Select this option to specify that the field will be available for selection on the Advanced Search page. This setting applies to primary and auxiliary tables. See Configuring Advanced Search [page 191] for additional information.

- **Automatically prepend wildcard to lookup text**

  > **Note:** This option is available only when the **Appears on lookup form and relational field value lookup** option is enabled.

  Select this option to automatically use wildcard characters both *before* and *after* the search text for this field, unless there are wildcard characters *within* the search text. This helps users search for keywords more easily.

  When this option is enabled, a percent sign (%) is displayed next to the field on the Lookup form.

  When this option is disabled, a wildcard character is automatically appended to the search text, unless the search text itself includes wildcard characters.

- **Include field in keyword searches**

  Select this option to let users search for particular text from keyword searches.

  > **Tip:** Improve the performance of your process app in the by disabling this option for any *Text* fields that you do not expect to be relevant in searches (especially those with the **Memo** option enabled).

URL Field Options

Style

- **Allow multiple URLs**

  Select this check box to enable users to add up to 100 URLs to a field.

Display

- **Span entire row on forms**

  Select this option to make this field the only field to appear on a single row.

Search & Query

- **Appears in report field lists**

  Select this option to include the field in lists on report forms. If you clear this option after the field is used in reports, the changed setting is ignored for those reports and the field is still shown until it is removed from the report definition.

- **Include field in keyword searches**

  Select this option to let users search for particular text from keyword searches.

- **Appears on searches for this table**

  Select this option to specify that the field will be available for selection on the Advanced Search page. This setting applies to primary and auxiliary tables. See Configuring Advanced Search [page 191] for additional information.

User Field Options

Style

- **Allow searching**

  Select this option to enable the value find feature for the field on submit, transition, and update forms. This enables users to search for values.

  > **Tip:** If the field will contain 200 or more selections, this option is recommended to ensure best performance. If you do not allow searching, fields that contain over 250 selection values are automatically set as searchable in the Editable Grid.

- **Single drop-down list**

  Select this option to let users select a value from a populated drop-down list.

Associated Roles

Select the roles that determine which values users can select. You must assign at least one role to a *User* or *Multi-User* field before you can set a default value for the field in SBM Application Administrator. For *User* fields, users can select a single value. For *Multi-User* and *Multi-Groups* fields, multiple values can be selected.

Display

- **Span entire row on forms**

  Select this option to make this field the only field to appear on a single row.

Search & Query

- **Appears in report field lists**

  Select this option to include the field in lists on report forms. If you clear this option after the field is used in reports, the changed setting is ignored for those reports and the field is still shown until it is removed from the report definition.

- **Appears on lookup form and relational field value lookup**

  Select this option to add the field to the Auxiliary Data search form and Relational Field Value Lookup form.

The field order for the Relational Field Value Lookup form for primary items is determined by the default field order of the table's first project in the project hierarchy. Projects are defined in Application Administrator.

- **Appears on searches for this table**
  Select this option to specify that the field will be available for selection on the Advanced Search page. This setting applies to primary and auxiliary tables. See Configuring Advanced Search [page 191] for additional information.

  > **Note:** When this option is selected, the field appears as a facet on the Work Center search results page for a single application. This option also adds the field to the Work Center search index, which enables users to search for work items using values in this field.

- **On lookup or query-at-runtime forms**

  - **Allow searching**

    This option enables the value find and relational field value lookup for the field. (See *Allow Searching* under *Style* above for a description of the *value find* feature.) The relational field value lookup feature provides an advanced searching mechanism that lets users find values for *Single Relational* and *Multi-Relational* fields. If this option is enabled, an additional search icon is available for the field.

  - **Show full list**

    This option lets users select a value from a drop-down list on the Relational Field Value Lookup form and Query-at-Runtime form for reports.

Attributes Tab of the Field Property Editor

Use this tab to specify attributes for any field except for the *Sub-Relational* type.

Element	Description
Defaults	**Default value**: Where available, specify the value (or values) to be preselected or pre-entered for this field.
	For some field types, default values must be specified in SBM Application Administrator as part of project configuration after the process app is deployed.
	For guidance on setting default values for specific field types, refer to Setting Default Values for Fields [page 218].
	Backfill to existing items: Specify that the default value will apply to any existing items that were created before the default value was set. This option is available for *Binary/Trinary*, *Date/Time*, *Multi-Selection*, *Numeric*, *Single Selection*, and *Text* fields.

Element	Description
Attributes	**Required**: Specify that users must set a value for the field. This option is disabled for *Binary/Trinary* fields. On forms, the label of a required field typically appears to the user in red or green italics, though that can be modified globally in SBM System Administrator. **Allow mass update**: Make the field available when users mass update items in the project or auxiliary table. Mass updates let users transition, update, or delete multiple primary items simultaneously and update or delete multiple auxiliary items simultaneously. See Configuring Mass Updates [page 187] for related information. This option is not supported for *File* and *URL* fields. **Require appended text**: Specify that users must append text to the field during every transition. This option is available when the **Required** check box is selected, and when **Journal** is selected on the **Options** tab. **Require appended file/URL:** Specify that users must add a file or URL to the field during every transition. This option is available for *File* and *URL* fields when the **Required** check box is selected, and when **Allow multiple files** or **Allow multiple URLs** is selected on the **Options** tab. **Read only**: Configure the field so that it can be viewed but not edited by users. **Append only:** For *File* and *URL* fields, specify that users can only add files or URLs, preventing users from modifying existing files or URLs.

Setting Default Values for Fields

The following sections provide guidance for setting default values for the following field types:

- Date/Time Fields [page 218]

- Selection Fields [page 219]

- Text Fields [page 220]

Date/Time Fields

This topic describes the date/time tool that is displayed when you click the down-arrow next to the Default Value field for a *Date/Time* field on Attributes Tab of the Field Property Editor [page 217].

You can type a default value in the field, but the date/time tool is a convenient way to prevent problems introduced by typing errors.

CAUTION:

⚠️   The "(Auto)" string is not supported as a default value for *Date/Time* fields.

Element	Description
Clear	Click this button to clear the previously specified value from the **Default value** field, so no default is specified. You can also delete the text in the **Default value** field in the Property Editor without invoking the tool.
List of values (Special Value tab)	Select a time period like **This Week** or **Last Month** from the **Start Of** or **End Of** column. Though **Now** is displayed in the **Start Of** column, it represents the current time at the moment of transition. **Note:** Assuming a default value of **Now**, the value stored in the database is equal to the number of seconds that have passed since 12:00:00 a.m. in the submitter's or modifier's time zone. As an alternative to the values listed, you can type `plus` *n* or `minus` *n* directly in the **Default value** field to offset the value of `Today` by a number of days. You can specify a date up to 5000 days from today (`plus 1`, `plus 2`, ..., or `plus 5000`) or up to 1000 days before today (`minus 1`, `minus 2`, ..., or `minus 1000`). **Note:** Clicking the column heads *does not* sort the list Instead, it clears the **Default value** field.
Exact Value	Select this tab to display the calendar tool, which includes a time field. When you have set the date and time, click **Accept**. Click **Special Value** to return to the keyword tool.

Selection Fields

Selection fields are broadly defined as any field where users can choose a predefined value. For SBM, these field types are considered selection fields:

- *Single Selection*
- *Multi-Selection*
- *Binary/Trinary*
- *User*
- *Multi-User*
- *Multi-Group*
- *Folder*
- *Single Relational*
- *Multi-Relational*

Values for these field types are defined in various interfaces. Default values are set in the same interface where the value is defined. For details, refer to About Selection Field Values [page 182].

Text Fields

You can use a Rich Text Editor to format default values for *Text* fields if these conditions apply:

- The field is set as a Memo or Journal type.

- The **Enable Rich Text** check box is selected on the **Options** tab for the field. Refer to Text Field Options [page 212].

- The **Enable HTML5 features** check box is selected on the Applications tab of the SBM Composer options dialog box. Refer to Application Options [page 486].

If you choose to manually enter HTML as a default value, be aware that obviously suspicious or dangerous HTML is not rendered at runtime.

The following tags are considered suspicious:

- `<applet, </applet`

- `<embed, </embed`

- `<form, </form`

- `<frame, </frame`

- `<iframe, </iframe`

- `<input, </input`

- `<script, </script`

- `<textarea, </testarea`

The following tags may not be rendered at runtime unless they are added using the Rich Text Editor. These tags will not be rendered if they contain suspicious attributes, such as `onload` or `onclick`:

- `<a, </a`

- `<img, </img` (Rendered only when added at runtime)

> **Remember:** Improper or invalid HTML could have a negative impact on the field and possibly on the entire form. Refer to the World Wide Web Consortium Web site at http://www.w3.org for information about HTML syntax.

Administrators can use the **Sanitize HTML Values in Memo Fields, Journal Fields, and Notes** option in SBM Application Administrator to configure a list of approved tags, tag attributes, and restricted styles that sanitizes HTML in the database prior to rendering it in text fields.

If the **Enable Rich Text** check box is cleared for the field, only plain text is allowed. Carriage or hard returns can be entered in the default values for Memo and Journal fields, but not for Fixed Length fields.

Note: You cannot change the default value for the system *Item ID* field.

Dependencies Tab of the Field Property Editor

Use this tab to specify field dependencies for *Single Relational*, *Single Selection*, or *User* fields.

Element	Description
Dependent Fields list	Select the fields that are dependent on the value selected in this field. See Configuring Field Dependencies [page 190] for details and examples.
Edit Value Restrictions	See Configuring Field Dependencies [page 190] for details and examples.

Forms Tab of the Field Property Editor

Use this tab to view the custom forms based on the table. The forms that contain the selected field are highlighted by bold type.

Element	Description
Name	The name of the form.
Type	The type of form
Contains field	Whether the form contains the selected field.
Add to Forms	Opens the Add *'Field Name' Field Type* Field to Forms [page 222] dialog box, which allows you to add the field to selected forms.
Reposition in Forms	Opens the Reposition *'Field Name' Field Type* Field in Forms [page 224] dialog box, which allows you to reposition the field on selected forms.
Remove from Forms	Opens the Remove *'Field Name' Field Type* Field from Forms [page 225] dialog box, which allows you to remove the field from selected forms.

Tip: Double-click a form to open it in the form editor, or right-click the form and then select **Open** or **Open in New Tab**.

Add/Reposition/Remove Field Dialog Box

Use this dialog box to add, remove, or reposition a field on multiple custom forms.

This dialog box has three modes:

- Add *'Field Name' Field Type* Field to Forms [page 222]

- Reposition *'Field Name' Field Type* Field in Forms [page 224]

- Remove *'Field Name' Field Type* Field from Forms [page 225]

> **Tip:** You can change the update action in the dialog box; the dialog box will dynamically switch to the applicable mode.

Add *'Field Name' Field Type* Field to Forms

Element	Description
Add field	The update action. To change the action, click it and then select **Reposition field** or **Remove field**.
Field	The field to add. By default, the field you selected in the table editor is selected. To change the field, click it and then select another field.

Element	Description
Relative position	Specify the relative position of the field. **Note:** The field placement depends on whether the **Span entire row on forms** option is selected on the **Options** tab of the field Property Editor. **above**: • If the **Span entire row on forms** option is selected and the row above the adjacent field control is "autosize" and empty, the field is placed in the empty row. Otherwise, an autosize row is added above the adjacent field control and the field is placed in the new row. • If the **Span entire row on forms** option is not selected, and the cells above the adjacent field control are empty and in an autosize row, the field is added directly above the adjacent field control. Otherwise, an autosize row is added and the field is placed directly above the adjacent field control. **after or below**: • If the **Span entire row on forms** option is selected and the row below the adjacent field control is "autosize" and empty, the field is placed in the empty row. Otherwise, an autosize row is added below the adjacent field control and the field is placed in the new row. • If the **Span entire row on forms** option is not selected and there is space to the right of the adjacent field control, the field is placed there. Otherwise, an autosize row is added and the field is placed directly below the adjacent field control. **below**: • If the **Span entire row on forms** option is selected and the row below the adjacent field control is "autosize" and empty, the field is placed in the empty row. Otherwise, an autosize row is added below the adjacent field control and the field is placed in the new row. • If the **Span entire row on forms** option is not selected, and the cells below the adjacent field are empty and in an autosize row, the field is placed in the empty row. Otherwise, an autosize row is added and the field is placed directly below the adjacent field control. **Note:** The field will have the same characteristics as the adjacent field. Therefore, if the label for the adjacent field is above the control, an additional row will be added if necessary to accommodate both the label and the control.

Element	Description
the adjacent field	Select the field that this field will be adjacent to when it is added to the form.
to selected forms	Lists the custom forms that do not already contain the field you want to add. By default, all forms that contain the adjacent field are selected. All other forms are shown in a lighter shade and their check boxes are disabled. You can manually select and clear check boxes to specify the forms that should include the field, or use the list described below to select forms by type or clear your selections. **Note:** You cannot update forms that are checked out by others; they are also shown in a lighter shade with disabled check boxes.
Select All, Clear All, Select State Forms, Select Transition Forms, Select Print Forms *(for primary table fields)* Select All, Clear All, Select View Forms, Select Edit Forms, Select Print Forms *(for auxiliary table fields)*	Specify whether you want to add the field to all forms or to all forms of a certain type, or whether you want to clear your selection and start over.
Add to Forms	Adds the field to the selected forms. To view an updated form, double-click it or right-click it and then select **Open** or **Open in New Tab**.
Close	Closes the dialog box.

Reposition *'Field Name' Field Type* Field in Forms

Element	Description
Reposition field	The update action. To change the action, click it and then select **Add field** or **Remove field**.
Field	The field to reposition. By default, the field you selected in the table editor is selected. To change the field, click it and then select another field.

Element	Description
Relative position	See description under Add *'Field Name' Field Type* Field to Forms [page 222], above.
the adjacent field	Select the field that this field will be adjacent to when it is repositioned on the form.
in selected forms	Lists the custom forms that contain the field you want to reposition. By default, all forms that contain the adjacent field are selected. All other forms are shown in a lighter shade and their check boxes are disabled. You can manually select and clear check boxes to specify the forms on which the field should be repositioned, or use the list described below to select forms by type or clear your selections. **Note:** You cannot update forms that are checked out by others; they are also shown in a lighter shade with disabled check boxes.
Select All, Clear All, Select State Forms, Select Transition Forms, Select Print Forms *(for primary table fields)* Select All, Clear All, Select View Forms, Select Edit Forms, Select Print Forms *(for auxiliary table fields)*	Specify whether you want to reposition the field on all forms or to all forms of a certain type, or whether you want to clear your selection and start over.
Reposition in Forms	Repositions the field on the selected forms. To view an updated form, double-click it or right-click it and then select **Open** or **Open in New Tab**.
Close	Closes the dialog box.

Remove *'Field Name' Field Type* Field from Forms

Element	Description
Remove field	The update action. To change the action, click it and then select **Add field** or **Reposition field**.

Element	Description
Field	The field to remove. By default, the field you selected in the table editor is selected. To change the field, click it and then select another field.
from selected forms	Lists the custom forms that contain the field you want to remove. By default, all forms that contain the adjacent field are selected. All other forms are shown in a lighter shade and their check boxes are disabled. You can manually select and clear check boxes to specify the forms from which the field should be removed, or use the list described below to select forms by type or clear your selections. **Note:** You cannot update forms that are checked out by others; they are also shown in a lighter shade with disabled check boxes.
Select All, Clear All, Select State Forms, Select Transition Forms, Select Print Forms *(for primary table fields)* Select All, Clear All, Select View Forms, Select Edit Forms, Select Print Forms *(for auxiliary table fields)*	Specify whether you want to remove the field from all forms or from all forms of a certain type, or whether you want to clear your selection and start over.
Remove from Forms	Removes the field from the selected forms.
Close	Closes the dialog box.

Field Dependency Tutorials

The following tutorials explain how to set up field dependencies:

- Single Selection Field Dependency Tutorial [page 227]

- User Field Dependency Tutorial [page 228]

- Relational Field Dependencies Tutorial [page 230]

Single Selection Field Dependency Tutorial

Prerequisites:

- This tutorial assumes you have an application workflow created in SBM Composer and are familiar with basic application tasks, such as adding fields and field selections.

- To test this tutorial in the SBM Work Center, you must have privileges to deploy to a running environment.

The following tutorial explains how to configure field dependencies that tailor the value list for a *Single Selection* field. You can reproduce this example for other field dependencies as needed.

In this example, you will create a *Product* field that has two available values: Product A and Product B. You will set dependencies to limit the values available in a *Version* field depending on the selection made in the *Product* field. For example, if a user selects Product A from the *Product* field, a set of available values unique to Product A are listed in the *Version* field.

This tutorial is configured exclusively in SBM Composer. You can log in to SBM Work Center to test the tutorial.

> **Tip:** You can override dependent field values for independent *Single Selection* fields in projects in Application Administrator.

To establish a dependency for two *Single Selection* fields:

1. In an application primary table in SBM Composer, create a *Single Selection* field named *Product*.

2. Select the **Options** tab for the *Product* field. Add the following values:

   - Product A

   - Product B

3. In the same table, create a *Single Selection* field named *Version*.

4. Select the **Options** tab for the *Version* field. Add the following values:

   - 2.0

   - 2.5

   - 3.0

   - 3.1

5. Select the *Product* field, and the select the **Dependencies** tab.

6. Select the *Version* field from the fields list to set it as the dependent field.

7. Select the **Edit Value Restrictions** link, and then select the workflow that contains the *Version* field. The **Dependencies** tab for the workflow opens.

8. Select **Product A** in the left pane, and then select the *Version* field check box in the middle pane. In the right pane, clear the **3.0** and **3.1** check boxes.

9. Select **Product B** in the left pane, and then select the *Version* field check box in the middle pane. In the right pane, clear the **2.0** and **2.5** check boxes.

10. Save, check in, publish, and deploy your changes.

11. Log in to SBM Work Center as a user who has privileges to view and update items in the application that contains the field dependency you added.

12. Submit an item into a project associated with the application workflow that contains the field dependency.

Values for the **Version** field are now dependent on the selection users make for the **Product** list. In this example, when Product A is selected from the **Products** list, the user can choose the None, 2.0, or 2.5 selections. When Product B is selected from the Products list, the user can choose the None, 3.0, or 3.1 selections.

> **Note:** You can further enhance the dependency relationship by setting a default value for the dependent field. For example, you can set the **Version** field to default to *2.5* when *Product A* is selected in the **Product** field.

User Field Dependency Tutorial

Prerequisites:

- This tutorial assumes you are familiar with basic application tasks, such as adding fields and field selections, and with checking in, publishing, and deploying process apps.

- You must use SBM Composer and Application Administrator to set up the dependency. To complete this tutorial, you must have privileges to deploy to a running environment.

- Your application should contain two *User* fields with valid user selections. The fields should be called *Manager* and *Employee*. Users who are values should have privileges to view, submit, and update items in the application.

The following tutorial explains how to configure field dependencies that tailor the selection list for a *User* field. Dependencies will be set that limit the selections available in an *Employee* field depending on the selection made in the *Manager* field. For example, if a user selects Kathy Manager from the *Manager* field, Chris Tester and Hans Tester are available selections for the *Employee* field; if the user selects Joe Manager from the *Manager* field, Laura Engineer and Newton Engineer are available as selections for the *Employee* field.

Establish the Dependency in SBM Composer

In this step, you will create two *User* fields and establish a dependency between those fields.

To establish a dependency for a *User* field:

1. Open a process app in SBM Composer.

2. From the App Explorer, select the primary table for your application.

3. Add a *User* field named *Manager*.

4. Add another *User* field named *Employee*.

5. Select the *Manager* field, and then select the **Dependencies** tab.

6. In the list of available dependent fields, select *Employee*.

7. Save, check in, publish, and deploy your changes.

Add User Selections for the Independent and Dependent Fields

In this step, you will add user selections to the fields created in the previous step. To illustrate the example, sample users are added to the field.

To add user selections:

1. In Application Administrator, edit the workflow that contains the *Manager* and *Employee* fields.

2. Select the **Default Fields** tab.

3. Click the *Manager* field.

4. Select the **Attributes** tab.

5. In the **Values** section, click **Manage User Selections**, and then add Kathy Manager and Joe Manager as selections.

6. Save your changes.

7. Click the *Employee* field.

8. Select the **Attributes** tab.

9. In the **Values** section, click **Manage User Selections**, and then add Laura Engineer, Newton Engineer, Chris Tester, and Hans Tester as selections.

10. Save your changes.

Set Dependent Field Selections

In Application Administrator, you will specify which field selections are available for the dependent field based on the selection users make for the independent field. For example, when a user selects Kathy Manager from the *Manager* field, Chris Tester and Hans Tester are the only selections available in the *Employee* field.

To set dependent field selections:

1. In Application Administrator, edit the project for which the dependent values should be set.

2. Select the **Default Fields** tab.

3. Click the *Manager* field.

4. Select the **Dependencies** tab.

5. From the **Select Value to Override** list, select Joe Manager.

6. In the **Fields Dependent on *Manager*** pane, select the *Employee* field.

7. Clear the check boxes next to Chris Tester and Hans Tester.

8. From the **Select Value to Override** list, select Kathy Manager.

9. In the **Fields Dependent on *Manager*** pane, select the *Employee* field.

10. Clear the check boxes next to Laura Engineer and Newton Engineer, and then click **OK**.

11. Save your changes.

Test the Dependency

To test the dependency:

1. Log into SBM Work Center as a user who has privileges to the application that contains the dependency.

2. Submit an item into the project that you modified in Application Administrator in the previous steps.

3. On the **Submit** from, select Kathy Manager from the *Manager* field. Notice that Chris Tester and Hans Tester are available as selections for the *Employee* field.

4. Select Joe Manager from the *Manager* field. Notice that Laura Engineer and Newton Engineer are available as selections for the *Employee* field.

Relational Field Dependencies Tutorial

Prerequisites:

- This tutorial assumes you are familiar with basic application tasks, such as adding tables, and with checking in, publishing, and deploying process apps. You should also understand how to add items to auxiliary tables using the Manage Data feature.

- You must use SBM Composer to set up the dependency. To complete this tutorial, you must have privileges to deploy to a running environment.

The following example explains how to configure field dependencies that tailor the value list for a *Single Relational* field added to a primary table. You can reproduce this example for other field dependencies as needed.

In this example, you will create two auxiliary tables: a *Products* table and a *Versions* table. You will then establish a relationship between the two tables that allows specific

version records in the *Versions* table to be available for each product in the *Products* table. For example, if Version 1 and Version 2 only apply to Product A, those versions are the only available values when Product A is selected from the *Products* field.

To set up this Relational field dependency:

1. In SBM Composer, create two auxiliary tables: a *Products* table and a *Versions* table. For this example, the *Products* table is the independent field table and the *Versions* table is the dependent field table.

2. Add a *Single Relational* field of the independent field type to the dependent field table. For this example, add a *Single Relational* field to the *Versions* table, and select Products from the **Table** list located on the **Options** tab.

3. Open the application's primary table, and then add a *Single Relational* field that will serve as the independent field. For this example, name the field *Products*, and then select "Products" from the **Table** list on the **Options** tab for the *Single Relational* field.

4. In the same table, add another *Single Relational* field that will serve as the dependent field. For this example, name the field *Versions*, and then select "Versions" from the **Table** list on the **Options** tab for the *Single Relational* field.

5. Select the *Single Relational* field that serves as the independent field. For this example, select the *Products* field.

6. Select the **Dependencies** tab, and then select the *Versions* table from the dependent fields list.

7. Click the **Edit Value Restrictions** link, and then select the workflow for which you will restrict dependent field values.

8. Verify that the *Products* field is selected in the left pane, and then select the *Versions* field in the middle pane.

9. Select the *Products* field from the right pane.

10. Grant privileges to the *Products* and *Versions* tables. You can do this in SBM Composer by granting role privileges for each table, or you can assign user or group privileges to the table in Application Administrator after you deploy the process app. For this tutorial, users should be able to submit, view, and update items in both tables.

11. Save, check in, publish, and deploy your changes.

12. Log in to SBM Work Center as a user who has privileges to manage the *Products* and *Versions* tables.

13. To open the Manage Data feature, click the user icon in the upper right corner, and then select **Manage Data**.

14. Populate the *Products* table with items. For this example, add "Product A" and "Product B" items.

15. Populate the *Versions* table with the following items. When you add these items, select the listed value from the *Products* field:

- Version 1 – Product A

- Version 2 – Product A

- Version 3 – Product B

- Version 4 – Product B

- Version 5 – Product B

16. Open the **Submit** form for a project used by the workflow in which you created the dependencies.

17. Click the **Find** button for the *Products* field. Notice that "Product A" and "Product B" are available as selections.

18. Select "Product A." Notice that for the *Versions* field, "Version 1" and "Version 2" are available.

19. Select "Product B" from the *Products* field. Notice that "Version 3," "Version 4," and "Version 5" are available as selections in the *Versions* field.

Chapter 13: Working with Forms

This section contains the following information:

About Forms

Forms provide access to information in primary and auxiliary items through three views:

- **State forms**

  A read-only view of primary items. For auxiliary items, these are referred to as view forms.

- **Transition forms**

  An editable form used to submit, transition, and update primary items. For auxiliary items, these are referred to as edit forms.

- **Print forms**

  A printable version of a state or view form. Many controls, such as the transition button bar and the **Actions** menu, are removed from print forms. If the form structure includes collapsed sections or inactive tabs, the hidden information is included in the generated print form so users can see it.

For each type of view, you can use the default "quick" form or a custom form. When a different form is needed, you can override the default form for each state and transition in SBM Composer. You can also use SBM Application Administrator to override forms for individual projects or states and transitions in those projects.

Quick Forms

Quick forms are automatically generated by the system and initially assigned to workflows, states, transitions, and primary and auxiliary tables. They use field privilege sections to determine field placement and security. Quick forms use standard formatting and require little maintenance.

Custom Forms

Custom forms offer great flexibility for changing the appearance and the behavior that users experience as they work with primary and auxiliary items. You can:

- Organize fields regardless of their privilege section.

- Change the form's visual appearance. For example, you may want to match forms to your organization's color scheme. You can also add your own images, hyperlinks, static text, buttons, and JavaScript files to custom forms.

- Use form actions to add dynamic behavior, such as hiding one field when a certain value is selected in a different field.

- Remove or modify transition buttons or the **Actions** drop-down list.

- Add widgets. For example, use the Embedded Report widget to add an SBM report to your form, or use the REST Grid widget to access external data through a REST service and display it in a tabular format in an application.

You can create a custom form with pre-populated sections or auto-sections. You can also base the custom form on another form or start with an empty form.

For details, refer to Customizing Forms [page 239].

Auto-sections

A custom form with *auto-sections* is automatically populated based on field privileges. With auto-sections, you can:

- Manually place some fields within any section, while letting the rest of the fields be automatically placed at runtime.

- Add logos, widgets, form actions, and other customizations without worrying about field placement.

- Add new fields to the primary table or to an auxiliary table, which then get automatically added to their respective privilege sections in all custom forms with auto-sections.

Security vs. Design

Custom forms enable you to control access to data and customize design and layout through two mechanisms: privilege sections and visual sections.

Privilege sections control access to fields placed on forms; visual sections control field placement on forms. For details, refer to Controlling Access to Data [page 244].

Modern Forms

Modern forms, introduced with SBM Composer 11.1, use technologies such as HTML5 and .css properties to achieve greater stability in layout and less reliance on scripting overall. Modern forms make it possible to offer the advanced features listed below:

- Responsive layout (see Working With Responsive Forms [page 269])

- Headers, footers, and sidebars with scrollable content (see Using Headers, Footers, and Sidebars [page 272])

- A runtime option that shows only required fields (see the Work Center online help)

Also, some of the historic defects present in legacy forms have been fixed with modern forms.

Legacy forms will continue to be supported; however, you will not be able to use the advanced features mentioned above. You may want to continue to use legacy forms if:

- You want to base new forms on existing forms that contain custom JavaScript, .css, or other page layout tweaks

- You are using pre-built solutions, or you have created forms based on solution forms

- You want your existing forms and new forms to be consistent

To convert your forms to modern forms:

1. Make sure the **Enable HTML5 features** check box is selected in the Form Options tab of the SBM Composer Options dialog box.

2. Create a new form based on your existing form, and make sure the **Legacy mode** check box is cleared.

3. Validate the process app and check for form errors.

> **Note:** If your forms contain custom JavaScript, .css, or tweaks to currently-deployed page layout, they will need to be reworked in order to convert them to modern forms (some new features such as headers/footers/sidebars may replace existing script). If your forms contain simple customizations such as form actions, or custom JavaScript use is limited to supported JSAPI functions, you can convert them as described above.

Quick Forms or Custom Forms?

Use the following table to determine whether to use quick forms or custom forms.

Scenario	Quick form	Custom Form
You need to get up and running quickly.	✔	
You want a low-maintenance solution for managing forms.	✔	
You want fields automatically grouped and ordered by privilege sections.	✔	✔

Scenario	Quick form	Custom Form
You want to use the quick forms as a starting point, and then make minor modifications, such as removing some of the Item Details sections or adding a logo or widget.		✔
You want flexibility in overall layout styles.		✔
You want to use form actions and widgets to control form behavior.		✔
You want to customize the form tool bar by hiding the Actions menu, removing items from the menu, and more.		✔
You want complete control over the form's appearance, security, and behavior.		✔

Navigating the Form Designer

Use the legend below the image to learn about parts of the form editor.
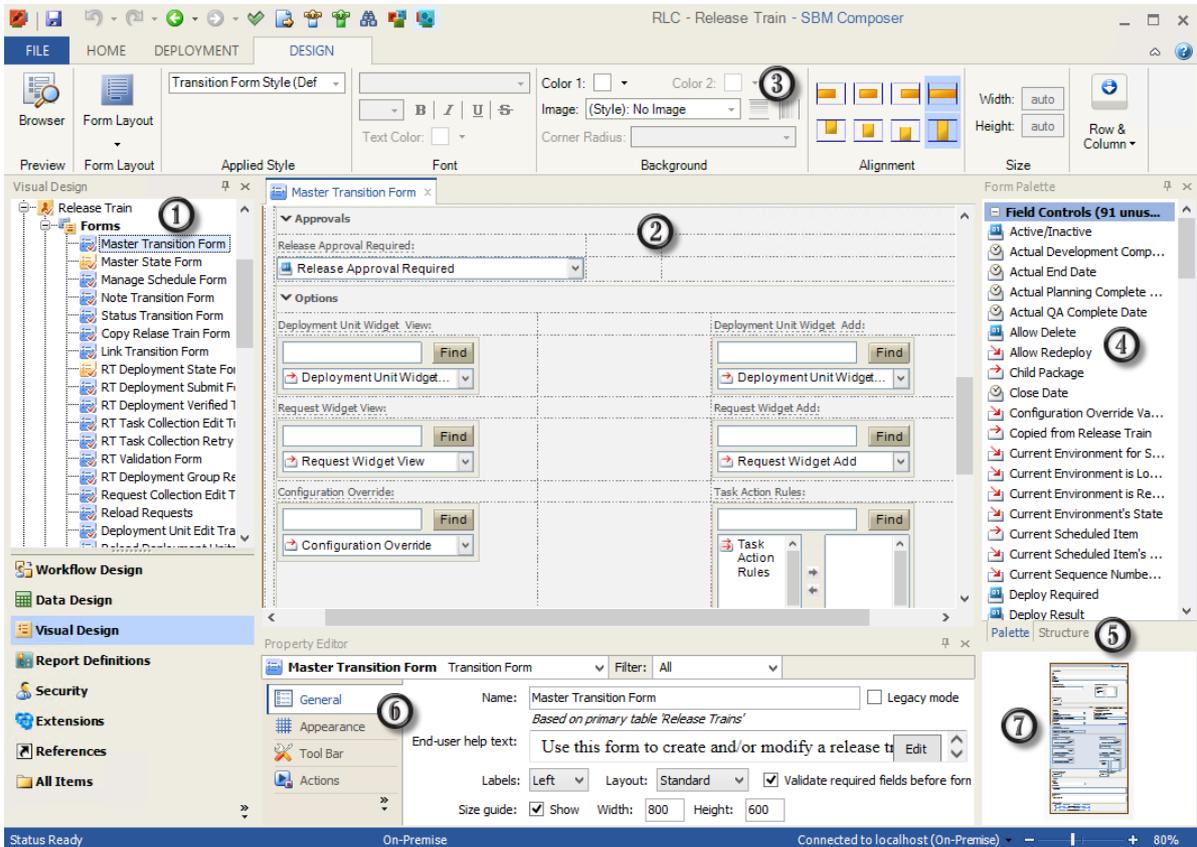
1. **Forms List**

    Click the Visual Design filter in the App Explorer to see all forms for the selected application. (For auxiliary tables, click the **Data Design** filter, then select the table. View, update, and print forms are available in the property editor.)

    Select a form to open it in the form editor; right-click on a form to view a menu of various options for working with forms.

2. **Form Editor**

    Drag and drop fields, controls, and more from the form palette onto the editor. You can then manage form layout, appearance, and behavior. Refer to Changing Form Appearance [page 241] and Managing Form Behavior [page 245].

3. **Form Ribbon**

    Use commands to preview your form and to access layout and style options. Refer to Design Tab of the Ribbon [page 497].

4. **Form Palette**

    Drag and drop these types of controls onto your form:

    - **Field Controls** represent all fields in the associated table. Each field control can be used only one time on the form.

- **Detail Controls** represent all Item Details sections that can appear for an item, such as Attachments, Change History, and Notes. Each detail control can be used only once on the form.

- **Container Controls** represent the visual sections that you can use to group controls together on the form.

- **Other Controls** represent the custom elements that you can include on the form, such as images, static text, and hyperlinks. These controls can be used as needed and are configured in the Property Editor.

- **Widgets** represent specialized widget types, including widgets that you can use to embed almost any valid HTML or JavaScript, the content found at any valid URL, and so on.

- **Custom Widgets** represent widgets that are imported as part of a form extension. This control type is available only if you have imported form extensions.

  Refer to Form Palette [page 312].

5. **Form Structure**

   Click to see a hierarchical tree structure with all form controls. Select a control in the tree to highlight it on the form or select a control on a form to highlight the control in the tree.

6. **Form Property Editor**

   Use these tabs in the form Property Editor to view and modify the various aspects of the selected form:

   - General [page 289]

   - Appearance Tab of the Form and Control Property Editor [page 297]

   - Tool Bar Tab of the Form Property Editor [page 298]

   - Actions Tab of the Form Property Editor [page 301]

   - Rows [page 302]

   - Columns [page 303]

   - Javascripts [page 304]

7. **Zoom Preview**

   Click in the box to move around in the form. Use the slider bar to change the zoom level. For details, refer to the following section.

To select a preset zoom level:

- In the **Zoom** area on the **Home** tab of the Ribbon, do one of the following:

  - Click **Zoom**, and select a zoom level.

  - Click **100%**.

To zoom in and out:

- Do one of the following:

  ▪ Below the zoom preview (in the bottom right corner), click the **+** and **−** buttons on the slider, or move the slider to the right and left.

  The current zoom level is displayed to the left of the slider.

  ▪ With the pointer over the workflow editor, press the Ctrl key while you move the mouse wheel backward and forward.

To fit the form to the window:

- In the **Zoom** area on the **Home** tab of the Ribbon, click **Zoom** and then select **Fit to Window**.

To move around in the form editor:

- Do one of the following:

  ▪ In the zoom preview (in the bottom right corner), drag the blue rectangle to the section of the form that you want to view in the form editor window.
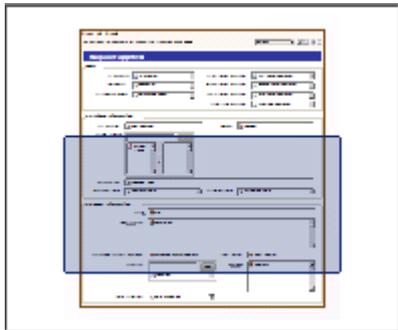
  ▪ In the form editor, with the pointer over the background (white space), press and hold the right mouse button while you drag the form graphic within the editor pane.

  Tip: By default, SBM Composer remembers the zoom level and position of the form within the editor pane after you close the process app. To override this, see the "Form Tab" section in Application Options [page 486].

Customizing Forms

This section provides guidance for working with custom forms and includes the following information:

- Creating Custom Forms [page 240]

- Changing Form Appearance [page 241]

- Controlling Access to Data [page 244]

- Managing Form Behavior [page 245]

- Previewing Forms [page 268]

Creating Custom Forms

Before you create a form, first determine where it will be used. The form may be automatically assigned to a process app element, depending on where it is created. You can also assign a form to elements after it is created.

- **To create a form for an application, and assign it to workflows, states, or transitions later:**

  In App Explorer, select **Visual Design**, and then right-click the **Forms** heading. Select **Add New**, and then select a form type.

- **To create a default form for a workflow:**

  In App Explorer, select **Workflow Design**, and then select the application workflow. Then select the **Forms** tab of the workflow Property Editor.

- **To create a form for a for a specific state or transition:**

  In App Explorer, select **Workflow Design**, and then select the application workflow. In the workflow editor, select the state or transition, and then select the **Form** tab of the state or transition Property Editor.

- **To create a form for a primary or auxiliary table:**

  In App Explorer, select the **Data Design** filter, then select the primary or auxiliary table. Select the **Forms** tab of the table Property Editor.

After you decide where to create the form, follow these steps to design it:

1. On the **Form Configuration** dialog box, choose the type of form (state, transition, print) as needed.

   Note: When you design a print form, note the following:

   - Consider the limitations of printing from a Web browser. If the page is too wide to fit on a printed page, information could be truncated or garbled.

   - If you specify a fixed size or vertical autofill for *Journal* fields, *Memo* fields, and containers, a scroll bar will automatically appear if the content does not fit in the allotted space. When the user prints the form, this may cause the overflow content to be cut off.

2. Select an option to create the form with pre-populated sections or auto-sections, to create the form based on another form, or to create an empty form. For details on these settings, refer to Form Configuration Dialog Box [page 288].

3. From the **Layout** drop-down list, select **Responsive** for a form layout that adjusts to the size of the screen (available only for modern forms), or select **Standard** for a non-responsive layout. For details, refer to Working With Responsive Forms [page 269].

4. If you do not want to create a modern form, select the **Legacy mode** check box to create a form using legacy technology (used in all releases prior to SBM Composer 11.1).

5. From the **Labels** drop-down list, choose to display labels to the left of form controls or on top of form controls.

> **Note:** By default, label text does not wrap on standard (non-responsive) forms. To enable label text to wrap, set the **Width** control of the label to a fixed value.

6. Select the number of columns for the form, from 1 to 4.

7. Click **OK**.

8. To change the overall layout for the form, select a layout from the **Form Layout** area on the **Design** tab of the Ribbon. For details on the options, refer to Form Layout [page 241].

9. To create visual sections on the form, drag container controls onto the form, configuring the rows and columns of the containers as needed. To create visual subsections, drag container controls into existing cells on the form. For details, refer to Form Controls [page 243].

10. To add data fields to the form, drag field controls into privilege or visual sections. Each field control and detail control can be used only once on the form. For details, refer to Controlling Access to Data [page 244].

11. To control form behavior, add form actions, widgets, or JavaScripts. For details, refer to Managing Form Behavior [page 245].

12. To view a mockup of the form, click **Preview** on the **Design** tab of the Ribbon. For details, refer to Previewing Forms [page 268].

Changing Form Appearance

Use the following mechanisms to control form appearance:

- Form Layout [page 241]

- Form Styling [page 242]

- Form Controls [page 243]

Form Layout

The first step to designing a form is to determine its layout, which is initially based on columns and rows.

You can use one of the predefined layout styles as described in Form Tools [page 497] or you can begin with a blank form and add your own columns and rows.

To learn about managing form layout, refer to:

- Working With Responsive Forms [page 269]

- Using Headers, Footers, and Sidebars [page 272]

Form Styling

Default form styles help ensure consistency in form design. Default styles are provided for state/view, transition/update, and print forms and for design elements, such as tabs and hyperlinks.

To see the list of default form styles, refer to About Styles [page 437].

You can use the default styles or modify them so they apply to every form you create for your process app. If you need to change the appearance for some forms, you can create custom styles. You can also override a particular style for an individual form or form element. This should be done rarely, however, to ensure consistency in form design.

Working with Styles

To modify default styles:

1. Select **Visual Design** in the App Explorer, and then select **Styles**.

2. Select a style in the **Form Related Styles** area.

3. Modify the style as needed.

4. To revert the styling, select **Default** for specific design settings, such as Background or Foreground/Text.

> **Tip:** You can revert most modified style settings to the default behavior as needed.

To create a new style:

1. Select **Visual Design** in the App Explorer, and then select **Styles**.

2. Do one of the following:

   - Right-click a style in the **Form Related Styles** area, then select **Duplicate**.

   - Right-click a style in the **Form Related Styles** area, then select **Create New**. Select the type of style to create.

3. Modify the style as needed.

To apply styles to forms:

1. Open a form or select the form element you want to apply a new style to.

2. Select the new style from the **Style** menu in the **Design** tab of the Ribbon.

   > **Note:** Default styles are noted for the form or the selected form element.

To override a form style:

1. Select the form element you want to modify.

2. Use options on the **Design** tab of the Ribbon to modify styling, such as font size and color.

Using Themes

You can use the SBM theme for design elements such as colors, fonts, and corner radius. Default and custom styles initially use theme values, which you can change as needed. In the styles editor, theme values are indicated by:

*   "(Theme)" preceding the value

*   A picture frame around the color

For example, the tab style shows the foreground/text color, background 2 color, and font as using theme values.

In the **Design** tab of the Ribbon, theme values are indicated in the same way, with the addition of "(Style):" preceding the value if a default or custom style is being used.

Form Controls

Once you define your basic layout and styling, you can drag and drop the following form palette elements onto your form:

*   Container controls that organize other controls and information. Container examples include expanders, sections, and tabs. Refer to Container Controls [page 314].

*   Field controls that present data to users based on their privileges and the placement of the field. Refer to Controlling Access to Data [page 244].

*   Buttons, hyperlinks, and other controls. Refer to Other Controls [page 315].

*   Detail controls to add supplementary information to your form, such as change history, notes, and time tracking. Refer to Detail Controls [page 312].

You can also modify the default toolbar and create custom transition controls. Refer to the following sections for details.

About Tool Bar Controls

Tool bar controls are available on every state and transition form by default. The tool bar shows the Item ID and project information at the top of forms and also contains the button bar and action bar.

You can customize the tool bar on custom forms for primary and auxiliary tables. For example, you can remove the **Add Item Link** item from the **Actions** drop-down list and instead map that action to a clickable image on the form. You can make individual information and actions available only on certain forms, or remove the tool bar, button bar, or action bar entirely.

For details, refer to Tool Bar Tab of the Form Property Editor [page 298] and Behavior Tab of the Control Property Editor [page 307].

About Custom Transition Controls

You can add custom transition controls to custom forms for primary and auxiliary tables. Custom transition controls can be used in addition to the standard transition buttons in the button bar at the top of the form. For example, you might want to put a custom transition button in the middle of a form, so the user can click it without having to scroll to the top of the form, but want to keep the standard button for that transition in the button bar.

A custom transition control can also be used instead of a button in the button bar. For example, you might have a state with two outgoing transitions, **Get Info from Submitter** and **Begin Work**. You could keep the button for the **Begin Work** transition in the button bar, but remove the **Get Info from Submitter** button from the button bar and instead add a custom button for it on the other side of the form, separate from the button bar.

> **Note:** Because a sub-workflow inherits the custom forms used by its parent workflow, custom transition controls are also inherited. If you want to add or remove custom transition controls to a form in a sub-workflow, duplicate the parent form in the sub-workflow, change the controls, and associate the new form with the sub-workflow.

For details, see Behavior Tab of the Control Property Editor [page 307] and Custom Transition Control Tutorials [page 282].

Controlling Access to Data

Each field in your application is associated with a privilege section. This makes the field available to users based on their privileges. For example, users assigned to a certain role may be able to view fields in the User and Manager privilege sections, but they can only update fields in the User section.

Privilege sections play an important role in form design, but custom forms offer the flexibility to organize fields any way you choose without compromising security.

You can:

- **Organize fields based on their privilege sections**

  Create custom forms based on quick forms, and do not move fields from visual sections that differ from their privilege section.

- **Organize fields in visual sections**

  Move field controls to a visual section, which may or may not be associated with a privilege section. In this case, user privileges determine if the field is visible. For example, the *Owner* field might be associated with the Manager privilege section, but you can move it into the Standard section on the form. In this case, users who have privileges to the Manager section would see the *Owner* field in the Standard section, but users without these privileges would not.

  This applies to visual sections with similar names to privilege sections and to independent containers, such as a tab control. For example, you can add a tab control with the label **Backlog Info** and place a field from the *User* field privilege section within that tab. The field is only visible to users with privileges to work with *User* fields.

- **Organize fields in both privilege and visual sections**

  Create custom forms with auto-sections, and move some fields to visual sections that differ from their privilege section, while enabling the rest of the fields to be automatically placed in their respective privilege sections.

  Note: Fields added to your application or auxiliary table are automatically added to custom forms with auto-sections. As you add fields to your application or auxiliary table, you should review your custom forms to ensure the field appears correctly to users.

Managing Form Behavior

SBM Composer provides the following features for controlling behavior on forms:

- **Form Actions**

  Provide a powerful and flexible means of dynamically changing and controlling form behavior without using scripts, Web services, or other programs. Refer to Using Form Actions [page 245].

- **Form Widgets**

  Allow you to embed data or programs, such as REST services or JavaScripts, into custom forms. Refer to Using Form Widgets [page 255].

- **Form Extensions**

  Provide extended capabilities through packaged form actions, custom widgets, and assets, such as JavaScripts, images, and .css and .html files. Refer to Using Form Extensions [page 266].

- **JavaScripts**

  Include your own JavaScript files in custom forms and make calls to them from certain form controls. This option is intended for advanced use cases that cannot be addressed with form actions. Refer to Using JavaScript [page 267].

Using Form Actions

Form actions provide a means of dynamically changing a form based on events, such as field value changes. Form actions fire when one or more events occur and when one or more defined conditions are met.

The following table provides some basic examples of condition-driven dynamic behavior that you can implement using form actions. For more advanced examples, refer to Advanced Form Action Examples [page 252].

Event	Condition	Action
Value of *Amount Due Employee* field on an expense report form changes.	If the value of the *Amount Due Employee* field is less than $500.00...	Make the *Approving Manager* field read-only.
IT technician selects **None** in the *Server* field.	If the value of the *Server* field is "None"...	Hide the *Server Version* field.
Value of *Age* field for a customer-reported defect changes.	If the value of the *Age* field is greater than or equal to 30...	Set the value of the **Escalate** field to "Yes."
Customer types `5551212` in the *Phone Number* field.	If the number of digits in the *Phone Number* field is less than 10...	Display `Area code is required` message.
Employee selects the **Vacation** check box.	If the **Vacation** check box is selected and the *Time Remaining* field is greater than or equal to 4 hours...	Expand the **Vacation** section on the form.
Customer clicks the **Gold** button.	If the value of the *Customer Type* field is "Gold" and the value of the *Account Balance* field is greater than $40,000.00...	Set the value of the **Rate Qualification Category** drop-down list to "Premium."

No JavaScript programming is required with form actions, but JavaScripts can be specified in the dialog as a condition (such as "if a JavaScript evaluates to true or false"), and as an action (such as "execute a JavaScript").

Alternatively, you can add JavaScripts directly to a form to create dynamic behavior. For information about using JavaScript to add dynamic features to forms, see Using JavaScript [page 267].

Working With Form Actions

Use the **Form Actions** dialog to action expressions that define events, conditions, and actions that result in dynamic behavior.

Consider the following information as you work with form actions:

- You cannot use a form action to set specific values for *User*, *Multi-User*, and *Multi-Group* fields because these values are not defined in SBM Composer.

  You can, however, set the value for *User* and *Multi-User* fields to the "current user" if the user is a valid value for the field. This includes *User* and *Multi-User* fields that are searchable. If the current user is not a valid selection, the action is ignored.

- You cannot use a form action to set the value of certain system fields such as *Submit Date* or *Owner*. If you try to set values directly using the `SetFieldValue` or `SetFieldValues` method in the SBM JavaScript library, the value is ignored when the form is submitted.

- You can order actions on the Actions Tab of the Form Property Editor [page 301]. However, if two actions conflict with each other (for example, two actions that specify different background text colors), the last action in the list that was executed takes effect.

- You cannot use the "a rule evaluates to true or false" condition for forms that are based on auxiliary tables.

- Some events, conditions, and actions used in action expressions are not available for print forms.

- Form actions are incompatible with the **Use Accessible Interface** User Profile display option in Work Center.

- You can specify an action to occur if a rule or JavaScript condition is false. For example, you can configure a message to appear if a rule that triggers a transition is false, to explain why the transition did not occur.

Adding or Editing a Form Action

To add or edit a form action:

1. Open the form.

2. Click the **Actions** tab in the form Property Editor.

3. To add a new behavior, click **New**. To edit an existing behavior, click **Edit**.
   The **Form Actions** dialog opens. There are five sections in the dialog:

   - **When** is used to define events. At least one "When" statement is required.

   - **If** is used to define conditions.

   - **Then** is used to define actions. At least one "Then" statement is required.

   - **Add Else If** provides additional conditions to evaluate if all of the previous conditions evaluate to "false."

   - **Add Else** provides an action to take if all "Else If" statements evaluate to "false."

   - Click the **X** to delete a condition or action statement. If you remove a condition, the specified action is always executed.

4. Click **Add Event**, **Add Condition**, or **Add Action** to create statements for the events, conditions, and actions that form the action expression. To edit an existing statement, click the links and change values.

   If you create multiple conditions within a single section, by default, the statements are connected with AND logical operators. To switch to OR logical operators, hover over the section, select the down arrow that appears at the top right of the section, and then select **'Or Conditions'**. To switch to AND logical operators, select **'And**

Conditions'. All conditions within a single section will use the logical operator you selected; you cannot combine AND and OR logical operators within a single section.

If you create multiple events within a single section, the statements are connected with OR logical operators.

5. The statement can include links, such as **a field**, **expanded**, or **<**. For example, if you click the **a field** link, a list of fields is presented and you can select or search for a field. If you click the **expanded** link, a list containing "expanded" and "collapsed" is presented. If you click **<**, a list of conditional operators is presented. Click each link and make a selection to complete the statement.

> **Important:** Design elements such as fields and controls must exist on the form before they can be used in events or actions. An exception to this is a condition that specifies a field value, such as **Age > 40**.

> **Note:** If a rule is used in a condition, and the rule uses an application variable, an administrator can override the value of the variable at the project level in SBM Application Administrator. For more information, see the *SBM Application Administrator Guide* or online help.

In some cases, a dialog box prompts you to provide additional information. The following table describes the dialog boxes.

Link Text	Description
a regular expression	Select **Custom** and enter the regular expression in the box; or select a predefined expression, such as **Match an email address**. Select the **Ignore case** check box if you want to make the regular expression case insensitive.
a time	Select **Specific time** and enter or scroll to the appropriate time; or select **Now**. Click **Clear** to remove your selection and return the time to its default value.
a color	Select a color from the color palette. Click **More Colors** for additional colors and the ability to use the RBG model to specify numeric values. To reopen the palette and change a color, click the RGB value in the statement.
a string	To display specific text, enter the text. To use an empty string, select **Empty string**.

Link Text	Description
a value	The dialog box depends on the design element. For example: • For a *Text* field, enter specific text, insert a string reference (see Using the String Builder Tool [page 277]), or select **Empty string**. • For a *Binary/Trinary* field, select one of the values. • For a *Selection* field or an input control such as a list box, move fields from the **Available Values** box to the **Selected Values** box. If more than 10 values are available, you can search for values to add to the statement. • For a *User* or *Multi-User* field, you can select "None" or "Current User." • For a JavaScript, enter the source code in the box if you need to extend the functionality provided by the wizard. The user will not see the source code. If a JavaScript is used in a condition ("a JavaScript evaluates to true or false"), use the following syntax: `return expression;` **Note:** If a JavaScript has no more than one semicolon, the system will prepend `return` if it is missing.

Link Text	Description
	Alternately, you can use the string builder tool to insert references to table fields and form controls. For information about this tool, see Using the String Builder Tool [page 277].

Important:

- If the code you type in this field includes any opening or closing braces ("{" or "}"), precede each of them with a backslash ("\"). Otherwise, SBM Composer tries to interpret the text inside of the braces as the name of a field or control on the form. This prevents the form from working properly. (If you paste the brace, you are prompted whether you want a backslash to be added. If you use the Treat '{' as a literal option in the string builder tool, a backslash is automatically added.)

- If you are using string substitution to initialize a JavaScript string constant, you must use the STRING_ESC() macro. This ensures that special characters are correctly encoded for use in this context.

- Use the string builder tool instead of including references to controls, fields, or other form artifacts in the JavaScript source code.

- When you create a generic form action or custom control, do not reference another control that exists on the form because the form action or custom control will be portable across forms that may not have the same fields and controls.

6. The **Enabled** check box is selected by default if the action expression is valid. If you do not want the action to be used right away, clear this check box.

7. Click the **OK** button to complete the expression and add it to the **Actions** tab so it can be used by the form.

Tip: To see a textual summary of the action, select the **Show action summaries** check box at the bottom of the **Actions** tab.

Copying Expressions

You can copy expressions within a form action and then modify them. For example, if you have an expression that enables certain fields depending on the value of a selection field and you want other fields to be enabled when the selection field contains a different value, you can duplicate the expression and modify the condition, action, or both as needed.

To copy an expression:

1. Select the action and then click **Edit**.

2. Rest your cursor over the section with the "if" and "then" statements and then click the down arrow at the top right of the selected section.

3. Do one of the following:

   - Select **Copy Conditions and Actions to new Else If** if you want to copy both the conditions and the actions.

   - Select **Copy Actions to new Else** if you want to copy only the actions.

4. Modify the new expressions as needed.

> **Note:** You can also copy form actions. For details, refer to Reusing Form Actions [page 251].

Reusing Form Actions

You can reuse a form action by exporting it to an .xml file and importing it into any other form. This is useful when you have multiple forms for which a single action is relevant or for limiting the amount of rework across multiple applications and process apps.

You can also copy a form action from a form in the same process app or in a referenced application and paste it into the open application.

To export and import a form action:

1. Navigate to the form in the source application, and then select the **Actions** tab.

2. Right-click the action in the source form, and then select **Export**. To export all of the form's actions, select **Export all**.

3. In the **Save As** dialog box, navigate to the directory where you want to save the exported file, and click **Save**.

4. Navigate to the target form in any application. The **Actions** tab should be automatically selected.

5. Right-click on the actions tab, and then select **Import**.

6. In the **Import Form Actions** dialog box, navigate to the file you exported and click **Open**.

7. SBM Composer attempts to resolve references to fields, controls, images, or rules. If references cannot be resolved, the action is disabled and you need to manually repair it.

To copy a form action:

1. Navigate to the form in the source application, and then select the **Actions** tab.

2. Right-click the action in the source form, and then select **Copy**.

3. Navigate to the target form. The **Actions** tab should be automatically selected.

4. Right-click on the actions list, and then select **Paste**.

5. SBM Composer attempts to resolve references to fields, controls, images, or rules. If references cannot be resolved, the action is disabled and you need to manually repair it.

Advanced Form Action Examples

- Using Transitions in Form Action Conditions [page 252]

- Calling Other Actions [page 253]

- Comparing and Copying Field Values in a Form Action [page 253]

Using Transitions in Form Action Conditions

You can use a condition in a form action that compares against the current transition. This enables you to create different behaviors for different transitions. You can then use a form with the same visual design for multiple transitions rather than create a different transition form for each unique form action.

For example, you can create a form action that sets a *Manager* field as required when the *Customer Priority* field is set to "Critical." You may want this action to fire only for the "Evaluate" and "Assign" transitions, but not for the "Complete" transition.

In this case, add this condition to your action expression:

the current transition is/is not in the specified list of transitions

You can then specify transitions for which the action is valid or invalid, and the form action is then fired only for applicable transitions.

You can also use "Else If" conditions to build more complex form actions based on the current transition. For example, you may create an action that shows or hides fields based on specific transitions:

This particular action hides the *Due Date* field for all transitions except "Assign" and "Approve."

Calling Other Actions

You can create a form action that calls another form action on the same form. This enables you to use a form action in different circumstances, without having to repeatedly create the action expression.

For example, you can create an action called "Show Manager Section Based on Priority." This action shows the *Manager* field section when the *Priority* field is set to "critical" and hides the *Manager* field section when any other value.

The action uses the event "When this action is called," as shown below:

You can then create other actions that call the "Show Manager Section Based on Priority" action based on a different action expression. For example, you can create an action that calls the "Show Manager Section Based on Priority" action when the value of *Priority* field changes.

Comparing and Copying Field Values in a Form Action

You can create a form action that compares the values of two fields. This enables you to create different behaviors based on if the values match.

For example, you can create a form action that compares two text fields when the Compare button is clicked. Depending on if the values match, the message "Values are the same" or "Values are different" appears.

```
When
    Compare_Button button is clicked
        Add Event
If
    Text1_EditBox control equals Text2_EditBox control's value
        Add Condition
Then
    pop up message 'Values are the same.'
        Add Action
Else
    pop up message 'Value are different.'
        Add Action
    Add Else If
```

You can also create a form action that assigns a value to a field using another field's value. This enables users to copy the contents of one field to another, saving time and preventing repetition. If users need to modify the field's copied value, this does not change the originating field's value.

For example, you can create a form action that sets the "Actual Start" date/time field to the value of the "Proposed Start" date/time field when the Copy Date button is clicked.

```
When
    Copy_Date button is clicked
        Add Event
Then
    set Actual Start field value/text to Proposed Start field's value
        Add Action
    Add If
```

The list of fields and controls available as a right-hand parameter is restricted by the field or control chosen as the left-hand parameter. The following guidelines apply to both comparing and copying field values:

- Left-hand *Text* fields, Buttons, EditBoxes, HyperLinks, Images, and Text controls/labels can be used with any right-hand field or control.

- Any left-hand field or control can be used with a right-hand *Text* field, Button, EditBox, HyperLink, Image, or Text control/label.

- Left-hand *Single Relational*, *Multi-Relational*, *User*, *Multi-User*, *Single Selection*, and *Multi-Selection* fields, as well as left-hand ComboBox and ListBox controls, cannot be used with any right-hand field or controls.

- *Binary* fields cannot be used with *Trinary* fields, and *Binary/Trinary* fields can be compared only with other *Binary/Trinary* fields that share the same control style (drop down, radio buttons, or checkbox).

- *Date/Time* fields are compatible if they share the same control style (date and time, date only, time of day, or elapsed time).

- A left-hand *Sub-Relational* field can be used with any right-hand field compatible with the field it references.

- All left-hand fields and controls not mentioned in the preceding items can be used with right-hand fields and controls of the same type.

- *Numeric* and *Summation* fields are cross-compatible.

- *Text* and *RichText* fields are cross-compatible. When a *Text* value is assigned to a *RichText* field, its line breaks are replaced with <br> tags. When a *RichText* value is assigned to a *Text* field, all markup tags are removed except for <br> tags, which are replaced with line breaks. *RichText* line breaks achieved by other means are lost in the conversion. Comparing the values of *Text* and *RichText* fields is not recommended.

Using Form Widgets

Form widgets are small applications that provide a unique function when placed on custom forms. For example, the Embedded Report widget adds a report to a form so users can view and interact with data from other items as they work with the form.

In the form editor, the palette contains widget types. Drag widgets onto the form, then use the tabs in the Property Editor to view and modify the various aspects of the selected widget.

Use the controls on the **Design** tab of the Ribbon to set the alignment and size of the selected widget. The **Alignment** options (left, right, top, bottom, center, fill) determine which handles are available for "stretching" the selected widget to the desired size. The **Size** options provide precise control.

For information on using the most common widgets, refer to:

- Using the Embedded Report Widget [page 256]

- Using the Relational Grid Widget [page 257]

- Using the REST Grid Widget [page 260]

- Using the PDF Widget [page 263]

Additional Widgets

Widget	Description
HTML/ Javascript	Lets you embed almost any valid HTML or JavaScript into a custom form. Refer to HTML/JavaScript Widget [page 327].
Web Page	Lets users view the content of an HTML page. Refer to Web Page Widget [page 339].

Using the Embedded Report Widget

The Embedded Report widget shows an SBM report on a transition, state, or print form. Because the report is embedded, users can view relevant information without having to navigate away from the form. For example, when the Embedded Report widget is based on a relational field, fields in a related item can be displayed in the widget.

The report can be:

- Defined in SBM Work Center. Refer to Referencing Reports Defined Outside SBM Composer [page 256].

- Defined in SBM Composer as an application report run against a primary or auxiliary table.

- Defined in SBM Composer as a report of related items based on specific relational fields.

For details on configuring reports for the widget, refer to Embedded Report Configuration Dialog Box [page 324].

> **Note:** For print forms, printing software may only print the current page of results, and will cut off content hidden by scrollbars on embedded reports. Consider using auto height to ensure the frame can expand freely.

Referencing Reports Defined Outside SBM Composer

When you embed a report defined in SBM Work Center, you should use a special report URL that contains a report reference name. This preserves the report URL if the process app is promoted to another environment (such as a production environment).

If you do not use the report reference name, the report URL will contain a report ID and field IDs (for Query at Runtime fields). Report IDs and QAR field IDs change when a process app is promoted to another environment, so the report will not work in the new environment.

To reference a report:

1. In SBM Work Center, create the report you want to reference.

2. Save the report, and in the **Save As** section on the page, type a report name in the **Title** box, and type a reference name in the **Reference Name** box. The reference name must be unique within an application, but can be the same as the report title.

   > **Restriction:**
   > - The **Reference Name** box is only present if you have "Regular User" product access with the Remote Administration privilege (granted in SBM Application Administrator) or if you have "Managed Administrator" product access with the Deploy privilege (granted in SBM Application Repository).
   >
   > - This box is disabled if the **Privilege Category** is **Private**.

3. In the confirmation message, click **Reference Link**. A message opens that contains the report URL.

4. Alternatively, run the report, click the **Copy URL to Clipboard** icon, and then select **Reference Link** in the dialog box that opens.

5. Press CTRL+C to copy the report URL to the Windows Clipboard.

6. In SBM Composer, drag an Embedded Report Widget [page 323] from the **Form Palette** to a custom form.

7. Click **Configure Report** on the **General** tab of the widget Property Editor.

8. Paste the URL into the Embedded Report Configuration Dialog Box [page 324] that opens.

9. The report URL may contain query-at-runtime parameters that enables users to select search criteria when the report runs. To map these parameters to these values, click the **Query** tab in the widget Property Editor, and follow the instructions in Using the String Builder Tool [page 277].

> **Tip:** When you embed a query-at-runtime report URL, note the following options:
>
> - You can include the part of the URL that specifies values for the query-at-runtime fields:
>
>   `"&HasRuntimeParams=1&F_TS_CONTACT="`
>
>   ...and bind form field values to the query-at-runtime parameters.
>
>   For example, if you added the Contact field to the form, you could bind the value of that field to the `&F_TS_CONTACT=` parameter, and the report on the form would show all the defects for this contact.
>
> - You can include the part of the URL that specifies values for the query-at-runtime fields:
>
>   `"&HasRuntimeParams=1&F_TS_CONTACT="`
>
>   ...and not bind form field values to the query-at-runtime parameters.
>
>   In this example, the report engine runs the report as if no value was selected in the query-at-runtime parameter for the report, and that part of the report query is ignored when building the report.
>
> - You can remove the part of the URL that specifies values for the query-at-runtime fields. This means SBM will run the report and ask the user for query-at-runtime field values.

Using the Relational Grid Widget

The relational grid enables you to replace *Single* and *Multi-Relational* field controls on custom state and transition forms with an embedded listing report that displays the related items and their fields (as opposed to viewing only the **Value display format** for related items). This enables users to view relevant information without having to navigate away from the form.

The report can be:

- Defined in SBM Composer as a report of related items based on specific relational fields.

- Defined in SBM Work Center.

In a state form, users can view the list of related items and view specific fields on those items in a columnar format (depending on the fields that are defined in the report definition).

In a transition form, users can select an item using a radio button control for *Single-Relational* fields, or they can select multiple items using check boxes for *Multi-Relational* fields, which updates the underlying relational field when the transition is finished.

For a report that is defined in SBM Composer, you can choose which fields to display, what search filter to apply to limit candidate records, and query at runtime values that enable dynamic filtering of related items at runtime.

Setting up a Relational Grid

Before you begin, determine the application and table that are associated with the relational field that you want to display using the grid. You can find these details on the **Options** tab of the relational field.

To set up a relational grid:

1. Create a **Report Definition** in SBM Composer.

   a. In the application you noted earlier, right-click on the associated table, and then select **Create new Report Definition for this table**.

   b. Drag fields from the **Report Palette** to define the columns that you want to display, the sort order to apply, and create a search filter to limit the items that are returned in the report.

   c. Optionally, use a query at runtime parameter in your search filter to dynamically filter the results at runtime.

2. Alternatively, create a report in SBM Work Center.

   a. In SBM Work Center, create the report you want to reference.

   b. Save the report, and in the **Save As** section on the page, type a report name in the **Title** box, and type a reference name in the **Reference Name** box. The reference name must be unique within an application, but can be the same as the report title.

   > **Restriction:**
   > - The **Reference Name** box is only present if you have "Regular User" product access with the Remote Administration privilege (granted in SBM Application Administrator) or if you have "Managed Administrator" product access with the Deploy privilege (granted in SBM Application Repository).
   >
   > - This box is disabled if the **Privilege Category** is **Private**.

   c. In the confirmation message, click **Reference Link**. A message opens that contains the report URL.

   d. Press CTRL+C to copy the report URL to the Windows Clipboard.

3. Configure the relational grid widget.

   a. Drag and drop a **Relational Grid** widget on to your form.

b. On the **General** tab, click **Configure**.

c. Select the relational field and report definition that you defined earlier, or paste the report URL if you created a report in SBM Work Center. If the target table is in a different process app and the application has not been referenced, you must add an application reference to make the report available. For details, see Chapter 17: Creating Application References [page 369].

d. **Select Automatically bind compatible parameters** to have SBM Composer attempt to map query at runtime parameters for you.

e. Click **OK** to finish.

4. Configure the report query to filter results.

a. Open the **Query** tab.

b. Select a project constraint to limit results to a particular project.

c. Map controls to the widget to enable dynamic filtering. You can add **EditBox** and **Button** controls to the form, and then map a query at runtime field in your report to the EditBox. This creates a search field that exposes the query at runtime capability. For example, if your report contains a search filter like:

```
Title contains '(Query at runtime)'
```

Add an **EditBox** (named `EscalatedIssue` in this example), and then map it to the **Title** field:
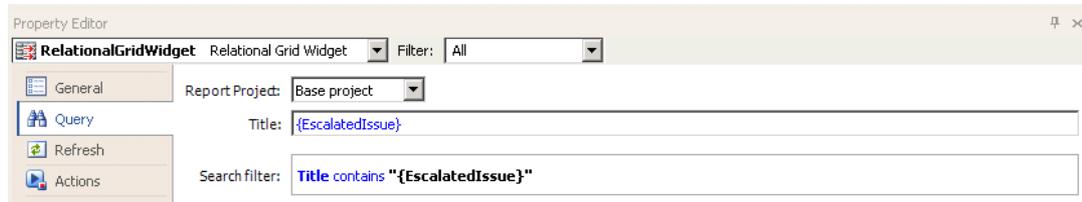
d. Open the **Refresh** tab, select the button control that you added, and then select **On click** to refresh the widget when the button is clicked.

At runtime, when a user enters text in the search field and clicks the button, the results will be filtered by the text that is entered.

Using Selected Values in the Relational Grid

This section provides a brief example that demonstrates how you can use values in selected items in the grid.

In transition forms, you can map the value or values of selected records to populate other controls. For example, you can add a **Text** control that displays the currently selected *Severity* values in the target table.

On the **Refresh** tab of the **Text** control, type { in the **Display text** field, and then select the relational grid and field:

Select **On data change** to have the text field contents update whenever the search button is clicked.

For *Single Relational* fields, because the display text of the text control can contain multiple references, you can construct any string from the contents of the related fields.

If the relational grid is associated with a *Multi-Relational* field, `{GridWidgetName.ColumnName}` is replaced by a comma-separated list of the values from the referenced field.

Considerations for State Forms

On state forms, the relational grid displays records that are not selected in the relational field unless you filter the results in the grid to only display records that are referenced by the relational field.

You can accomplish this by creating a relational field in the child item that points back to the parent item. You can then create a report that displays only the selected records by using a query such as:

```
Title contains (Query at runtime)
AND
ParentItem in (Query at runtime)
```

In the widget's **Query** tab, add the parent's record ID to restrict the results:

```
ParentItem: {_RecordId}
```

This ensures that the relational grid displays only the records contained in the *Multi-Relational* field when it is used on a state form.

Using the REST Grid Widget

Use the REST Grid widget to pull data from any Web service that supports the REST format and display that data on custom forms.

For advanced REST grid functions, see the *SBM JavaScript Library Guide*.

REST Grid Widget Examples

Example 1

As part of creating a product incident request, a service representative needs to select the product that an incident is related to from a list. Instead of replicating the product data in SBM to populate the list, use a REST Grid widget to retrieve the information dynamically from the company's product catalog. The REST Grid widget can also display product images, so the representative has a visual way to select the product.

Example 2

A writer for a financial magazine has an item assigned to him for each date he needs to submit an article for publication. The custom form for the **New** state has a REST Grid widget that returns the financial questions people asked on the magazine's Web site. The writer uses these questions as ideas for the content of the article.

Follow these steps for basic widget configuration:

1. Add or edit a custom state or transition form.

2. Drag the **REST Grid** widget from the **Widgets** section of the **Form Palette** to the form editor.

3. Complete the **General** tab of the widget Property Editor:

   a. Type a name in the **Control name** box.

   b. In the **Caption** box, provide information you want to show in the widget title bar.

   c. Click the **Configure URL** button. The **REST Service Configuration** dialog box opens.

   d. Configure the REST service, using the information in REST Service Configuration Dialog Box [page 468] for guidance.

4. On the **Result** tab of the widget Property Editor, choose to render the widget as a grid or tiles, then move **Available data** items to the **Grid columns** or **Tile Content** pane.

5. Select other options on the **Result** tab, using the information in Result Tab [page 332] for guidance.

Binding to REST Widget Data

In the REST Grid widget, users can access the data in the grid to populate controls such as edit boxes. Users select the row in the grid whose data they want to put in the control.

> **Note:** If the form is a primary table form (for example, in a workflow), then only primary table data can be accessed. If the form is for an auxiliary table, then only auxiliary table data can be accessed.

To bind to widget data:

1. Drag a widget from the **Form Palette** to the form and configure it.

   > **Note:** For information about configuring the REST Grid widget, see REST Grid Widget [page 330].

2. Drag a control such as an edit box from the **Form Palette** to the form and give it a label. For example, if you are going to populate the edit box with addresses from the REST Grid widget, give the label the name `Addresses`.

3. Enlarge the widget so that it consumes the whole form. To do so, perform the following steps:

   a. Select the form. Sizing indicators are displayed for each column and row.

b. Click the sizing indicator for the column containing the widget until it changes to **%**. Repeat this step for the row.

c. Select the widget and click the **Autofill Vertically** icon in the **Alignment** area.

> **Note:** For more information about resizing, see Resizing Columns and Rows [page 274].

4. In the Property Editor for the widget, on the **Results** tab, select **Enable row selection**, if it is not already selected.

> **Note:** If you clear this check box, the **Grid columns** table contains an additional **Action on click** column. This option is not available if you are binding to widget data, because when you bind to widget data, there are other links, and it would be unclear what you are clicking.

5. Still in the Property Editor, on the **Results** tab, move fields you want to bind from the **Available data** table to the **Grid columns** table.

6. Select the control on the form.

7. In the Property Editor, click the **Refresh** tab.

8. In the **Contents** field, use the string builder tool to enter the grid data you want users to select to populate this edit box.

> **Note:** For information about using the string builder tool, see Using the String Builder Tool [page 277].

Users can now select a row of data, and the control is populated with that data.

Intercepting and Manipulating REST Data

In SBM Composer, you can intercept data retrieved through the REST Grid widget and manipulate it before it is displayed on the form. For example, you can use this in situations where the REST calls do not return data in a format that the REST Grid is capable of processing. Another example is where the REST caller runs different reports that return different columns, and you need to map specific columns to data on the form before you display it.

To intercept the data, you use the `AddResultCallback()` function call in JavaScript. You can then use additional JavaScript code to manipulate the data as needed. For example, if your REST data returns data in a hierarchical way through an array, you could use JavaScript to convert it to a flat data structure so that it can be displayed as a list format in the REST Grid widget.

For information on `AddResultCallback()`, refer to *SBM JavaScript Library Guide*.

You can include the JavaScript in one of the following ways:

- Attach a JavaScript file to the form. Refer to Using JavaScript [page 267].

- Place an HTML/Javascript widget on the form. The HTML/Javascript widget must be placed farther down on the form than the REST widget with which it interacts. Also, since Javascript attached to a form run after any scripts included in an HTML/ Javascript widget, they can overwrite changes made from within HTML/JavaScript widgets. Refer to HTML/JavaScript Widget [page 327].

Using the PDF Widget

Use the PDF widget to provide a PDF document with application data. You can embed the PDF document in a custom form or open it in a new browser window.

The widget includes standard PDF document features, such as printing and saving to file. This means users can have a printed version of application data, and can e-mail a saved version of the PDF document containing the application data to others.

The document presented to users is based on a template added to the PDF widget on your form. You can map fields in your application to fields in the template, or you can provide static values. For guidelines, refer to Field Mapping Rules [page 264].

The following guidelines apply to the PDF widget:

- PDF templates created with Adobe® Acrobat® 9 Pro, Adobe® Acrobat® X Pro, and Adobe® Acrobat® XI Pro are supported. The PDF documents must be PDF version 1.4 or later.

- To view PDF documents and use their features, users must have Adobe® Reader® installed on their computers. To view PDF documents in preview mode or test your process app, you must have this application installed as well. If this application is not installed, a link to the Adobe Reader download page is displayed.

- To view PDF documents inside of an SBM form, users must use a browser that supports the Adobe Acrobat plug-in, as well as have Adobe Reader installed. Mozilla Firefox and Microsoft Internet Explorer support the Adobe Acrobat plug-in and display PDF documents as expected. It is highly recommended that users use the latest versions of Firefox or Internet Explorer and the Adobe Acrobat plug-in.

  Note that the Adobe Acrobat plug-in is not compatible with Google Chrome and Microsoft Edge. See the Adobe website for information on which browsers are compatible with the Adobe Acrobat plug-in.

- In preview mode, the PDF template is displayed, but the input fields are not populated with runtime data.

- The input field names can contain a maximum of 128 characters.

- To support Unicode characters, make sure that "plain text" is not enabled for fields during the creation of the PDF document.

- If the number of characters in a text field is too large, scroll bars appear so users can see all of the text. However, if this text field is mapped to an input field in a PDF template, the extra text is truncated and cannot be seen. Similarly, if users add an image that is too large for the mapped input field, it will not be displayed in the PDF document.

  To prevent this, make sure the input field in the template is large enough to accommodate the expected amount of text or an image.

- By default, the PDF widget supports application data that contains rich text. For details, see Rich Text Support [page 265].

- Users can manually edit a PDF document after it has been populated by SBM field values; however, any changes made will not be synchronized back to SBM. Note that text fields without rich text are editable, while text fields with rich text are read-only.

For a sample template, refer to S137510 in the Knowledgebase.

Field Mapping Rules

All SBM field types can be used in the PDF widget, except for *Summation* and *Secondary Owner*.

All special fields are supported. These fields are available from the string builder tool. (For more information, see Using the String Builder Tool [page 277].) Special fields include the following:

- `_height`

- `_width`

- `_ProjectID`

- `_RecordID`

- `_TableID`

The following PDF field types are supported:

- Check Box

- Drop-down List/Combo Box

- List Box

- Radio Button

- Text

SBM fields can be mapped to any PDF field type, but the values must match exactly. For example:

- If you are mapping a SBM *Single Selection* field to a PDF List Box field type, you must create the field with the same values in each application. For example, if you add **Defect**, **Enhancement**, and **Internal Task** as values for a *Single Selection* field called "Request Type" in SBM Composer, you must enter the same values as list items in the List Box field in the PDF form.

- Similarly, if you are mapping a SBM *Single Selection* field to a PDF Radio Box field type, the values must match. If you add **Vacation**, **Sick**, and **Jury Duty** as values for a *Single-Selection* field called "Time-Off Type" in SBM Composer, you must enter the same values as item names in the PDF form.

  Note: An exception to this rule is when you map a SBM *Binary* field to a Check Box PDF field type that is used to toggle between two settings. In this situation, the mapping is handled internally, and the values do not have to match. There is no adverse effect if the PDF form requires values of "On" and "Off", and the SBM *Binary* field requires "Yes" and "No."

Rich Text Support

The PDF widget automatically displays application data using rich text in addition to plain text. Rich text is available for *Memo* and *Journal* fields.

> **Note:**
>
> - To display rich text in the PDF widget, you must map *Memo* and *Journal* fields to PDF text fields with the **Multi-line** property enabled.
>
> - For *Journal* fields, only the first entry appears in the resulting PDF. Additional information about the author of that entry (including the name, user icon, and date/time stamp) is not included.

To disable rich text, set the **RichTextEnabled** option to **false** in the `config.properties` file, located in the following directory:

installDir`\SBM\Common\Tomcat` *x.x*`\server\default\webapps\DocGenerator\WEB-INF\classes`

The following rich text options are supported:

- Font

- Font size

- Font color

- Background color

- Bold

- Italic

- Strikethrough

- Underline

- Superscript/subscript

- Hyperlinks

- Bullet list

- Source code

> **Note:** In SBM Work Center, users can click the source code option in the Rich Text Editor to manually insert HTML tags. If users use this option to insert a table, the table is displayed in the PDF document with fixed-width columns, even if the table columns in SBM were specified with different widths.

The following fonts are supported:

- Arial Black

- Courier

- Courier New

- Comic Sans MS

- Helvetica

- Impact

- Lucinda Grande

- Lucinda Sans

- Tahoma

- Times

- Times New Roman

- Verdana

If a non-supported font is used, the PDF widget uses the default font specified in the `config.properties` file. By default, Arial is used.

Plain Text Formatting

For plain text output by the PDF widget, you can customize the formatting by editing the `html_formatter.properties` file, located in the following directory:

installDir\SBM\Common\Tomcat *x.x*\server\default\webapps\DocGenerator\WEB-INF\classes

This file contains the HTML tags used in PDF documents and their default values. See the file for instructions on how to customize these values.

Error Messages

The following error messages could be generated during the configuration of the PDF widget.

- `PDF Template does not contain field with name.`

- `PDF Template contains field of unknown type.`

- `Invalid values.`

- `No field mapping found in the PDF Template.`

- `Unable to read fields from PDF Template.`

To correct these problems, modify or replace the PDF document being used as the template.

Using Form Extensions

Form extensions are packages that contain custom form actions, custom widgets, and assets, such as JavaScripts, images, and .css and .html files. Form extensions are available from Community website.

You can add form extensions to an application, then use any of its elements on custom forms.

To import a form extension:

1. Download the extension from Community website.

2. In SBM Composer, select **Extensions** from the App Explorer, and then right-click **Form Extensions**.

3. Select **Add New Form Extension**.

4. Click **Add to Library**. This ensures the extension is available for other process apps.

5. Navigate to file you downloaded, then click **Open**.

6. Review the documentation provided on the editor to learn about the extension's capabilities.

> **Note:** You will receive a warning if the extension you are adding was developed in a later version of SBM Composer than you are using. You can continue to add the extension, but certain features may not be available.

7. Click **Add** to add the extension to your process app.

You can also add form extensions directly to your local library:

```
My Documents\SBM Composer\Library\Extensions
```

This ensures extensions are available for every process app.

Using JavaScript

If you are unable to create a form action that meets your needs, you can include JavaScript files in custom forms and make calls to them from certain form controls. The scripts are executed when the form opens or when the form control is clicked.

JavaScripts are executed at runtime. To execute JavaScripts during form preview in SBM Composer, select the **Enable JavaScripts** check box in the Form Preview Dialog Box [page 317].)

Perform the following procedure if you want to use the JavaScript across forms. For a single form, use the HTML/JavaScript Widget [page 327].

To add a JavaScript file to an application:

1. In App Explorer, open the application to which you want to add a JavaScript file.

2. Right-click the **JavaScripts** heading, and select **Add New JavaScript**.

3. Do one of the following:

   • To create a new JavaScript file, type or paste the source code into the empty box.

   • To import a JavaScript file, click **Import**, locate the file, and then click **Open**. The source code is displayed, and the path to the file is added to the **Source file** box.

You can use the JavaScript file when you select the following:

- **Custom form**: In the **JavaScripts** tab of the form Property Editor, you can select the file from the **Add** list. You can also import a JavaScript file and include it in the form by clicking **Import**. See JavaScripts Tab of the Form Property Editor [page 304] for details.

  > **Note:** You cannot directly add JavaScripts to a print form; however, you can include JavaScript code in print form actions. Refer to Using Form Actions [page 245].

- **Button, HyperLink, or Image control**: In the **URL** field on the **General** tab of the control Property Editor, you can make a call to a JavaScript file. For buttons, enter the function to be called for the `onclick` event. For **HyperLink** and **Image** controls, create a JavaScript URL by typing `javascript:function()`.

Previewing Forms

You can preview state, transition, and print forms before you deploy them. You can:

- Choose to preview forms as they will appear in SBM Work Center or on different mobile devices (if the application is enabled for mobile access).

- Preview forms as they will appear to users assigned to different roles and in different contexts, such as a particular state. You can also preview forms without any role restrictions.

- Specify whether JavaScripts included in the form should be executed or suppressed during form preview.

- Preview the form as it will appear in different Web browsers (Internet Explorer, Mozilla FireFox, or Google Chrome).

- Resize a responsive form to preview how it will appear in different screen sizes. See Working With Responsive Forms [page 269].

To preview a form, do one of the following:

- Select the form in the Forms list, and then click **Browser** in the **Preview** area of the **Design** tab. For mobile forms, click **Mobile** and select a mobile device type from the list.

- Right-click a state or transition, and then select **Preview Form**.

- Select a workflow, state, transition, or auxiliary table, and click **Preview** on the **Form** tab of the Property Editor. Alternatively, you can press F5 while in the form editor.

The form opens in the Form Preview Dialog Box [page 317].

Using Preview Mode to Test Your Workflow

In preview mode, you can navigate through the state and transition forms in your workflow so you can experience the user flow without having to deploy the process app and test it in a runtime environment.

To test your workflow:

1. In the Workflow editor, right-click a state near the beginning of your process, and then select **Preview State Form**.

2. Click a transition control on the form. The form associated with the transition opens.

   Note: By default, transition controls are buttons that are automatically placed at the top of the form. You can replace these buttons with transition controls (buttons, hyperlinks, or images) that can be placed anywhere on the form. For more information, see Behavior Tab of the Control Property Editor [page 307].

3. Click **OK** to mimic the transition completion or click **Cancel** to return to the state form.

4. Continue to click transition controls on forms as you navigate through the workflow.

   Note: This feature is not available when you select a form in App Explorer and then click **Browser** in the **Preview** area of the **Design** tab. It is only available from the workflow, as described in this procedure.

Tips for Working With Forms

The following topics provide helpful tips for working with forms.

- Working With Responsive Forms [page 269]
- Using Headers, Footers, and Sidebars [page 272]
- Working With Mobile Forms [page 273]
- Resizing Columns and Rows [page 274]
- Spanning Columns and Rows [page 276]
- Selecting Parent Controls or Cells on a Form [page 277]
- Using the String Builder Tool [page 277]

Videos:

- Copying Forms Between Applications
- Updating Fields on Custom Forms

Working With Responsive Forms

Responsive forms, available only for modern forms, enable the layout to adjust to the screen size by wrapping rows to new lines as needed. For smaller screen sizes like tablets and mobile phones, responsive forms eliminate the need to scroll horizontally to view the contents of the form. All scrolling is vertical.

Rather than the auto/percent/fixed width used by standard forms, responsive rows split their total width into 12 implicit blocks, and distribute those blocks between their cells. These block sizes correspond to percentages of the total width.

A typical layout contains two field columns with widths of 2 - 4 - 2 - 4 blocks, as shown below:
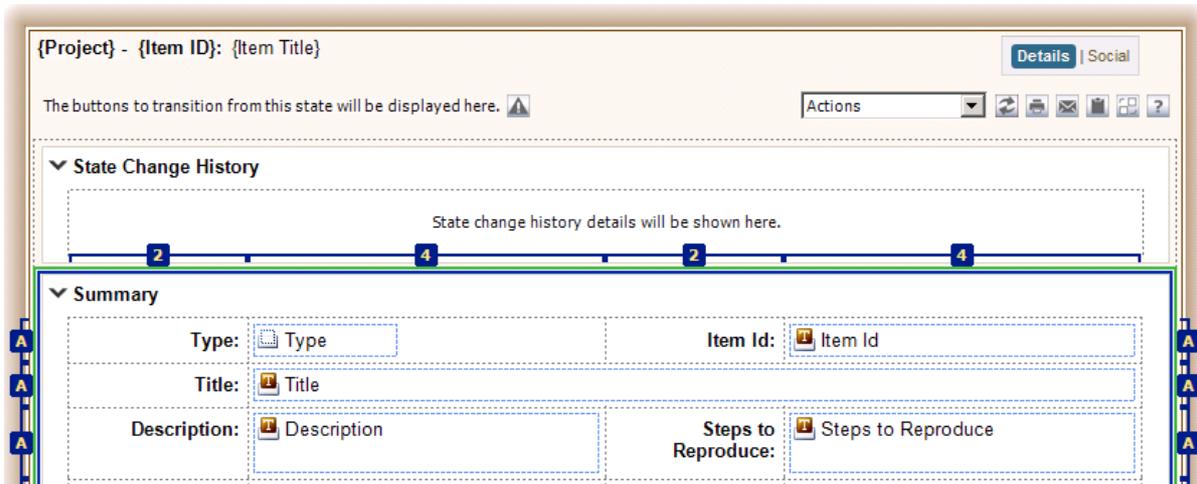
Previewing Responsive Forms

When you preview a responsive form, you can resize the form to test how it will appear in different screen sizes. Below you can see how the rows wrap as the screen width decreases:
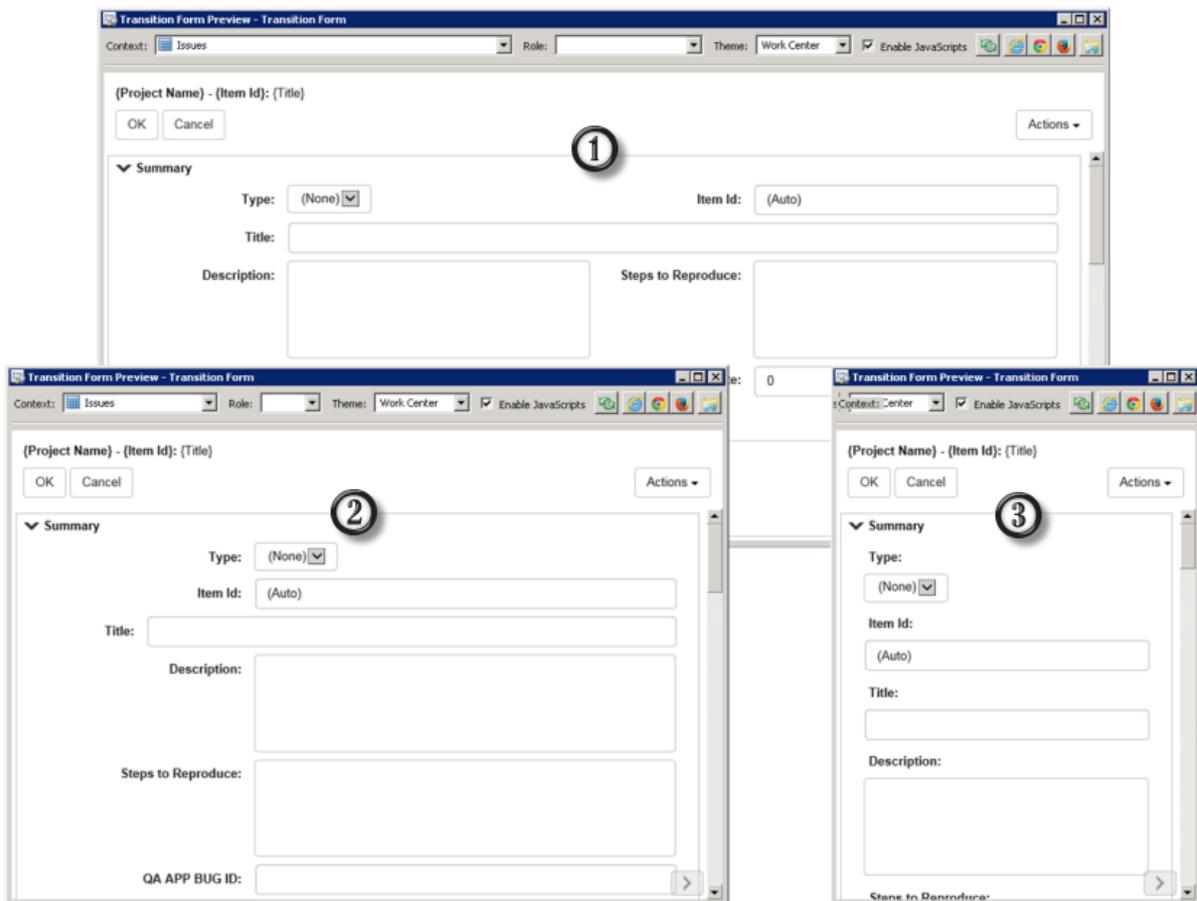
1. On screens that are 1024 pixels wide or greater, responsive cells will appear as shown in SBM Composer.

2. Between 720-1023 pixels wide, responsive cells will double in width, causing each row to split down the middle and wrap into a second row.

3. Below 720 pixels, all cells render as their own row for maximum horizontal space. All labels and fields are forced to left align for readability.

Responsive Forms and Double Cells

Responsive forms use double cells to combine the label and the field into a single element so they don't separate as cells move around at different sizes. This is simple enough for left-placed labels, but when using top-placed labels, the entire pair of rows must be combined due to HTML layout limitations. Elements other than field/label pairs will be similarly tied together and can wrap together unexpectedly.

To make sure the double cells are working as expected, preview your responsive forms and turn on the **Toggle responsive cell borders** option. When you hover over any double cells, a solid blue border extends across both linked cells, while its component controls have dashed borders.

The example below shows a horizontal double cell with a left-placed label.

The next example shows vertical double cells with top-placed labels. As you can see, there is extra content in one of the rows that is not a field/label pair. This is flagged in the Preview Tools window.

Therefore, when using top-placed labels, use only label/field pairs in the same row. This will prevent the creation of vertical double cells that contain anything other than the label/field pairs.

Notes on Working With Responsive Forms

- A form can include a mix of responsive and standard containers. On the General tab of the Property Editor, you can choose a responsive or standard layout for the selected container, or choose to use the form's layout.

- You can specify only **AutoSize** or **Fixed** for responsive row types. **Fixed** is applied as the minimum row height.

- Responsive tab controls will be converted to "pills," so that tabs with long names or many tabs can wrap more freely on smaller screens.

- You cannot place controls as new columns in responsive-layout containers. To add a new column manually, right-click and select **Add Column**, or press Ctrl+Shift+left/right arrow keys. These will be constrained to a maximum of 12 columns.

Using Headers, Footers, and Sidebars

Headers, footers, and sidebars are available with modern forms. With this feature, you can ensure that important elements remain visible on the screen at all times. For example, you may want to keep certain buttons or controls visible while users scroll or change tabs in the main form area.

After you add a header, footer, or sidebar, you can drag and drop form palette elements onto them, just as you can with the main form area.

In the example below, the toolbar has been hidden on the form. The left sidebar includes a subset of transitions and actions. The custom header reflects the type of incident being submitted.

To add headers, footers, and sidebars:

1. Select the form.

2. Click the **Appearance** tab on the form Property Editor.

3. Under Form features, select the check boxes for the areas you want to add.

4. For sidebars, you can specify the width, and for headers and footers, you can specify the height.

5. Drag and drop form palette elements to each area as needed.

6. To specify the background color or an image for the area, use the **Background** area of the ribbon. To specify these settings for all headers, footers, and sidebars, use the Styles Editor.

Headers, Footers, and Sidebars on Print Forms

Printers will cut off overflowing contents of any internal scrollbars, including the scrolling content cell normally created by using headers, footers, and sidebars. In order for the main contents to fully appear on print forms, headers, footers, and sidebars will render as ordinary (non-fixed) table elements, exactly as if you created an extra row or column instead of using them.

Working With Mobile Forms

You can enable your application to be accessible through the mobile app. The mobile app provides two views for each form, simple and advanced. By default, the simple view displays only required fields, while the advanced view displays all fields and controls (except widgets).

Additionally, you can:

- Modify which individual fields and controls are displayed

- Show or hide transitions on mobile state forms

- Create form actions with a mobile form condition

To enable mobile access:

1. Select the application in App Explorer.

2. In the Application Editor, select the **Enable options for SBM mobile app** check box.

To set the visibility of controls on mobile forms:

1. In the form, select the field, detail, container, or other control.

2. Click the **Mobile** tab in the control Property Editor.

3. Select an option to show or hide the control on all mobile forms (both simple and advanced). You can also choose to show the control only on advanced mobile forms.

To set the visibility of transitions on mobile state forms:

1. In the workflow, select the transition.

2. Click the **Options** tab in the transition Property Editor.

3. Select an option to show or hide the transition on all mobile state forms (both simple and advanced). You can also choose to show the transition only on advanced mobile state forms.

To create a form action with a mobile form condition:

1. Open the form.

2. Click the **Actions** tab in the form Property Editor.

3. Click **New**.

4. Click **Add Condition** and select "the current form is or is not a mobile form."

5. Complete the rest of the form action.

To preview mobile forms:

1. Open the form.

2. Click **Mobile** in the **Preview** area of the **Design** tab, and then select a mobile device type from the list.

3. Select an option from the **Mobile view** list to preview the form as it will be seen in the simple view or the advanced view.

Resizing Columns and Rows

You can manipulate the size of columns and rows in a selected form or container control.

You can choose one of three options for each column and row:

- **Fixed**

  Specify, in pixels, the width of a column or the height of a row.

- **Percentage**

  Specify the percentage of the remaining width that the columns or height that the rows can occupy after the fixed and autosized columns and rows are allotted.

- **AutoSize**

  Automatically sets the width of the column or height of the row to fit its contents. Justification options are not applicable to columns and rows set to autosize.

  Empty columns and rows with this sizing behavior are visible in the form editor, but are invisible in form preview mode and at runtime. In the form editor, such columns and rows have a red hatch pattern on them, and are displayed in red in the **Columns** and **Rows** tabs of the form Property Editor. A warning about empty columns or rows is generated when you validate the process app.

  > **Note:** For fixed and percentage options, if the specified size is smaller than the minimum content size, the column or row assumes the size of the content.

To see whether a column or row is fixed, autosized, or a percentage, click the form. An **F** indicates that the column or row is of fixed size, an **A** indicates that a column row is autosized, and a **%** sign indicates that the size of the column or row is measured as a percentage of the total width (height) of the container.
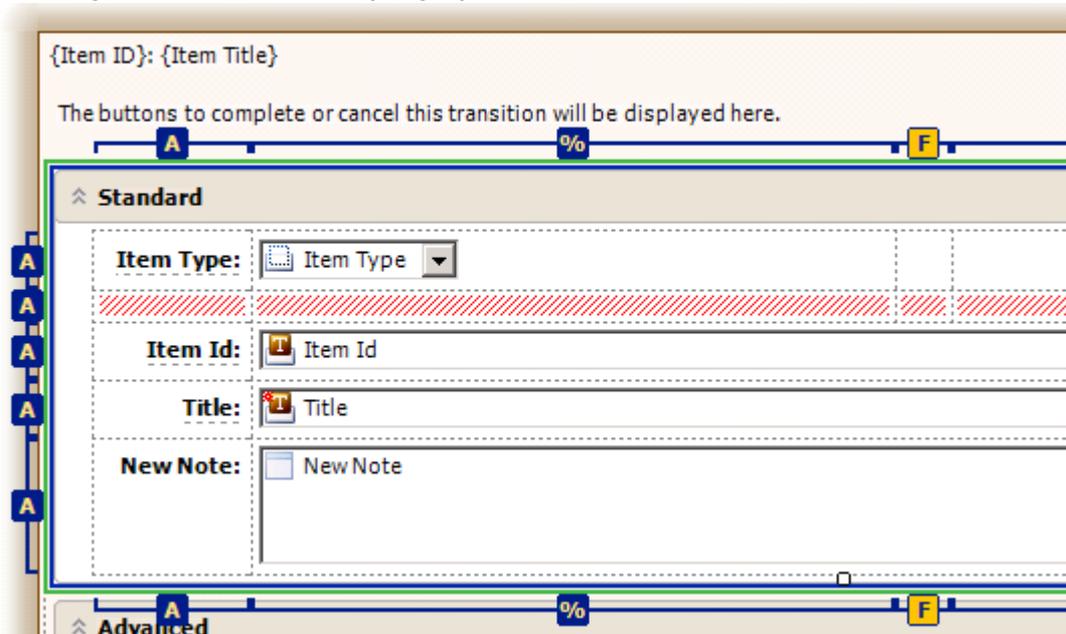
To resize rows and columns:

1. Select the form.

2. To change a sizing behavior, do one of the following:

   - In the Property Editor, select the **Type** field in one of the rows. In the menu that opens, select the sizing behavior you want.

- Click the form and then click **F**, **A**, or **%**. This toggles between sizing behaviors.

- Right-click a column or row, select **Column Sizing** or **Row Sizing**, and then select **Fixed**, **Percentage**, or **AutoSize**.

3. To resize a fixed or percentage column or row from the Property Editor, in the **Width** or **Height** column, type the number of pixels for the column or row or the percentage of the total width (height) that the column or row should occupy on the form.

4. To resize a fixed or percentage column or row by using the drag-and-drop operation, hover the cursor over the right edge of the column or the bottom edge of the row. The cursor turns into the splitter shape, and a bold dotted line highlights the edge and displays the size of the column or row in pixels or a percentage. Drag the line to change the size of the column or row.

Spanning Columns and Rows

You can span rows or columns by dragging a cell over adjacent, empty cells. You use green, circular "handles" to perform the drag operation. A black arrow on top of each handle indicates the directions in which you can drag.

If the arrow does not have a pointer, you cannot drag in that direction. For example, in the following illustration, the arrow points to the right, so you can only drag the cell to the right. The black arrow only appears when you hover over a green handle.

The following illustration shows a row that was spanned by dragging the cell to the right over three cells. The arrow is bidirectional, so you can drag the handle on the right side of the selection to the right or to the left.
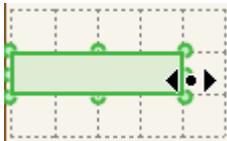
When you select a single control, you can resize the control using the square handles, or the cell using the round handles.
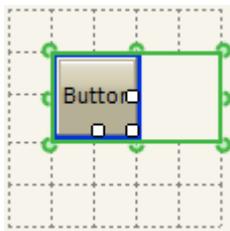
You can select contiguous cells and drag them at the same time. The following illustration shows three selected cells.
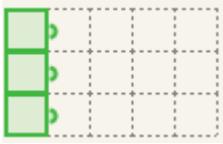
The following illustration shows the effect of dragging the three cells at the same time. There are handles at the left boundary of the form, because you could drag the cells from the left to the right.

You cannot drag the selected cell in the following illustration to the right, because it would overlap the combo box control. (The handles on the right indicate that you can drag the cell to the left to make it narrower.
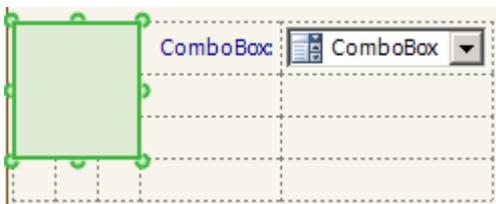
Selecting Parent Controls or Cells on a Form

From a menu, you can select a parent control or cell that you want to modify in a form. This lets you see the parent hierarchy for that control and makes it easy to select a control that is behind another control on the form.

> **Note:** You can also select a control, and then press the Esc key to select the next highest control in the hierarchy.

To select containers:

1. Select a control in a form, or the form itself.

2. Right-click and select **Select Element**. In the menu that opens, a hierarchical list of controls is displayed. The parent control is at the bottom of the list.

3. Click a control in the menu to select it in the form. If you click **(Selected cell)** or **(Containing cell)**, the cell in which the control was placed is selected.

> **Note:** If you drag a field control (or label control), the corresponding label control (or field control) is also moved.

Using the String Builder Tool

The "string builder" tool lets you insert references to table fields and form controls, and insert references to grid data in grid-style widgets. This creates dynamic forms and widgets that are driven by application and grid widget data and the actions of the user. You can combine static data and dynamic data elements in the string. The dynamic data

elements represent replaceable text that drives the appearance of input data. The dynamic data is either stored in the database table or on the form.

The string builder tool also enables you to associate a translatable string with a form control or widget. Once you have defined a translatable string on a custom form (via the **Localization** tab of the form Property Editor), the string is selectable anywhere the string builder appears on the form.

You can also use the string builder tool to insert HTML tags. This functionality is available in fields that contain text, such as the **Content** field in the HTML/JavaScript widget, and the **Display text** field in a Text control.

Note the following points:

- If you use the string builder tool in JavaScript code in the HTML/JavaScript widget, the code will be generated into an iframe (inline frame). If the code is in an iframe, the widget cannot reference other parts of the form. Therefore, the string builder tool should not be used in JavaScript code.

- The **Treat '{' as a literal** option in the string builder tool menu is used to start typing JavaScript code. To use { as a literal character, it must be "escaped" with a backslash (\). The backslash is automatically added when you select this option from the string builder tool menu.

To use the string builder tool:

1. Type { in one of the fields that says **Type { to add a dynamic value**.

2. In the menu that opens, select an appropriate application variable, control, field, endpoint, or data element.

The following examples illustrate how the string builder tool works.

- For the Web Page widget, **URL** field, you could type the URL for a social networking site, and then type {, select **Candidate First Name**, type +, type {, and then select **Candidate Last Name**. This creates a URL that takes the user (for example, a hiring manager) to the Web page for that candidate.

- For the HTML/Javascript widget, **Content** field, you could type `zips={` and then select **Title**. **Title** is a text box on the form in which users type their zip code to retrieve weather information.

- For the REST Grid widget, **URL** field, after the endpoint portion of the URL you could type {, and then select an application variable that will set the resource portion of the path. An administrator could then override the application variable value in SBM Composer on a project-by-project basis. At runtime, both the endpoint and the application variable would be resolved resulting in REST calls with different paths for different projects.

- For the HTML/JavaScript widget, **Content** field, you could type {, and then select *#EndPointName.param* to select any output parameter from an endpoint listed in the Custom Endpoint Library, where *param* could be any of the following: AuthType, ID, Password, URL, and UserName. For example, if you are programming using the SBM JavaScript Library `SecureRESTServiceWrapper` object, you will be able to reference endpoint information directly in the JavaScript using the *ID* parameter.

For information on `RESTServiceWrapper` and `SecureRESTServiceWrapper`, refer to *SBM JavaScript Library Guide*.

Managing Form Maintenance

SBM Composer offers several methods for maintaining custom forms:

- As your data needs change, you can add, remove, or reposition fields on multiple custom forms in the same application or auxiliary table. Refer to Updating Fields on Multiple Custom Forms [page 279]

- To help standardize forms within and across applications, you can copy and paste these elements:

  - **Forms** - Refer to Copying Forms [page 280].

  - **Form Controls** - Refer to Copying and Moving Controls [page 281].

  - **Form Actions** - Refer to Reusing Form Actions [page 251].

- To deploy forms independently of the process app that uses the forms, create a separate process app to store these external forms. Refer to Using External Forms [page 282].

Key Benefits

- Make custom form maintenance easy and quick.

- Standardize form styling across all applications.

- Lower the cost of customizing solutions.

Updating Fields on Multiple Custom Forms

You can add, remove, or reposition fields on multiple custom forms at once, with one click of a button. This simplifies the task of updating forms, because you do not have to perform the action manually on each individual form. For example, suppose you add a field to the primary table in a process app, and that field needs to be on all state and transition forms that are based on that table. The field is not automatically added to custom forms, as it is for quick forms. This feature allows you to quickly and easily add the field to selected custom forms in the location you specify.

> **Note:** New fields are automatically added to custom forms that include auto-sections. By default, the fields are placed and ordered in their specified privilege section.

See form updates in action

A list of all custom forms based on the containing primary or auxiliary table is presented in the field Property Editor. After you select an action (add, remove, or reposition), a dialog box opens. The list of forms in the dialog box is initially filtered according to the action. For example, if you want to add the *Severity* field, the list includes only those forms that do not already contain that field. For "add" and "reposition" actions, the list is filtered again after you select an adjacent field. For example, if you want to place the *Severity* field next to the *Business Priority* field, the forms that do not contain the *Business Priority* field will be disabled.

> **Note:** The updates are performed in a dialog box that has three modes, one for each update action. The following procedures describe how to open the appropriate mode from the field Property Editor. Alternatively, you can switch to another mode from within the dialog box.

To add a field to custom forms:

1. Select the field in the table editor.

2. Click the **Forms** tab in the field Property Editor.

3. Click **Add to Forms**.

4. Complete the Add *'Field Name' Field Type* Field to Forms [page 222] dialog box.

Removing a Field

To remove a field from custom forms:

1. Select the field in the table editor.

2. Click the **Forms** tab in the field Property Editor.

3. Click **Remove from Forms**.

4. Complete the Remove *'Field Name' Field Type* Field from Forms [page 225] dialog box.

Repositioning a Field

To change the position of a field on custom forms:

1. Select the field in the table editor.

2. Click the **Forms** tab in the field Property Editor.

3. Click **Reposition in Forms**.

4. Complete the Reposition *'Field Name' Field Type* Field in Forms [page 224] dialog box.

Copying Forms

You can copy a form in the same application or in a different applications.

To copy and paste a form in the same application, right-click on the form name, and then click **Duplicate**.

See form copies in action

When you copy and paste forms across applications, consider this information:

- SBM Composer attempts to match fields on the form to those in the target application. If matches cannot be made based on field name and database name, field controls are disabled.

- You should copy images and styles before copying forms. If images are not found in the target application, they are reset to none; if styles are not found, the default style is used.

- If a form contains controls configured to perform a transition, the list of transitions is cleared after the copy.

To copy and paste a form from a different application:

1. If you are copying a form from a different process app and a reference to the source application does not already exist, create one following the steps in Defining a Reference [page 371]. (Once you have copied forms, you can remove the reference.)

2. In the source application, right-click the form you want to copy, and then select **Copy**.

3. In the target application, right-click on the **Forms** header in the App Explorer, and then select **Paste**.

4. Choose which table to associate the form with.

5. Click **OK**.

6. Associate controls with fields as needed. To do so, right click on a control, select **Change Associated Field**, and then select a field to associate with the control. You can choose from fields that are not already used and that are compatible with the selected control.

Copying and Moving Controls

You can copy or move one or more controls from one part of a custom form to another. You can also copy or move them from one custom form to another one, even if the forms are in different applications in the same process app. This makes it easy to maintain custom forms.

To copy or move a control:

1. Open the source form.

2. Select the control on the form.

3. Right-click and select **Copy** if you want to copy the control, or **Cut** if you want to move the control.

4. If you want to copy or move the control to another form, open the target form.

5. Right-click an empty cell and select **Paste.**

Note the following points:

- All of the controls within a **Container** control (for example, a **GroupBox**) are copied or moved with it.

- You cannot add a control that is invalid for the target form (for example, a **State Change History** section cannot be added to a transition form).

- The target form must have enough empty space to accommodate the controls you want to paste.

- If you paste a control from the **Field Controls** section of the **Form Palette**, the existing field is removed. There can be only one instance of such a control on a form. For example, if you copy the **Title** control and paste it on the same form, only the pasted version remains.

- If you copy a control from the **Field Controls** section to a form in another application, the control is copied only if a matching control is found.

- If you select multiple controls, the **Paste** operation must be performed in the location relative to where you performed the **Copy** or **Move** operation. For example, suppose you have a row with three cells, and each cell contains a button. Select the three buttons, right-click the last button, and then select **Copy**. To paste, select the last cell in an empty row and then select **Paste**. If you select the first cell, you get a message that the target space does not have enough empty cells. This behavior is consistent with how the drag-and-drop operation works in the form editor.

- If row or column spanning was applied to source cells, the spanning is maintained in the target cells, if there is enough space. If there is not enough space, the existing cell size is used.

Using External Forms

An external form is based on a primary table that is not in the current application. It can be a table in another application in the current process app, or in a table in a referenced application (except for system auxiliary tables in the Global Application).

External forms are used to deploy forms separately from the process apps in which they are used. This enables you to more quickly test form changes.

To use external forms, create a separate process app that contains only external forms related to a table in an existing application. This "forms" process app can be deployed without redeploying the process app that defines the workflow. You can then use SBM Application Administrator to associate the forms with applicable states and transitions in the workflow.

To enable external forms, select the **Show external tables in Form Configuration dialog** option on the **Form Options** tab in SBM Composer Options [page 479]. After you do this, you can create the form as described in Creating Custom Forms [page 240] and Form Configuration Dialog Box [page 288].

> **Tip:** In App Explorer and in the forms list, a different icon is displayed for forms based on external tables.

Custom Transition Control Tutorials

This section contains the following tutorials:

Tutorial: Adding a Custom Transition Button to a State Form

This tutorial demonstrates how to add a custom transition button to a state form in a location that is different from the standard transition button bar. It also demonstrates how to use a variable for the button label, so if the transition name changes in the workflow, the button label is automatically updated.

Prerequisites:

Before you start this tutorial, create a new process app from the **Application Process App** template in the Create New Process App Dialog Box [page 67], and add the following states and transitions to the application workflow:

- In Dispatch ("active" state)

- In Progress ("active" state)

- Waiting for User ("active" state)

- Resolved ("active" state)

- Closed ("completed" state)

- Dispatch ("regular" transition from **New** state to **In Dispatch** state)

- Assign ("regular" transition from **In Dispatch** state to **In Progress** state)

- Request Info ("regular" transition from **In Progress** state to **Waiting for User** state)

- Provide Info ("regular" transition from **Waiting for User** state to **In Progress** state)

- Resolve ("regular" transition from **In Progress** state to **Resolved** state)

- Close ("regular" transition from **Resolved** state to **Closed** state)

To add a custom transition button:

1. Click the **In Progress** state in the workflow editor.

2. Click the **Form** tab on the state Property Editor.

3. Click **New** in the State form area. In the **Form Configuration** dialog box, select **Based on 'quick form' for: In Progress Regular State**.

4. In the form editor, perform the following steps:

     a. Select the top section of the form, right-click, and then select **Add Row Above**.

     b. Drag a **Button** control from the **Form Palette** onto the new row.

     c. Click the **Align right** icon in the **Alignment** section on the **Design** tab of the Ribbon.

5. On the **Behavior** tab of the form Property Editor, perform the following steps:

     a. Select **Perform a transition**.

     b. Click **Add**.

     c. In the **Add Transition** dialog box, make sure the **In Progress** state is selected, select the **Request Info** transition, and then click **OK**.

     d. Select the **Show transition name** check box. The transition name will be displayed as the button label. If the name of the transition changes in the workflow, the button label will change automatically.

6. On the **Toolbar** tab of the form Property Editor, make sure the **Remove transition buttons matching custom transition controls** check box is selected. This option removes the **Request Info** transition button from the button bar.

7. To see the changes you made, click **Preview** in the Ribbon.

On the custom state form, the custom **Request Info** transition button is at the top right of the form, and the standard transition buttons (**Resolve**, **Update**, and **Delete**) are in the button bar at the top of the form.

Tutorial: Repeating Transition Buttons at the Bottom of a Long State Form

This tutorial demonstrates how to add custom transition buttons at the bottom of a long custom state form, so users do not have to scroll to the top of the form to click a standard transition button in the button bar. It also demonstrates how to use a single custom transition button for multiple transitions.

Prerequisites:

Before you start this tutorial, create a new process app from the **Application Process App** template in the Create New Process App Dialog Box [page 67], and add the following states and transitions to the application workflow:

- Assigned ("active" state)

- Work Started ("active" state)

- In Peer Review ("active" state)

- In QA ("active" state)

- Closed ("completed" state)

- Assign ("regular" transition)

- Start Work ("regular" transition)

- Review ("regular" transition)

- Test ("regular" transition)

- Close ("regular" transition)

To repeat standard transition buttons:

1. Create a custom state form. In the **Form Configuration** dialog box, select **Based on 'quick form' for:** *application* **Table**. (For instructions, see Creating Custom Forms [page 240].)

2. On the **Forms** tab of the workflow Property Editor, select the form you just created from the **Default state form** list, then click **Edit**. The state form opens in the form editor.

3. On the **Toolbar** tab of the form Property Editor, clear the **Remove transition buttons matching custom transition controls** check box. This step is necessary so buttons can be at both the top (button bar) and bottom of the form.

4. Drag a **Panel** control from the **Container Controls** section of the **Form Palette** to the bottom of the form, and drop it on the green row that appears. The Insert Dialog Box [page 316] opens.

5. In the **Insert Panel** dialog box, select **3** columns, **1** row, and **All autoSize**. Click **OK**.

6. Drag a **Button** control from the **Other Controls** section of the **Form Palette** to each column in the new row.

7. Select the button in the first column. (This button will be used for all outgoing transitions you added to the application workflow.) Click the **Behavior** tab of the form Property Editor, and perform the following steps:

   a. Select **Perform a transition**.

b. Select **Show transition name**. This option lets you add a single transition button for all transitions you added to the workflow. The button label changes automatically based on the current state. It also changes automatically if the transition name is changed in the application workflow.

8. Still on the **Behavior** tab, click **Add**. In the Add Transition Dialog Box [page 310], perform the following steps:

   a. Select the application workflow and the **New** state. Because there is only one outgoing transition from this state, the **Assign** transition is already selected. Click **OK**.

   b. Click **Add** again. Select the **Assigned** state. (The **Start Work** transition is selected.) Click **OK**.

   c. Click **Add** again. Select the **Work Started** state. (The **Review** transition is selected). Click **OK**.

   d. Click **Add** again. Select the **In Peer Review** state. (The **Test** transition is selected.) Click **OK**.

   e. Click **Add** again. Select the **In QA** state. (The **Close** transition is selected.) Click **OK**.

   > **Note:** If you do not select **Show transition name**, you must type the button label in the **Display text** box on the **General** tab of the button Property Editor.

9. Select the button in the second column. (This button will be used to repeat the **Update** button, which is for a built-in transition and on the standard button bar). Perform the following steps:

   a. Click the **General** tab of the button Property Editor.

   b. Type `Update` in the **Display text** box.

   c. Click the **Behavior** tab of the button Property Editor.

   d. Select **Perform a transition**.

   e. Clear **Show transition name**. This ensures that the label you typed on the **General** tab will be displayed on the button instead of the transition name.

   f. Click **Add**. In the **Add Transition** dialog box, select the **[Any]** state and the **Update** transition. Click **OK**.

10. Repeat the previous step for the button in the third column, but type `Delete` in the **Display text** box, and in the **Add Transition** dialog box, select the **Delete** transition. (**Delete** is also a built-in transition.)

11. To see the changes you made, click **Preview** in the Ribbon.

On the custom state form, the transition buttons appear at both the top and bottom of the form. (The standard transition button bar is at the top of the form, and the custom transtition buttons are at the bottom of the form.)

Tutorial: Replacing Standard Transition Buttons with Hyperlinks on a Transition Form

This tutorial demonstrates how to replace the standard **OK** and **Cancel** buttons on a transition form with hyperlinks and change their labels.

Prerequisites:

Before you start this tutorial, create a basic process app from the **Application Process App** template in the Create New Process App Dialog Box [page 67], and add the following states and transitions to the application workflow.

- Assigned ("active" state)
- In Progress ("active" state)
- Closed ("completed" state)
- Assign ("regular" transition)
- Start Work ("regular" transition)
- Close ("regular" transition)

To replace standard transition buttons with hyperlinks and change their labels:

1. Click **Element** in the **New** section on the **Home** tab of the Ribbon, and then click **Transition Form**.

2. In the Form Configuration Dialog Box [page 288], select **Based on 'quick' form for <application name> Table**.

3. On the **Forms** tab of the workflow Property Editor, select the form you just created from the **Default transition form** list, and then click **Edit**. The form opens in the form editor.

4. Select the top section of the form, right-click, and then select **Add Row Above**.

5. Drag a **Panel** control from the **Container Controls** section of the **Form Palette** to the new row.

6. In the **Insert Panel** dialog box, select **2** columns, **1** row, and **All autoSize**.

7. On the **Toolbar** tab of the form Property Editor, clear the **Show button bar** check box. This ensures that the hyperlinks you add will replace the standard buttons.

8. Drag a **HyperLink** control from the **Other Controls** section of the **Form Palette** to the first column in the new row.

   a. On the **Behavior** tab of the hyperlink Property Editor, select **Submit form**.

   b. On the **General** tab of the hyperlink Property Editor, in the **Display text** box, type `Submit`).

9. Drag another **HyperLink** control to the second column of the new row.

    a. On the **Behavior** tab of the hyperlink Property Editor, select **Cancel form**.

    b. On the **General** tab of the hyperlink Property Editor, in the **Display text** box, type `Abandon`).

10. Optionally, repeat steps 4 through 9, but add a row to the bottom of the form. This ensures the hyperlinks are available at the top and the bottom of the form.

11. To see the changes you made, click **Preview** in the Ribbon.

On the custom transition form, the standard **OK** and **Cancel** buttons are replaced with hyperlinks. **Submit** replaces the standard **OK** button label, and **Abandon** replaces the standard **Cancel** button label.

Form Settings

The following sections provide descriptions of settings used to create forms. Sections are organized by property editors.

Form Configuration Dialog Box

Use this dialog box to provide basic parameters for a new custom form.

Element	Description
Name	The name of the new custom form. Each form must have a unique name. If you type the name of an existing form, a number is appended to the name (for example, "State Form 2").
Type	Select **State form**, **Transition form**, or **Print form**.
Layout	Select **Responsive** for a form layout that adjusts to the size of the screen, (available only for modern forms), or select **Standard** for a non-responsive layout. For details, refer to Working With Responsive Forms [page 269]. You can modify this setting for specific container controls after the form is created.
Legacy mode	Select this check box if you want to create a form using legacy technology (used in all releases prior to SBM Composer 11.1). By default, this check box is selected if **Enable HTML5 features** is disabled in the Form Options tab of the SBM Composer Options dialog box, and cleared if **Enable HTML5 features** is enabled.

Element	Description
Options	Select one of the following:
	Form with pre-populated sections for: Creates a form for the primary table, an application workflow or sub-workflow, or an auxiliary table in the open process app. By default, the content of the form is determined by the privilege section specified for each field in the table. Select a workflow if there are field overrides at the workflow level.
	Form with auto-sections for: Creates a form for the primary table or an auxiliary table. The sections in the form will be automatically populated at runtime based on field privileges. You can leverage the automatic layout of fields while still customizing the form as needed.
	Empty form for: Creates an empty form based on the selected table.
	Based on another form: Creates a form based on the selected custom form in the open process app. In most cases, the form types do not have to match, so you can create the form from either a state form or a transition form.
	Note: If the "external forms" option is selected on the **Forms** tab in SBM Composer Options [page 479], the first three options also contain tables external to the current application. This includes both primary and auxiliary tables from other applications in the open process app and from referenced applications (except for system auxiliary tables from the Global Application). Forms that are based on external tables are included in the last option.
Labels	Choose to place labels to the left of form controls or on top of form controls.
	You can modify this setting for specific container controls after the form is created. For details, refer to General Tab of the Form and Control Property Editor [page 289].
Field columns	Specify the number of field columns for the form.
Preview	Displays a read-only version of the blank form or the form on which you are basing the new form.
Actual size	If this check box is selected, the size of the form in the **Preview** element is the same as the size of the form in the form editor.
	If this check box is cleared, the form is sized to fit in its entirety in the **Preview** element.

General Tab of the Form and Control Property Editor

The information and options that appear on the **General** tab depend on what you selected in the form or control editor:

General Options for Forms

Control	Description
Name	The name of the form. The table that the form is based on is displayed under the name. If the table is external to the current application, its application is also displayed.
Legacy mode	Select this check box if you want the form to use legacy technology (used in all releases prior to SBM Composer 11.1).
End-user help text	Click **Edit** to open an HTML editor and add optional comments or notes about the purpose of the form. The text you enter is displayed to users. For details, refer to Chapter 24: Providing Custom End-user Help [page 443].
Options *(Transition form)*	Select the **Validate required fields before form submit** check box to perform client-side validation on all fields. (The SBM Application Engine always performs server-side validation of required fields regardless of this setting.) The validation takes place when the user clicks the **OK** button on the form. Selecting this check box can improve performance because it eliminates the need for the server to return the form when validation fails, because the client-side validation will catch invalid data before submitting the form to the server. The validation takes place on fields that the workflow requires (those fields marked as **Required** in the field Property Editor) and on fields that the SBM JavaScript Library conditionally requires. (JavaScript that performs this validation is present on every custom transition and state form.) If this check box is not selected, client-side validation takes place on only those fields that the JavaScript Library conditionally requires. **Note:** Workflow fields can be overridden as required or not required at the workflow level in SBM Composer and for projects in SBM Application Administrator.

Control	Description
Size guide	Select the **Show** check box to display a shaded boundary around the portion of the form that will be visible to users in a single screen. When the option is selected, use the **Width** and **Height** controls to set the desired size. Instead of typing a specific number in the **Width** and **Height** controls, you can type `a` to automatically set the width of the column or the height of the row to fit the contents of the column or row. The **Size guide** option does not affect the behavior of the form. It is simply an aid in designing forms that minimize the need for scrolling. **Tip:** If you know the likely size of your users' screens, set the **Size guide** dimensions a little smaller. If you design your form to fit within the **Size guide** parameters, it is more likely that your users will be able to see and use all the controls on the form without having to scroll.

General Options for Field Controls

Control	Description
Linked to field '*field-name*' in table '*table-name*'.	Click this link (in the upper right corner of the Property Editor, next to the drop-down list of form components) to view the field's definition in the '*table-name*' table. Click on the Quick Access toolbar to return to the field's properties in the form editor.
Name	SBM Composer automatically assigns unique names to field controls.
Description	An area for optional comments about the control.
Display text	For a field label, the display name of the field. For a field control, the value depends on the context, as visible in the form preview.

Control	Description
Custom display text	Select this check box to change the **Display text** for a field label. This is useful if you want the label on the form to differ from the database field name. For example, you might want to use the same label on more than one field. You could have a form with an employee section and a manager section. Each section needs a *First Name*, *Last Name*, and *Employee Number* field, one set for the employee and one set for the employee's manager. These are distinct fields in the table, but using the same label for each pair of fields simplifies the form. **Note:** If you do not use this option, the only way to have duplicate labels for fields is to have multiple fields with the same name in the table. This is problematic because the fields will be indistinguishable in areas such as the form, rule, report, and application variable editors.
Input mask	*(Text fields with the **Fixed length** style on a transition form)* **Important:** If an input mask is used, the **Include jQuery plugin** check box must be selected on the **JavaScripts** tab of the form Property Editor. Enter or select a mask to make it easier for users to enter data in the correct format. For example, if you select the **Date** input mask, when users click in the text box, two forward slashes appear. After two digits are entered for the month (or day), the cursor moves to the right of the first forward slash so two more digits can be entered for the day (or month). The cursor then moves to the right of the second forward slash so the four-digit year can be entered. The other preformatted input masks are **Phone number**, **Phone number with optional extension**, and **Social Security number**. If you select **Custom input mask**, type a mask using the following syntax: • **a** represents an alphabetic character (A-Z, a-z) • **9** represents a numeric character (0-9) • ***** represents an alphanumeric character (A-Z, a-z, 0-9) • **?** indicates that anything that follows is optional. Use **?** to designate an optional part of the mask, such as a phone extension: (999) 999-9999? x99999. Leave blank or select **(No input mask)** to allow all input.

Control	Description
Options	*("Multi" field types)* Select the **Show as dual list boxes in design/ preview mode** check box to see two list boxes (with arrows for adding or removing values) in the form editor and form preview, to match what the user will see.
	("Multi" field types) Select the **Display dual list boxes vertically (HTML5 only)** check box to align two list boxes vertically instead of horizontally. This option applies only when HTML5 is enabled in SBM Application Administrator.

General Options for Detail Controls

Control	Description
Name	The control name must be unique within the form. **Note:** SBM Composer automatically assigns unique names to detail controls.
Description	An area for optional comments about the control.
Display text	The label to display on the form. To modify the default text, select **Custom**.
Container type	Depending on the detail control, you may be able to change its container type.
Show collapsed on startup	Use to specify whether detail controls that use the expander type should be collapsed initially when the form opens. (This option is not available on print forms.)
Options for specific Detail controls	*(Attachments Detail control)* **Hide if no attachments have been added**: Clear this check box if you want the **Attachments** section to appear on a custom form, even if there are no attachments. *(Notes Detail control)* **Hide if no notes have been added**: Clear this check box if you want the **Notes** section to appear on a custom form, even if there are no notes.

General Options for Container Controls

Control	Description
Name	The control name must be unique within the form. **Note:** SBM Composer automatically assigns unique names to detail controls.
Bound privilege section	Optionally, choose to bind one or more privilege sections to the control. This enables you to use the privilege section label as the control label. To change the label, select the **Custom** check box, and then modify the display text.
Description	An area for optional comments about the control.
Display text	The label to display on the form. Not available for panel controls.
Container type	Lists the available types for your control. You can change the type for most controls, except for tab controls.
Labels	Choose to place labels to the left of controls or on top of controls for the selected container. By default, the label setting for the form is used, but you can override the form setting for each container. You also can customize the location of specific control labels in the container. For example, you can place some labels in a container at the top of the control and others at the left. If you change the Label setting for the container, however, your custom locations are removed for all controls in the container.
Layout	Choose **Responsive** for a container layout that adjusts to the size of the screen (available only for modern forms), or choose **Standard** for a non-responsive layout. By default, the container inherits the form's layout.
Show collapsed on startup	*(Expander controls)* Use to specify whether the control should be collapsed initially when the form opens. (This option is not available on print forms.)
Show lines between rows	Display thin lines between rows within this container. This can make a wide form easier for your users to read.
Auto-populated section	Turns on or off auto-section behavior of the container. If checked, a placeholder row is displayed, and fields will be automatically populated at runtime by privilege section.

General Options for Other Controls

Control	Description
Name	The control (or label) name must be unique within the form.
Description	An area for optional comments about the control.
Value(s)	*(Combo Box, List Box)* The values to be available for users to choose from on a form. Enter one value on each line.
Default text	*(Edit Box)* The text you want to display to users when they first open the form. You could use this field to provide a default value or a prompt for the data you want them to enter. If you select the **Multi-line edit box** option, the default text (and anything your users type in its place) will wrap. If you select the **Password** option, the default text (and anything your users type in its place) will appear as a string of black dots.
Options	*(Edit Box)* Select **Edit box** to create a single-line text entry field, and then select **Password** if you want characters in the field to be replaced by black dots. Select **Multi-line edit box** to create a multiple-line text area, and use the **Width** and **Height** controls (in the **Size** area on the **Design** tab of the Ribbon) to control the dimensions of the text area. *(Combo Box, List Box)* Select **Combo box** to create a drop-down list, in which only the first item listed for **Value(s)** above is initially displayed. The other values are displayed when users click the arrow at the right side of the combo box. Select **List box** to create a scrolling list. Select **Allow multiple selection** to use the `GetFieldValues`, `SetFieldValues`, `GetMultiListValues`, and `SetMultiListValues` JavaScript API functions with the list. (See the *SBM JavaScript Library Guide* for information about these functions.) Use the **Height (in rows)** control to specify how many of the items in the **Value(s)** control above will be displayed at a time. **Note:** Values from these options are only available when the form is run and are not available for use in scripts.

Control	Description
Display text	*(Button)* The text to be displayed on the button.
	(HyperLink) The clickable text for the link.
	(Image) Text that will be displayed as a tool tip when the user hovers over the image or in place of the image if it cannot be loaded for some reason. This text may be recognized by text-to-speech software (also known as *screen readers*).
	(Text) The text to be displayed. The text you type here can include HTML tags (`<b>` and `</b>`, for example). They will be included in the HTML code for the page that is rendered at runtime.
	(Edit/Combo/List Box label) The text to be displayed next to the associated **Edit box**, **Combo box**, or **List box** control. SBM Composer creates a default label when you place one of these controls on your form, and then change that label text to reflect the purpose of the control.
Tooltip text	(*Button*, *Hyperlink*, *Image*) Enter the text you want to display to your users when they hover over the button, hyperlink, or image.
Image	*(Image)* The image to display. Select an image already added to this application, or select **(New image...)** to add a new image to the application and use it for this control.
Display image as button	*(Image)* Applies button styling to an image.

General Options for Headers, Footers, and Sidebars

Control	Description
Labels	Choose to place labels to the left of controls or on top of controls for the selected header, footer, or sidebar. By default, the label setting for the form is used, but you can override the form setting.
	You also can customize the location of specific control labels in the header, footer, or sidebar. For example, you can place some labels at the top of the control and others at the left. If you change the Label setting for the header, footer, or sidebar, however, your custom locations are removed for all controls in the header, footer, or sidebar.
Layout	Choose **Responsive** for a layout that adjusts to the size of the screen (available only for modern forms), or choose **Standard** for a non-responsive layout. By default, the header, footer, or sidebar inherits the form's layout.

Appearance Tab of the Form and Control Property Editor

Use this tab to fine-tune the appearance of the overall form, as well as detail controls, container controls, headers, footers, and sidebars.

Element	Description
Form features *(Forms)*	Select the check boxes to add an empty header, footer, left sidebar, and right sidebar to your form. After you select the check boxes, you can further specify the height or width: • Select **Fixed**, **Percentage**, or **AutoSize**. • For fixed, type the number of pixels for the sidebar width or the header or footer height. • For percentage, type the percentage of width (sidebars) or height (footers or headers) in relation to the total form width or height. • For autosize, the width and height are irrelevant and cannot be set.
Table settings *(Forms, Container Controls, Headers, Footers, Sidebars)*	**Cell spacing**: Specify the number of pixels that should be between cells. **Cell margins**: Specify the number of pixels that should surround the content of cells. **Restore defaults**: Restores the default value of 0 for the cell spacing, and 4 for the cell margins.
Container padding *(Container Controls, Detail Controls, Headers, Footers, Sidebars)*	**Top:** Specify the number of pixels that should be at the top of the container. **Left:** Specify the number of pixels that should be at the left of the container. **Bottom:** Specify the number of pixels that should be at the bottom of the container. **Right:** Specify the number of pixels that should be at the right of the container. **Restore defaults**: Restores the default values of 0 for the top, 20 for the left, 0 for the bottom, and 20 for the right.

Tool Bar Tab of the Form Property Editor

Use this tab to customize the tool bar in the form's header area.

Tip: To remove the entire toolbar, clear all of the check boxes in the following table.

Control	Description
Options	Select or clear check boxes as needed to show or hide:
	• **Show item identifier**
	The item ID, title, and transition name (transition forms only).
	• **Show project item identifier**
	The project and singular item name.
	• **Show button bar**
	The standard transition button bar at the top of the form. You might want to clear this check box if you added custom transition controls and want to limit the available transitions to those associated with those controls.
	CAUTION:
	If you clear this check box on transition forms, and do not map the **OK** and **Cancel** buttons to other controls on the form, you will get a validation error for each button, because the user will be unable to submit a form or cancel a transition before submitting a form.
	• **Show action bar**
	The **Actions** drop-down list that contains the selected standard options, the icons for the selected optional actions, and the selected report navigation actions (described below).
	• **Show viewport**
	(State forms only.) The **Details/Social** toggle.
	• **Show transition sequence indicator**
	(State forms only.) The icon that when hovered over, displays the last transition that you executed for any item.
	• **Remove transition buttons matching custom transition controls**
	(State forms only.) Standard transition buttons in the button bar for those transitions that have custom transition controls.

Control	Description
Standard Actions	The selected standard actions will be items in the **Actions** drop-down list in the action bar. Select or clear check boxes as needed. The following is a brief description of what each action allows a user to do. **Add Note**: Add a note to a primary or auxiliary item. **Add URL**: Add a URL to an external Web site or to a page within SBM. **Add File**: Add a file attachment to a primary or auxiliary item. **Add Item Link**: Create a link between items. **Add Item Notification**: Subscribe to an e-mail notification pertaining to a primary or auxiliary item. *(State forms and submit forms only)* **Add to Folder**: Create a link to the item in a specified folder. **Add File Association**: Add a version control file association to a primary item. **Link Subtasks**: Establish a principal/subtask relationship among existing primary items. **Link to Principal**: Links a subtask item to a principal item. **Unlink Principal**: Breaks a principal/subtask relationship by removing the link between a subtask item from its principal task. **Tip:** To show or hide all of these actions, select or clear the **Standard Actions** check box. **Note:** All of these actions are available for mapping to a button, image, or hyperlink control, even if the check boxes for them are cleared or if the **Show action bar** check box is cleared. For more information, see Behavior Tab of the Control Property Editor [page 307].

Control	Description
Optional Actions	Icons for the selected actions will be displayed in the action bar. Select or clear check boxes as needed. The following is a brief description of what each action allows a user to do.
	(State forms only) **Reload Item**: Reload any item that might be waiting for an update from an orchestration workflow or Web service.
	(State forms only) **Display Printable View**: Open a printer friendly view of the form in another window.
	Click to Send E-mail: Send an e-mail message about the item.
	Manage External Users: Enables SBM users to update external user details on items that are created by anonymous e-mail submit. Note that this option only appears for users with the **External Communication** privilege.
	(State forms and submit forms only) **Copy URL to Clipboard**: Copy the URL for the item details to the Windows Clipboard.
	Get Help for this Application: Show the visual workflow that the current primary item follows, and show information about the workflow and the design elements it contains, such as transitions and states. See Chapter 24: Providing Custom End-user Help [page 443] for more information.
	Get Help for this Form: Show information about the form and the design elements it contains, such as fields. See Chapter 24: Providing Custom End-user Help [page 443] for more information.
	Tip: To show or hide all of these actions, select or clear the **Optional Actions** check box.
	Note: All of these actions are available for mapping to a button, image, or hyperlink control, even if the check boxes for them are cleared or if the **Show action bar** check box is cleared. For more information, see Behavior Tab of the Control Property Editor [page 307].

Control	Description
Report Actions (*State forms only*)	Links for the selected actions for items in report or search results will be displayed above the button bar. Select or clear check boxes as needed. The following is a brief description of what each action allows a user to do. **Back to Results**: Return to the report or search results. **First Item**: Go to the first item in the report or search results. **Previous Item**: Go to the previous item in the report or search results. **Next Item**: Go to the next item in the report or search results. **Last Item**: Go to the last item in the report or search results. **Tip:** To show or hide all of these actions, select or clear the **Report Actions** check box. **Note:** All of these actions are available for mapping to a button, image, or hyperlink control, even if the check boxes for them are cleared or if the **Show action bar** check box is cleared. For more information, see Behavior Tab of the Control Property Editor [page 307].
Submit Text (*Transition forms only*)	By default, the user clicks **OK** to submit a form. Type something else in this box if you want to change the label on the button.
Cancel Text (*Transition forms only*)	By default, the user clicks **Cancel** to cancel a transition before a form is submitted. Type something else in this box if you want to change the label on the button.

Note: Some actions (for example, Add Item Notification, Link Principal, Unlink Principal, Click to Send E-mail) are not available for Submit transition forms. These actions will not appear in the **Actions** drop-down list on Submit transition forms, even if they are selected on this tab.

Actions Tab of the Form Property Editor

Use this tab to create action expressions that drive dynamic behavior in the form. See Using Form Actions [page 245] and Working With Form Actions [page 246] for more information.

Control	Description
Action	The name of the action expression.

Control	Description
Description	An area for an optional description of the action expression.
Enabled	The **Enabled** check box is selected by default if the expression defined for the action is valid. If you do not want the action to be used right away, clear this check box. If the expression is invalid, the action is crossed out and the check box is cleared.
New	Opens the Form Logic dialog box, where you can define a new action expression.
Edit	Opens the Form Logic dialog box, where you can edit the selected action expression.
Delete	Deletes the selected action expression.
Move up, Move down	Changes the order in which the dynamic behavior is performed. **Note:** If there is a conflicting action, such as two actions that change the background text color with different colors, then the action that was executed last takes effect.
Show action summaries	Displays textual summaries of the actions.

Rows Tab of the Form and Control Property Editor

Use this tab to manipulate the rows in the selected form, container control, header, footer, or sidebar.

Element	Description
Type	Select **Fixed**, **Percentage**, or **AutoSize**. **Note:** For responsive forms, you can specify only **Fixed** or **AutoSize**.

Element	Description
Height	For fixed rows, type the number of pixels. For percentage rows, type the percentage of total height you want this row to occupy. Other rows are adjusted as needed for a total of 100%. For autosize rows, the height is irrelevant and cannot be set. **Note:** For responsive forms, if you selected **Fixed**, the value you specify here is applied as the minimum height of the row. **Note:** For container controls on modern forms, if you specify a fixed height or vertical autofill, a scroll bar will automatically appear if the content does not fit in the allotted space.
Add row above/ below	Adds a row above or below the selected cell.
Delete row	Deletes the entire row containing the selected cell.

Columns Tab of the Form and Control Property Editor

Use this tab to manipulate the columns in the selected form, container control, header, footer, or sidebar.

Element	Description
Type	Select **Fixed**, **Percentage**, or **AutoSize**. **Note:** For responsive forms, the type is specified as **Responsive Block** and cannot be changed.
Width	For fixed columns, type the number of pixels. For percentage columns, type the percentage of total width you want this column to occupy. Other columns are adjusted as needed for a total of 100%. For autosize columns, the width is irrelevant and cannot be set. For mixed (autosize and percentage) columns, the label portion is wide enough to accommodate the longest label, and the field portion is allocated by percentage. **Note:** For responsive forms, the width is specified as **12** and cannot be changed. **Note:** For container controls on modern forms, if you specify a fixed width, a scroll bar will automatically appear if the content does not fit in the allotted space.

Element	Description
Add column left/right	Adds a column to the left or right of the selected cell.
Delete column	Deletes the entire column containing the selected cell.

JavaScripts Tab of the Form Property Editor

Use this tab to include your own JavaScripts in a state or transition form. The SBM Server executes the scripts when users open the form.

> **Note:**
>
> - By default, the scripts are not executed during form preview. For more information, see Form Preview Dialog Box [page 317].
>
> - The **JavaScripts** tab is not available for print forms.

Element	Description
Include jQuery plugin	Loads the jQuery library plugin. This check box must be selected if an SLA widget or Social widget is added to the form. (See Detail Controls [page 312] for information about these widgets.) It must also be selected if an input mask is specified for a text field on a transition form. (See General Options for Field Controls [page 291] for information about the input mask control.)
Add	If you select **New**, opens the JavaScript editor, where you can create or import a JavaScript file. The JavaScript file is automatically included in the selected form. If you select a JavaScript file from the drop-down list, that file is included in the selected form. This list includes files under the **JavaScripts** heading in App Explorer that are not already included in the form.
Import	Imports a JavaScript file into the application and includes it in the selected form.
Remove	Removes the selected JavaScript file from the form (but not from the application).
Move up/down	Moves the selected JavaScript file higher or lower in the list. The order of the files in the list determines the order in which they are included by the SBM Server. This is important if you are reusing JavaScript files in several different forms. Files in the list may require variables to be defined or scripts to be executed in previous includes for them to function properly.

Localization Tab of the Form Property Editor

Use this tab to define translatable strings in a form. Once they are defined, you can use the strings in the following places:

- Form actions

- String Builder tool in form controls and widgets

Once you use the strings on a form and deploy the process app, you can access the strings in Application Administrator:

1. Log in to Application Administrator, and then open the **Localization** feature.

2. Select the **Allow translation for MLS Runtime Objects and Design Objects** check box on the **Settings** tab.

3. Select the **Values** tab.

   The strings appear in the **Form Strings** section within your process app. The strings will remain in Application Administrator even if they are no longer in use on a form.

Control	Description
String name	The name of the translatable string.
Base text	The text shown to users if a translation doesn't exist for the specified locale.
Add	Adds a new string that can be accessed on the custom form.
Delete More | Delete All	Deletes the selected string or all strings. To delete a string that is currently in use, uncheck **Prevent the deletion if the localizable string is in use** in the warning that appears. To locate where a string is in use, right-click the string and select **Where Used**.
Move up, Move down	Changes the order of the strings.
More | Export	Exports the list of strings in CSV format. Typically, you would use this option to first create a file that contains the default English strings. You can then import the file after the strings are translated.
More | Import	Imports a list of strings in CSV format.

Refresh Tab of the Control Property Editor

Use this tab to specify the form data that should be displayed in an edit box, hyperlink, image, or text control, and define when the data should be refreshed.

> **Note:** Instead of typing or pasting text into the **Contents**, **Image Url**, and **Display text** fields, use the string builder tool to insert references to table fields and form controls, and to insert HTML tags. For more information, see Using the String Builder Tool [page 277].

Element	Description
Contents	Enter the content that should be displayed in the **Edit Box** control.
Display text	Enter the text that should be displayed in the **HyperLink** or **Text** control.
Refreshable Items	For **Image** controls, you can refresh both the image URL and the display text. **Image URL:** Select this option and enter the URL of the image that should be displayed on refresh in the **Image** control. **Display Text:** Select this option and enter the text that should be displayed on refresh in the **Image** control.
On page load	If this check box is selected, the content is retrieved when the form is initially displayed at runtime or in the form preview. > **Note:** This check box is automatically selected if the **Contents**, **Image Url**, or **Display text** field does not refer to any data fields, and if the form does not include any buttons, images, or hyperlinks.
On data change	If this check box is selected, the content is updated when the field that maps to the **Contents**, **Image Url**, or **Display text** field is changed at runtime or in the form preview. > **Note:** This check box is disabled if the fields listed above do not refer to any data field.
On click	In the drop-down list box, select one of the buttons, images, or hyperlinks that are on this form. The content is updated when the control is clicked at runtime or in the form preview. > **Note:** This check box is disabled and the drop-down list is hidden if the form does not include any button, image, or hyperlink.

Behavior Tab of the Control Property Editor

Use this tab to specify the content and appearance of a Web page or popup window that opens when users click a button, hyperlink, or image control.

Behavior Options

Element	Description
Open URL (all form types)	
URL	The URL of the Web page that opens when the user clicks the control. Type `{` if you want to insert references to table fields or other form controls.
	The contents of substitution parameters in the URL are escaped if the substitution parameter is preceded by an equal sign (=). For example, for this URL:
	`http://www.google.com/#2q={fieldname}`
	The contents of the substitution parameter {fieldname} will be escaped at runtime. For example, if the value is "abcdef" (including the quotes) after processing the query will look like this:
	`http://www.google.com/#2q=%22abcdef%22`
	For this URL:
	`http://www.google.com/#{fieldname}`
	The value of the {fieldname} substitution parameter will not be escaped.
Target	Whether the Web page should open in the same window or in a separate window.
	You can also type the name of a window or an iframe (inline frame) in this field. For example, suppose you have a **Help** button for two fields on the form. You type `Help` in the **Target** box for each button instead of selecting **New Window** or **Same Window**, and enter a unique URL for each button. When a user clicks the **Help** button for the first field, information about the first field is displayed in a new window. When the user clicks the **Help** button for the second field, information about the second field replaces the information in the window.
Open custom popup (all form types)	

Element	Description
Body HTML	The content of the popup window that opens with the user clicks the control. You can type HTML code if you want the text formatted, or type text if you do not want the text formatted. There is no limit to the number of characters you can type.
	Type { if you want to insert references to table fields or other form controls. By default, referenced fields will render as plain text or rich text according to how the **Enable Rich Text** option is set in the field Property Editor. You can override this behavior in SBM 11.4 and later (see "Options" below).
	Important: When HTML5 is enabled, the popup is a dialog box instead of a separate browser window. This enables you to leverage the form's custom JavaScript and CSS in the dialog, as well as initiate transitions from the dialog box, such as submitting an item.
Options	Default values for features that specify the appearance of the popup window. You can modify the options as needed.
	When HTML5 is enabled, you can specify a wider set of options for the dialog box. For the list of options, refer to solution S142150.
Reset	Restores the default options.
Perform a transition *(state forms)*	
Mapping	Shows which transition will be executed for a specific state when users click the custom transition control. You can use custom transition controls in addition to or instead of the standard transition buttons at the top of a custom form. These custom controls can have arbitrary labels and can be placed anywhere on the form.
	You can associate multiple controls with a single transition. This is useful when you want to repeat controls in different parts of a long form, so users do not have to scroll to the standard button bar at the top of the form.
	Note: See Custom Transition Control Tutorials [page 282] for a specific use case and instructions.
	You can see the application workflow where the transition was created in the **Defining application workflow** column.
	Note: Mapping also applies to all sub-workflows of the defining application workflow.

Element	Description
Show transition name	If you select this option, you can configure the control to use the transition name as its label. This is useful when using a single control to trigger different transitions depending on the current state. This option is also useful for a single transition, because if the transition name changes in the application workflow, the label changes automatically.

> **Tip:** It is recommended that you type `auto` in the **Height** and **Width** boxes in the **Size** section on the **Design** tab of the Ribbon. If you specify fixed values instead, the label for a long transition name could be truncated.

If you do not select this check box, the display text entered on the **General** tab of the control Property Editor is shown. |
| Show transition name as tooltip | Select this option to display the transition name when users hover over the control. |
| Allow extended transition button mapping for this form | Select this check box to map any state in the selected workflow to an outgoing transition, regardless of the states that apply to the form. This enables you to assign the form to a state or a project in SBM Application Administrator. If you map the form to an invalid state and transition combination, it appears in italics text. If you clear the **Allow extended transition button mapping for this form** check box, the mapping appears in red text.

The **Allow extended transition button mapping for this form** option applies to all custom transition actions on a form. |
| Open | Opens the application workflow for the selected row in the **Transitions** list. The transition in the selected row is selected in the workflow editor. |
| Add | Opens the Add Transition Dialog Box [page 310], which lets you define state and transition mapping for the control.

You can add transition mapping only if the custom control is on a form that is associated with a state or that is the default state form for the workflow or if the **Allow extended transition button mapping for this form** check box is selected. If none of these conditions apply, you can only add mapping for the system Update and Delete transitions.

> **Tip:** If you selected **Show transition name** and want to have a single control for multiple transitions, add each transition (one at a time) to this tab. |
| Edit | Opens the **Edit Transition** so you can modify existing transition mappings. |
| Remove | Removes the selected row from the **Transitions** list. |

Element	Description
Perform a Tool Bar action *(all form types)*	
Tool Bar Action	Select the action you want to map to this control. All actions are available, even if on the Tool Bar Tab of the Form Property Editor [page 298], you removed the action bar from the form or removed the actions from the action bar. **Note:** Some actions (for example, Add Item Notification, Link Principal, Unlink Principal, Click to Send Email) are not available for Submit transition forms. If these actions are mapped to custom controls, they will appear on the form, but clicking them will do nothing.
Submit form *(transition forms and edit forms)*	Performs the same function as the standard **OK** button. The control can be placed anywhere on the form, and can have a unique label. **Note:** On the **General** tab of the form Property Editor, specify whether the standard **OK** button should remain in the button bar on the top of the form.
Cancel form *(transition forms and edit forms)*	Performs the same function as the standard **Cancel** button. The control can be placed anywhere on the form, and can have a unique label. **Note:** On the **General** tab of the form Property Editor, specify whether the standard **Cancel** button should remain in the button bar on the form.
Update item *(view forms)*	Performs the same function as the standard **Update** button. The control can be placed anywhere on the form, and can have a unique label. **Note:** On the **General** tab of the form Property Editor, specify whether the standard **Update** button should remain in the button bar at the top of the form.
Delete item *(view forms)*	Performs the same function as the standard **Delete** button. The control can be placed anywhere on the form, and can have a unique label. **Note:** On the **General** tab of the form Property Editor, specify whether the standard **Delete** button should remain in the button bar at the top of the form.

Add Transition Dialog Box

Use this dialog box to specify the transition that should be executed when users click a button, hyperlink, or image control. For more information, see Behavior Tab of the Control Property Editor [page 307].

By default, the transitions you can add are based on the form's context:

- If a form is used by specific states in the workflow, you can only add transitions that exit those states or out of the "Any" state.

- If a form is used as a default state form for a workflow, you can select any state that is not already mapped to a transition, and then select a transition that exits that state. In this case, you can map a transition for each state where the button, hyperlink, or image will be shown.

- You can add transitions that start in the same state but are defined in sibling workflows. You are warned if your mapping conflicts with another mapping for the same control.

If you want to map transitions for any state regardless of the form's context, first select the **Allow extended transition button mapping for this form** check box on the **Behavior** tab. This enables you to map any state in the selected workflow to a transition exiting that state. You might use this option if you want to assign the form to a state or project in SBM Application Administrator rather than to a state or workflow in SBM Composer. If you map a transition that is not applicable for a particular state form, the control is hidden from users.

> **Tip:** To map the system Update or Delete transitions to a control, select the "Any" state. Once you do this, you cannot map other transitions to the control. If you have other state/transition mappings assigned to a control, you must remove them before you can map to the "Any" state.

The following options are available on the **Add Transition** dialog box:

Element	Description
Application workflow	If there are multiple application workflows in the process app, select the workflow whose default custom state form contains the control.
	Select **All Application Workflows** to map the system Update or Delete transitions for all workflows in your application.
State	Select the state with the outgoing transition that should be executed when the user clicks the control.
Transition	Select the outgoing transition that should be executed when the user clicks the control.

Mobile Tab of the Control Property Editor

If the application is enabled for mobile access, the **Mobile** tab is available for fields, details, containers, and other controls. It is not available for widgets. Use this tab to set the visibility of fields, details, containers, or other controls when the form is viewed via the SBM mobile app.

Element	Description
Mobile client forms	Select one of the following options: • **Always show:** Show on all mobile forms (both simple and advanced). • **Always hide:** Hide on all mobile forms. • **Hide in simple view** (available only for fields and other controls)**:** Show on advanced mobile forms only.

Form Palette

In the form editor, you drag controls from the **Form Palette** and organize them on the form. You then use the control Property Editor to configure the controls. This topic describes the controls in each section of the **Form Palette**.

Field Controls

The controls in the **Field Controls** section represent fields in your primary table. You can use each field control one time in each form.

Note: For information about field control options, see General Tab of the Form and Control Property Editor [page 289].

Detail Controls

The controls in the **Detail Controls** section represent sections that contain detailed information about items. You can use each detail control one time in each form.

Some sections are displayed on the form only if they contain something. For example, the **Item Notifications** section is not displayed until a notification is added to the item. This applies to the **Attachments**, **Notes**, **Item Notifications**, **Subtasks**, and **Version Control** sections. The **Attachments** and **Notes** sections can be seen, however, if you clear an option on the **General** tab of the **Attachments** and **Notes** Property Editors.

Note: If you are using the SourceBridge integration, you must include the Version Control section on a custom form.

Control	Description
Attachments	Section that contains file attachments, item links, and URLs.
Change History	Section that contains information pertaining to changes to items. For example, a change history entry is added to an item when it is submitted and each time it is transitioned or updated.
Notes	Section that contains notes and e-mail messages attached to an item.
Item Notifications	Section that contains item notifications that a user subscribed to and the history of notifications for the item.

Control	Description
State Change History	Section that contains a graphical or tabular representation of the states and transitions a primary item moved through as it is tracked through a workflow. **Note:** This control is used on state forms only.
Subtasks	Section that contains links to subtasks or principal primary items.
Version Control	Section that contains source control information associated with a primary item. This section is used if your system has an integration to a version control tool.
New Note	Control that lets users type in a comment when they transition an item. On the **Options** tab of the transition Property Editor, you can specify whether this field should be displayed, and whether it should be a mandatory field. **Note:** This control is used on transition forms only.
Integrations	Section for external URL integrations.
SLA Widget	Control that contains Service Level Agreement details. This control is available only for state forms. **Important:** If you add this control to a form, select the **Include jQuery plugin** option on the JavaScripts Tab of the Form Property Editor [page 304].
Social Widget	Control that displays users ("experts") who have worked with an item. This control is only available for state forms. This control includes a popup that opens at the left side of the control if there is room on the form; otherwise it opens on the right side. To make sure that users can see the popup, do not place the widget in a single-column row, and place it in a column that has enough room on either side for the popup to open. **Important:** If you add this control to a form, do the following: • Select the **Enable searching for Social Widget** option on the Options Tab of the Table Property Editor [page 173] for the primary table. • Select the **Include jQuery plugin** option on the JavaScripts Tab of the Form Property Editor [page 304].

Control	Description
Time Capture Widget	Control that allows individual users to add an entry for the time they spent on an item and displays the total amount of time spent on the item by all users. This widget allows you to place the control in a specific area of the form.
	If the Time Capture feature is enabled and you do not add the control to custom forms, and for quick forms, this control is automatically placed below the **Change History** section (state forms) or above the **Standard Fields** section (transition forms) and is collapsed.
	If the feature is disabled, the control will not appear on custom forms, even if you explicitly added it to them.

Container Controls

The controls in the **Container Controls** section represent types of containers that hold other controls or information. For example, a **History** tab could contain the **Change History** detail control. Container controls serve the same purpose, but differ in appearance. A container control can contain another other container controls. For example, a **System** tab could contain group boxes that contain different types of system information.

For information about container control options, see General Tab of the Form and Control Property Editor [page 289] and Container Control Options [page 318].

> **Note:** For modern forms, containers are automatically hidden at runtime in the following cases:
>
> - The user does not have permissions for any of the controls in the container.
>
> - The container does not contain any controls (Preview Tools will warn you if an empty container have been hidden).

Control	Description
Expander	An expander control can be expanded to show its contents, or collapsed to hide the contents. Users can use expander controls to view or hide information based on their interests.
Section	A section on a print form that shows all of its content because it cannot be collapsed and cannot contain tabs.
GroupBox	A group box provides groups related information in an outlined area on the form.
Panel	A panel is a rectangular area that holds related information.
Tab	A user can click a tab in a series of tab controls to view different types of information. Tab controls save space on the form. This control is not available on print forms.

For print forms, only **Section**, **GroupBox**, and **Panel** are available in the **Container Controls** section, and the **Show collapsed on startup** option for them is disabled.

Other Controls

The **Other Controls** section contains controls that users interact with. It also lets you add images to a form.

Button, **HyperLink**, and **Image** controls can be used to perform various functions, depending on whether they are on a state form or transition form. They can be used in addition to or instead of standard controls, such as transition buttons and **Actions** drop-down list items. For more information, see Behavior Tab of the Control Property Editor [page 307].

For information about options for these controls, see General Tab of the Form and Control Property Editor [page 289].

Control	Description
Button	A button that opens the Web page you specify, opens a popup window, or executes a transition.
	Note: Standard buttons that users click at the top of a transition form to transition items are automatically added to a button bar at the top of state and transition forms. These buttons can be removed if custom transition controls are added to the form. (See **Important** note, above.)
ComboBox	Lets the user select an item from a list.
EditBox	Lets the user type or edit items in a field. For example, this could be a **Password** field.
HyperLink	A link that opens the Web page you specify, opens a popup window, or executes a transition.
Image	Lets you add an image to the form. If configured, opens the Web page you specify, opens a popup window, or executes a transition.
List Box	Lets the user select an item from a list.
Text	Lets the user type information into a field. For example, this could be a **Description** field in which the user types the description of a software problem.

For print forms, only **Image** and **Text** are available in the **Other Controls** section, and on the **Refresh** tab of their Property Editors, **On page load** is the only refresh option.

> **Note:** Images are printed only if users enable background printing in their browsers.

Widgets

The controls in the **Widgets** section represent widgets (small applications that provide a special function). For detailed information about the available widgets, see Using Form Widgets [page 255]. Custom widgets may also be shown if you have imported form extensions.

For print forms, only the **Embedded Report** widget is available in the **Widgets** section.

Insert Dialog Box

Use this dialog box to specify the layout, label position, and number of rows and columns in the expander, group box, panel, tab, or print form section you are dragging to the form.

Note: This dialog box provides a starting point. In the Property Editor for the form, and from the context menu that opens when you right-click a row or column, you can add and delete rows and columns, or change the width of columns and the height of rows on an existing form.

JavaScript Editor

The JavaScript editor lets you create a new JavaScript file, or view and modify an imported JavaScript file.

Element	Description
Name	By default, this is the file name of the imported JavaScript file without the `.js` extension. You can change this name, if you prefer.
Description	An area for an optional description of the JavaScript file.
Source file	The original location and file name of the imported JavaScript file.
Import	Replaces the JavaScript file with a different or updated file. Click this button to locate and import the file.
Save to file	Creates a JavaScript file from the source code in the *Source code* box and saves it to the location you specify. **Note:** Use this command only if you want to use the file outside of SBM Composer.
Code	An area for typing, pasting, and editing JavaScript source code.
Script size	The size of the JavaScript file, in number of rows and bytes.

Form Preview Dialog Box

This dialog box shows how the selected form would look to users.

> **Note:** You can preview a state form from the Form Preview Dialog Box [page 317] if the selected role has any "view item" privilege granted to it in the Roles Editor [page 409]. You can preview a transition form if the selected role has any "transition item" privilege. (An exception to this is the "Update" transition type, which requires any "update item" privilege.)

Element	Description
Context	Select the context to preview. Context options depend on where the form is used. For example, you can select a state or transition that uses or inherits the form. If the form is set as a default form for a workflow, you can select any state or transition that is not assigned a different form. This enables you to see how the custom form will appear to users as items move through the workflow.
Role	Select from the roles authorized to view the form. This enables you to see how the form will appear to users based on the selected context and their role permissions.
Mobile view *(Mobile forms only)*	For mobile forms, you can preview the form in the simple view or the advanced view. By default, only required fields appear in the simple view, while all fields and controls appear in the advanced view.
Theme	Choose to preview the form in Classic or Work Center style. The **Enable HTML5 Features** option determines the appearance of preview mode for the "Classic" style. This setting is located on the **Applications - Form Options** tab of the SBM Composer options dialog. For details, refer to Form Options [page 486].
Enable JavaScripts	Specify whether you want JavaScripts that are included in the form to be executed or suppressed during form preview. > **Note:** This setting is set globally on the Application Options [page 486] dialog box, and is selected by default. Some scripts are too complex to show during form preview; you can clear the check box as needed for individual forms.

Element	Description
Preview Tools *(Modern forms only)*	For responsive forms, you can resize the form to preview how the contents will wrap on smaller screens. You can also click the **Toggle responsive cell borders** option to show cell borders during preview. Warning messages can also appear here. For example, you could be warned about vertical double cells containing content other than label/field pairs, or informed if an empty row or container has been automatically hidden.

Container Control Options

In the form editor, you can right-click a container control and select from options relevant to that control.

> **Note:** You can also access container control options in the control Property Editor. For information about these options, see General Options for Container Controls [page 294].

Expander Control Options

Option	Description
Collapse	Collapses the control on the form. > **Note:** The Collapse option collapses the control on the form, but not in Preview mode or at runtime. To collapse the control in these places, select **Show Collapsed on Startup** on the **General** tab of the control Property Editor.
Select Element	For information about the **Select Element** option, see Selecting Parent Controls or Cells on a Form [page 277].
Row Sizing	Lets you select the sizing behavior for the rows in the control. For more information, see Resizing Columns and Rows [page 274].
Column Sizing	Lets you select the sizing behavior for the columns in the control. For more information, see Resizing Columns and Rows [page 274].
Add Row Above	If you select the control, adds a row to the form above the control. If you select a cell within the control, adds a row to the control, above the selected cell.
Add Row Below	If you select the control, adds a row to the form below the control. If you select a cell within the control, adds a row to the control, below the selected cell.

Option	Description
Add Column Left	If you select the control, adds a column to the form to the left of the control. If you select a cell within the control, adds a column to the control, to the left of the selected cell.
Add Column Right	If you select the control, adds a column to the form to the right of the control. If you select a cell within the control, adds a column to the control, to the right of the selected cell.
Delete Row(s)	If you select the control, deletes the row containing the control. If you select a cell within the control, deletes the row below the selected cell. You can select multiple cells to delete multiple rows.
Delete Column(s)	If you select the control, deletes the column containing the control. If you select a cell within the control, deletes the column to the right of the selected cell. You can select multiple cells to delete multiple columns.
Container Options	Changes the container control to the selected type of container control.
Delete	Deletes the selected container control.
Edit Text	Lets you change the label on the container control.
Show Properties	Shows the Property Editor for the container control, if not already shown.

GroupBox Control Options

Option	Description
Select Element	For information about the **Select Element** option, see Selecting Parent Controls or Cells on a Form [page 277].
Row Sizing	Lets you select the sizing behavior for the rows in the control. For more information, see Resizing Columns and Rows [page 274].
Column Sizing	Lets you select the sizing behavior for the columns in the control. For more information, see Resizing Columns and Rows [page 274].
Add Row Above	If you select the control, adds a row to the form above the control. If you select a cell within the control, adds a row to the control, above the selected cell.

Option	Description
Add Row Below	If you select the control, adds a row to the form below the control. If you select a cell within the control, adds a row to the control, below the selected cell.
Add Column Left	If you select the control, adds a column to the form to the left of the control. If you select a cell within the control, adds a column to the control, to the left of the selected cell.
Add Column Right	If you select the control, adds a column to the form to the right of the control. If you select a cell within the control, adds a column to the control, to the right of the selected cell.
Delete Row(s)	If you select the control, deletes the row containing the control. If you select a cell within the control, deletes the row below the selected cell. You can select multiple cells to delete multiple rows.
Delete Column(s)	If you select the control, deletes the column containing the control. If you select a cell within the control, deletes the column to the right of the selected cell. You can select multiple cells to delete multiple columns.
Container Options	Changes the container control to the selected type of container control.
Delete	Deletes the selected container control.
Edit Text	Lets you change the label on the container control.
Show Properties	Shows the Property Editor for the container control, if not already shown.

Panel Control Options

Option	Description
Select Element	For information about the **Select Element** option, see Selecting Parent Controls or Cells on a Form [page 277].
Row Sizing	Lets you select the sizing behavior for the rows in the control. For more information, see Resizing Columns and Rows [page 274].
Column Sizing	Lets you select the sizing behavior for the columns in the control. For more information, see Resizing Columns and Rows [page 274].

Option	Description
Add Row Above	If you select the control, adds a row to the form above the control. If you select a cell within the control, adds a row to the control, above the selected cell.
Add Row Below	If you select the control, adds a row to the form below the control. If you select a cell within the control, adds a row to the control, below the selected cell.
Add Column Left	If you select the control, adds a column to the form to the left of the control. If you select a cell within the control, adds a column to the control, to the left of the selected cell.
Add Column Right	If you select the control, adds a column to the form to the right of the control. If you select a cell within the control, adds a column to the control, to the right of the selected cell.
Delete Row(s)	If you select the control, deletes the row containing the control. If you select a cell within the control, deletes the row below the selected cell. You can select multiple cells to delete multiple rows.
Delete Column(s)	If you select the control, deletes the column containing the control. If you select a cell within the control, deletes the column to the right of the selected cell. You can select multiple cells to delete multiple columns.
Container Options	Changes the container control to the selected type of container control.
Delete	Deletes the selected container control.
Show Properties	Shows the Property Editor for the container control, if not already shown.

Tab Control Options

Option	Description
Add New Tab	Adds a new tab to the control, to the right of the rightmost tab.
Select Element	For information about the **Select Element** option, see Selecting Parent Controls or Cells on a Form [page 277].

Option	Description
Row Sizing	Lets you select the sizing behavior for the rows in the control. For more information, see Resizing Columns and Rows [page 274].
Column Sizing	Lets you select the sizing behavior for the columns in the control. For more information, see Resizing Columns and Rows [page 274].
Add Row Above	If you select the control, adds a row to the form above the control. If you select a cell within the control, adds a row to the control, above the selected cell.
Add Row Below	If you select the control, adds a row to the form below the control. If you select a cell within the control, adds a row to the control, below the selected cell.
Add Column Left	If you select the control, adds a column to the form to the left of the control. If you select a cell within the control, adds a column to the control, to the left of the selected cell.
Add Column Right	If you select the control, adds a column to the form to the right of the control. If you select a cell within the control, adds a column to the control, to the right of the selected cell.
Delete Row(s)	If you select the control, deletes the row containing the control. If you select a cell within the control, deletes the row below the selected cell. You can select multiple cells to delete multiple rows.
Delete Column(s)	If you select the control, deletes the column containing the control. If you select a cell within the control, deletes the column to the right of the selected cell. You can select multiple cells to delete multiple columns.
Delete	Deletes the selected container control.
Show Properties	Shows the Property Editor for the container control, if not already shown.

Widget Settings

This section contains the following topics:

- Embedded Report Widget [page 323]

- Embedded Report Configuration Dialog Box [page 324]

- HTML/JavaScript Widget [page 327]

- PDF Widget [page 329]

- REST Grid Widget [page 330]

- Relational Grid Widget [page 335]

- Relational Grid Configuration Dialog Box [page 338]

- Web Page Widget [page 339]

- Widget Refresh Tab [page 339]

Embedded Report Widget

This topic describes Embedded Report widget settings. For details on using the widget, refer to Using the Embedded Report Widget [page 256].

General Tab

Note: If the widget is configured as a **Relational grid** type report, it uses the grid style. You can customize the colors and text in the Styles Editor [page 439].

Field	Description
Name	Type the name by which the widget is uniquely identified. No other widget or control on this form can have the same name.
Caption	Type the text to appear in the title bar above the widget. If you leave this field blank, the widget will not have a title bar.
Report	Displays the name of the configured report. If none has been configured, click **Configure Report**, and complete the dialog box that opens as described in Embedded Report Configuration Dialog Box [page 324]. Click **Clear** to remove the configuration and start over.
Options	**Border**: Select this check box to draw a thin line around the widget. **Scroll bars**: Select this check box to include scroll bars in the widget. This provides more content than the allotted space. **Check box selection**: Select this option to place check boxes next to each row in the report. If the check box is cleared, users select multiple items by using the **CTRL** key. This option is only available for Relational grid type reports, and only for widgets based on *Multi-Relational* fields.
Description	Type an optional description of the widget.

Query Tab

Field	Description	Applicable To
Report Template	Indicates the report template that will be used at runtime. You can specify a different template as long as it is stored in the `reports` directory on the SBM server and is compatible with your report type. Specify the template in the following format: `reports/`*templateName* For details on creating custom report templates, refer to *SBM System Administrator Guide*.	SBM Composer report definitions
Report Project	You can select one of the following options: • **Current project** specifies the project into which the item was submitted. • **Base project** specifies the header project at the top of the project hierarchy. • **Top level project** specifies the project immediately below the base project in the hierarchy. • **Named project** specifies any other project in the hierarchy. This should be the internal name of the project as shown in SBM Application Administrator.	SBM Composer report definitions based on a primary table
FIELD_NAME	Set values for the query-at-runtime parameters for the report. You can use static values or be bound to other field or control values. See Using the String Builder Tool [page 277] for information about binding values. **Important:** When you bind certain types of fields, two binding values are presented: *fieldname* and *fieldname*[ID]. When you bind to a field that has been configured with the Query at Runtime search feature, you should use *fieldname*[ID] if it is available.	SBM Composer report definitions and Relational grid

Refresh Tab

See Widget Refresh Tab [page 339].

Embedded Report Configuration Dialog Box

This dialog box opens when you click **Configure Report** on the **General** tab of the Property Editor for the Embedded Report Widget [page 323] and the Relational Grid Widget [page 335] (when you select the **Reference Link** report type). You must define or create the report before you can complete this dialog box.

The following describes the configuration settings for a new embedded report.

For details on using the widget, refer to Using the Embedded Report Widget [page 256].

Element	Description
Report type	Select one of the following options: • **Composer report definition** Embed an application report against a primary or auxiliary table. • **Reference Link** Embed a report created in SBM Work Center. • **Relational grid** Embed an application report of related items based on a relational field. This option applies only to the Embedded Report widget.
Application, Report (**Composer report definition** option)	Select the application in which the report was defined and the report definition. The application and report can be in this process app or a referenced application in another process app.

Element	Description
Relational field, Report (**Relational grid** option)	Select the relational field you want to report against and the report definition. The relational field can be mapped to a primary or auxiliary table in this application or to a table in a referenced application in another process app. The report definition must be based on this table. For example, if in Process App A you want to display a list of relational field values for Auxiliary Table A located in Process App B: 1. Open Process App B. 2. Select Auxiliary Table A in the **Data Design** area of the App Explorer. 3. Right-click, and then select **Create Report Definition for this table**. 4. Create the report definition. 5. Close Process App B, and open Process App A. 6. Create a reference to Process App B. (For steps, refer to Defining a Reference [page 371].) 7. Add a *Single Relational* or *Multi-Relational* field based on Auxiliary Table A. 8. Configure the embedded report widget in Process App A to use the report created in Process App B.
Report address (**Reference Link** option for the Embedded Reports widget)	Copy the report URL from SBM Work Center and then paste it into this box. After you close the dialog box, the **Report name** field on the **General** tab of the widget Property Editor is populated with the report name that you specified in SBM Work Center. An example URL is: `tmtrack.dll?ReportPage&template=reports%2Flist&ReportRef=` `→UBG_ISSUES.NewIncomingIssues&HasRuntimeParams=` `→1&F_TS_ACTIVEINACTIVE=&embedded` **Important:** After you save a report in SBM Work Center, you can obtain a special URL that contains the report reference name. The report reference name must be used if you want the embedded report to work after the process app containing the report is promoted to another environment. For more information, see Using the Embedded Report Widget [page 256].

Element	Description
Relational field, Report URL (**Reference Link** option for the Relational Grid widget)	Select the relational field you want to report against and the report URL. For details on the report URL, see the description for the *Report address* (above).
Inputs	Shows the input parameters for the report. Change parameters on the **Query** tab of the widget Property Editor.
Automatically bind compatible parameters	Select this option if you want fields that are configured as "query-at-runtime" to be automatically bound on the **Query** tab of the widget Property Editor when you complete this configuration dialog box.

HTML/JavaScript Widget

This topic describes how to configure the HTML/JavaScript widget.

General Tab

Field	Description
Control name	Type the name by which the widget is uniquely identified. No other widget or control on this form can have the same name.
Caption	Type the text to appear in the title bar above the widget. If you leave this field blank, the widget will not have a title bar.

Field	Description
Content	Type or paste the HTML or JavaScript code in this field. When the form is displayed, the code is rendered by the Web browser.

Alternately, you can use the string builder tool to insert references to table fields and form controls. For information about this tool, see Using the String Builder Tool [page 277].

> **Note:** If you use the string builder tool in JavaScript code in the HTML/JavaScript widget, the code needs to be in an iframe (inline frame). If the code is in an iframe, it cannot reference other parts of the form. Therefore, the string builder tool should not be used in JavaScript code.

> **Important:**
>
> • If the code you type in this field includes any opening or closing braces ("{" or "}"), precede each of them with a backslash ("\"). Otherwise, SBM Composer tries to interpret the text inside of the braces as the name of a field or control on the form. This prevents the form from working properly. (If you paste the brace, you are prompted whether you want a backslash to be added. If you use the Treat '{' as a literal option in the string builder tool, a backslash is automatically added.)
>
> • If you are using string substitution to initialize a JavaScript string constant, you must use the `STRING_ESC()` macro. This ensures that special characters are correctly encoded for use in this context.
>
> The example below shows the content of a HTML/Javascript widget on a form with a single EditBox. The first reference is inside the HTML and therefore does not need any escaping, while the second reference is inside the JavaScript and needs to be escaped in order to be correctly processed.

Name: HTMLJavascriptWidget

Caption: Line: 1 Position: 1

Content:
```
<div>{EditBox}</div>
<script>
 var s = "STRING_ESC({EditBox})";
 s = htmlDecode(s);
 alert(s);
</script>
```

Field	Description
Options	**Border:** Select this check box to draw a thin line around the widget. **Scroll bars:** Select this check box to include scroll bars in the widget. This provides more content than the allotted space.
Description	Type an optional description of the widget.

Refresh Tab

See .

PDF Widget

This topic describes PDF widget options. For details on using the widget, refer to .

General Tab

Field	Description
Name	Type the name by which the widget is uniquely identified. No other widget or control on this form can have the same name.
Caption	Type the text to appear in the title bar above the widget. If you leave this field blank, the widget will not have a title bar.
Template	Browse to a PDF template that contains interactive form fields that will be filled in when the application runs. **Important:** You are asked to select another PDF template if you select a PDF template that does not allow interactive form fields to be filled in or that requires a password or certificate to open.
Display in	**Form:** Select this option if you want the PDF document to be displayed within the form. **New Window:** Select this option if you want the PDF document to be displayed in a new window.
Options	**Border:** Select this check box to draw a thin line around the widget. **Scroll bars:** Select this check box to include scroll bars in the widget. This provides more content than the allotted space.
Description	Type an optional description of the widget.

Parameters Tab

The **Parameters** tab contains the input fields that are in the PDF document you imported. You map table fields and form controls into the fields on this tab.

For example, suppose you have an Item Type field on a Submit form. In the PDF document, you also have an **Item Type** input field. In this field on the **Parameters** tab, type { and then select **Item Type**. When a user selects a value for the *Item Type* field, the field in the PDF document is populated with the same value.

> **Important:** The mapping of table fields and controls is preserved when you reimport a PDF document that was used as a template. For example, mapping is not affected if you change the type of a field in the PDF document, as long as the new field type contains inputs with the same names as the associated field in the SBM form. For example, you can change a Combo Box control to a List Box control, as long as the values that users can select are the same as the values configured for the associated field in SBM Composer.

> **Note:**
>
> - Required PDF input fields are displayed on this tab with a red asterisk (*).
>
> - For information about mapping fields and controls, see Using the String Builder Tool [page 277].
>
> - You can type default values for fields on the form instead of mapping.

Refresh Tab

Use the **Refresh** tab to update values that refer to external data, such as field values. If the widget does not refer to any external data, the **Refresh** tab options are disabled.

Field	Description
Refresh contents	**On page load:** If this check box is selected, the content of the widget is retrieved when the form is initially displayed at runtime or in the form preview. > **Note:** This check box is selected by default. **On data change:** If this check box is selected, the content of the widget is updated when fields that map to values on the **Parameters** tab are changed at runtime or in the form preview. **On click:** In the drop-down list box, select one of the buttons, images, or hyperlinks that are on this form. The content of the widget is updated when the control is clicked at runtime or in the form preview.

REST Grid Widget

This topic describes REST Grid widget settings. For details on using the widget, refer to Using the REST Grid Widget [page 260].

General Tab

Field	Description
Name	Type the name by which the widget is uniquely identified. No other widget or control on this form can have the same name.
Caption	Type the text to appear in the title bar above the widget. If you leave this field blank, the widget will not have a title bar.
URL	Displays a URL to a configured REST service. If none has been configured, click **Configure URL**. Complete the dialog box that opens as described in REST Service Configuration Dialog Box [page 468]. Click the **Clear** button to remove the URL and start over.
Paging	Select paging options to include buttons at the top of the widget that let users navigate between pages in the result set. Select from the following paging options: **Off:** No buttons for paging. **On (Server-Side):** Paging is done on the server side, so that there is never more data returned to users' machines than they are currently viewing. If this option is selected, the configured REST service must support paging and users cannot click on column headers to sort data in ascending or descending order. For information on configuring the REST service to support this, refer to Query Tab [page 332] **Parameters/ Values** entry. **On (Client-Side):** Paging is done on the client side, and the returned results are stored in the browser. When you select this option, the **Results per page** option is also enabled, where you specify the number of records to return per page. For example, if **Results per page** is set to 10, at runtime the REST service might return 100 records to the browser, and user would use the paging buttons to page through 10 at a time. **Note:** When using the client-side option, you must properly filter your data before returning it so that the number of results returned is not so large that it impacts browser performance.
Options	**Border**: Select this check box to draw a thin line around the widget. **Scroll bars**: Select this check box to include scroll bars in the widget. This provides more content than the allotted space. **Include lower page bar**: Select this check box if you want to include paging buttons at both the top and the bottom of the widget.
Description	Type an optional description of the widget.

Note: This widget uses the grid style. You can customize the colors and text in the Styles Editor [page 439].

Query Tab

Use this tab to override the parameters and values for the REST service.

Field	Description
URL	Indicates the base REST URL. During deployment, the endpoint is generated using this URL.
Parameters/ Values	URL parameters are listed below the URL. Values can be static values or be bound to other field or control values.

For the Parameters/Values row:

Important:

If **Server-Side** paging is selected on the **General** tab, you must map a parameter to keep track of the paging to the `RESTGridWidget._currentPage` or `RESTGridWidget._currentIndex` parameter.

Otherwise, during validation, you receive a warning message, and when the user pages, the previous results are displayed. (See Using the String Builder Tool [page 277] for instructions.)

This parameter mapping is not necessary for **Client-Side** paging.

Result Tab

Note: During widget configuration, you will receive a warning message if the amount of data in the result set exceeds 100 rows. The message includes ways to limit the result set.

Field	Description
Render as	• **Grid**: Select this option if you want the search results to be presented as an HTML table in which each row contains one returned item, with a titled column for each data element you select under **Grid columns**. • **Tiles**: Select this option if you want the search results to be presented as an HTML table in which each table cell contains all the content for a single item, and each row contains the number of cells you specify in the per row list. *N* **per row**: This list is enabled if you selected the **Tiles** option. Specify the number of tiles, or table cells, that you want on each row. For example, if you specify 2, there are will be five rows in the initial presentation (assuming that there are at least ten items to be returned). **Note:** If you specify a number other than 1, 2, 5, or 10 (that is, a number evenly divisible by ten), the last row will contain fewer tiles than the other rows. **Enable row selection**: Lets users populate a control such as a text box with the data from a selected row. If this check box is cleared, the **Columns the grid will display** table contains an additional **Action on click** column. **Enable multi-selection**: Lets users select multiple rows in the REST grid by pressing Ctrl+click or Shift+click. If you map this to a single text field, the field will be populated with a comma-separated list (for example, "value1,value2,value3"). **Autoselect first row:** Select this option to automatically select the first row of the grid on refresh.
Output: Available Data	Shows the data elements you can include in the result set. Bold nodes indicate valid elements that you can move to the **Grid columns** or to the **Tile content** pane. As you type in the box above the list, SBM Composer filters the list to show only the data elements with names that contain the characters you typed.

Field	Description
Grid Columns	• **Value** Shows the data elements from the widget that you can include in the result set. • **Caption** Accept the default column name or type another name. • **Display as** Select **Text**, **Url**, or **Image**. • **Visibility** Select **Visible**, **Always Visible**, **Hidden**, or **Always Hidden**. Columns set to *Visible* or *Hidden* are available for end users to show or hide at runtime. (To show or hide columns, the end users click the down arrow on the right side of the column headings and select or deselect check boxes beside the available columns. The columns they choose to show or hide will persist when they view the widget instance again in the same browser. They can click **Show All** to select all columns for display.) Columns set to *Always Hidden* are not selectable to display, but the data is still available to be mapped as input to other fields or used in JavaScript code. Columns set to *Always Visible* cannot be removed from the display by end users. The **Action on click** column is not available if you are binding to widget data, because when you bind to widget data, there are other links, and users would find it difficult to determine what they are clicking. **Note:** For more information about binding to widget data, see Binding to REST Widget Data [page 261].

Field	Description
Tile Content	Type or paste the content for each tile in the result set.
	Alternatively, you can use the string builder tool to insert references to table fields and form controls.
	For example, you could use the following code to put a small HTML table in each tile:
	``` <table width='100%'> <tr> <td rowspan='2' width='80px'>{SmallImage}</td> <td>{ItemAttributes.Title}</td> </tr> <tr> <td>{OfferSummary.LowestNewPrice}</td> </tr> </table> ```

Refresh Tab

See Widget Refresh Tab [page 339].

Relational Grid Widget

This topic describes Relational Grid widget settings. For details on using the widget, refer to Using the Relational Grid Widget [page 257].

General Tab

Field	Description
Name	Type the name by which the widget is uniquely identified. No other widget or control on this form can have the same name.
Caption	Type the text to appear in the title bar above the widget.
	If you leave this field blank, the widget will not have a title bar.
Report	Displays the name of the configured report. If none has been configured, click **Configure Report**, and complete the dialog box that opens as described in Relational Grid Configuration Dialog Box [page 338].
	Click **Clear** to remove the configuration and start over.

Field	Description
Options	**Border**: Select this check box to draw a thin line around the widget.
	Scroll bars: Select this check box to include scroll bars in the widget. This provides more content than the allotted space.
	Check box selection: Select this option to place check boxes next to each row in the report. If the check box is cleared, users select multiple items by using the **CTRL** key. This option is only available for widgets based on *Multi-Relational* fields.
	Radio button selection: Select this option to use a radio button to make the selection. If this check box is cleared, users select the item by clicking on it (which is not as visually obvious as the radio button). This option is only available for widgets based on *Single Relational* fields.
	Title: Select this option to show the title ("*Report* for field *relational field*") in the relational grid header.
	Details: Select this option to show item count and applied filters in the relational grid header.
	Requery: Select this check box to show the **Requery** link in the relational grid footer.
	Select/Unselect all: Select this option to show the **Select All** and **Unselect All** links in the relational grid footer. This option is only available on transition or edit forms, and for widgets based on *Multi-Relational* fields.
	Clear: Select this option to show the **Clear** link in the relational grid footer. This option is only available on transition or edit forms.
	Show only selected item: Select **yes** to show only the items the user has currently selected in the relational grid or **no** to show all items. For transition forms, the additional option **if read only** limits the relational grid to show the selected read-only items as well as editable items.
	Display 'Show only selected items' checkbox: Select an option to show or hide the **Show only selected items** checkbox in the relational grid footer. For transition forms, the additional options **if editable** and **if read only** enable you to show the checkbox depending on if the items in the relational grid are editable or read only.
Description	Type an optional description of the widget.

Query Tab

Field	Description	Applicable To
Report Project	You can select one of the following options: • **Current project** specifies the project into which the item was submitted. • **Base project** specifies the header project at the top of the project hierarchy. • **Top level project** specifies the project immediately below the base project in the hierarchy. • **Named project** specifies any other project in the hierarchy. This should be the internal name of the project as shown in SBM Application Administrator.	SBM Composer report definitions based on a primary table
FIELD_NAME	Set values for the query-at-runtime parameters for the report. You can use static values or be bound to other field or control values. See Using the String Builder Tool [page 277] for information about binding values. **Important:** When you bind certain types of fields, two binding values are presented: *fieldname* and *fieldname*[ID]. When you bind to a field that has been configured with the Query at Runtime search feature, you should use *fieldname*[ID] if it is available.	SBM Composer report definitions and SBM Work Center reports
Searchable	Enables users to filter the contents of the relational grid by entering a value for a specific field. To be searchable, the field must be defined as a column and as a query-at-runtime search filter in the report definition. The following fields and field types are searchable: *Item Id*, *Text*, *Project*, *Company*, *User*, *Multi-User*, *Multi-Group*, *Single Selection*, *Multi-Selection*, *Folder*, and *Numeric* (exact value, not "contains"). These field types are not searchable: *File*, *URL*, *Single Relational*, *Multi-Relational*, and *Sub-Relational*. **Note:** If a field is used multiple times as a search filter, you cannot make it searchable. This prevents confusion about which search filter is being applied to the grid.	SBM Composer report definitions

Refresh Tab

See Widget Refresh Tab [page 339].

Relational Grid Configuration Dialog Box

This dialog box opens when you click **Configure** on the **General** tab of the Property Editor for the Relational Grid Widget [page 335]. You must select a relational field and report before you can complete this dialog box.

The following describes the configuration settings for a new relational grid report based on an SBM Composer report definition. To configure a relational grid report based on an external report (**Reference Link** report type), refer to Embedded Report Configuration Dialog Box [page 324].

For details on using the widget, refer to Using the Relational Grid Widget [page 257].

Element	Description
Relational field	Select the relational field you want to report against. The relational field can be mapped to a primary or auxiliary table in this application or to a table in a referenced application in another process app.
Report	Select the report definition. The report definition must be based on the table that is referenced by the relational field. For example, if in Process App A you want to display a list of relational field values for Auxiliary Table A located in Process App B: 1. Open Process App B. 2. Select Auxiliary Table A in the **Data Design** area of the App Explorer. 3. Right-click, and then select **Create Report Definition for this table**. 4. Create the report definition. 5. Close Process App B, and open Process App A. 6. Create a reference to Process App B. (For steps, refer to Defining a Reference [page 371].) 7. Add a *Single Relational* or *Multi-Relational* field based on Auxiliary Table A. 8. Configure the relational grid in Process App A to use the report created in Process App B.
Inputs	Shows the input parameters for the report. Change parameters on the **Query** tab of the widget Property Editor.
Automatically bind compatible parameters	Select this option if you want fields that are configured as "query-at-runtime" to be automatically bound on the **Query** tab of the widget Property Editor when you complete this configuration dialog box.

Web Page Widget

This topic describes how to configure the Web Page widget.

> **Note:** Do not use this widget for SBM report URLs. Use the Embedded Report Widget [page 323] instead.

General Tab

Field	Description
Control name	Type the name by which the widget is uniquely identified. No other widget or control on this form can have the same name.
Caption	Type the text to appear in the title bar above the widget. If you leave this field blank, the widget will not have a title bar.
URL	Type the address of the Web page you want to display. Alternately, you can use the string builder tool to insert references to table fields and form controls. (For information about this tool, see Using the String Builder Tool [page 277].) When the form is displayed, the widget is replaced with the content from that URL (along with a caption, a border, and scroll bars, if you enable those options.)
Options	**Border:** Select this check box to draw a thin line around the widget. **Scroll bars:** Select this check box to include scroll bars in the widget. This provides more content than the allotted space. **Use SSO Authentication:** Select this check box to enable Single Sign-On (SSO) authentication for SSO-enabled URLs.
Description	Type an optional description for the widget.

Refresh Tab

See Widget Refresh Tab [page 339].

Widget Refresh Tab

This topic describes how to configure the **Refresh** tab in the Property Editor for each widget.

Refresh Tab

Use the **Refresh** tab to specify when values that refer to external data, such as field values, are updated. If the widget does not refer to any external data, the **Refresh** tab options are disabled.

The following options are available for the **Refresh contents** field on the **Refresh** tab.

On page load: If this check box is selected, the content of the widget is retrieved when the form is initially displayed at runtime or in the form preview.

> **Note:** This check box is selected by default.

On data change: If this check box is selected, the content of the widget is updated when the values of certain fields are changed at runtime or in the form preview. These are the fields that are bound to database fields in the widget Property Editors. (See Using the String Builder Tool [page 277] for information about binding fields.)

On click: In the drop-down list box, select one of the buttons, images, or hyperlinks that are on this form. The content of the widget is updated when the control is clicked at runtime or in the form preview.

SBM Composer Extensions Editor

The **SBM Composer Extensions Editor** enables you to view extensions you have imported into your system. You can also export and reimport extensions.

For details, refer to Using Form Extensions [page 266].

The following table describes extension settings, which cannot be modified.

Element	Description
Name	Indicates the extension name.
Revision	Indicates the extension version.
Author	Indicates the extension author. Extensions may be created by SBM Development or by third-party solution developers.
Copyright	Shows copyright information provided by the extension author.
Size	Indicates the size of the extension.
Description	Shows a brief description provided by the extension author.
Documentation	Shows detailed information provided by the author about the extension.
File	Indicates the location of the extension file.
Reimport	Click this button to reimport the extension. You may need to do this if a new version of an extension is available, for example.
Export	Click this button to export the extension.

Chapter 14: Using Conditional Routing

This section contains the following information:

- About Conditional Routing [page 341]

- Using Conditional Routing [page 344]

- Conditional Routing Settings [page 345]

- Conditional Routing Tutorial [page 347]

About Conditional Routing

This section contains the following topics:

- Conditional Routing Overview [page 341]

- About Decisions [page 342]

- Use Cases: Conditional Routing [page 343]

- Best Practices: Conditional Routing [page 343]

Conditional Routing Overview

Conditional routing lets designers create application workflows that automatically route an item to a particular state based on visually designed rules that evaluate item data. They are used in decisions to determine which outgoing transition is executed. (For information about rules, see About Rules [page 351].)

One or more incoming transitions lead from one or more source states to a decision, and two or more outgoing transitions lead from the decision to their respective target states. The rules are evaluated in the order you specify in the Rules Tab of the Decision Property Editor [page 346]. If a rule evaluates to "true," the associated transition is taken.

The last transition in the Property Editor list must be mapped to an "Otherwise" rule. This transition is taken if the last rule evaluates to "true" or if no rule evaluates to "true."

> **Note:** For step-by-step instructions for using conditional routing, see Conditional Routing Tutorial [page 347]. For examples of how conditional routing can be used, see Use Cases: Conditional Routing [page 343]. For ways to implement conditional routing most effectively, see Best Practices: Conditional Routing [page 343].

Key Benefits

- Conditional routing is an intuitive way to route items based on certain conditions. You no longer need to use actions and scripts to accomplish this.

- The decision in the workflow editor lets you see the routing options at a glance.

- A rules editor lets you build rule expressions graphically.

- The Property Editor for the decision displays the rule associated with each outgoing transition, and optionally shows detailed information about the rule and its associated transition. You can open the rules editor from this Property Editor to add and edit rules.

- Application variables make it easy to create and maintain rules.

About Decisions

Decisions enable conditional routing in application workflows or sub-workflows. The decision is included in the **Workflow Palette**, and is added to a workflow by a drag-and-drop operation, the same way states and transitions are added to a workflow.

A decision has one or more incoming transitions and two or more outgoing transitions. An incoming transition connects a source state to the decision, and the outgoing transitions connect the decision to target states. When you add a decision to a workflow, two outgoing transitions are automatically added: **Otherwise** and **Branch**.

The decision evaluates rules that are associated with each outgoing transition, and the transition associated with the first rule that evaluates as "true" is executed. If a role or item type is restricted from using an outgoing transition, the workflow evaluates the rule associated with the first non-restricted outgoing transition. If no outgoing transition is allowed for a role or item type, then the transition associated with the "Otherwise" rule is executed. For information about rules, see Rule Overview [page 351].

If a decision fails to complete when the application runs, the item that entered the step will remain in the source state. This prevents changes that were saved to the item by completed actions or by the user in a transition form from being lost. (Decision failures are caused by a user abandoning or cancelling a transition form or by an error.)

When the incoming transition to a decision is the default **Submit** transition, the transition privileges that apply to that transition also apply to all outgoing transitions from the decision. This allows the workflow to proceed to the target state even when the submitter has limited transition privileges.

Restriction:

- You cannot use the **Post** or **Subtask** transition type for incoming transitions to a decision.

- You cannot use a **Copy** or **Update** transition type on a decision.

- Using a "Transition", "Orchestration Workflow (continue executing)," or "Trigger" action type on an incoming transition to a decision is not currently supported.

- You cannot restrict the **Otherwise** transition from a decision by role or type.

- There are some field comparison limitations in expressions used to create decision rules. For details, see Rule Operators [page 359].

- You cannot use the "restrict by rule" feature on an outgoing transition from a decision.

Use Cases: Conditional Routing

The following use cases illustrate ways in which conditional routing can be used.

Expense Reporting

Employees use an application to submit expense reports. A decision named **Amount?** has an incoming **Evaluate** transition and three outgoing transitions: **Send to A/P**, **Get Mgr Approval**, and **Get VP Approval**. The workflow evaluates the rules that are associated with the **Amount?** decision.

- If the reimbursement amount is less than or equal to $100.00, no approval is required, so the **Send to A/P** transition is executed.

- If the reimbursement amount is more than $100.00 but less than or equal to $1000.00, a manager's approval is required, so the **Get Mgr Approval** transition is executed.

- If neither of the preceding rules evaluate to "true," then the reimbursement amount is more than $1000.00. In this case, a vice president's approval is required, so the **Get VP Approval** transition is executed.

New Hire Processing

Human resources uses a new hire application to request new computers. A decision named **Employee Type?** has an incoming **Evaluate** transition and two outgoing transitions: **Send to Facilities** and **Send to IT**. The workflow evaluates the rules that are associated with the **Employee Type?** decision.

- If the employment type is "Local," office space is needed for the employee, so the **Send to Facilities** transition is executed. After the office space is allocated, the **Send to IT** transition is executed so a computer can be ordered for the employee.

- If the employee type is not "Local," the employee must be "Remote." In this case a computer is needed, but not office space, so the **Send to IT** transition is executed.

Issue Escalation

IT uses an incident management application to handle trouble tickets and escalate them to a manager if they have been open for more than 14 days. A decision named **Age?** has two outgoing transitions: **Escalate** and **Otherwise**. The workflow evaluates the rules that are associated with the **Age?** decision. If the *Submit Date* in the rule is **Now - 14 days**, the **Escalate** transition is executed. If not, the **Otherwise** transition is executed and the workflow continues in its normal path.

> **Note:** If the value of the *Submit Date* field is converted into an application variable, an administrator can override it in SBM Application Administrator. For example, if a policy change lowers the escalation time to 7 days, the change can be made without changing the rule in SBM Composer. For more information, see the *SBM Application Administrator Guide* or online help.

Best Practices: Conditional Routing

This topic lists ways to implement conditional routing most effectively.

- Cover the entire range of values in your rules. For example, if an item should be routed based on a numeric value, and the user specifies a value that is not covered

by a rule, it must take the transition associated with the **Otherwise** rule. This could result in unintended routing.

- Make rules mutually exclusive. For example, if an item in a defect management application should be routed based on severity, and the severity level can be 1 through 5, make sure each level is covered by only one rule. Do not assign 1 and 2 to one rule, and assign 2 and 3 to another rule.

- Use application variables. They offer the following advantages:

  - You can change the value of an application variable and all of the rules that use the application variable are automatically updated.

  - You can prevent gaps that can cause unintended routing. For example, in an expense reimbursement application, one rule covers reimbursements that are between $0.00 and $100.00, inclusive. Another rule covers reimbursements greater than $100.00 but less than $1000.00.

    If you change the upper limit of the first rule to $90.00, but forget to change the lower limit of the second rule accordingly, then a reimbursement of $95.00 must take the transition associated with the **Otherwise** rule. You can avoid this unintended routing if you instead assign 100 to an application variable, and use that variable in both rules.

- To prevent infinite loops:

  - Do not have two decisions, where each decision has an outgoing transition to the other decision.

  - Do not put a transition action on an outgoing transition from a decision that causes the incoming transition to that decision to be executed.

- Do not restrict the "Otherwise" transition from a decision by role, group, or item type.

- A decision must have one incoming transition from the **Any** state before it can have an outgoing transition back to the **Any** state.

Using Conditional Routing

You can add decisions to an application workflow or sub-workflow, and then modify its properties and rename or delete the decision.

To add a decision:

1. From the **Workflow Palette**, drag the **Decision** item to the workflow editor.

2. Type a name for the decision, and press the Enter key.

3. To reposition the decision, drag it to a new location in the editor.

   Note: You can also double-click the decision in the **Workflow Palette** to add the decision to the workflow. The decision is added to the right of a selected state, on the right side of a selected transition, or to the center of the workflow editor if no state or transition is selected.

To modify a decision's properties:

1. Select the decision in the workflow editor.

2. Modify its properties using the tabs of the step Property Editor, which are described in Conditional Routing Settings [page 345].

To rename a decision:

*   Right-click the decision in the workflow editor, select **Rename**, type the new name, and press the Enter key.

To delete a decision:

*   Right-click the decision in the workflow editor, and select **Delete**.

Conditional Routing Settings

The following sections provide descriptions of settings used for conditional routing. Sections are organized by property editors.

General Tab of the Decision Property Editor

Element	Description
Name	Type or change the decision name, if necessary.
End-user help text	Click **Edit** to open an HTML editor and add optional comments or notes about the purpose of the decision. The text you enter is displayed to users. For details, refer to Chapter 24: Providing Custom End-user Help [page 443].

Rules Tab of the Decision Property Editor

Note: For information about rules, see Rule Overview [page 351].

Element	Description
Rule	Click a row in the **Rule** column to open the rule menu. All rules that are defined for the application are listed at the bottom of the menu. To map an existing rule to the transition, select the rule from the menu. To create a new rule for this transition, select **(New rule)**. The rule editor opens. One transition must be mapped to the "Otherwise" rule. This transition is executed if none of the other rules evaluates as "true." **Important:** The rules are evaluated in the order in which they are listed here, so it is critical that the order is correct. (See **Move up, Move down** later in this table.)
Transition	The outgoing transitions from this decision.
To state	Shows the target state for this transition.
Status	Shows whether the transition is enabled or disabled. Disabled transitions are highlighted in a light gray color on the workflow editor.
Hide disabled transitions	Removes disabled transitions from the **Transition** column.
Override transition ordering	*Sub-workflows only:* Enables the **Move up** and **Move down** buttons so you can change the transition order in a sub-workflow without changing the order in the parent workflow.
Show rule summaries	Expands the area under each row and provides a textual summary of the rule.
Show transition summaries	Expands the area under each row and shows the form the transition uses, the state the workflow moves to after the transition is executed, and other information about the transtition (if any).
Open rule	Opens the rule editor for the selected rule.
Move up, Move down	Changes the order in which rules are evaluated. **Note:** After you map the "Otherwise" rule to a transition, it is moved to the bottom of the list, because that transition is taken if no other rule evaluates as "true."

Conditional Routing Tutorial

This tutorial shows you how to route an item in an application workflow based on the amount of an expense reimbursement request.

- If the amount is less than $100.00, no approval is required, so the item is routed directly to the **Accounts Payable** and **Processing** states, where the reimbursement is processed. After the reimbursement is processed, the item is closed.

- If the amount is between $100.00 and $1000.00, inclusive, a manager's approval is required, so the item is routed to the **Manager** state. The manager approves the request and transitions the item to the **Accounts Payable** state.

- If the amount is more than $1000.00, a vice president's approval is required, so the item is routed to the **Vice President** state. The vice president approves the request and transitions the item to the **Accounts Payable** state.

> **Note:** This tutorial does not demonstrate all ways to configure decisions and creating rules and application variables. For complete information about these tasks, see Using Conditional Routing [page 344], Creating a Rule [page 353], and Creating an Application Variable [page 367].

To use conditional routing:

1. Create a process app from the Create New Process App Dialog Box [page 67] and name it `Expense Reimbursement`.

2. In the application workflow, add states and a decision as follows:

   a. Drag a **Decision** from the **Workflow Palette** to the workflow editor, and name it `Amount?`.

   b. Drag four "active" states from the **Workflow Palette** to the workflow editor and give them the following names:

      - **Accounts Payable**

      - **Manager**

      - **Vice President**

      - **Processing**

   c. Drag a "completed" state to the workflow and name it **Closed**.

3. Rename the **Branch** transition from the **Amount?** decision to `Send to A/P`, and connect it to the **Accounts Payable** state.

4. Rename the **Otherwise** transition from the **Amount?** decision to **Send to VP**, and connect it to the **Vice President** state.

5. Add "regular" transitions from the **Workflow Palette** as shown in the following table.

> **Note:** Quick transitions do not have forms associated with them and do not require user input. You can drag **Quick** from the **Transitions** section of the **Workflow Palette**, or make a regular transition a quick transition by selecting the **Quick transition (do not show a form)** check box on the **Options** tab or **Form** tab of the transition Property Editor. Outgoing "regular" transitions from a decision are configured as quick transitions by default.

Source	Target	Name	Quick Transition?
New	Amount?	Evaluate	Yes
Amount?	Manager	Send to Mgr	Yes (default)
Manager	Accounts Payable	Send to A/P	No
Vice President	Accounts Payable	Send to A/P	No
Accounts Payable	Processing	Process	No
Processing	Closed	Close	No

6. Add a *Numeric* field to the **Expense Reimbursement** table:

   a. Name the field **Amount**.

   b. On the **Options** tab of the field Property Editor, select **Fixed precision**, specify two digits to be displayed after the decimal point, and specify `$` as the prefix.

7. Select the **Amount?** decision in the workflow editor, and then select the **Rules** tab of the decision Property Editor.

8. Select the **Send to A/P** transition, and then click the arrow in the **Rule** column.

9. Select **(New rule)**. The rule editor opens.

10. On the **General** tab of the rule Property Editor, type `A/P Direct` in the **Name** box.

11. Drag an *Amount* field from the **Rule Palette** onto the **Rule: A/P Direct** block in the rule editor.

   a. Select the **>** conditional operator.

   b. Make sure **Value** is selected in the drop-down list, and type `0` in the box.

12. Drag an **AND** logical operator from the **Rule Palette** onto the **Amount** field.

13. Drag another *Amount* field onto the bottom line of the **AND** operator.

> **Note:** You can optionally skip the previous step and instead drag the other *Amount* field onto the existing *Amount* block. This automatically adds the **AND** logical operator to the expression.

a. Select the **<=** comparison operator.

b. Make sure **Value** is selected in the drop-down list, type `100.00` in the box, and then click **Convert to variable**.

> **Note:** Designers can use the same application variable in multiple rules. They can also change the default value of an application variable, and all rules that reference that field will use the new value. For more information, see About Application Variables [page 367].

14. Click **Edit application variables**.

15. In the application variable editor, rename **Amount Variable** to **Threshold**.

16. Click the **Navigate Backward** button in the quick access toolbar at the top of the SBM Composer window two times.

17. On the **Rules** tab of the **Amount?** decision Property Editor, select the **Send to Mgr** transition, and then click the arrow in the **Rule** column.

18. Select **(New rule)**.

19. On the **General** tab of the rule Property Editor, type `Manager` in the **Name** box.

20. Drag an *Amount* field from the **Rule Palette** to the **Rule: Manager** block in the rule editor.

    a. Select the **>** comparison operator.

    b. Click **Variable** in the drop-down list at the right of the expression, and then select **Threshold**.

21. Drag an **AND** logical operator from the **Rule Palette** onto the **Amount** field.

22. Drag an *Amount* field onto the bottom line of the **AND** operator.

    a. Select the **<=** comparison operator.

    b. Make sure **Value** is selected in the drop-down list, and type `1000.00` in the box.

23. Click the **Navigate Backward** button .

24. On the **Rules** tab of the **Amount?** decision Property Editor, select the **Send to VP** transition, and click the arrow in the **Rule** column.

25. Select **[Otherwise]**. One "otherwise" rule must be mapped to a transition. This transition is executed if the rule mapped to it evaluates to "true," or if no rule evaluates to "true." In this case, any amount over $1000.00 will use this rule and the **Send to VP** transition will be executed.

26. Validate, publish, and deploy the process app.

Chapter 15: Defining Rules

This section contains the following information:

- About Rules [page 351]
- Working with Rules [page 353]
- Rules Settings [page 365]

About Rules

This section contains the following topics:

- Rule Overview [page 351]
- Rule Editor [page 352]
- Restrictions by Rule for a Transition [page 352]

Rule Overview

A rule is an expression or set of expressions that can be used in the following contexts:

- **Conditional routing**

  Rules evaluate how an item should be routed from a decision in an application workflow. For more information, see Conditional Routing Overview [page 341].

- **Transitions**

  Rules can determine whether a transition will be available to users. If the rule evaluates to true, and if no other restrictions on the transition apply, the transition appears on the form for the source state.

- **Form actions**

  Rules can be included in action expressions that define events, conditions, and actions that result in dynamic behavior on a form. For more information, see Working With Form Actions [page 246].

- **Report filters**

  Most rules that you create in SBM Composer can be reused as report filters as well. A report filter is a predefined set of conditions that users can insert into a report definition to limit report results at runtime. This means you can build a complex rule once, and then users can reuse it in one or more SBM reports.

Rules contain evaluation criteria that includes fields from the primary table, comparison operators appropriate for the field type, and values to compare. A value can be converted into an application variable. See About Application Variables [page 367] for the advantages of using application variables.

Rule Editor

The rule editor is primarily used to construct and edit rules that determine:

- How an item should be routed from a decision in an application workflow.

- Whether a transition should be available on a state form.

Rule Palette

The **Rule Palette** includes the following types of controls:

- **Logical Operators** include AND and OR search operators. These operators are used to specify search operators for fields and to group expressions to set a sequence for evaluating rules.

- **Fields** include the fields that can be evaluated. These fields are in the primary table. If you add or remove fields from the primary table, they are added or removed from this section.

Rule Expression

In the rule editor, drag fields and logical operators from the **Rule Palette** to the **Rule: *rule name*** block. Then specify the value, field, or application variable you want to compare to complete the logic of the rule. The **Rule summary** block contains a read-only string that represents the rule as you create it. The name of the rule is displayed on the tab in the rules editor.

> **Note:** Only compatible field types can be evaluated. See Field-to-Field Comparisons [page 358] for a list of the field types that can be evaluated against the field you dragged from the **Rule Palette**.

Restrictions by Rule for a Transition

You can associate a rule with a transition that determines whether the transition will be available to users. If the rule evaluates to true, and if no other restrictions on the transition apply, the transition will appear on the form for the source state.

> **Restriction:**
> - You cannot use a rule to restrict an outgoing transition from a decision, because these transitions have their own rules associated with them. For more information, see About Decisions [page 342].
>
> - You cannot use a rule to restrict an outgoing transition from the Submit state.

To restrict a transition using a rule:

1. Open the workflow containing the transition.

2. Select the transition.

3. In the transition Property Editor, click the **Restrict By Rule** tab. See Restrict By Rule Tab of the Transition Property Editor [page 166] for details.

Use Cases

The following list describes some use cases for restricting transitions by rules.

- **Item aging:** The **Expedite** transition should be available only when an item is at least ten days old.

- **Priority service:** The **Calculate Discount** transition should be available only for priority service customers.

- **Item escalation:** The **Escalate** transition should be available only for a high severity item that was submitted by a new customer.

Working with Rules

This section includes the following topics:

- Creating a Rule [page 353]

- Modifying Rule Properties [page 354]

- Creating Expressions for Rules [page 354]

Creating a Rule

Note: The first five steps of this procedure describe the various ways you can add a rule to an application workflow.

To create a rule:

1. To add a rule directly from a decision:

   a. Click the decision in the application workflow.

   b. Click the **Rules** tab in the decision Property Editor.

   c. Click a cell in the **Rule** column.

   d. Select **New rule**.

2. To add a rule directly from a transition:

   a. Click the transition in the application workflow.

   b. Click the **Restrict by Rule** tab in the transition Property Editor.

   c. Click **New**.

3. To add a rule from App Explorer:

   a. Right-click the **Rules** heading in App Explorer.

   b. Select **Add New Rule**.

4. To add a rule from the rule editor:

   a. Right-click in the rules list on the **Rules** tab.

   b. Select **Add New Rule**.

5. To add a rule from the Ribbon:

   a. Click **Element** in the **New** section of the Ribbon.

   b. Select **Rule**.

6. Now that the new rule has been added, build an expression that the workflow will use to evaluate the rule. For instructions, see Creating Expressions for Rules [page 354].

7. On the **General** tab of the rule Property Editor, type a name and optional description for the rule. Select the **Validate for use in a report filter** check box to ensure that the rule can be used as a report filter after you deploy the process app. This option is selected by default for all rules that you create; clear the check box if you intend to create rules that will not be used as report filters.

Modifying Rule Properties

To modify rule properties:

- Select the rule in App Explorer (or click the **Rules** heading, and then double-click the rule in the list).

   The Property Editor below the rule editor lets you view and edit the name and description of the selected rule.

   > **Tip:** If the Property Editor is not visible, select **Property Editor** in the **Common Views** area on the **Home** tab of the Ribbon.

Creating Expressions for Rules

A rule consists of one or more field expressions that are joined by logical operators. A field expression consists of a field, followed by a comparison operator, followed by a literal value, another field, or an application variable.

For example, to create a rule that evaluates whether the value of a *Numeric* field is equal to or more than 500 but less than 1000, create the following expression:

Amount >= 500

AND

Amount < 1000

The expression contains three panes:

- The left pane contains the field you want to evaluate.

- The middle pane contains the available comparison operators for the field type.

- The right pane contains a field value editor that lets you specify the value, field value, or application variable value that will be compared to the field.

For example:
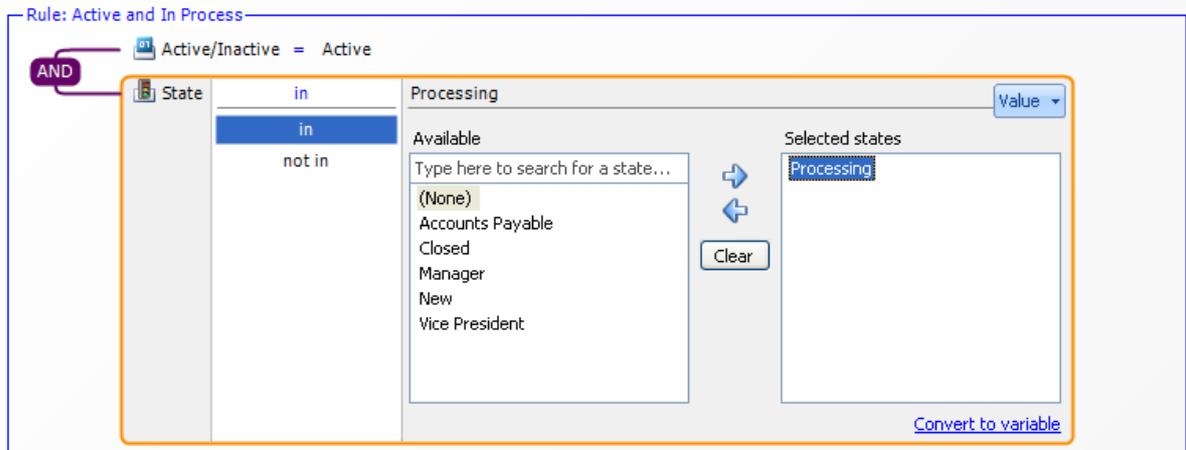
To create expressions for rules:

1. Drag a field or logical operator from the **Rule Palette** and drop it in the **Rule: *name*** block.

2. If you added a logical operator, drag a field from the **Rule Palette** onto each branch of the operator. For a description of logical operators, see Rule Operators [page 359].

3. Comparison operators for most field types can be changed. Optionally, click the field and select another comparison operator from the popup window that opens. For a description of comparison operators, see Rule Operators [page 359].

4. Do one of the following to complete the expression.

   • If you want to evaluate the value of the field, select **Value** from the drop-down list at the top right corner of the rule, and then select or type the value or values. For example, if you are working with an *Item Type* field, the types that are available as a selection for the field are displayed in the menu that opens. You can select a type from the list.

   • If you want to compare the value of this field to the value of another field, select **Field** from the drop-down list at the top-right corner of the rule, and then select the field.

      Note: Only compatible field types are available. For a list, see Field-to-Field Comparisons [page 358].

   • If you want to compare the value of this field to the value of an application variable, select **Variable** from the drop-down list on the top right corner of the rule panel, and then select the variable.

      Note: Only those application variables that are compatible with the field are available.

5. To combine expressions and establish the order in which they are evaluated, drag an AND or OR logical operator from the **Rule Palette** onto the expression. This inserts the logical operator in the place where the expression was and moves the expression to the first child of the new operator. For more information about using the drag-

and-drop operation in various scenarios, see Drag-and-Drop Behavior [page 398]. For examples of rule logic, see Rule Logic Examples [page 363].

> **Tip:** You can drag a field onto another field. This creates a new comparison expression that is "AND'd" with the existing one.

You can view a string that represents the rule in the **Rule summary** block.

Field Options and Values

The type of field that you use in an expression determines the options and values that are available to you. The following list describes common features of fields.

- **Current user** – All *User*, *Multi-User*, and *Multi-Group* fields have a special "Current user" field value. The expression evaluates to "true" when the user appears in that field. For example, add the *Submitter* system field from the **Fields** section of the **Rule Palette**. Select the **in** operator, and then select the **Current user** check box.

  This expression will evaluate as true only if the "Current User" submitted the issue. In *Multi-Group* fields, the expression will evaluate to "true" if the current user is a member of one or more groups associated with the field.

- **None** – With certain field types, you can select the **None** check box. This selection evaluates all items meeting the rest of the rule criteria that do not have a value for the selected field.

- **Clear** – Most fields have a **Clear** button. Use this button to clear selected values or to return to the default value of **Undefined**.

- **Value, Field, Variable** – All fields have a drop-down list containing these selections. You can compare the field value to a specific value, the value of another field, or the value of an application variable. Fields and variables that are not supported in a given expression are not included as selections. (For a list of supported field types, see Field-to-Field Comparisons [page 358].)

> **Note:** See Rule Operators [page 359] for a list of restrictions on comparisons used in expressions.

The following table lists field types and describes the options that are available to them when **Value** is selected in the drop-down list.

Field Type	Options
Selection/ User	If you can select values for a field, you can search for specific values or select multiple values to add to your condition.
	If you select a selection or user field, the values that are available as selections for the field are available in the **Available values** column. Use the right arrow button to move them to the **Selected values** column.
	In some cases, a text box contains **Type here to search for a value**. You can type a few letters in this box to filter the list of values. When you do this, the only values that are displayed are those that contain the letters you typed.
Date/ Time	To specify a date relative to the current date, select an option, such as **Start of Last Week** on the **Special value** tab.
	If you want to specify a certain number of days before or after the current date, select the **Now** checkbox on the **Special value** tab, select **+** or **-**, and type or select the number of days. For example, if you want to evaluate whether a Help Desk incident was submitted in the last 30 days, select the ≤ operator, select the **Now** check box, select **−**, and then type 30 in the **days** combo box. The expression becomes **Date ≤ 30 days ago**.
	To specify an exact date and time, click a date and select a time on the **Exact value** tab, and then click **Accept**.
Non-Selection	If a field does not let you select a specific value, type a value in the text box. For example, if you select a *Text* field, such as the *Description* field, type characters to search for. *Date/Time*, *Numeric*, and *Summation* fields also let you specify a value.
	In some cases, the text box contains **Enter a wildcard pattern**. For information about using wildcards, see Wildcards [page 396].
Binary	If you select a *Binary* field, such as the *Active/Inactive* field, select one value for your rule criteria.
Trinary	If you select a *Trinary* field, select one, two, or three values for your rule criteria.
File	If you select a *File* field, you can specify the **is Empty** value (find items in which a file has not been provided by a user) or the **is Not empty** value (find items in which a file has been provided).
URL	If you select a *URL* field, you can specify the **is Empty** value (find items in which a URL has not been provided by a user) or the **is Not empty** value (find items in which a URL has been provided).

Field-to-Field Comparisons

The following table shows the SBM field types that can be compared to each other in a rule expression.

> **Note:** See Creating Expressions for Rules [page 354] for more information about field-to-field comparisons.

Field Type	Field Subtype	Compared Field Type
Binary/ Trinary	Binary, Trinary	Binary/Trinary with same subtype The comparison is done against the positions of the selections in the field control, not against their textual values. For example, if **First value** in one binary field is "No," and **First value** in the compared binary field is "False," the rule will evaluate to true if **No** and **False** are selected.
Date/Time	Date and Time, Date Only, Elapsed Time, Time of Day	Date/Time with same subtype
File	N/A	File The comparison is done against the actual filename and size, not against the file content.
Folder	N/A	Not supported
Multi-Group	N/A	Multi-Group
Multi-Relational	N/A	Not supported
Multi-Selection	N/A	Not supported for field-to-field comparison; supported for field-to-value and field-to-variable comparisons.
Multi-User	N/A	Multi-User with same selection mode on **Options** tab of field Property Editor (**Individual users** or **Groups & users**)
Numeric	Fixed, Integer, Float	Numeric or Summation

Field Type	Field Subtype	Compared Field Type
Project	N/A	Not supported
Single Relational	N/A	Not supported
Single Selection	N/A	Not supported for field-to-field comparison; supported for field-to-value and field-to-variable comparisons.
State	N/A	Not supported for field-to-field comparison; supported for field-to-value and field-to-variable comparisons.
Sub-Relational	N/A	Selected **Sub-field** on **Options** tab of field Property Editor
Summation	N/A	Summation or any Numeric field type
Text	Fixed Length, Memo, Journal	Text with any subtype
URL	N/A	URL The comparison is done against the actual URL, not against the URL content.
User	N/A	User, Multi-User, Multi-Group

Rule Operators

Logical operators let you set a sequence for evaluating conditions. *Comparison* operators let you select a function to compare the value of a field to a specific value, the value of another field, or the value of an application variable.

> **Restriction:**
> - In field-to-field comparisons, you can only select fields that are suitable to a given field type. For a list of supported field types, see Field-to-Field Comparisons [page 358].
> - You can only select comparison operators that are suitable for a given field type. For example, you can select a **>** operator for a *Numeric* field, but not for a *Multi-User* field. For a list of supported operators, see Field Comparison Operators [page 361].

Logical Operators

The following table describes the AND and OR logical operators.

Operator	Description
AND	Drag **AND** from the **Logical Operators** section of the **Rule Palette** onto the expression. A rule using this operator evaluates as "true" when all expressions it joins evaluate to "true." For example, the condition "Amount > 100 " AND "Amount <= 500" evaluates whether the *Amount* field value is between 101 and 500.
OR	Drag **OR** from the **Logical Operators** section in the **Rule Palette** onto the condition. A rule using this operator evaluates as "true" when any expression it joins evaluates to "true." For example, the conditions "Owner in Joe Manager" OR "State in New" evaluates whether the item is in the "New" state *or* is owned by Joe Manager.

Comparison Operators

The following table describes the comparison operators.

Operator	Description
= (equal to)	An expression using this operator will evaluate to "true" if the value of the field on the left is the same as the value on the right.
> (greater than)	An expression using this operator will evaluate to "true" if the value of the field on the left is greater than the value on the right.
< (less than)	An expression using this operator will evaluate to "true" if the value of the field on the left is less than the value on the right.
>= (greater than or equal to)	An expression using this operator will evaluate to "true" if the value of the field on the left is greater than or equal to the value on the right.
<= (less than or equal to)	An expression using this operator will evaluate to "true" if the value of the field on the left is less than or equal to the value on the right.
<> (not equal to)	An expression using this operator will evaluate to "true" if the value of the field on the left is not the same as the value on the left.
in	An expression using this operator will evaluate to "true" if the field on the left contains one or more of the values specified on the right.

Operator	Description
not in	An expression using this operator will evaluate to "true" if the field on the left does not contain any of the values specified on the right.
contains all	An expression using this operator will evaluate to "true" if the field on the left contains all of the values specified on the right.
contains any	An expression using this operator will evaluate to "true" if the field on the left contains any of the values specified on the right.
does not contain all	An expression using this operator will evaluate to "true" if the field on the left does not contain all of the values specified on the right.
does not contain any	An expression using this operator will evaluate to "true" if the *Multi-Selection* or *Multi-User* field on the left does not contain any of the values specified on the right.
contains	An expression using this operator will evaluate to "true" if the field contains the specified text.
like	An expression using this operator will evaluate to "true" if the first field on the left matches the specified wildcard expression.
not contains	An expression using this operator will evaluate to "true" if the *Text* field on the left does not contain the specified text.
not like	An expression using this operator will evaluate to "true" if the field on the left does not match the specified wildcard expression.
like (zero-filled)	An expression using this operator will evaluate to "true" if the field on the left matches the specified wildcard expression, including leading zeroes. For example, if there are leading zeroes on an Item ID number, they do not have to be specified for the expression to evaluate as "true."
is empty	An expression using this operator will evaluate to "true" if the field value is empty (in which values have not been entered or not provided by a user).
is not empty	An expression using this operator will evaluate to "true" if the field value is not empty (in which values have been provided).

Field Comparison Operators

The following table lists the comparison operators that are available for field-to-value comparisons and field-to-field comparisons in rule expressions.

Field Type	Field-to-Value Operators	Field-to-Field Operators
Binary/ Trinary	Binary: =; Trinary: in, not in	=, <>
Date/Time	=, <>, >, >=, <, <=	Same as "Field-to-Value Operators"
File	contains, not contains, is empty, is not empty	=, <>
Multi-Group	contains any, does not contain any	Same as "Field-to-Value Operators"
Multi-Selection	contains all, does not contain all, contains any, does not contain any	Not supported
Multi-User	contains any, does not contain any	Same as "Field-to-Value Operators"
Numeric	=, <>, >, >=, <, <=	Same as "Field-to-Value Operators"
Single Selection	in, not in	Not supported
State	in, not in	Not supported
Sub-Relational	All operators applicable to the referenced *Sub-Relational* field type.	Same as "Field-to-Value Operators"
Summation	=, <>, >, >=, <, <=	Same as "Field-to-Value Operators"
Text	contains, not contains, like, not like	=, <>
URL	contains, not contains, is empty, is not empty	=, <>
User	in, not in	Same as "Field-to-Value Operators"

Wildcards

You can use wildcard characters expressions for rules if **Enter a wildcard pattern** is in the text box.

The following guidelines apply to wildcard characters:

- Asterisks (*) and percent signs (%) serve as wildcard characters. A wildcard character matches zero or more consecutive characters.

- Underscores (_) match a single character.

- Wildcard characters are automatically applied to the beginning and end of your criteria. Items containing all of your criteria are returned.

- You can override automatic wildcards by including at least one wildcard in your criteria. Wildcard characters can be placed anywhere in the box and could return different results depending on the location of the wildcard character. For example, *ed returns all items ending in ed, while ed* returns all items beginning with ed.

- To find all items, type a wildcard character (* or %) or leave the box empty.

- Leading and trailing spaces are removed.

Rule Logic Examples

The following examples illustrate how rules are constructed.

Note: AND and OR operators always have at least two child nodes. The nodes are either fields or **Drag a field here** placeholders.

(1 or ((2 or 3) and (4 or 5)) or (6 and 7)) and 8

1 and 2 and 3 and 4 and 5 and (6 or 7)

(1 and 2) or 3

Rule: Untitled 3

Active/Inactive = Inactive

AND ②

Owner in (None)

OR

State not in New

1 and (2 or 3)

Rule: Untitled 4

Active/Inactive = Active

AND

Owner in (Current User)

OR

State in New

(1 or 2 or 3) and 4

Rule: Untitled 5

Active/Inactive = Active

OR

State not in (None)

AND

Owner in (Current User)

Submitter not in (Current User)

(1 and (2 or 3)) or 4

Rule: Untitled 6

Active/Inactive = Inactive

AND

State in (None)

OR

Title contains Field

OR

Owner in (None)
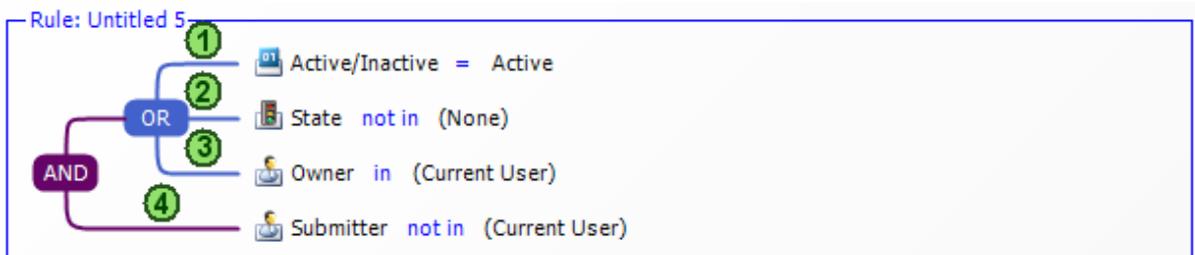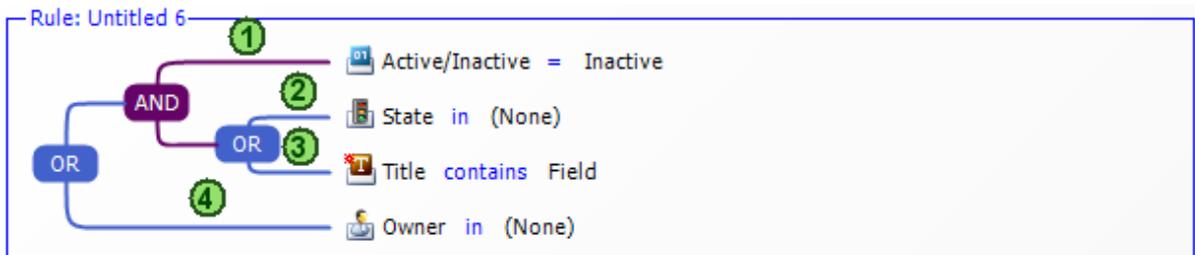
Drag-and-Drop Behavior

The following points describe how to use the drag-and-drop operation to add fields and operators to a rule expression.

Drag-and-Drop Operation	Behavior
AND to AND	To add a new expression to an AND group, drag a field from the palette onto another field in the group. The new field is added above the field onto which you dragged.

Drag-and-Drop Operation	Behavior
OR to AND	To add an OR group to an AND group, drag the **OR** operator to one of the lines extending from the AND group. If there is a field on the top line, the OR group is added to the bottom line of the AND group. If no field is on the top line, the OR group is added to the top line of the AND group.
OR to OR	To add a new expression to an OR group, drag a field from the palette onto another field in the group. The new field is added above the field onto which you dragged.
AND to OR	To add an AND group to an OR group, drag the **AND** operator to one of the lines extending from the OR group. If there is a field on the top line, the AND group is added to the bottom line of the OR group. If no field is on the top line, the AND group is added to the top line of the OR group.
Existing Field	To move an existing field within an AND or OR group, drag it to another field. The existing field is placed above the field onto which you dragged it.

Rules Settings

The following table describes the information and options that are displayed on the **General** tab of the rule Property Editor.

Element	Description
Name	The name of the rule that was provided when the rule was created.
Description	An optional description of the rule.
Validate for use in a report filter	Ensures that the rule can be used as a report filter. This option is selected by default for all new rules that you create; clear the check box if you intend to create rules that will not be used as report filters.

A rule that contains an application variable or field-to-field comparison is not available for use as a report filter at runtime. Use the **Validate for use in a report filter** option depending on your intent for the rule:

- Select this check box to receive warnings while you work with the rule and validation errors on deployment (if the rule should continue to be coupled with and used as a report filter). This ensures that any change to the rule will still allow it to be used as a report filter.

- Clear the check box if you intend to use the rule for purposes other than as a report filter. For example, clear the check box if you want to use an application variable or field-to-field comparison in the rule. As a result, for any report filter in Work Center that originated from a rule in SBM Composer, the following occurs:

- After you clear the check box, change the rule, and then deploy the process app, the report filter is not deleted from the reports that use it; instead, the report filter is simply decoupled from the rule, and henceforth managed exclusively in Work Center. The filter is now editable in Work Center and set to **Public** access.

- After you clear the check box, any further changes to the rule in SBM Composer will have no impact on the filter that exists at runtime, and the report filter and its conditions are now editable in Work Center. Changes made to the report filter in Work Center will have no effect on the rule in SBM Composer.

Chapter 16: Defining Application Variables

This section contains the following information:

About Application Variables

An application variable stores the value of a primary field. You can create an application variable explicitly, or convert a value in a rule expression to an application variable.

Restriction: You cannot create an application variable for *File* and *URL* fields.

Note: For information about rules, see Rule Overview [page 351].

Key Benefits

- Designers can assign a default value to an application variable once, and then reuse the application variable in multiple rules in the process app.

- Designers can change the default value of an application variable, and all rules that reference that variable will use the new default value.

- Designers can use the values in application variables to control the behavior of controls, fields, and widgets on custom forms. Form customization by way of JavaScript, form actions, form extensions, or custom controls can reference application variables to dynamically change the behavior of the form and its display.

- Administrators can override the value of an application variable for individual projects in SBM Application Administrator. This allows administrators to tailor workflows based on business needs for various projects. For example, items in one project may need to move to the **Review** state based on one date; items in another project may move to this state based on a different date. For more information, see the *SBM Application Administrator Guide* or online help.

Creating an Application Variable

To create an application variable:

1. Do one of the following:

   a. Right-click the left pane of the Application Variable Editor [page 368] and then select **Add New Field Application Variable**.

   b. In the Rule Editor [page 352], click the field for which you want to create the variable, and then click **Convert to variable**.

2. Complete the fields in the right pane of the Application Variable Editor [page 368] to specify a default value for the variable.

3. Click in an empty area of the right pane to refresh the left pane of the application variable editor.

> **Note:** The application variables are ordered by value under headings in the left pane. For example, all application variables with an "Undefined" value are listed under the **(Undefined)** heading.

Application Variable Editor

Use the application variable editor to add application variables and specify the default values of new or existing application variables.

The left pane of the editor contains a list of the application variables and their current default values. The right pane of the editor is where you assign or modify the default values.

Element	Description
Name	The name of the application variable.
Description	A description of the application variable.
Settings	The field associated with the application variable and its data type. Click the **Edit field** link to navigate to the table editor and open the Property Editor for the field so you can change the field settings.
Value	The default value that will be stored in the application variable. If you want to change the value, type it in the box below the value or select **Empty string**. To change what you typed, click **Clear**.

Chapter 17: Creating Application References

This section contains the following information:

About References

You create references to make an association in your process app to a table, field, or other design element in an application contained in another process app. You cannot create a reference to such a design element unless you first create a reference to the application that contains the design element. This topic contains examples of typical ways in which you can use references.

Example: Using a Post Transition

A tester submits a ticket to engineering, requesting an enhancement to a feature. This ticket is assigned to an engineer, who starts making code changes. The engineer realizes that the enhancement requires documentation changes, so he posts it to the documentation department. A new ticket is created for documentation, and a writer is assigned to it.

In this example, one process app handles requests to the engineering department. The other process app handles requests to the documentation department. In the application workflow for the engineering process app, you use a **Post** transition to create a reference between the referencing process app (engineering) to the referenced process app (documentation).

Example: Using a Subtask Transition

A company has a new hire, and the IT department needs to prepare for the employee's first day of work. The preparation includes assigning a cubicle for the employee. The hiring manager submits an issue to the employee onboarding application, and the issue is assigned to the IT manager. The IT manager assigns the issue to a system administrator, who posts it to the facilities department to handle the cubicle assignment. After the cubicle is assigned, the issue is transitioned to the **Cubicle Assigned** state and reassigned to the system administrator.

In this example, one process app handles the overall employee onboarding process. Another process app handles the facility department processes. In the application workflow for the employee onboarding process app, you use a **Subtask** transition from a **Create Network Account** state to a **Cubicle Assigned** state. The **Subtask** transition creates a reference between the referencing process app (employee onboarding) to the referenced process app (facilities).

Example: Using a Relational Field

A customer calls Technical Support to report that a feature does not work as expected. The technical support representative submits an issue to a defect management application.

The engineering manager realizes that the solution is complex, and schedules it to be implemented during the next release. In the meantime, the technical support engineer submits an issue to a knowledge base application. This issue describes the problem and provides a temporary solution.

In this example, the issue defect management application and knowledge base applications are in separate process apps. A *Single-Relational* field called "Associated Issue" is on the submit form in the knowledge base application. This field references the Issues table in the defect management application. When the technical support engineer submits the knowledge base issue, he selects the customer-reported issue from a list of issues.

Example: Using the Global Process App

A customer support representative is working with Acme Corporation, who reported a problem with a product. The customer support representative submits an issue to the incident management application on behalf of Acme Corporation. On the transition form, the customer support representative selects **Acme Corporation** from the **Requestor** drop-down list.

In this example, the one process app handles the incident management process. The Global Process App contains a Company auxiliary table, which includes a *Company* field. You create a reference to the Global Process App by mapping the *Requestor* field in the incident management process app to the Company auxiliary table in the Global Process App.

> **Important:** You can create references to fields in Global Process App tables, but you cannot create references from tables in the Global Process App.

About Resolving References

This topic describes some use cases for resolving references.

> **Important:** When you *import* a set of process apps (by pointing to **Import and Export** from the **File** menu and then selecting **Import**), the process apps keep the same internal identifiers and the references will be preserved. When you *create* a set of process apps (using **New** from the **File** menu), the process apps get new internal identifiers and you must resolve the references explicitly.

Use Case: Two Existing Process Apps

You have an existing process app with a reference to a field in another existing process app. The field in the referenced process app was deleted, so the reference becomes invalid. When you try to deploy the referencing process app, you get an error message about the missing field.

When you double-click the error message, the Property Editor for the missing field is displayed. In the Property Editor, you can select another field to use as a reference.

Use Case: Ignoring Unresolved References

You create a new process app that meant to work with another process app, so it contains unresolved references.

You do not need the other process app now, so you decide to ignore the unresolved references. To do this, right-click the unresolved reference in App Explorer and select **Ignore Reference**. The reference and any design elements in the referenced application are marked through, and when you validate the process app, no unresolved reference warnings are displayed in the Validation Results.

Working with References

This section contains the following topics:

- Defining a Reference [page 371]

- Viewing References [page 372]

- Refreshing References [page 372]

- Removing References [page 372]

- Examining References [page 373]

- Modifying a Referenced Application [page 373]

- Resolving References [page 373]

- Changing a Reference Resolution [page 374]

- Ignoring Unresolved References [page 374]

Defining a Reference

To define a reference:

1. In App Explorer, right-click the **References** heading or the name of an existing reference, and select **Add Application Reference**. The **Add Application Reference** dialog box opens.

2. Select **Local Cache** or **Repository** to indicate whether the process app you want to reference is stored in your computer's file system or checked in to the repository from SBM Composer.

3. Select the process app that contains the application that you want to reference, and then click **Add**.

   Note: To define a reference, the referenced process app must have been checked in. For example, if you want to create a new process app and create a reference to the Global Process App, you must open the Global Process App in SBM Composer and then check it back in before you can see it in the repository and reference it. You must also have the "View" privilege (set in Application Repository) for the process app you want to reference if the process app is in the repository, not in the Local Cache.

CAUTION:

⚠️ If you add a reference to an application in the open process app, the application becomes inaccessible.

4. Create references to the referenced application. For example:

   - For a **Post** transition, on the **Post Options** tab of the transition Property Editor, select the referenced application in the **Post application** list. If there are multiple tables in the referenced application, select the table you want to reference in the **Post table** list.

   - For a *Multi-Relational* or *Single Relational* field, on the **Options** tab of the field Property Editor, select the referenced application in the **Application** list. If there are multiple tables in the referenced application, select the table you want to reference in the **Table** list.

Viewing References

To view a list of application references:

1. Click **+** next to the **References** heading to expand the list in App Explorer.

2. If you want to see details about a referenced application, double-click the application in the list to open the application editor.

Refreshing References

If a design element in a referenced application changes after the reference is defined, you need to "refresh" the reference definition in your process app. Subsequent validation shows whether the changed design element will still serve the intended purpose in your process app. Refreshing a reference has an effect similar to that of the **Get Latest of All** command.

To refresh a single reference definition:

   - In App Explorer, under the **References** heading, right-click the application name and select **Refresh**.

To refresh all reference definitions:

   - In App Explorer, right-click the **References** heading, and select **Refresh All References**.

   📝 **Note:** You can use the **Refresh All References** command if there is at least one reference to a process app in the repository.

Removing References

To remove an application reference:

1. In App Explorer, under the **References** heading, right-click the application name.

2. Select **Remove**.

The application reference is removed if it is not used by any design element references. If it is used by design element references, a dialog box opens that lists all affected design elements. Eliminate all such design element references, and then remove the application reference.

Examining References

To examine a referenced application:

In App Explorer, under the **References** heading, select the referenced application or navigate to a design element in the application as described in Using the App Explorer [page 31]. Design elements in a referenced application are read-only. If you need to make changes, click the yellow alert bar at the top of the applicable editor to open the related process app.

Modifying a Referenced Application

To modify a referenced application:

1. In App Explorer, right-click the **References** heading, and then select **Open Related Process App**.

2. Respond to prompts regarding saving changes to the open process app and checking in design elements.

Resolving References

This topic explains how to resolve an unresolved reference to a design element in another process app. When you validate a process app, a list of any unresolved references is displayed in the Validation Results at the bottom of the SBM Composer window.

> **Tip:** In many cases, the process app containing a referenced application is missing from the repository (that is, the referenced application existed when the reference was created, but was later deleted). After you resolve the application reference, the references to other design elements in the referenced application are usually resolved automatically.

> **Note:** You might not need to resolve references. See Ignoring Unresolved References [page 374] for more information.

Unresolved references are displayed with a red X. The following illustration shows an unresolved application reference as it appears in App Explorer.

To resolve an unresolved reference:

1. After validating the process app, double-click the message in the Validation Results or right-click the message and select **Go to Location**. SBM Composer takes you to the location where you can resolve the reference. For example, for an unresolved application reference, SBM Composer takes you to the application editor. For an

unresolved table reference, SBM Composer takes you to the Property Editor for the field in the referenced table.

2. If this is an unresolved reference to an application, perform the following steps:

   a. Right-click the reference in App Explorer and select **Resolve**, or click the message at the top of the application editor. The **Resolve Reference to Application** dialog box opens.

   b. Complete the dialog box as described in Resolve Reference to Application Dialog Box [page 376].

   c. Validate the process app again.

3. Resolve any remaining resolved references as appropriate. For example, if a referenced field is missing from a table, either restore the field in the table, or in the Property Editor for the field, select another application and field that resolves the reference.

Changing a Reference Resolution

You can select a resolved application reference and change it to another application. When you do this, all element references (such as relational fields, **Post** transition options, and **Subtask** transition options) are automatically resolved.

To change a reference:

1. Right-click the reference under the **References** heading in App Explorer—**All Items** or App Explorer—**References**, and select **Change**. The **Change Reference** dialog box opens.

2. Complete the dialog box as described in Change Reference Dialog Box [page 375].

Ignoring Unresolved References

You might not need to resolve a reference to something in another process app. For example:

- You do not need the other process app.

- Your process app runs correctly, even though it is missing some unneeded functionality.

- You want to see if the process app is suitable before you make the reference.

To ignore an unresolved reference:

1. Right-click the reference in App Explorer and select **Ignore Reference**. The reference is marked through and grayed out.

2. All references to design elements in the referenced application should also be ignored. Click **Validate** and make sure that all messages have cleared from the Validation Results.

Reference Settings

The following sections provide descriptions of application reference settings. Sections are organized by property editors.

Add Application Reference Dialog Box

Use this dialog box to select an application that contains the table, field, or other design element with which you need to make an association.

Element	Description
Look in	Select whether the application is in a process app that is stored in the Local Cache or in a process app that was checked in to the repository from SBM Composer. Click the column headings to sort the process apps by name, application, category, updater, and date updated. A **+** next to a process app name means that it contains applications. (You cannot define references based on orchestrations.) Click **+** to list its applications. Then select the application that contains the design element you want to use and click **Add** (or double-click the application name).

Change Reference Dialog Box

Use this dialog box to change the application that contains the table, field, or other design element with which you need to make an association.

Element	Description
Look in	Select whether the application is in a process app that is stored in the Local Cache or in a process app that was checked in to the repository from SBM Composer. Click the column headings to sort the process apps by name, application, category, updater, and date updated. A **+** next to a process app name means that it contains applications. (You cannot define references based on orchestrations.) Applications that can be resolved automatically are displayed in bold. Click **+** to list its applications. Then select the application that contains the design element that you want to use, and click **Add** (or double-click the application name).

Referenced Applications Dialog Box

This dialog box lists any applications to which the process app that you are deploying has references.

You receive warning messages when you try to deploy a process app with a reference to a missing application. You can still deploy the process app, but references are lost if they are not resolved before a process app is deployed.

Note: Applications that are unresolved references are not listed in this dialog box.

Resolve Reference to Application Dialog Box

Use this dialog box to locate the application that is likely to resolve the unresolved reference.

CAUTION:

Save any changes you make to the process app before you resolve an application reference, because the only way you can "undo" the application reference resolution is to close the process app without saving changes.

Note: There are situations in which you do not need to resolve references. For more information, see Resolving References [page 373].

Element	Description
Look In	Select whether the application is in a process app that is stored in the Local Cache or in a process app that was checked in to the repository from SBM Composer.
	Click the column headings to sort the process apps by name, application, category, updater, and date updated.
	A **+** next to a process app name means that it contains applications. (You cannot define references based on orchestrations.)
	Click **+** to list its applications. Then select the application that contains the design element that you want to use, and click **Resolve** (or double-click the application name). Compatible applications (that is, applications that have the same design number) are shown in bold. For information about design numbers, see Managing Internal Identity and Design Numbers [page 62].

Chapter 18: Understanding Inheritance and Overrides

This section contains the following information:

- Inheritance Rules [page 377]

- Overrides Overview [page 378]

- Finding Overrides in SBM Composer [page 380]

- Field Overrides Window [page 382]

Inheritance Rules

Each application can have multiple workflows, which may or may not be related through inheritance.

When a project is assigned to a workflow, the elements of that workflow (such as fields, forms, states, and transitions) are automatically inherited by the project. These elements and their properties are then inherited throughout the project hierarchy.

The following figure shows the basic inheritance path. In this example, workflow settings are inherited by the project assigned to the workflow and to its sub-project.
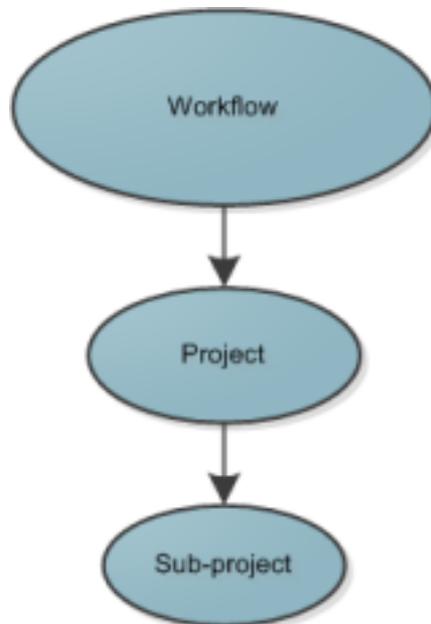
You can use inheritance to quickly establish workflow and project hierarchies. When you assign a workflow to a parent project in SBM Application Administrator, all sub-projects of the parent project automatically use the same workflow.

The following example shows the inheritance path for two sibling workflows in an application.
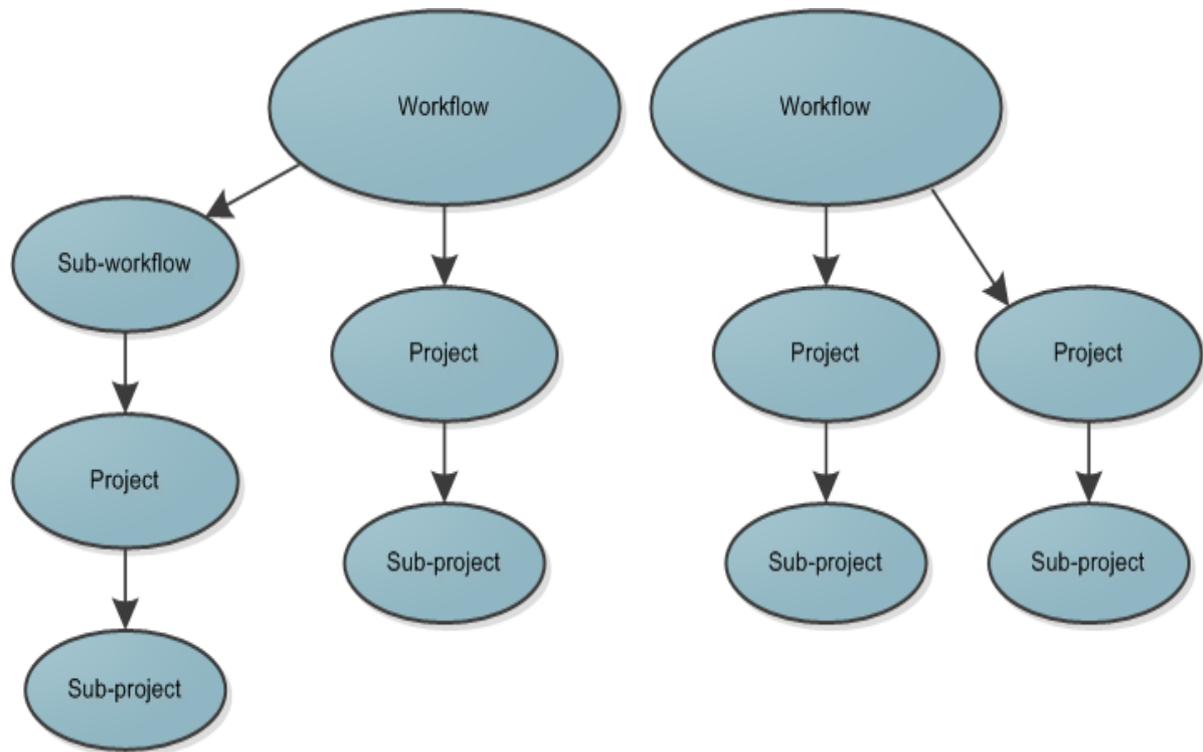
Overrides Overview

Use overrides to tailor your overall process by modifying inherited elements in workflows and projects.

For example:

- A workflow may include a transition that is not needed in all sub-workflows. You can edit sub-workflows and disable the transition as needed.

- A workflow handles requests that come into the Support organization, but different support managers need to be set as default values for the **Support Lead** field. In SBM Composer, enable overrides for the workflow. After you deploy the process app, you can specify **Mark** (the Support manager) the default value for the **Support Lead** field, but override the default value for different projects assigned to the workflow.

- You want to make sure that users always provide a value for a **Customer Name** field in a project used to report customer issues. You can set the field as required work the workflow, and then override this setting for a sub-project that employees use to report internal issues.

- In an application that handles time-off requests, one sub-workflow is used for vacation time and another is used for sick time. The vacation sub-workflow needs a **Vacation Time Used** field, and the sick time sub-workflow needs a **Sick Time Used** field. You can specify different forms for the states and transitions in each of the two sub-workflows to present the applicable fields to users.

Workflow overrides are applied in SBM Composer; all project and a few workflow overrides are applied in SBM Application Administrator.

Changes you make in a workflow are not reflected for end users if the project has settings that override the workflow's settings. A state or transition override at the project level has the final authority regarding what users see. For example, if you set a default value for a selection field in a workflow, that value might not be available to users if a project or transition in a project has its own override for that field.

To ease application maintenance, always define workflow settings at the highest level possible, and then override them for sub-workflows and projects only when necessary.

Applying Workflow Overrides

You can override the following workflow attributes in SBM Composer:

- Field properties

- Field privileges for workflows, states, and transitions

- Default form selection for workflows and for individual states and transitions

- Transition ordering for states

- Time Capture settings

For details, refer to Finding Overrides in SBM Composer [page 380].

> **Note:** A duplicated workflow is considered a sibling workflow and inherits the fields, states, transitions, and forms in the original workflow.

Applying Project Overrides

You can perform the following override tasks in SBM Application Administrator.

Task Type	Tasks
Workflow	Set and override default values for *User*, *Multi-User*, and *Multi-Group* fields. The **Override field properties** check box must be selected for each field on the **Field Overrides** tab of the application workflow editor.

Task Type	Tasks
Project	Override the following project attributes: • Default state forms • Default transition forms • Default project, state, and transition field ordering when quick forms are used • Field attributes for projects and transition fields • Display options for fields in projects and for transition fields (except *Binary/Trinary*, *Date/Time*, *Numeric*, and *Text* fields) • Dependent field selections for independent *Single Selection* fields • Dependent field selections for *User* fields • Time Capture settings
State	Override the following state attributes in projects: • Inherited state form • Default field ordering quick forms are used • Transition button ordering
Transition	Override the following transition attributes in projects: • Transition authentication settings for transitions • Inherited transition form • Project settings for Post, Subtask, and Copy transitions • Default field ordering for transitions when quick forms are used • Calculations for *Date/Time* and *Numeric* fields • Transition button ordering • Default values for transition fields

Finding Overrides in SBM Composer

Use the following information to determine where overrides are applied for various design elements.

Field Overrides

You can locate where fields have been overriden for a workflow or transition:

1. Do one of the following:

   - Select the primary table.

   - Open the workflow or transition Property Editor, and then select the **Field Overrides** tab.

2. Right-click the field that you want to find overrides for, and then select **Overrides**.

3. Use the Field Overrides window to view the workflows and transitions that contain an override for the selected field. For details, refer to Field Overrides Window [page 382].

You can also view field overrides on the **Field Overrides** tab for workflows and transitions.

Field overrides for workflows and transitions are represented by:

- **Bold** type means the **Override field properties for *field-name*** check box is selected, regardless of whether any overrides are applied to the field.

- *Italics* type means the status of a value for a selection field was overridden at the workflow level.

- ***Bold italics*** type means that both of these override conditions exist for the field.

Form Overrides

Use the icons on the workflow editor to determine if a state or transition uses a custom form or a quick form. A square on any icon indicates that the form is inherited. For example:
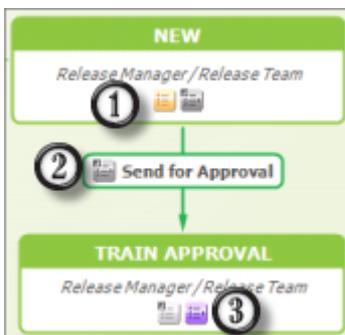
1. A colored icon indicates that a state or transition is using a custom form.

2. A gray icon indicates that a "quick form" is being used.

3. The second icon for states represents print forms. A colored icon indicates that a custom form is used.

You can also see inheritance information on the Forms tab for individual sub-workflows, states, and transitions.

Field Privilege Overrides

You can locate where a field belongs to a privilege section that has been overriden for a workflow, state, or transition. Overrides include when the field order has changed in a privilege section, or when fields have been moved to different privilege sections.

1. Do one of the following:

   - Select the primary table.

   - Open the workflow, state, or transition Property Editor, and then select the **Field Privileges** tab.

2. Right-click the field that you want to find privilege overrides for, and then select **Privileges**.

3. Use the Field Privileges window to view the workflows, states, and transitions that contain a privilege override. For details, refer to Field Privileges Window [page 383].

You can also view field privilege overrides on the **Field Privileges** tab for workflows, states, and transitions. If the **Override field privileges** check box is selected for any of these design elements, the field privileges have likely been overridden. Scroll through the list to see field placement in privilege sections.

Time Capture Settings

You can view Time Capture overrides on the **General** tab for sub-workflows. If time capture is set to on or off for a sub-workflow, the settings have been overridden.

Transition Ordering for States

You can view transition ordering on the **Transitions** tab for specific states. If the **Override transition ordering** check box is selected, the transition ordering has likely been overridden. Scroll through the list to see the transition order for the state.

Field Overrides Window

Use the Field Overrides window to locate where a selected field has been overriden. To view overrides for multiple fields, you can open multiple instances of the Field Overrides window at a time.

Element	Description
Field	The selected field. To view the overrides for another field, select the field from the drop-down list.
Mode	Enables you to switch between field overrides or privilege overrides for the selected field.
Options	These options hide various levels of information (non-overridden items, affected items). By default, these options are all selected. To retain your option settings, click the **Save Options** button.

Element	Description
Overrides Area	This area displays where the selected field is used in workflows and transitions. • **Bold** type means the workflow or transition contains an override for the selected field. • *Italics* type means the workflow or transition has inherited the field override. • Grayed out type means the workflow or transition does not contain an override for the selected field, and has not inherited a field override either.
Go to property editor	To open the field in the applicable property editor, select the field, workflow, or transition and then click **Go to property editor**. Alternatively, you can right-click an item and select **Go to property editor**, or double-click an item to open the field in the property editor.

Field Privileges Window

Use the Field Privileges window to locate where a selected field belongs to a privilege section that has been overriden. To view overrides for multiple fields, you can open multiple instances of the Field Privileges window at a time.

Element	Description
Field	The selected field. To view the privilege overrides for another field, select the field from the drop-down list.
Mode	Enables you to switch between field overrides or privilege overrides for the selected field.
Options	These options hide various levels of information (non-overridden items, affected items). By default, these options are all selected. To retain your option settings, click the **Save Options** button.

Element	Description
Privilege Override Area	This area groups the workflows, states, and transitions by privilege section for the selected field. • **Bold** type means the workflow, state, or transition has an overriden field order or privilege section. • *Italics* type means the workflow or transition has inherited the overriden field order or privilege section. • Grayed out type means the workflow or transition does not have an overriden field order or privilege section, and has not inherited an overriden field order or privilege section either.
Go to property editor	To open the field in the applicable property editor, select the field, workflow, state, or transition and then click **Go to property editor**. Alternatively, you can double-click the item in the privilege override area to open it in the property editor.

Chapter 19: Designing Application Reports

The following section includes the following information:

- About Application Reports [page 385]

- Working with Application Reports [page 386]

- Application Report Settings [page 399]

About Application Reports

Application reports are *listing* reports that you define in SBM Composer. Listing reports return textual lists of primary or auxiliary items based on the display, sorting, and search options you select.

Application reports contain all of the settings that are part of a regular listing report. They also contain search criteria and fields that use the "Query at runtime" condition if potential values, such as user information, is not known to SBM Composer. This information is obtained from the users in response to prompts when they run reports.

> **Note:** For more information about the "Query at runtime" condition, see Basic Conditions [page 394].

Project names are based on application workflow names. Application reports are created for base projects from the report definition defined in SBM Composer when the process app is deployed.

Application reports are a starting point for users to pre-configure regular reports. Users can generate any number of regular reports from a single application report, and can modify and save them as regular reports, but cannot save them as application reports. To change application reports, you must use SBM Composer and redeploy the changes.

The privilege level for a role determines whether a user can see an application report and is set in SBM Composer when the application is created. The privilege level is a combination of the "Run reports" privileges set for the table, the **Privilege category** set in the definition of the application report, and table and workflow privileges.

> **Note:** Application reports are automatically created, updated, and deleted as a result of deploying a process app, and are not promoted like regular reports.

Use Case

You create an application and need an application report that a user (for example, an administrator in this case) can use to create various regular reports that pertain to the application. Your application report includes an *Owner* field. You use the "Query at runtime" condition on this field. This means that the administrator is prompted to specify an owner for the report when he or she runs the report. The administrator then saves the report as a regular report, which other users can run.

Working with Application Reports

This section contains the following topics:

- Creating an Application Report [page 386]

- Selecting Fields to Display as Columns [page 386]

- Sorting Search Results [page 387]

- Using Search Filters [page 388]

Creating an Application Report

To create an application report:

1. To create a report for a primary or auxiliary table, right-click a primary or auxiliary table in App Explorer and select **Create Report Definition for this table**.

2. The three blocks described in the following table are displayed in the report definition editor.

Section	Description
Columns	Contains options that let you select which columns to display in the report. For more information, see Selecting Fields to Display as Columns [page 386].
Sort Order	Lets you specify sorting options for the items returned in the report. For more information, see Sorting Search Results [page 387].
Search Filter	Lets you narrow your search for items. For more information, see Using Search Filters [page 388].

Selecting Fields to Display as Columns

The **Columns** block lets you select which fields to display as columns in the report. The *Item Id* and *Title* fields are included by default, but can be removed.

> **Tip:** The fields you select to display as columns in a report can be different from the fields you select for sorting or the fields you use in the filter.

To select the fields to display as columns in the report:

1. Drag a field from the **Report Palette** to the **Columns** block. Green up and down arrows indicate where the field is to be dropped if you release the mouse.
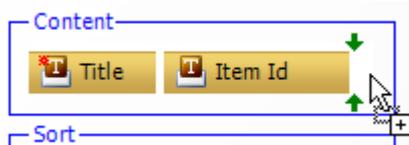
2. Release the mouse when you are satisfied with the location of the field in the block.

> **Note:** You can have only one instance of a field in the **Columns** block. If you drag a field that is already in the block, the field moves to the right side of the block and is selected (green).

3. If you change your mind and want to delete a field, select the field and then press the Delete key.

4. To change the order of the fields, select a field and use the drag-and-drop operation to move it.

5. To change the width of a column, select the right edge of the field and drag it until it is the size you want. The width of the column, in pixels, is displayed at the top right corner of the field. Use the splitter cursor to drag the dotted line to the right or to the left. When the field reaches its minimum width, **min** is shown instead of the number of pixels.
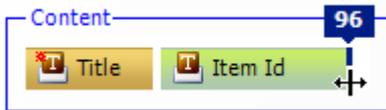
> **Note:** The minimum width is determined by the display text. You can reset this to auto size by dragging the right edge of the field to the left or by right-clicking the field and selecting **Reset to Auto Size**.

Sorting Search Results

Sort options let you sort search results based on values in selected fields. For example, you can sort items by the state they are in or by their active or inactive status.

> **Note:** The **Project (hierarchy)** field is displayed is displayed if you select **Always perform a primary sort by project** on the **Options** tab of the Property Editor.

> **Tip:** The fields you select for sorting can be different from the fields you select to display as columns or the fields you use in the filter.

To sort search results:

1. Drag a field from the **Report Palette** to the **Sort By** field in the **Sort Order** block.

> **Note:** The sort fields cannot include *File, URL, MultiGroup, MultiRelational, MultiSelection, MultiUser*, or non-fixed length *Text* fields. *SubRelational* fields use the field type that they reference, and follow the rules stated in the previous sentence.

2. If you want to sort by additional fields, drag them to the **then by** fields. You can specify a maximum of four fields.

3. To change the sort order for each field, click the up or down arrow next to the field. This determines whether the search results in each column are sorted in ascending or descending order.

Using Search Filters

A search filter is a collection of conditions and logical operators. Search filters (also known as "expressions") let you narrow your search for items. You can define a search filter by making selections from the list of operators and fields in the **Report Palette**. In expressions, fields are known as "conditions."

For example, to create a Listing report that includes items that are in the **StartState** state and that are assigned to the current user, create the following expression:

State = StartState

AND

Owner in (Current User)

> **Tip:** The fields you select for filtering can be different from the fields you select to display as columns and the fields you use to sort returned data.

To use search filters:

1. Drag a field from the **Report Palette** to the **Search Filter** block.

2. Some operators can be changed. Click the field and select another operator from the popup window that opens. For a description of operators, see Report Operators [page 388].

3. Select the field value or field values for the condition. This creates an expression. For example, if you are working with an *Approvers* field, the names that are available as selections for the field are displayed in the menu that opens. You can select one or more names from the list.

4. To group expressions to set a sequence for evaluating conditions, drag a logical operator from the **Report Palette** onto the expression. This inserts the logical operator in the place where the expression was and moves the expression to the first child of the new operator. For more information about using the drag-and-drop operation in various scenarios, see Drag-and-Drop Behavior [page 398]. For examples of report logic, see Report Logic Examples [page 397].

> **Note:** You can view a query string that represents the search filter in the **Search Filter Summary** block.

Report Operators

Logical operators let you set a sequence for evaluating conditions. *Condition* operators let you select a function for a selected field.

In the report definition editor, **Search Filter** block, AND operators are purple and OR operators are blue. The selected operator and line are lighter in color than unselected operators and lines.

Logical Operators

Operator	Description
AND	The AND operator returns all items that meet all conditions defined in the search parameters. For example, the conditions "Owner in Joe Manager" AND "State in New" return all items that are in the "New" state *and* are owned by Joe Manager.
OR	The OR operator returns any items that meet the conditions defined in the search parameters. For example, the conditions "Owner in Joe Manager" OR "State in New" return any items that are in the "New" state *or* are owned by Joe Manager.

Condition Operators

Operator	Description
= (equal to)	Use the equal to operator to find exact values. For example, select the *Active/Inactive* field with the = operator and the Active value to display all active items in the specified project. For *Date/Time* fields, dates used with this operator are treated as a Date-only field. For example, select the *Submit Date/Time* field with the = operator and type yesterday's date in the **Value** menu to find all of the items submitted into the specified project yesterday.
> (greater than)	Use the greater-than operator to find larger values than the value specified. For example, select the *Submit Date/Time* field with the > operator and type the last day of last year in the **Value** menu to find all of the items submitted into the specified project for this calendar year. Use this operator with the less-than operator to define all items between two specified dates.
< (less than)	Use the less-than operator to find smaller values than the value specified. For example, select the *Submit Date/Time* field and the < operator, and then type 2/27/2009 in the **Value** menu to return all items with a submit date and time up to 2/26/2009 11:59:59 p.m. Use this operator with the greater than operator (>) to define all items between two specified dates.

Operator	Description
>= (greater than or equal to)	User the greater-than-or-equal-to operator to find identical and greater values than the value specified. For example, select the *Submit Date/Time* field with the >= operator and type the last day of last year in the **Value** menu to find all of the items submitted into the specified project this calendar year. Use this operator with the less-than sign to define all items between two specified dates.
<= (less than or equal to)	Use the less-than-or-equal-to operator to find identical and lesser values than the value specified. For example, select the *Submit Date/Time* field with the <= operator and type the first day of this year in the **Value** menu to find all of the items submitted into the specified project before this calendar year. **Note:** For Query At Runtime searches, the time portion of the date you select from the calendar defaults to 12:00:00 AM. Therefore, to search for items submitted on 6/30/2011, select 07/01/2011 as the less than or equal to date/time (or enter the date/time manually as 6/30/2011 11:59:59 PM) and select 6/30/2011 as the greater than or equal to date/time.
<> (not equal to)	Use the not-equal-to operator to find values not equal to the value specified. For *Date/Time* fields, dates used with this operator are treated as Date-only fields. For example, select the *Submit Date/Time* field with the <> operator and type a date in the **Value** menu to find all items that were not submitted into the specified project on that date. For *Date/Time* fields, dates used with this operator are treated as Date-only fields.
contains all	Use this operator to select one or more values to return items that contain all values in the field. For example, select the "v2.0", "v2.1," and "V3.0" values as your search criteria for a *Change in Version* field, and then select "Contains All" from the operator drop-down list. Items in which "v2.0", "v2.1," **AND** "V3.0" are selected for the *Change in Version* field are returned. Multiple search conditions are allowed.
contains any	Use this operator to select one or more values to return items that contain any values in the field. Multiple conditions are allowed. For example, select the "v2.0", "v2.1," and "V3.0" values as your search criteria for a *Change in Version* field, and then select "Contains Any" from the operator drop-down list. Items in which "v2.0," "v2.1," **OR**"V3.0" are selected for the *Change in Version* field are returned. Multiple search conditions are allowed.

Operator	Description
does not contain all	Use this operator to select one or more values to return items that do not contain all specified values in the field. Multiple conditions are allowed. For example, select the "v2.0", "v2.1," and "V3.0" values as your search criteria for a *Change in Version* field, and then select "Does Not Contain All" from the operator drop-down list. Items in which "v2.0," "v2.1," **AND** "V3.0" are not selected for the *Change in Version* field are returned. Multiple search conditions are allowed.
does not contain any	Use this operator to select one or more values to return items that do not contain any values in the field. Multiple conditions are allowed. For example, select the "v2.0," "v2.1," and "V3.0" values as your search criteria for a *Change in Version* field, and then select "Does Not Contains Any" from the operator drop-down list. Items in which "v2.0," "v2.1," **OR** "V3.0" are not selected for the *Change in Version* field are returned. Multiple search conditions are allowed.
contains	Use this operator to search for keywords in a *Text* field or *Sub-Relational* field or the Item Type Prefix option. SBM automatically includes wildcard characters at the beginning and the end of the search criteria. For example, type the value *icons* for a title search and the "contains" operator to return all items that contain the word "icons" in the title. Use this operator to search for exact phrases or single keywords in the *Text* field, or for searching by items by a specific item prefix. Use this operator to search for exact phrases or single keywords in the *Text* or *Sub-Relational* field, or for searching by items by a specific item prefix.
like	Like is a comparison expression that returns data that is like the value selected. This operator gives you complete control over how SBM uses wildcard characters. For example, select the *Title* field and the "Like" operator, and then type a word in the **Value** menu followed by an asterisk to display items that have this word only at the beginning of the title. Use this operator to search for multiple phrases or keywords throughout any *Text* or *Sub-Relational* field or the Item Type Prefix option. For example, select the *Title* field and the "like" operator, and then type *change*modern* to return items that contain both the words "change" and "modern" in the title, and where "change" precedes "modern."

Operator	Description
not contains	Use this operator to return items that do not contain specified keywords in the queried *Text* or *Sub-Relational* field or the Item Type Prefix option. When you run the report, SBM automatically includes wildcard characters at the beginning and the end of the search criteria. For example, type the value *icon* and use the "Not Contains" operator for a title search to return items that do not contain the word *icon* in the title. Use this operator to search for exact phrases or single keywords not in the field or item prefix.
not like	This is a comparison expression that returns data that is not like the selected value. Select the **Title** field and the "not like" operator, and then type the first or last word of the title to exclude that item from the report. You can use an asterisk (*) in the search as a wildcard character. For example, type the value `*icons*` for a title search to return items that do not contain the word icons. Use this operator to search for multiple phrases or keywords not in the field or item prefix.
like (zero-filled)	Use this operator to include leading zeros in your criteria for the Item ID field. For example, if there are leading zeroes on the ID number, they may be left off your search criteria. For example, to find item ID number "BUG00017," type **17**. To search for multiple items by Item ID, separate each Item ID with a space.
not like (zero-filled)	Use this operator to not include leading zeros in your criteria for the *Item ID* field. For example, if there are leading zeroes on the ID number, they must be included in your criteria. For example, to find item ID number "BUG00017," you must type `00017`.
is empty	Use this operator to find items with field values that are empty (in which values have not been entered or not provided by a user). For example, select the **Description** field with the **is empty** operator to return items that do not have any content in the **Description** field.
is not empty	Use this operator to find items with field values that are not empty (in which values have been provided). For example, select the **Secondary Owner** field with the **is not empty** operator to return items that have been assigned a secondary owner user or group.
: (colon)	Use the : (colon) operator to specify all keywords that should be included in a search against all *Text* fields enabled for searching by your administrator. For example, if you select the **Text Fields With All Keywords** option and type graphic and file as field values, all items that contain the word graphic AND file are returned. The colon can be used on Text Fields With All Keywords option

Operator	Description
: (colon)	Use the : (colon) operator to specify any keyword that should be included in a search against all *Text* fields enabled for searching by your administrator. For example, if you select the **Text Fields With Any Keywords** option and type graphic and file as field values, all items that contain the word graphic OR file are returned. The colon can be used on Text Fields With Any Keywords option

Available Operators

Field Type	Operators
Binary	=
Company, Contact, Folder, Item Type, Project, Single Relational, Single Selection, State, Sub-Relational, Trinary, User	contains any, does not contain any
DateTime, Numeric, Summation	=, <>, >, >=, <, <=
Multi-Group, Multi-Relational, Multi-Selection, Multi-User	contains all, contains any, does not contain all, does not contain any
Text, Item Type, Prefix Option	contains, like, not contains, not like
Item Id	contains, like, like (zero-filled), not contains, not like
Sub-Relational	The available operators depend on the selected sub-field type.
File, URL	contains, not contains, is empty, is not empty

Creating Empty and Is Not Empty Filters

In Work Center, users can specify Is Empty or Is Not Empty operators. To create an application report with the same functionality, refer to the following field types and operators:

- Text fields with "like (empty string)" will search for Is Empty

- Text fields with "not like (empty string)" will search for Is Not Empty

- Multi-User fields with "contains any (None)" will search for Is Empty

- Multi-User fields with "does not contain any (None)" will search for Is Not Empty

- Multi-Group fields with "contains any (None)" will search for Is Empty

- Multi-Group fields with "does not contain any (None)" will search for Is Not Empty

- Multi-Relational fields with "contains any/all (None)" will search for Is Empty

- Multi-Relational fields with "does not contain any/all (None)" will search for Is Not Empty

- Multi-Selection fields with "contains any/all (None)" will search for Is Empty

- Multi-Selection fields with "does not contain any/all (None)" will search for Is Not Empty

- File and URL fields with "is Empty" will search for Is Empty

- File and URL fields with "is Not empty" will search for Is Not Empty

Basic Conditions

Reports that use basic conditions let you define search filters by selecting fields and search criteria for those fields. When you click a field that you dragged from the **Table Palette** to the **Search Filter** block, a panel opens that contains the conditions that make up the search criteria for your report. An expression is created after you define the search criteria. Multiple expressions are linked with a solid line and contain either an AND or an OR operator. You can select an expression to edit it.

> **Note:** For information about custom fields and custom field types, see Field Types [page 177]. For information about operators, see Report Operators [page 388].
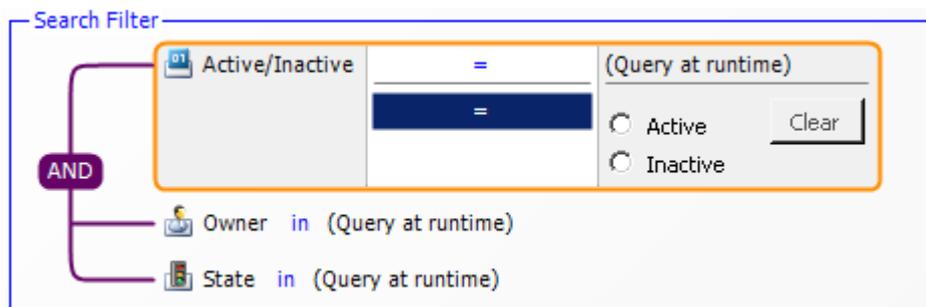
The type of field that you drag determines the options and values that are available to you. The following list describes common features of conditions.

- **Current User** – All *User*, *Multi-User*, and *Multi-Group* fields have a special "Current User" field value that matches all records where the user running the report appears in that field. For example, add the *Submitter* system field from the **Fields** section of the **Report Palette**. Select the **in** operator, and then select the **Current User** check box.

  This expression will match only records in which the user running the report (the "Current User") submitted the issue. In *Multi-Group* fields, a field is matched only when the current user is a member of one or more groups associated with the field.

- **None** – With certain field types, you can select the **None** check box. This selection returns all items meeting the rest of the report criteria that do not have a value for the selected field.

- **Query at runtime** – All fields, except *Summation* and *Numeric* fields, have a special "query at runtime" field value. You can use this when the value you want to use for the search filter is not known at design time. The user will be prompted to select the value when the report is run.

  For example, a *User* field contains users that are defined in SBM Application Administrator after deployment and therefore cannot be specified at design time. You use the "query at runtime" value in SBM Composer, and then, after the process app is deployed, users can set the values when they create a regular listing report from the application report.

- **Clear** – Most fields have a **Clear** button in the popup window. Use this button to clear selected values or to return to the default value of **Query at runtime**.

The following table lists the field types and describes the options that are available to them.

Field Type	Options
Selection, User	If you can select values for a field, you can search for specific values or select multiple values to add to your condition.
	If you select a selection field, the values that are available as selections for the field are available in the **Available Values** column. Use the right arrow button to move them to the **Selected Values** column.
	In some cases, a text box contains **Type here to search for a value**. You can type a few letters in this box to filter the list of values.
Date/ Time	Select an option, such as **Start of Last Week** on the **Special Value** tab.
	Alternately, on the **Exact Value** tab, click a date and select a time, and then click **Accept**.
	If you want to specify an absolute number, select the **Now** checkbox, select **+** or **-**, and type or select the number of days. For example, a Help Desk manager wants a report showing all incidents that were submitted in the last 30 days. You select the ≤ operator, select the **Now** check box, select **−**, and then type 30 in the **days** combo box. The expression becomes **Date ≤ 30 days ago.**.
Relational	SBM Composer does not have access to runtime data. Therefore, the only options are **Query at runtime** and **None**.

Field Type	Options
Non-Selection	If a field does not let you select a specific value, type a value in the text box. For example, if you select a *Text* field, such as the *Description* field, type a keyword or keywords to search for. *Date/Time*, *Numeric*, and *Summation* fields also let you specify a value. In some cases, the text box contains **Enter a wildcard pattern**. For information about using wildcards, see Wildcards [page 396].
Binary	If you select a *Binary* field, such as the *Active/Inactive* field, select one value for your search criteria.
Trinary	If you select a *Trinary* field, select one, two, or three values for your search criteria.
File	If you select a *File* field, you can specify the **is Empty** value (find items in which a file has not been provided by a user) or the **is Not empty** value (find items in which a file has been provided).
URL	If you select a *URL* field, you can specify the **is Empty** value (find items in which a URL has not been provided by a user) or the **is Not empty** value (find items in which a URL has been provided).

Wildcards

You can use wildcard characters to search for items and field values if **Enter a wildcard pattern** is in the text box in the search filter for the field.

The following guidelines apply to wildcard characters:

- Asterisks (*) and percent signs (%) serve as wildcard characters. A wildcard character matches zero or more consecutive characters.

- Underscores (_) match a single character.

- Wildcard characters are automatically applied to the beginning and end of your criteria. Items containing all of your criteria are returned.

- You can override automatic wildcards by including at least one wildcard in your criteria. Wildcard characters can be placed anywhere in the box and could return different results depending on the location of the wildcard character. For example, *ed returns all items ending in ed, while ed* returns all items beginning with ed.

- To find all items, type a wildcard character (* or %) or leave the box empty.

- Leading and trailing spaces are removed.

Report Logic Examples

The following examples illustrate how reports are constructed.

> **Note:** AND and OR operators always have at least two child nodes. The nodes are either fields or **Drag a field here** placeholders.

(1 or ((2 or 3) and (4 or 5)) or (6 and 7)) and 8
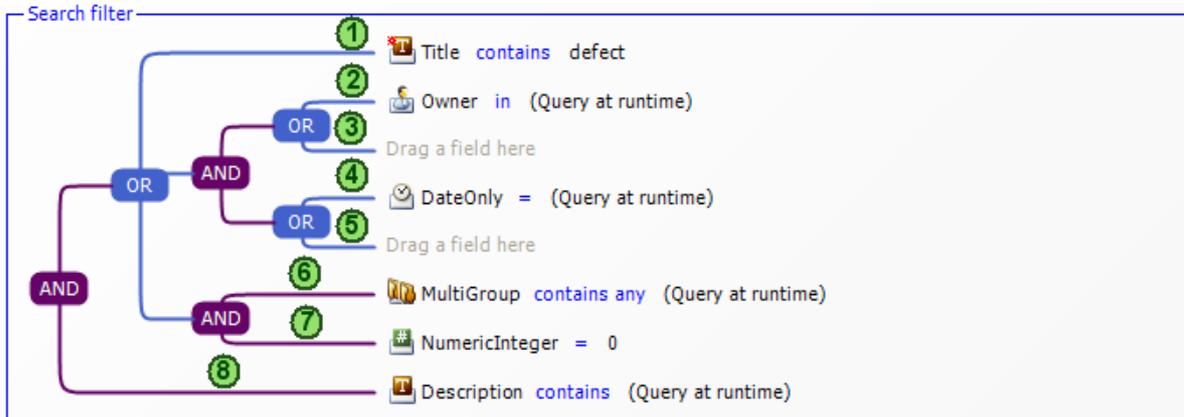
1 and 2 and 3 and 4 and 5 and (6 or 7)
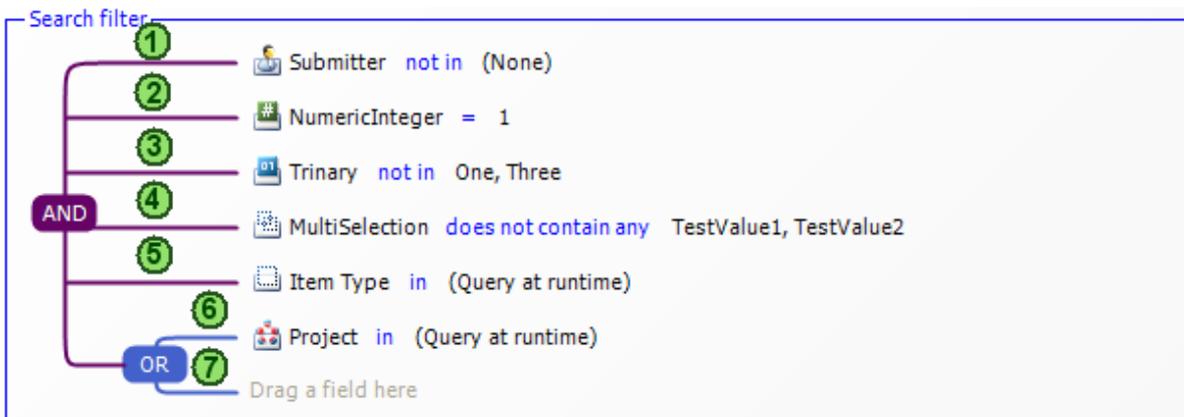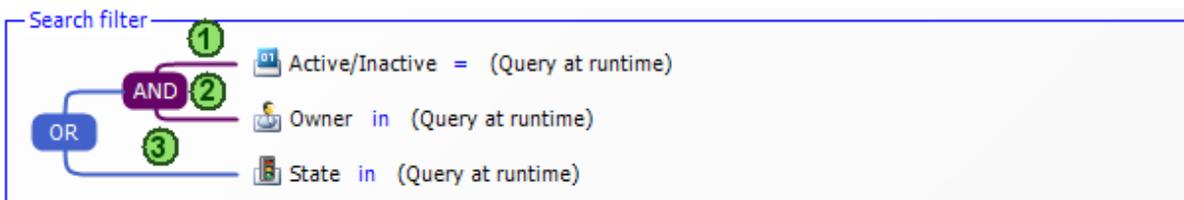
(1 and 2) or 3

1 and (2 or 3)
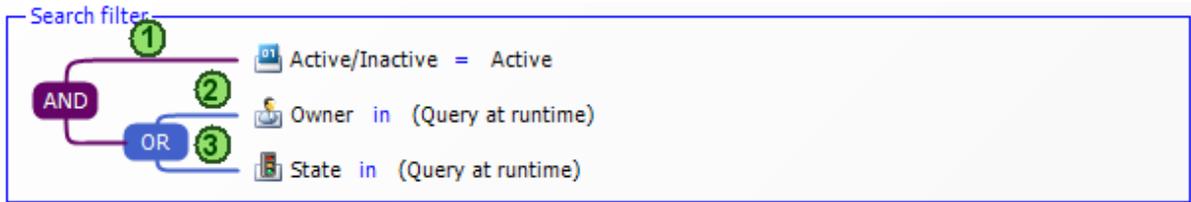
(1 or 2 or 3) and 4
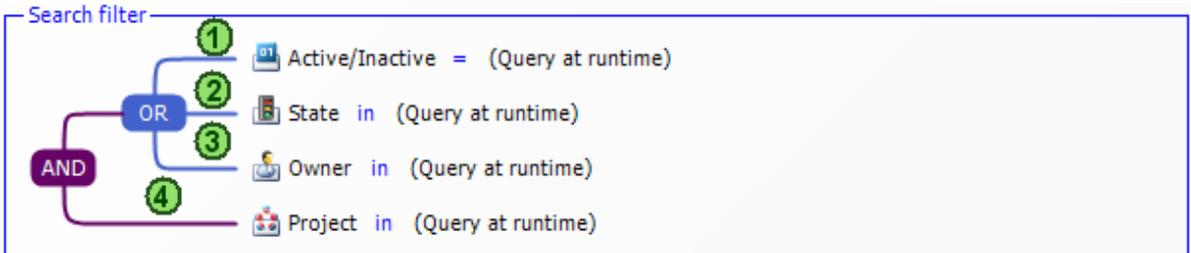
(1 and (2 or 3)) or 4
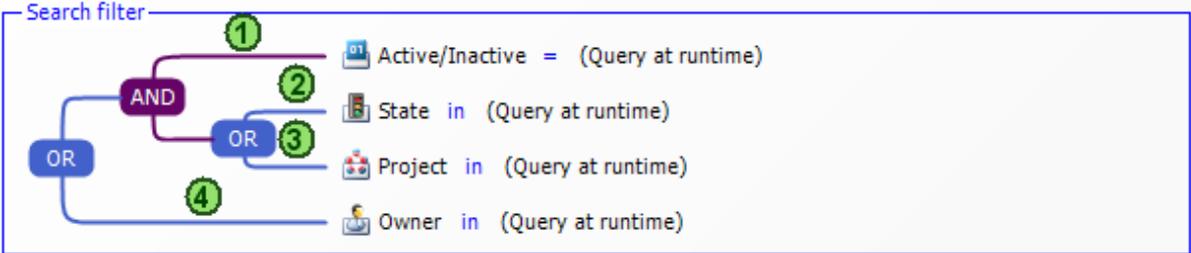
Drag-and-Drop Behavior

The following points describe how to use the drag-and-drop operation to add fields and operators to a search filter.

Drag-and-Drop Operation	Behavior
AND to AND	To add a new expression to an AND group, drag a field from the palette onto another field in the group. The new field is added above the field onto which you dragged.
OR to AND	To add an OR group to an AND group, drag the **OR** operator to one of the lines extending from the AND group. If there is a field on the top line, the OR group is added to the bottom line of the AND group. If no field is on the top line, the OR group is added to the top line of the AND group.
OR to OR	To add a new expression to an OR group, drag a field from the palette onto another field in the group. The new field is added above the field onto which you dragged.

Drag-and-Drop Operation	Behavior
AND to OR	To add an AND group to an OR group, drag the **AND** operator to one of the lines extending from the OR group. If there is a field on the top line, the AND group is added to the bottom line of the OR group. If no field is on the top line, the AND group is added to the top line of the OR group.
Existing Field	To move an existing field within an AND or OR group, drag it to another field. The existing field is placed above the field onto which you dragged it.

Application Report Settings

The following sections provide descriptions of settings used to define application reports.

Report Definition Editor

Use the report definition editor to design application reports.

In the report definition editor, drag fields to the **Columns** and **Sort Order** blocks, and drag fields and operators to the **Search Filter** block to create the format and logic of the application report. Additional options are available when you right-click in the report definition editor and right-click its fields and operators. For example, in the **Columns** block, you can right-click a column and select **Move Down**. This moves the column to the immediate right.

If you selected the **Always perform a primary sort by project** check box on the **Options** tab in the report definition Property Editor, **Project (hierarchy)** is the first sort criteria in the **Sort Order** block. This is relevant if you include multiple projects in the report search filter.

The **Associated Attachments** column in the **Columns** block is used to display any attachments that are present (for example, items, notes, URLs, and files). The **Search Filter Summary** block contains a read-only query string that represents the search filter as you build it.

The name of the report definition is displayed on the tab in the report definition editor. The name of the table the report is associated with is displayed in parentheses after the report definition name.

Report Palette

The **Report Palette** includes the following types of controls:

- **Logical Operators** include AND and OR search operators. These operators are used to specify search operators for fields and to group conditions to set a sequence for evaluating conditions. Operators can only be added to the **Search Filter** block.

- **Fields** include the fields that can be queried. These fields are in the primary table. If you add or remove fields from the primary table, they are added or removed from this section. Fields can be added to the **Columns**, **Search Order**, or **Search Filter** blocks.

Report Definition Property Editor

Use the following tabs in the report definition Property Editor to view and modify report options.

General Tab of the Report Definition Property Editor

Element	Description
Name	The name of the report definition that was provided when the report definition was created. **Primary** or **Auxiliary** is appended to the name in the report definition editor to indicate the type of table that the report is created against.
Uses table	The table (primary or auxiliary) that the report is created against.
Type	**Listing Report** is displayed as a read-only field, because listing reports are the only type of report you can create from SBM Composer.
Privilege category	Select **Guest**, **User**, or **Manager**. This determines the report privileges. Privileges determine which users can access the different levels of reports.
Description	Type a description of the report definition.

Options Tab of the Report Definition Property Editor

Element	Description
Searching and Sorting	**Include items from sub-projects**: Select this check box to include subprojects of the selected project in the query. If one of your preferred projects is a parent project, but its subprojects are not in your preferred projects list, subprojects of the parent are searched if this check box is selected. **Enable dynamic column sorting**: Select this check box to display column headers as links. Click to sort the data in ascending or descending order. The administrator determines if this option is available. If the listing report is part of a multi-view report, dynamic column sorting is not available. **Always perform a primary sort by project**: If you include multiple projects in the report search filter, select this check box to sort items by project hierarchy. If you clear this check box and do not provide other sorting criteria, items are listed randomly. **Note:** These check boxes are disabled for reports that use auxiliary tables.
Display	**Hide project titles**: Select this check box to prevent the project hierarchy headers from being displayed in the report. This check box is disabled for reports that use auxiliary tables. **Remove line breaks from memo/text fields**: Select this check box to remove line breaks from *Text* and *Memo* fields in the report. This option is useful if you plan to export the report data to another application, such as Microsoft Excel.
Columns of Linked Data	**Include linked files from attachments**: Select this check box to display links to any files attached to items. Links are displayed only if users have privileges to view attachments. **Include linked URLs from attachments**: Select this check box to display any URLs attached to items. Links are displayed only if users have privileges to view attachments. **Include linked notes**: Select this check box to display any notes attached to items. Links are displayed only if users have privileges to view notes. **Include linked items**: Select this check box to display any links to other items associated with the item being viewed. Users can click the item link to view item details. Links are only displayed if users have privileges to view the linked item. **Note:** If any of these check boxes are selected, an additional column labeled **Associated Attachments** appears at the end of the report columns.

Calculations Tab of the Report Definition Property Editor

Users can include calculations on *Numeric*, *Binary/Trinary*, and *Date/Time* fields in their report results. The available calculations are addition, subtraction, multiplication, and division.

To add a calculation to a report:

1. Select the **Add column** check box.

2. In the **Display** box, type the name that will be displayed as the Custom Header in the report.

3. In the **Formula** drop-down list, do one of the following:

   - Click the **Fields** tab and select a field that will serve as the first operator.

   - Click the **Special Date** tab and select a value such as **Start of Next Week** or **End of This Month**.

4. Select an operand from the next drop-down list.

5. In the second calculation drop-down list, select a field that will serve as the second operator.

The result type is displayed in the **Result Type** column. For example, if you specified **Description = Last Modified Date - Submit Date**, the **Result Type** is **Elapsed Time**.

> **Note:** Each selected calculated column is added to the field palette. To appear in the report, they must be added to the **Columns** or **Sort Order** block.

Chapter 20: Creating Roles

This section contains the following information:

About Roles

Watch It

Roles are created in SBM Composer as part of a process app, which can comprise multiple applications. Roles span the applications within the process app, and serve two functions:

- They are a named collection of privileges. The privileges secure user actions and data access. For example, a role named User could be a collection of privileges suitable for someone to whom items are assigned but who has no administrative tasks. Someone with that role could be unable to execute some transitions and view some fields on forms.

- They are a means to populate selection lists for *User*, *Multi-User*, and *Multi-Group* fields. You associate roles with these fields in SBM Composer.

The two functions can be connected or disconnected. When the functions are connected, a role provides privileges and populates a selection list. When the functions are disconnected, a role either populates a selection list with no privileges, or only provides privileges.

Privileges fall into two broad categories:

- System privileges control a user's ability to deploy and promote process apps, and perform actions that affect application configuration.

- Application privileges define what a user can view and act on within an application. These privileges allow fine-grained control over items, attachments, notes, reports, workflows, and fields.

Roles vs. Groups

Roles are distinct from groups, which are named collections of users. Administrators can use groups to identify a set of users based on criteria other than job function. A group could be created for a particular project, for example, or for a division within the company. You can assign roles to a group.

> **Note:** If you associate a user or group with a role, and the role contains privileges that conflict with the user or group level of product access, those privileges are not granted to the user or group.

Users and groups are associated with roles for particular projects in SBM Application Administrator.

> **Note:** When groups or users are copied, role assignments are copied with them. Also, when groups or users are imported through LDAP using a template group or user, the role assignments associated with the template are copied to the new group or user.

Use Cases for Roles

In SBM, the process design environment (SBM Composer) is independent of the administrative environment where you are concerned with specific projects, users, and groups. At design time, you might not know which specific users or groups will interact with the application that you are creating or modifying; however, you still need to be concerned with privileges from a design standpoint. Roles are the solution to this problem, because in the design environment, a role is associated with privileges but not with specific users or groups.

If you are administering an application in the runtime environment, you must assign actual users (individually or as members of groups) to roles for each project. A given user or group could have different roles in different projects. For example, two projects could have two different people assigned to the Build Manager role. The *role* is the same, but the *person filling the role* for a specific project (or for a specific deployment of the process app) is different.

A good place to start when deciding what roles to create is to look at the user-type fields in your primary items—particularly fields that determine ownership for items. For example, if you have an Engineer field that determines ownership of an item in a particular state, Engineer might be an appropriate role to create.

The following sections describe various ways you can use roles.

Creating Specialized Roles

Every application has two default roles: Administrator and User. These broad roles could be adequate in simple applications. However, most applications require more specialized roles.

In an employee time-off application, there could be Employee, Manager, and Payroll roles. The Manager and Payroll roles have privileges that are not part of the Employee role. For example, the Manager role could have the Delete Items privilege. The Advanced Fields section might be for payroll purposes only, so the Payroll role would have the Advanced Fields privileges.

Employees requesting vacation time have the Employee role, the employee's manager who approves or rejects the request has the Employee and Manager roles, and the payroll clerk has the Employee and Payroll roles. Because privileges are cumulative, the manager has all of the privileges granted by the Employee role, as well as any additional privileges granted by the Manager role. Similarly, the payroll clerk has all of the privileges granted by the Employee role, as well as any additional privileges granted by the Payroll role.

Assigning Privileges to Groups

Administrators create groups to limit the number of privilege sets that they need to maintain. For example, there could be a group called Development Team. The members of the group include individual users associated with the Engineering, Quality Assurance, and Documentation roles. Suppose a system privilege needs to be added for these users. An

administrator can use SBM Application Administrator to quickly add the system privilege to the group, and the privilege set for all of the individual users would be updated.

Tip: It is best to use groups to assign system privileges only, and to use roles to assign other privileges.

Associating Roles with a User Field

In an issue management application, after engineers finish fixing an issue, they transition the issue from the Fixing Issue state to the Peer Review state. In this state, another engineer reviews the code that the first engineer changed to fix the issue. The owner of the Peer Review state is Peer Reviewer, which is a *User* field associated with the Engineer role.

The engineer selects the peer reviewer from a list box that is populated with users with the Engineer role.

Associating Roles with a Multi-User Field

In a change request application, after a technical analyst determines that a change request should be considered by the Change Approval Board (CAB), he or she transitions the change request to the In Review state. The owner of this state is Reviewers, which is a *Multi-User* field associated with the CAB Members role. This role has privileges to own and view change requests, but not submit or delete them.

The technical analyst selects the appropriate board members from a list box that is populated with all users with the CAB Members role.

Associating Roles with a Multi-Group Field

Multi-Group fields are useful when you want to take advantage of queueing. Queueing takes place when users evaluate issues in a backlog and assign certain issues to themselves.

For example, in a help desk application for a company, employees submit issues for technical problems they are experiencing. The issues move into a Backlog state. The secondary owner for this state is a *Multi-Group* field called *Employee Location*. This field is associated with two roles: Technician and Manager. The Manager role is associated with the field because managers often work part-time as managers and part-time as technicians.

The *Employee Location* field is populated by groups that represent the various company sites. Each group is comprised of technicians and managers who are located at the site. All of the technicians and managers at the selected site own the issues in the Backlog state, and can assign the issues to themselves.

Associating Multiple Roles with a User

Privileges are additive and can be indifferent to the role-based transition restrictions specified on the Restrict By Role Tab of the Transition Property Editor [page 165]. This means that all privileges granted to all roles are initially put into a pool, and after that, role-based transition restrictions are considered. For example, suppose the Manager role is not restricted from the **Escalate** transition, and has the "Transition Item if Owner" privilege. Laura is associated with the Manager role, but she is not the owner of the source state. Therefore, she should not be able to access the **Escalate** transition.

However, Laura is associated with another role, the Engineer role. The Engineer role is restricted from the **Escalate** transition, but has the "Transition All Items" privilege. The

Engineer role privileges are added to the pool of privileges granted to Laura. Therefore, Laura can access this transition, even though she is not the owner of the source state.

Roles and Ownership

Every state in a workflow must have one primary owner and can also have a secondary owner. The primary owner field must be a *User* field. The secondary owner field can be a *User*, *Multi-User*, or *Multi-Group* field. The fields can be populated with users who are assigned the role or roles associated with the field. You specify the roles that are associated with the fields in the Property Editor for the field in SBM Composer.

The selected owner is responsible for performing an action and then moving the item to the next state. For example, a new item in an issue tracking application starts in a Submit state, and then moves to an Assigned state. The owner of the Assigned state could be someone with the Development Lead role. He or she assigns the item to someone with the Engineer role. The item moves through various states in the workflow until it is closed.

The following diagram illustrates the steps involved in the process of establishing the ownership of items in an application.
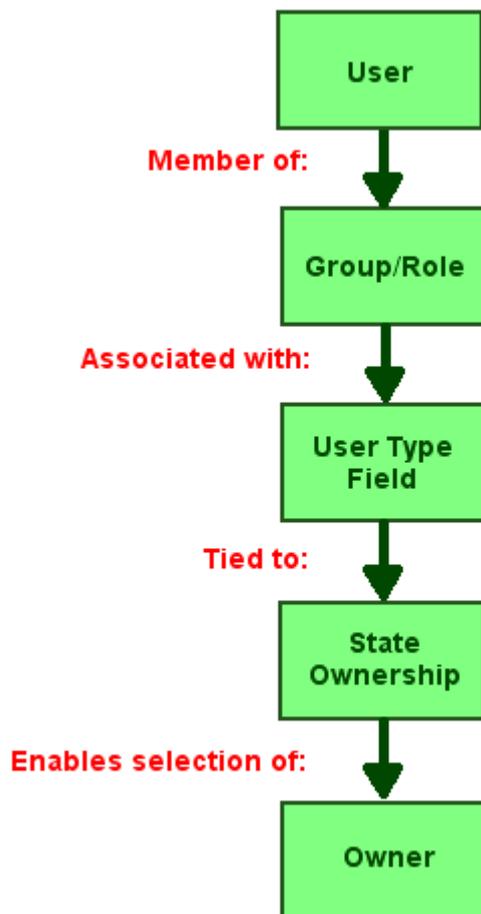
Comparing Role Privileges and User/Group Privileges

Roles let a designer (working in SBM Composer) work with privileges abstractly—that is, in relation to a function (such as QA Manager) rather than actual people. In contrast, an administrator (working in SBM Application Administrator) assigns privileges to actual users and groups of users. The administrator can assign privileges directly to users or groups or by associating a user or group with one or more of the roles created in SBM Composer.

The privileges available to the designer when creating or editing roles are all available to the administrator as user and group privileges, and the names are all close to identical (for example, the "View Fields in the 'User' Section" role privilege maps to the "View User Fields" user/group privilege). However, not all privileges available in SBM Application Administrator are available as role privileges. For example, administrative privileges for modifying a user profile and various system privileges are user/group privileges and are available only in Application Administrator.

Generally, role privileges alone are not enough for complex process app implementation, and you will likely want to combine roles with additional user/group privileges applied in Application Administrator.

Role privileges for primary tables are based on projects. The privileges can be categorized as follows:

- **Advanced** - View

- **All** - Submit, Own, Delete, View

- **Archived Items** - View, Restore

- **Attachments** - View, Add, Edit, Set, Delete

- **Change History** - View

- **Hidden** - View

- **Item** - View, Update, Transition, Mass Update

- **Manager** - View

- **Notes** - View, Add, Edit, Set, Delete

- **Principal** - Link/Unlink

- **Principal and Subtasks** - View

- **Reports** - Manage, Create, Modify, Run, Delete, Create/Modify Advanced SQL Queries

- **Section** - View, Edit

- **State Change** - View

- **Subtasks** - Link

- **System** - View

- **User** - View

- **Version Control History** - Manage

- **Workflow** - View

Role privileges for auxiliary tables are not based on projects. The privileges can be categorized as follows:

- **Advanced** - View

- **All** - Submit, Update, Delete, View

- **Attachments** - View, Add, Edit, Set, Delete

- **Change History** - View

- **Hidden** - View

- **Item** - Mass Update

- **Manager** - View

- **Notes** - View, Add, Edit, Set, Delete

- **Reports** - Manage, Create, Modify, Run, Delete, Access, Create/Modify Advanced SQL Queries.

- **Section** - View, Edit

- **User** - View

Creating and Editing Roles

Note: To edit an existing role (for example, to change its name), you must have the role checked out.

To create a new role:

1. Right-click the **Roles** heading in App Explorer, and select **Add New Role**.

   Tip: You can import a shared role instead of creating a new one. For details, see Shared Roles [page 409].

2. In the role Property Editor, give the role a name (maximum of 64 characters), and then enter a description for it.

   Note: The description is optional, but it can be useful for clarifying what functions are to be performed by users assigned to this role.

3. Select the **Shared** check box if you want to import the role into another application in the process app. The role in the other application will have the same privileges. For details, see Shared Roles [page 409].

4. Associate privileges with the role by selecting or clearing the check boxes in the roles editor. The privileges determine what users with this role can do in the various categories (such as items and fields).

> **Note:** The set of available privileges from which to select is different depending on whether you are creating a role for a primary table or auxiliary table. This is because the activities that you can perform for the two types of table are different.

> **Tip:**
>
> - To select all listed privileges at once, click **Check all**. To clear all the check boxes, click **Uncheck all**.
>
> - To group privileges, click a column header in the editor. To sort privileges alphabetically within a group, click the **Privileges** column header before you click the other column header, or press the Shift key while clicking the group column header. The sorting is remembered in the next SBM Composer session.

5. Save your changes by clicking the **Save locally** icon, or click **Check In** on the **Repository** tab of the Ribbon to both save your changes locally and check them in to the repository.

Shared Roles

There are some roles that need to be in multiple applications in a process app. A role that is marked as "shared" can be imported into any other application in a process app. The name and privileges granted to the shared role will apply to the role in the other applications.

> **Note:** You can change the name of the role in the new application and change its privileges later; however, you must clear the check box in order to rename a shared role.

If the **Shared** check box is selected for any role in any application:

- After you add another application to the process app, you are prompted whether you want to add the shared roles from existing applications to the new application.

- When you right-click the **Roles** heading in App Explorer, you can select **Add Shared Role** and then select a shared role as the role (or as the basis for the role) you want to add to the other application.

Shared roles do not apply to roles in the Global Process App because the Global Application is the only application that exists in the Global Process App.

Roles Editor

In the roles editor, you create and edit roles, giving each role a name (such as QA Manager) and a description, and associating privileges with the role. You can also specify whether the role should be used globally. For more information, see Creating and Editing Roles [page 408].

> **Note:** By default, all privileges for primary tables and auxiliary tables are granted to the Administrator role.

Chapter 21: Using Actions

This section contains the following information:

About Actions

You can define an action that executes when an initiating item is transitioned from one state to another in your workflow.

The following table describes the action types.

Action Type	Description
Transition	Causes a transition to be executed when the workflow reaches a transition or state on which the action is defined.
	For example, you could use a transition action to transition the subtasks for an item based on the value of a *Binary/Trinary* or *Single Selection* field.
	For more information, see Transition Action Tutorials [page 421].
Trigger	Causes associated primary items to be automatically transitioned as they move through a workflow. Associated items can be relational field selections, principal or subtask relationships, or item link attachments.
	For example, you could have two items that are related but stored in separate projects. The two items can be linked to each other and transitioned together as they move through the workflow. When one item is transitioned, any other items that are linked to the current item are also transitioned.
Script	Causes a script to be run when the workflow reaches a transition or state on which the action is defined.
	For example, you could have a script that copies the value of a *Title* field to a *Description* field when a Submit transition is executed.

Action Type	Description
Web Service	Causes a Web service to be invoked when the workflow reaches a transition or state on which the action is defined. For example, you could use a Web service operation to create items in an auxiliary table when the workflow reaches a transition or state on which the action is defined.
Orchestration Workflow	Causes an orchestration workflow to be invoked when the workflow reaches a transition or state on which an action is defined. For example, you could invoke an orchestration workflow that creates a set of items based on the values a user selects from a *Multi-Relational* field. The action is defined on the transition that is executed after the user selects the items.

Restriction: Do not use "transition", "continue executing (asynchronous) orchestration workflow," or "trigger" action types on an incoming transition to a decision.

Considerations for Using Actions

Consider the following when you set up actions:

- Actions can be performed on items transitioned manually by users, by other actions, or by mass updates.

- Actions must be defined at the level at which a state or transition was created and cannot be overridden for a sub-workflow or project.

- All actions must conform to the workflow. In other words, you cannot use an action to execute a transition that is not already defined in the workflow.

- Actions can be defined for all transition types, but only **Regular**, **Copy**, **Post**, and **Subtask** transitions can be executed as a result of an action.

- To ensure that actions execute properly, make sure there are no required fields for the transition that will execute as a result of an action. If necessary, provide default values for data in fields required for any transition that contains an action.

- You can define multiple actions for a transition. To the extent possible, actions are executed in the order in which they are listed on the **Actions** tab of the Property Editor for the selected transition or state. Use **Move up** and **Move down** to reorder actions as needed.

- If actions are used to execute multiple transitions on an item, Item Type restrictions may prevent subsequent transitions from executing.

- If you have multiple transition actions, the first one that evaluates as true prevents the evaluation of further actions.

- If you have an orchestration action, it must be last (it cannot be ordered with the other actions), and the orchestration workflow will not be invoked if a previous

transition action was executed. If the orchestration workflow must always run, the setup must be designed to account for that (for example, have the orchestration action on the transition, and find a way to move the transition actions to the next state).

- When there are multiple asynchronous orchestration actions on the same transition, only one event of each event type is raised. Events start orchestration workflows that will run simultaneously depending on available resources, so the ordering of event actions on the **Actions** tab has no effect on the execution order of the orchestration workflows.

  - If you specify multiple local event actions, the orchestration workflows that are specified will all be triggered by that one event.

  - If you specify multiple external event actions, then only one event of each type will be raised, and the data mapping specified by the first one in the list that is of that type will be used.

    Note: An event type is a distinct combination of ProductName, Version, ProductInstance, EventType, and ObjectType. This combination defines the matching criteria for an event, whether it is defined as a local event or an external event.

- When a transition action invokes a **Post** or **Subtask** transition and you do not specify a post-item project in SBM Application Administrator, if only one valid project is available, the submit defaults to the valid project. If there are multiple projects available, the submit does not occur and an error is generated.

- When a transition action affects parent or child items, the affected item must be in a state that is valid for that transition. For example, suppose a "Referenced items in relational field" transition action defines that a transition from State 2 to State 3 be executed on the specified parent item. If the parent item is not in State 2 when the transition with the action is executed, it will remain in its current state instead of moving to State 3.

- If an action fails, an error is recorded in the Event Viewer and displayed to users in the **Item Details** frame for the item that initiated the action.

- Do not use "transition," "continue executing (asynchronous) orchestration workflow," or "trigger" action types on an incoming transition to a decision.

- The SBM Application Engine executes Web service functions, scripts, transition attribute scripts, transition actions and state actions, and events, in the following order:

  1. Web service function for the pre-transition context

  2. Script for the pre-transition context

  3. Transition attribute scripts for the pre-transition context

  4. Transition executed by users

  5. Script for the post-transition context

  6. Transition attribute scripts for the post-transition context

7. Web service function for the post-transition context

8. Script for the post-state context

9. Web service function for the post-state context

10. Script for the pre-state context

11. Web service function for the pre-state context

12. Transition completed and recorded in the database

13. Transition actions

14. Events are emitted

15. Subtasks and posted items are submitted

16. State actions are performed

Required field attributes are validated after the transition is executed by a user (step 4) and before the post-transition context (step 5).

Using the Action Wizard

Use this wizard to define the action you want to associate with the selected state or transition. You can set up some action types to be executed before or after a transition, upon entry to a state, or upon exit from a state.

For a state or transition, available actions include invocation of a "wait for reply (synchronous)" orchestration workflow, transition of a linked item, execution of a script, and execution of a Web service method. For a transition, "continue executing (asynchronous)" orchestration workflow) and trigger actions are also available.

On each page of the wizard, you refine the action rule description.

- For a transition, the description includes which item will be affected, the conditions under which the action will occur, and the transition to be executed.

- For a "continue executing (asynchronous)" orchestration workflow, the description includes which item will be affected, the conditions under which the action will occur, and whether to use the default "Event without Reply" event (to invoke an orchestration designed to work with this application) or an external event definition (to invoke any other orchestration). The description also includes when the execution occurs (for example, before or after a transition).

  For asynchronous orchestration workflows, you can also specify the event type as "transition" or "action". Transition event types use the application workflow name, name of the transition start state, and name of the transition to construct the event type string. Action event types also add the orchestration workflow name to the event type string. This allows for specific binding between the event type and orchestration workflow in the Orchestration event map.

- For a "wait for reply (synchronous)" orchestration workflow, the description includes which item will be affected, when the execution occurs (before or after a transition, for example), and the orchestration workflow to be executed.

- For a Web service, the description includes which item will be affected, when the execution occurs (before or after a transition, for example), and the Web service method to be executed.

- For a script, the description includes which item will be affected, when the execution occurs (before or after a transition, for example), and the script to be executed.

- For a trigger, the description includes which item will be affected, the conditions under which the action will occur, and the trigger to be executed.

Some aspects of the description (such as orchestration workflow type, field names, values, and relationships) must be specified. Click links in the description to make your selections.

Click **Next** and **Back** to modify your choices until you are satisfied with the action rule description. Click **Finish** to save the description.

The **Action Wizard** includes the following tasks:

- Selecting the Action Type [page 415]

- Selecting the Affected Item [page 416]

- Selecting the Timing [page 418]

- Selecting the Condition [page 419]

- Selecting the Action [page 421]

- Selecting the External Event [page 421]

Selecting the Action Type

On this screen, select the type of action that should be performed for the selected state or transition. For states and transitions, you can select a "wait for reply (synchronous)" orchestration workflow, script, transition, or Web service. For transitions, you can also select a "continue executing (asynchronous)" orchestration workflow or a trigger.

Note: If you are editing an existing action, you cannot change its action type.

Action Type	Description
Orchestration Workflow	The orchestration workflow that should be invoked as a result of this action. For transitions, you can also specify whether the application workflow should continue processing as soon as the selected orchestration workflow is invoked ("continue executing [asynchronous]"), or wait for a reply from the orchestration workflow before continuing ("wait for reply [synchronous]"). For asynchronous workflows, you can also specify whether the workflow uses the local event or an external event definition.
	For states, the application workflow always waits for a reply from the selected orchestration workflow before processing continues.

Action Type	Description
Script	The defined script that should be executed as a result of this action.
Transition	The transition that should be executed as a result of this action. Some combination of **Regular**, **Copy**, **Post**, and **Subtask** transitions will be available, depending on the affected items you select later in the **Action Wizard**.
Trigger	The trigger that should be executed as a result of this action. Triggers can only be executed for transitions.
Web Service	The defined Web service that should be invoked as a result of this action.

Selecting the Affected Item

On this screen, select the item to be affected by this action.

Action Type	Affected Item
Orchestration workflow ("continue executing [asynchronous]")	**Items that reference this item using relational field:** Invokes the orchestration workflow on items in which the current item is selected as a value. Only relational fields that point to the primary table can be specified. **Referenced items in relational field:** Invokes the orchestration workflow on the principal item of a subtask when the specified condition is met. Only relational fields that point to the primary table can be specified. **This item:** Invokes the orchestration workflow on the current or initiating item.

Action Type	Affected Item
Transition	**Items that reference this item using relational field:** Defines actions for the parent in parent-child transitions. This enables users to establish a parent-child relationship between multiple items and enables you to define an action that executes transitions on items in which the current item is selected as a value. The current item can be selected as a value in multiple items. The action is applied to all items that meet the criteria. **Referenced items in relational field:** Defines actions for child items in parent-child transitions. This lets users define a parent-child relationship between items and lets you define an action that executes transitions on items selected as values in the field. **This item:** Defines subsequent actions for the initiating item. For most transition types, **This item** refers to the initiating item. For **Copy** transitions, **This item** refers to a copied item created as a result of the **Copy** transition. Note that if you are setting action attributes for a state: • This item is the only option available to you, and refers to either the principal item or the item affected by a trigger. • The affected item will be transitioned only if its current state is valid for the transition defined in the action (that is, the transition defined in the action must emanate from the current state).

Action Type	Affected Item
Trigger	**Items that reference this item using relational field:** Executes the trigger on items in which the current item is selected as a value. Select a primary *Multi-Relational* field to execute the trigger on items that have the current item selected as a value in this field when the condition for the action is met. **Principal item:** Executes the trigger on the principal item of a subtask when the specified condition is met. This option is not available for **Copy** transitions. **Referenced items in relational field:** Executes the trigger on items selected as values in the selected primary *Single Relational* or primary *Multi-Relational* field.Select a field to execute the trigger on items that are selected as a value in that field. **Subtask item** (triggers): Executes the trigger on subtasks when the specified condition is met. This option is not available for Copy transitions. **This item:** For Regular transition types, executes the trigger on the current or initiating item. For Copy transition types, executes the trigger on the copy that was created as a result of the transition. **Triggerable linked items:** Executes the trigger on linked items that have enabled triggers.

Selecting the Timing

On this screen, specify the timing of the action.

Action Type	Timing
Orchestration workflow ("wait for reply [synchronous]")	**After:** Invokes the orchestration workflow on exit from the selected state or after the selected transition is executed. **Before:** Invokes the orchestration workflow on entry to the selected state or before the selected transition is executed.
Script	**After:** Runs the script on exit from the selected state or after the selected transition is executed. **Before:** Runs the script on entry to the selected state or before the selected transition is executed.

Action Type	Timing
Web Service	**After:** Invokes the Web service method on exit from the selected state or after the selected transition is executed. **Before:** Invokes the Web service method on entry to the selected state or before the selected transition occurs.

> **Note:** For a transition or state, you can define only one **After** and one **Before** action. If you defined one or the other for the selected state or transition, that choice is disabled. Click the disabled choice to see where it is used.

Selecting the Condition

On this screen, select the condition that causes this action to be executed. The listed conditions depend on the affected item and the selected action type.

Sibling's Field Value

Performs an action on items where the current item is selected as a value in a primary *Multi-Relational* field. **All are**, **none are**, **half are**, and **most are** are available as operators. (**Most are** means "more than half.") The value of this field is evaluated for other selected items in the *Multi-Relational* field in which the current item is a selection.

Affected Item	Action Types
Items that reference this item using relational field	Orchestration workflow ("continue executing [asynchronous]"), Transition

Field Items Value

Performs an action on items that are selected as values in a primary *Relational* field. The *Binary/Trinary* field is contained in the same table as the selected primary *Relational* field. If the primary *Relational* field is a *Single Relational* field, **is** and **is not** are available as operators. If the primary *Relational* field is a *Multi-Relational* field, **all are**, **none are**, **half are**, and **most are** are available as operators. (**Most are** means "more than half.")

Affected Item	Action Types
Referenced items in relational field	Orchestration workflow ("continue executing" [asynchronous]), Transition

Sibling Tasks

Performs the action on the affected item, which is assumed to be a subtask, evaluating the status of sibling subtasks as defined by their value in the *Binary/Trinary* or *Single Selection* field used to determine subtask status. Three values are available: **In Progress**, **Completed**, and **Rejected**. You can set the action to execute when all, none, half, or most subtasks meet one of two or three specified values. (Most means "more than half".)

Affected Items	Action Types
This item	Orchestration workflow ("continue executing [asynchronous]"), Transition, Trigger
Principal item	Trigger
Subtask item	Trigger

Subtask's Status

Performs the action based on the value of the *Binary/Trinary* or *Single Selection* field used to determine subtask status. Three values are available: **In Progress**, **Completed**, and **Rejected**. You can set the action to execute when all, none, half, or most subtasks meet the specified value. (Most means "more than half.")

Affected Items	Action Types
This item	Orchestration workflow ("continue executing [asynchronous]"), Transition, Trigger
Subtask item	Trigger

Trigger Received

States only: Lets you enable a trigger for a state. You must also set the trigger to fire during one or more transitions.

Affected Item	Action Type
This item	Trigger

Unconditionally

The action is always performed on items satisfying the condition.

Affected Items	Action Types
Referenced items in relational field	Orchestration workflow ("continue executing [asynchronous]"), Transition
This item	Orchestration workflow ("continue executing [asynchronous]"), Transition, Trigger
Principal item	Trigger
Subtask item	Trigger

Affected Items	Action Types
Triggerable linked items	Trigger

Selecting the Action

On this screen, select the action to be taken:

- The orchestration workflow or Web service method to be invoked

- The script, trigger, or transition to be executed

If the workflow, method, or script you need is not listed, you can create it from this screen.

Selecting the External Event

On this screen, select the event definition that causes this action to be executed. Only event definitions that were exported from an orchestration or application and then imported into the calling application are available. After you select an event definition, click **data mapping** in the rule description, and then map orchestration inputs to application data in the Service Mappings Dialog Box [page 459].

> **Note:** For more information, see the orchestration links information in the *SBM Orchestration Guide*.

Transition Action Tutorials

This section contains the following tutorials:

- Tutorial: Basing an Action on a Single Selection Field [page 421]

- Tutorial: Defining Subtask-Driven Actions [page 423]

- Tutorial: Defining Child-Driven Actions [page 426]

- Tutorial: Defining Parent-Driven Actions [page 428]

- Tutorial: Submitting Multiple Primary Items [page 430]

Tutorial: Basing an Action on a Single Selection Field

You can create an action that executes a specific transition when a user selects a value in a field. For example, when a user selects a value from a *Category* field, the item is transitioned to a specific state based on that value. If users select **New Order** from the *Category* field, the **Place New Order** transition is executed and the item is sent to an **Order Placed** state.

Prerequisites:

The following must be set up before you perform this procedure:

- Application workflow renamed to `Single Select`.

- *Single Selection* field named "Category" with **New Order**, **Service**, and **Account Management** values

- **Assigned** state

- **Assign** transition that leads from the **New** state to the **Assigned** state

- **Order Placed** state

- **Place New Order** transition that leads from the **Assigned** state to the **Order Placed** state

- **Closed** inactive state

- **Close** transition that leads from the **Order Placed** state to the **Closed** state

To set up an action that is based on a *Single Selection* field:

1. In the application workflow editor, right-click the **Assign** transition, and select **Show Actions**.

2. On the **Actions** tab of the workflow Property Editor, click **New**. The **Action Wizard** opens.

3. Select **Transition** and then click **Next**.

4. Select **This item** and then click **Next**.

5. Select **Field Value**.

6. Click **specified** and select **Category**. The **value** link changes to one of the values for the *Category* field.

7. If that value is not **New Order**, click the value and then select **New Order**.

8. Click **Next**.

9. Select **SingleSelect:Place New Order(Assigned->Order Placed)**.

10. Click **Finish**.

11. Deploy the process app.

12. Test the process app.

    a. In SBM Work Center, submit an item into the Single Select project.

    b. In the **Title** box, type a name for the item.

    c. In the **Category** list, select **New Order**.

d. Click the **Assign** transition button.

The **Place Order** transition is automatically executed.

Tutorial: Defining Subtask-Driven Actions

This example describes how to add a **Subtask** transition that posts an item from one application to another. After you add the **Subtask** transition, you can define actions that transition principal tasks based on the subtask's values in a *Binary/Trinary* or *Single Selection* field used to determine subtask status. For example, in a Human Resources (HR) workflow that requires an Information Technology (IT) department to purchase a computer for a new employee, you can create a **Subtask** transition that submits a task into the IT workflow for the purchase. After IT closes its task in its workflow, the task in the HR workflow is automatically closed.

Prerequisites:

The following must be set up before you perform this procedure:

- Two applications, named `New Hires` and `IT Tickets`

- One application workflow for each application, named `New Hires` and `Asset Management`, respectively.

- *Binary/Trinary* field named `Subtask Status` in the **IT Tickets** table, with **In Progress** and **Completed** values.

- In the **New Hires** workflow:
  - **Waiting for IT** active state
  - **Send to IT** subtask transition that leads from the **New** state to the **Waiting for IT** state.
  - **Closed** completed state
  - **Resolved by IT** regular transition that leads from the **Waiting for IT** state to the **Closed** state.

- In the **Asset Management** workflow:
  - **In Progress** active state
  - **Assign** regular transition that leads from the **New** state to the **In Progress** state
  - **Closed** completed state
  - **Close** regular transition that leads from the **In Progress** state to the **Closed** completed state

> **Note:** A process app that defines subtask-driven actions requires at least two applications. The applications can be in the same process app, or can be in another process app, if the current process app contains references to them. See About References [page 369] for more information.

To set up a subtask-driven action:

1. Configure the **Send to IT** transition in the **New Hires** workflow.

   a. On the **Options** tab of the transition Property Editor, select the **Quick Transition** check box.

   b. On the **Post Options** tab of the transition Property Editor:
      - In the **Post Application** list, select **IT Tickets**.
      - In the **Post Table** list, select **IT Tickets**.
      - In the **Use submit transition** list, select **Asset Management : Submit**.
      - For **When finished, show**, select **New Item**.

- In the **Item Link Type** list, select **2-Way, no triggers**.

2. Set the subtask status in the **Asset Management** workflow.

   a. Select the **Completed** state. This is the state to which the subtasks are transitioned.

   b. On the **General** tab of the state Property Editor, in the **Subtask** list, select **Subtask Status**.

   c. Select the **Close** transition.

   d. On the **Field Overrides** tab of the transition Property Editor, select the *Subtask Status* field, select the **Override field properties** check box, select **Set to Default**, and then select **Completed** in the **Default Value** list.

   e. Still on the **Field Overrides** tab, make sure that there are no required fields for this transition, or set default values for fields that require values.

3. Create an action in the **New Hires** workflow that will transition principal items, based on the value of the *Subtask Status* field.

   a. Select the **Waiting for IT** state. This is the state in which principal items remain while they wait for subtasks to complete.

   b. On the **Actions** tab of the state Property Editor, click **New**.

   c. In the **Action Wizard** that opens, select **Transition**, and then click **Next**.

   d. Select **This item**, and then click **Next**.

   e. Select **Subtask's Status**.

   f. Select **all are**, select **Completed**, and then click **Next**.

   g. Select the **New Hires:Resolved by IT (Waiting for IT->Close)** transition, and then click **Finish**.

   h. Click the **Resolved by IT** transition.

   i. On the **Field Overrides** tab of the transition Property Editor, make sure there are no restrictions, such as Item Type restrictions, on the transition.

4. Test the process app.

a. In SBM Work Center, submit an issue into the New Hires project.

b. Click the **Send to IT** transition button.

c. Submit an issue into the IT Tickets project from the window that opens.

d. Click the **Assign** transition button.

e. Click the **Close** transition button. The IT Tickets subtask item is in the **Closed** state.

f. Click the principal item in the **Subtasks** or **Attachments** section. The New Hires principal item opens, and is in the **Closed** state.

> **Note:** In SBM Work Center, users are presented with a list of projects into which they have privileges to submit items. If only one project is valid for submission, the submit form opens for that project. If multiple projects are valid for submission, you can specify a single project by selecting a value in the **Use submit transition** field in the transition Property Editor. For more information, see Post Options Tab of the Transition Property Editor [page 161]. In addition, in SBM Application Administrator, administrators can configure transitions in projects to select a specific project into which all items are submitted.

Tutorial: Defining Child-Driven Actions

This example explains how to define an action that transitions a parent item when its child items are completed. It includes adding fields to drive the action; defining a transition action that creates child items, and defining the transition action that closes a parent item, based on the status of its children.

Prerequisites:

The following must be set up before you perform this procedure:

- *Multi-Relational* field named `Children`

- *Single Relational* field named `Parent`

- Application workflow named `Parent`:

  - **In Progress** state

  - **Assign** transition that leads from the **New** state to the **In Progress** state

  - **Child Created** state

  - **Create Child** Post transition that leads from the **In Progress** state to the **Child Created** state.

  - **Closed** state

  - **Close** transition that leads from the **Child Created** state to the **Closed** state.

- Application workflow named `Child`:

  - **In Progress** state

  - **Assign** transition that leads from the **New** state to the **In Progress** state

  - **Close** transition that leads from the **In Progress** state to the **Closed** state

To set up a child-driven action:

1. Configure the **Create Child** transition in the **Parent** workflow.

   a. In the **New Item in Original Original Item's Field** list, select **Children**.

   b. In the **Set Original Item in New Item's Field** list, select **Parent**.

   c. In the **Use Submit Transition** list, select **Child : Submit**.

   d. In the **Item Link Type** list, select **2-Way, no triggers**.

2. Create an action that will close the parent item when the status of every child item is **Completed**.

   a. In the **Child** workflow, right-click the **Close** transition, and select **Show Actions**.

   b. On the **Actions** tab of the transition Property Editor, click **New**.

   c. In the **Action Wizard** that opens, select **Transition**, and then click **Next**.

   d. Select **Items that reference this item using relational field**.

   e. Click the **specified** link, select **Children**, and then click **Next**.

    f. Select **Sibling's Field Value**.

    g. Click the **specified** link and then select **Active/Inactive**.

    h. Make sure the next link reads **all are**.

    i. Click the **Active** link, select **Inactive**, and then click **Next**.

    j. Select **Parent:Close (Child Created->Closed)**, and then click **Finish**.

3. Deploy the process app.

4. Test the process app.

    a. In SBM Work Center, submit an item into the Parent project.

    b. Click the **Assign** transition button.

    c. Click the **Create Child** transition button.

    d. Submit an item into the Child project from the window that opens.

    e. Click the **Assign** transition button.

    f. Click the **Close** transition button.

    g. Select all items in the **Children** field and then click **OK**.

    h. Click the parent item in the **Attachments** section. The parent item opens, and is in the **Closed** state. This is because when the children moved to the **Closed** state, the value of the *Active/Inactive* field becomes **Inactive**, and the action rule specifies that the parent item should close when the children items are all *Inactive*.

Tutorial: Defining Parent-Driven Actions

This example explains how to define an action that closes a child item when its parent is completed.

Prerequisites:

The following must be set up before you perform this procedure:

- *Single Relational* field called `Child`

- Application workflow named `Parent Workflow`:

  - **In Progress** state

  - **Assign** transtion that leads from the **New** state to the **In Progress** state

  - **Closed** state

  - **Close** transtion that leads from the **In Progress** state to the **Closed** state

- Application workflow named `Child Workflow` with the same steps and transitions as the **Parent Workflow**

To set up a parent-driven action:

1. Create an action that will close the child item when the value of the *Active/Inactive* field in the parent item is **Inactive**.

   a. In the **Parent** workflow, right-click the **Close** transition, and select **Show Actions**.

   b. On the **Actions** tab of the transition Property Editor, click **New**.

   c. In the **Action Wizard** that opens, select **Transition**, and then click **Next**.

   d. Select **Referenced items in relational field**.

   e. Click the **specified** link and then select **Child**.

   f. Select **Field value**.

   g. Click the **specified** link and select **Active/Inactive**.

   h. Make sure the next link reads **is**.

   i. Click the **Active** link, select **Inactive**, and then click **Next**.

   j. Select **Child:Close (In Progress->Close)**, and then click **Finish**.

2. Deploy the process app.

3. Test the process app.

   a. In SBM Work Center, submit an item into the **Parent Workflow**.

   b. Click **Assign**.

   c. Submit an item into the **Child Workflow**.

   d. Click **Assign**.

e. In the **Parent Workflow**, select the child item in the **Child** field, and then click **Close**. In the **Child** field in the **Standard Fields** section, the child item has **(Inactive)** after its name. This means the item was closed.

Tutorial: Submitting Multiple Primary Items

This example shows how to submit multiple primary items based on a single transition action. You will create a workflow and configure it so users create two items in two separate projects when a third item is submitted. In this example, the three items are linked together, but no parent/child or principal/subtask relationship is established, and the items move independently through their workflows.

Prerequisites:

The following must be set up before you perform this procedure:

- Application workflow named `Software Issues`

  - **Submit Defect** Regular transition that leads from the **Submit** state to the **New** state

  - **Submit Enhancement** Regular transition that leads from the **Submit** state to the **New** state

- Application workflow named `Product Development`

  - **Defect Post** Post transition that leads from the **Any** state back to itself

  - **Enhancement Post** Post transition that leads from the **Any** state back to itself

Consider the following information when you use actions to submit multiple primary items.

- The example in this topic uses **Post** transitions to submit multiple items. You can also use **Copy** or **Subtask** transitions.

- The example is defined within a single application. You can use **Subtask** or **Post** transitions to set up multiple submissions across applications as needed. Make sure users have submit privileges into the projects specified in the **Subtask** or **Post** transitions.

- When you implement multiple submissions in your system, consider how you want data to be mapped. By default, some field values automatically map to the new items, depending on the type of transition you use to submit new items, and whether you are submitting items into the same primary table or to a different primary table. You can specify your own mapping or use the default mapping. You can also define default field values for the **Copy**, **Post**, or **Subtask** transitions.

  For details, refer to Mapping Fields for Post, Publish, Copy, or Subtask Transitions [page 150].

To submit multiple primary items:

1. Configure the transitions in the **Software Issues** workflow:

   a. Click the **Submit Defect** transition.

b. In the transition Property Editor, on the **Options** tab, select the **Quick Transition** check box.

c. Click the **Submit Enhancement** transition.

d. In the transition Property Editor, on the **Options** tab, select the **Quick Transition** check box.

e. Save the process app.

2. Configure the transitions in the **Product Development** workflow:

a. Click the **Defect Post** transition.

b. In the transition Property Editor, on the **Options** tab, select the **Quick Transition** and **Hide transition button on state form** check boxes.

c. On the **Post Options** tab, in the **Use submit transition** list, select **Software Issues : Submit Defect**.

d. Still on the **Post Options** tab, In the **Item Link Type** list, select **2-Way, no triggers**.

e. Click the **Post Enhancement** transition.

f. In the transition Property Editor, on the **Options** tab, select the **Quick Transition** and **Hide transition button on state form** check boxes.

g. On the **Post Options** tab, in the **Use submit transition** list, select **Software Issues : Submit Enhancement**.

h. Still on the **Post Options** tab, in the **Item Link Type** list, select **2-Way, no triggers**.

3. Create actions to allow multiple submissions:

a. In the **Product Development** workflow, right-click the **Submit** transition, and then select **Show Actions**.

b. In the transition Property Editor, on the **Actions** tab, click **New**. The **Action Wizard** opens.

c. Select **Transition**, and then click **Next**.

d. Select **This Item**, and then click **Next**.

e. Select **Unconditionally**, and then click **Next**.

f. Select **Product Development:Defect Post([Any])->([Any])** and then click **Finish**.

g. In the transition Property Editor, on the **Actions** tab, click **New** again.

h. In the **Action Wizard**, repeat steps c, d, and e.

i. Select **Product Development:Enhancement Post([Any])->([Any])** and then click **Finish**.

4. Deploy the process app.

5. In SBM Application Administrator, add two subprojects to the Software Issues project.

   a. On the **Projects** tab, select **Software Issues Project**.

   b. Click **Add**.

   c. In the **Add Project** dialog box, type **Defects** in the **Project Name** box.

   d. Repeat steps b and c, except type **Enhancements** in the **Project Name** box.

6. Still in Application Administrator, edit the **Post** transitions in the Product Development project.

   a. On the **Projects** tab, select **Product Development Project**.

   b. Click **Edit**.

   c. In the **Edit Project** dialog box, click the **Transitions** tab.

   d. Select the **Defect Post** transition, and then click **Edit**.

   e. In the **Edit Transition** dialog box, select the **Override** check box above the **Post Item Project** box.

   f. Select **Defects** under **Software Issues Project**, and then click **OK**.

   g. Repeat steps d, e, and f, except select the **Enhancement Post** transition, and select **Enhancements** under **Software Issues Project**.

7. Test the process app:

   a. In SBM Work Center, submit an issue into the Product Development project. The **Attachments** and **Change History** sections indicate that when you submitted issue 000225, issues 000226 and 000227 were automatically submitted. Issue 000226 was submitted into the Defects project, and issue 000227 was submitted into the Enhancements project.

Orchestration Action Tutorials

This section contains the following tutorials:

-

Tutorial: Calling an Orchestration Workflow from Any Application

An orchestration is often created based on a particular application. If that application calls the orchestration workflow, no mapping is required between the primary table fields and the event inputs for the orchestration, because this is done automatically. However, you can call an orchestration from any application. In this case, the primary data fields from the calling application must be mapped to the event inputs. For an overview of this feature, see the orchestration links topic in the *SBM Orchestration Guide*.

To call an orchestration workflow from any application:

1. Identify the orchestration workflow you want to call.

2. Export the event definition mapped to that orchestration workflow from the **Application Links** collection in the orchestration:

    a. In App Explorer, select the event definition. The event definition editor opens.

    b. On the **General** tab of the event definition Property Editor, click **Export event definition**. The **Save As** dialog box opens.

    c. Save the event definition as an `.mtd` or a `.wsdl` file.

3. Import that event definition into the application where you want to raise the event.

    a. In App Explorer, right-click the **External Events** subheading under the **Orchestration Links** heading for the application, and then select **Add New External Event**. The Web Service Configuration Dialog Box [page 459] opens.

    b. In the **WSDL** field, navigate to the `.mtd` or `.wsdl` file you exported, and then click **OK**. The event definition is added to App Explorer under the **External Events** subheading.

4. Open the application workflow in the calling application.

5. Create an orchestration workflow action on the transition where you want to raise the event:

    a. Click the transition in the application workflow.

    b. Click the **Actions** tab in the transition Property Editor. The **Action Wizard** opens.

    c. Make sure that **Orchestration Workflow** is selected as the action type.

    d. In the rule description, make sure that **and continue executing (asynchronous)** link is selected. Click **using the local event**, change it to **an external event**, and then click **Next**.

    e. Select the affected item, complete the rest of the wizard screen, and then click **Next**.

    f. Select the condition, complete the rest of the wizard screen, and then click **Next**.

    g. Select the external event definition associated with the orchestration workflow you want to invoke.

h. Specify the object type and event type. This is used to determine which orchestration workflow to invoke when the same event definition is used by multiple orchestration workflows.

i. Click the **data mapping** link to specify the source of the data passed in the event as described in Service Mappings Dialog Box [page 459].

Chapter 22: Using Images

This section contains the following information:

- Images and Icons [page 435]

- Adding Images and Icons [page 435]

- Image Editor [page 436]

Images and Icons

You can use custom images and icons throughout an application. For example, you can add your company logo to a custom form, and have your company Web site open when users click this logo. Alternatively, you can select an image to appear as the background for a custom form.

You can also replace certain system-provided icons for primary and auxiliary tables.

Adding Images and Icons

This topic describes how to add images and icons to an application. For more information, see Images and Icons [page 435].

To add an image or icon to an application:

1. Select the application to which you want to add an image or icon.

2. In the App Explorer, right-click the **Images** heading, and select **Add New Image** or **Add New Icon**.

3. Locate the image or icon file and click **Open**. For images, the supported file types are `.bmp`, `.jpg`, `.gif`, and `.png`. For icons, only `.ico` files are supported.

4. After you add the file, the image or icon opens in the image editor. Modify its properties as needed. See Image Editor [page 436] for details.

The image or icon is now available in the following locations:

- **Icons** tab of the table Property Editor (icons only)

- Property Editor for an image control on a custom form

- Background area for a custom form (images only)

> **Note:** You can also add images or icons in the context of a specific custom form or table.

Image Editor

The image editor lets you view and modify the properties of an imported image or icon.

Element	Description
Name	By default, this is the file name of the imported image or icon without the extension. You can change this name.
Description	An area for optional comments about the image or icon.
Details	The format, width, and height of the imported image or icon in pixels. For icons, all sizes in the file are shown.
Actual size	If you select this check box, the image or icon is shown in the **Image** area in the editor at 100% of its size. If you clear this check box, the image or icon is adjusted to the best fit for the available space in this area.
	Actual size places the image at the top left, and scroll bars are present if the image or icon is larger than the available space. This option is useful when you need to see details of images.
	Best fit centers the image or icon if it is larger than the available space, and then shrinks it to fit.
	By default, this check box is cleared. Its setting (selected or cleared) is maintained during an SBM Composer session; that is, other images or icons are displayed the same way. However, when you restart SBM Composer, the check box will initially be cleared.
Image	An area that displays the image or icon.
File	The original location and file name of the imported image or icon.
Reimport	Replaces the image or icon with a different or updated file. Click this button to locate and reimport the file.
Export	Exports the image or icon. Click this button to save the image or icon to a file.

Chapter 23: Working With Styles

This section contains the following information:

About Styles

Styles specify the appearance of the following elements in an application:

- **Workflow annotations:** The style is applied to annotations you add to an application workflow. You can override default styles for annotations on the **Appearance** tab of the Ribbon.

- **Application workflows:** The style is applied to the workflow editor.

- **Decisions:** The style is applied to decisions you add to an application workflow.

- **Forms**: Default styles are applied to print forms and print form sections, state forms, and transition forms. Default styles are also applied to the controls and widgets that forms contain. The following table shows the default styles that are applied to controls and widgets.

Control Type	Control Name	Default Style
Detail	Integrations	Expander Style
Detail	SLA Widget, Social Widget	Control Style
Container	Expander, GroupBox, Panel	Expander Style
Container	Tab	Tab Style
Other	Button, ComboBox, Image, EditBox, ListBox	Control Style
Other	Text	Label Style
Other	HyperLink	HyperLink Style

Control Type	Control Name	Default Style
Widgets	Embedded Report (see exception below), HTML/Javascript, PDF, Web Page	Control Style
Widgets	Embedded Report (when the **Relational grid** report type is configured), REST Grid	Grid Style
Header	Header	Header Style
Footer	Footer	Footer Style
Left Sidebar	Left Sidebar	Left Sidebar Style
Right Sidebar	Right Sidebar	Right Sidebar Style

Note: You can override default styles on the **Design** tab of the Ribbon.

- **States:** The styles are applied to the active and inactive states in an application workflow.

- **Swimlanes:** The style is applied to swimlanes in an application workflow. You can override default styles for individual swimlanes on the **Appearance** tab of the Ribbon.

   Important: When you add a new swimlane, the background color and line settings you specified in the style editor are overridden. To apply these settings to new swimlanes, select the swimlane, and then select the desired style on the **Style** area on the **Appearance** tab.

- **Transition styles:** The styles are applied to the Exception, Normal, Optional, and Preferred transition styles in the workflow editor and in the visual workflow visible to users. You can set transition styles for individual transitions on the **Design** tab in the workflow editor. For details, refer to Transition Styles [page 112].

Customizing Styles

This topic describes how to create a new style or change an existing style.

To add or customize a style:

1. In App Explorer, select the application for which you want to customize a style.

2. Click the **Styles** heading in App Explorer.

3. Do one of the following:

- In the **Name** area in the left pane, select the style that you want to customize.

- To add a new style, right-click in the **Name** area, and then select **Add New** and the style type.

> **Note:** Not all style types can be added.

- To copy an existing style, right-click the style, and select **Duplicate**.

4. Complete the fields in the styles editor as needed. For details, refer to Styles Editor [page 439].

Styles Editor

The styles editor lets you customize and add default styles for the design elements listed in About Styles [page 437].

Element	Description	Notes
Name	The name of the style.	Read-only for system-provided styles; editable for custom styles.
Description	An area for an optional description of the style.	Applies to all styles.
Foreground/ Text	Specifies the color for text in an annotation, transition, or form control, or for an annotation border. Click the down arrow to select a standard color; click **More Colors** to specify a custom RGB value; click **Default** to restore the default color.	• Applies to all styles except print form, state form, transition form, header, footer, and left/right sidebar styles. • Can use an SBM theme value (form-related styles only).
Background 1	Specifies the background color for annotations, forms, or form components. Click the down arrow to select a standard color; click **More Colors** to specify a custom RGB value; click **Default** to restore the default color.	• Applies to all styles. • Can use an SBM theme value (form-related styles only).

Element	Description	Notes
Background 2	Specifies a secondary color. For expander containers, detail controls, and print sections, the second color is used in the header; for tabs, the second color is used to differentiate between selected and unselected tabs.	• Applies to tab and expander containers and print form sections. • Can use an SBM theme value.
Corner Radius	Specifies a corner radius to containers and detail controls. For print form sections, you must apply a background color for the radius to be visible on the form. For panels, the corner radius is visible when you apply a background image or the background color is different than the panel color. To use this option, the **Enable HTML5 features** check box must be selected. Refer to Form Options [page 486].	• Applies to tab and expander containers and print form sections. • Can use an SBM theme value.
Line style, Line width	Specifies the style and line width (in pixels) for an annotation border or transition arrow.	Applies to all styles except form-related styles.
Font	Specifies the text styling. Click the browse button to modify the text font, size, and other attributes.	• Applies to all styles except application workflow, print form, state form, transition form, header, footer, and left/right sidebar styles. • Can use an SBM theme value (form-related styles only).

Element	Description	Notes
Image	Specifies an image to use as the background for an annotation or form style. Select an existing image or select **Add** to import an image.	Applies to annotation, expander, print form, print form section, state form, tab, transition form, header, footer, left/right sidebar, and swimlane styles.
Row color, Alternate row color, Highlighted row color, Selected row color	Specifies the color for various rows in a grid-style widget. The alternate row color can differ from the row color so contiguous rows can be better identified. A highlighted row is the row over which the mouse pointer is hovering, and selected rows are the rows that are currently selected in the grid. Click the down arrow to select a standard color; click **More Colors** to specify a custom RGB value; click **Default** to restore the default color.	• Applies to the grid style. • Can use an SBM theme value.

Chapter 24: Providing Custom End-user Help

SBM Composer enables you to document your applications as you develop and maintain them. Information you provide for certain application elements, such as workflows, fields, states, and transitions, are then presented to users as they work with items.

Key Benefits

- Quickly develop end-user help content about using your individual processes

- Deliver updated content as your processes change

- Present information to users as they work with applications

- Store and deploy help content with your process apps rather than in separate repositories

- Link to external documents or help systems from within your custom application help

Creating Custom Process App Help

As you work with process apps in SBM Composer, information you provide in the **End-user help text** setting for the following elements is displayed to users:

- Fields

- Workflows

- States

- Transitions

- Decisions

- Forms

The **End-user help text** setting is located on the General properties tab for each element. Click **Edit** to open an HTML editor, which enables you to easily format text. For details, refer to Using the End-User Help Editor [page 445].
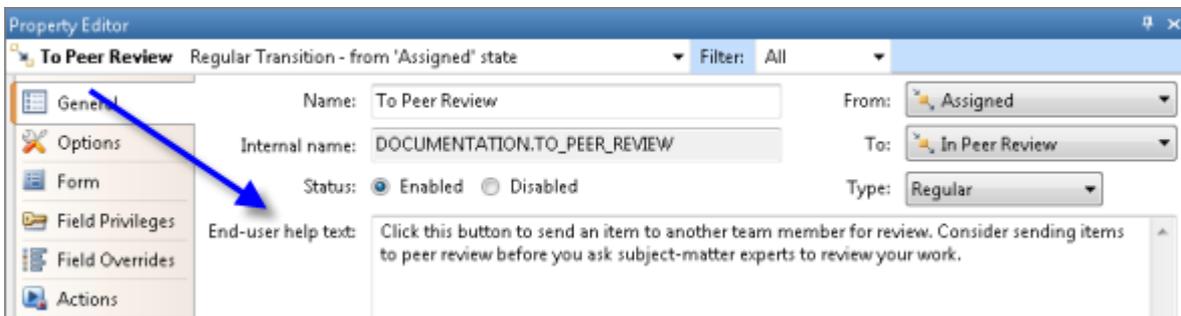
For applications, this setting is in the Application Editor. In this case, the description appears to users.

Once you deploy your process apps, users will see the information as described in Creating Custom Process App Help [page 443].

Considerations for Providing Custom Help

- You can use the HTML editor to link to external documents or help systems. Links are not active in field hover text and transitions, but they are active on the Form and Workflow help pages.

- You can add help text to the **End-user help text** setting for each field, state, decision, transition, and workflow. If you do not provide content for a field, state, transition, or decision, the element is not listed in the help.

- You cannot provide custom help text for the system Update and Delete transitions; default help text is provided, however.

- Use the **Find End-User Help Text** dialog box to search for existing help content in your process app. For details, refer to Finding End-user Help Text [page 446].

- HTML tags are not rendered in Application tab hover text.

- Privileges control some of the content users see:

  - Workflow help is visible to all users who can view a primary item and have the "View Workflow Graphically" privilege.

  - Form help and field-level help are based on users' field privileges. For example, users who do not have privileges to see fields in the Advanced section will not see help content for those fields.

Viewing Custom Help for Items

Custom help is available to users from state and transition forms for primary and auxiliary items. Custom help is available on quick forms and custom forms.

> **Note:** If you have removed the help and workflow icons from custom forms, users will not have access to Workflow or Form help.

Depending on the content you provide in SBM Composer, help is available in the following areas for items:

- **Workflow Help**

  Contains information describing the process items follow, including a graphical view. Information about each transition, state, and decision is also shown.

  Depending on administrative settings, open workflow help from any item by selecting the **Get help for this application** link on the **Actions** drop-down list, or clicking the workflow icon (⬚).

- **Transition Help**

  Hover over any transition button on a state form to view provided help content.

- **Form Help**

For transition forms, provides information about the transition and fields on the form.

For state forms, provides information about the current state, available transitions, and fields on the form.

Depending on administrative settings, open form help from any item from the **Actions** drop-down list or from the help icon.

Custom form information is also shown, if it was provided.

- **Field-Level Help**

  Hover over the name of any field that includes a dotted underline to view a shortened version of the provided help for that field.
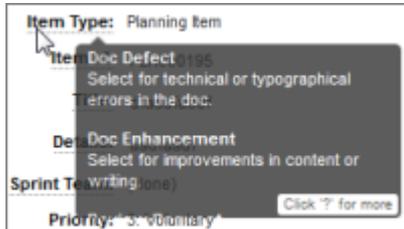
  As users move through field content on the help page, the field is highlighted on the form. When they hover over fields on the form that contain help content, the help page automatically moves to those fields.

Using the End-User Help Editor

This editor enables you to easily add, edit, and format end-user help text. To open the editor, click **Edit** on the **End-user help text** box on the **General** tab for applications, workflows, states, transitions, decisions, fields, and forms.

The End-user help editor uses standard formatting controls, including those for text, tables, links, attachments.

Tips for using the editor:

- To quickly edit help text for multiple entities, open the editor, and then search for help using the **Find End-user Help Text** dialog box from the **Home** tab on the Ribbon. As you scroll through the search results, the help text is placed in the editor. If you move to another entity before saving your changes, you are prompted to save them.

- While you can enter a large amount of content for each application entity, you might want to link to external documents instead. Use the hyperlink icon ( ) to link to external documents. Make sure the documents you link to are stored in a location visible to your end-users.

- Click **Edit Source** to view and edit HTML tags.

- To clear formatting, click the eraser icon ( ).

For details on creating custom end-user help, refer to Creating Custom Process App Help [page 443].

Finding End-user Help Text

The **Find End-user Help Text** dialog box is helpful for identifying design elements that do not have help text and for reviewing and modifying existing help text. End-user help text is entered in the **End-user help text** field on the **General** tab for:

- Primary and auxiliary table fields
- Application workflows
- Custom forms
- Transitions
- States
- Decisions

For details about how users access the help, see Creating Custom Process App Help [page 443].

This dialog box opens when you click **End-User Help** in the **Find** area on the **Home** tab of the Ribbon. The following settings are available:

Element	Description
Find what	**Contains**: Type the text you want to find, or use the drop-down list to reuse a previous search string.
	Empty: Select this option to find only design elements with no help text.
	Not empty: Select this option to find only design elements with help text.
Look in	Specify the scope of the search. You can search the entire process app, all application workflows, all tables, or all forms.
Match options	Select the **Match case** check box to restrict the search to items that match the case of the word or words in the **Find what** box. Select the **Match whole word** check box to restrict the search to items that match the entire word or words in the **Find what** box.

Part 4: Extended Functions

This section contains the following information:

Chapter 25: Using Scripts

This section contains the following information:

- About Scripts [page 449]

- Working with Scripts [page 449]

About Scripts

SBM provides two scripting languages that offer a degree of power and flexibility beyond that available through the standard administration interfaces:

- SBM ModScript – A powerful scripting language that is modeled after ChaiScript and contains extensions to support SBM. Programmers can use SBM ModScript to implement custom features in SBM. For details on SBM ModScript, refer to the *SBM ModScript Reference Guide*.

- SBM AppScript – The legacy SBM scripting language modeled after VBScript 4.0. For details on SBM AppScript, refer to the *SBM AppScript Reference Guide*.

You can associate scripts that implement custom features with transitions, notifications, and the self-registration form. You can also set up scripts that run when a user visits a special SBM URL.

Working with Scripts

This section contains the following topics:

- Creating and Editing Scripts [page 449]

- Validating Scripts [page 449]

Creating and Editing Scripts

In the script editor, you create and edit scripts.

> **Note:** To edit an existing script, you must have the script checked out.

For more information about creating and editing scripts, refer to the SBM ModScript or SBM AppScript reference sections in the online help.

Validating Scripts

The Validation Output pane below the script editor displays script validation status and results, and includes features to help troubleshoot validation errors. For more information, see Chapter 37: Troubleshooting Scripts [page 551].

In addition, the Validation Results can list scripts that failed validation when the process app containing the scripts is validated. For details, see Viewing Validation Results and Log Information [page 33].

To validate a specific script:

1. Select the script under the **Scripts** heading in App Explorer.

2. Right-click in the script editor and then select **Validate**.

To validate all scripts:

1. Select the **Validate Scripts during Process App validation** check box on the
   General Options [page 479] tab of the SBM Composer Options [page 479] dialog
   box.

2. Validate the process app containing the scripts (see Validating a Process App [page
   50]).

3. Double-click a message in the Validation Results to open the Validation Output pane
   and see details about the applicable script.

> **Note:** To optimize performance, the validation checks only those scripts that
> changed or were added since the process app was last validated.

> **Tip:** When the Validation Output pane and Validation Results are both open, you
> can arrange them by dragging, just as you can arrange editor tabs in the editor
> pane. For details, see Working with Panels [page 34].

Chapter 26: Creating Triggers

This section contains the following information:

- About Triggers [page 451]
- Working with Triggers [page 451]

About Triggers

Triggers can be set up so that they execute automatically when used with transition types that automatically link items (**Copy**, **Post**, **Subtask**, and **External Post**). You can also create triggers that users manually activate when they create a link between two items.

Create triggers in the Global Process App and reference them from the process app from which you want the trigger to fire. Triggers can be used to:

- Transition items that are selected as values in the primary relational field or in which a primary relational field is the selected value.
- Transition items that have principal or subtask relationships.

CAUTION:

⚠️ Triggers will fail when enabled for transitions that have unspecified values for required fields. For best results, provide default values for fields that are set as required for transitions in which triggers are enabled.

Working with Triggers

This section contains the following information:

- Setting up a New Trigger [page 451]
- Setting up Automatic Triggers [page 453]

Setting up a New Trigger

To set up a new trigger:

1. Create the trigger.

    a. Open the Global Process App.

    b. In App Explorer, select the **Triggers** heading.

    c. Right-click an existing trigger or an empty area in the left pane of the triggers editor.

    d. Select **Add New Trigger**.

    e. Type a name for the trigger in the **Name** field on the right side of the triggers editor.

2. Add a reference to the Global Application in the process app for which you want to create a trigger.

   a. Open the other process app.

   b. In App Explorer, right-click the **References** heading.

   c. Select **Add Application Reference** and select the Global Process App.

3. Enable the trigger for a specific state.

   a. In App Explorer, select the workflow containing the state for which you want to enable a trigger.

   b. In the workflow editor, select the state.

   c. In the Property Editor for the state, click the **Actions** tab, and then click **New**.

   d. In the **Action Wizard** that opens, select **Transition**, and then click **Next**.

   e. Click **Next** again.

   f. Select **Trigger Received**.

   g. Click **specified**, select the trigger you just created, and then click **Next**.

   h. Select the transition you want to invoke, and then click **Finish**.

4. Set the trigger to fire for a specific transition.

   a. In the workflow editor, select the transition you specified above.

   b. On the **Actions** tab of the transition Property Editor, click **New**.

   c. In the **Action Wizard** that opens, select **Trigger**, and then click **Next**.

   d. Select **Triggerable linked items**, and then click **Next**.

   e. Select the conditions under which you want the trigger to be invoked.

      - **Field Value:** Click **specified**, and select a field to be checked. Click the other links to define the condition (is, is not) and the value.

      - **Sibling Tasks** or **Subtask's Status:** Click the links to define the condition (all are, half are, most are, none are) and the value.

      - **Unconditionally:** *(No additional settings are required.)*

   f. Click **Next**.

   g. Select the trigger, and then click **Finish**.

The trigger is available to be manually activated by users when they transition linked items as specified by the trigger.

Setting up Automatic Triggers

You can create triggers that automatically transition items without intervention from the user. They can be used with **Copy**, **Post**, and **Subtask** transition types, because these transition types can automatically create primary item links.

> **Tip:** In some cases, it may be easier to use a **Subtask** transition than to manually create triggers. **Subtask** transitions create a new primary item in the specified project. The new item is linked to the original item and can be set up so that when the new item reaches an inactive state, the original item is transitioned from a waiting state to another state in the workflow. However, the transition occurs only when the new item reaches an inactive state; if you want a transition to fire at an active state, you should use a trigger.

To set up an automatic trigger:

1. In the workflow editor, add a **Copy**, **Post**, or **Subtask** transition to the workflow in which the trigger should be enabled.

2. On the **Post Options** tab of the transition Property Editor, select an item link type.

3. On the **Options** tab of the Property Editor, select **Hide transition button on state form**. This specifies that the transition be used only with triggers, automatically and with no intervention from users, and hides the transition button from users.

Chapter 27: Using Web Services

This section contains the following information:

- About Web Services [page 455]

- About the Web Services List [page 456]

- Web Service Settings [page 456]

About Web Services

You can add Web services to your process apps, making their defined operations available as actions assigned to states and transitions in your application workflows. (See Chapter 21: Using Actions [page 411] for details.) To use this feature, you should be familiar with how Web services work and with the particular operations, inputs, and outputs of the Web services you are calling.

Using the Web service operation property editor, you can map each operation's inputs and outputs to application fields. In applications, Web services add capabilities to workflows that are primarily human-centered. That is, the workflows, forms, and tables are intended to structure and capture information that comes from people. Examples include issue and defect tracking.

> **Note:** By comparison, Web services in orchestrations are organized into a workflow that is primarily machine-centered. They're arranged using control flow structures such as repeating loops, decision steps, and fault handlers. See the *SBM Orchestration Guide* for details.

Here are a few points to remember when using Web services in an application:

- Support for development efforts writing Web services is provided by Professional Services. Questions regarding use of Web services operations in orchestration processes as documented are handled by Support.

- Authentication for Web services is specified for the service's endpoint in either SBM Composer (in the Deploy Process App Dialog Box [page 74]) or Application Repository.

- Web service calls from an application are executed synchronously, which means that SBM waits for each Web service call to return (or for the specified time-out period to expire).

  > **Note:** Web service calls from an orchestration workflow are executed asynchronously, so your workflow can go on to something else after the Web service is called.

- Input data is passed to the Web service in UTF-8. Output data from the Web service is assumed to be in UTF-8.

- WSDLs may fail to import if they are not formatted correctly or if they contain functionality that is not supported by SBM.

- All errors and Web service invocation faults are recorded to the Event Viewer on the SBM Server.

> **Note:** You can also use the Web Services API to create integrations that create, read, update, and delete SBM items. See the *SBM Web Services Developer's Guide* for an overview of the services and how to use them.

About the Web Services List

The list of Web services is displayed in the editor pane when you click the **Web Services** heading in App Explorer. You can do the following:

- Select a listed Web service to view or edit it in the Web service Property Editor.

- Double-click a listed Web service to open it in the Web service editor as well.

- Click the **Find** icon on the **Home** tab of the Ribbon to search for a specific Web service in the list.

Web Service Settings

This section contains the following topics:

Web Service Editor

The Web service editor is displayed when you select a Web service in App Explorer (or in the list of Web services that is displayed when you select the **Web Services** heading in App Explorer). Use it to view (though not actually modify) information about the selected Web service.

Use the Web service Property Editor to view and edit the information about the selected Web service.

Select one of the listed operations to view their properties in the Web service operation Property Editor.

General Tab of the Web Service Property Editor

Element	Description
Name	Shows the name of the Web service.
Description	An optional description of the Web service.
WSDL	Shows the location from which the WSDL file was imported.
Reimport/ Refresh	Reimports the WSDL file and updates it if there is a later version available. **Note:** If you add an event field or change the name of an existing event field in an application's primary table, you could need to reimport the event definition WSDL file before you publish or export the process app.
Documentation	Displays the documentation for the Web service, as specified in the WSDL file.

General Tab of the Web Service Operation Property Editor

Element	Description
Name	Shows the name of the Web service operation.
Description	Shows the description of the Web service operation.
Documentation	Lists the documentation for the Web service, as specified in the WSDL file.

Tip: Click the banner (bearing the Web service's name) at the top of the Web service editor to display General Tab of the Web Service Property Editor [page 457], where you can view and edit properties of the Web service itself.

Inputs Tab of the Web Service Operation Property Editor

Although you cannot edit this information, you can see what inputs are required by the Web service operation. For example, a stock quote Web service might require an input string identifying the stock symbol.

When the Web service is included as part of an orchestration workflow, the **Data Mapping** tab of the orchestration workflow Property Editor displays similar information. Editable values, such as a password string, can be edited there.

Note: The drop-down list at the top of the Property Editor lets you view other operations in the Web service.

Element	Description
Elements	Shows the data elements defined as inputs to the selected Web service.
Type	Shows each input's data type, such as integer, Boolean, and complex types.
Vertical divider	Clicking the vertical divider to the right of the **Type** column displays additional information for the selected data element (such as its type and namespace).

Tip: Click the banner (bearing the Web service's name) at the top of the Web service editor to display General Tab of the Web Service Property Editor [page 457], where you can view and edit properties of the Web service itself.

Outputs Tab of the Web Service Operation Property Editor

Although you cannot edit any of the information, you can see what outputs are returned by the Web service operation. For example, a stock quote Web service could return an output string representing the quoted stock.

When the Web service is included as part of an orchestration workflow, the **Data Mapping** tab of the orchestration workflow Property Editor displays similar information.

Note: The drop-down list at the top of the Property Editor lets you view other operations in the Web service.

Element	Description
Elements	Shows the data elements defined as outputs from the selected Web service.
Type	Shows each output's data type, such as integer, Boolean, and complex types.
Vertical divider	Clicking the vertical divider to the right of the **Type** column displays additional information for the selected data element (such as its type and namespace).

Tip: Click the banner (bearing the Web service's name) at the top of the Web service editor to display General Tab of the Web Service Property Editor [page 457], where you can view and edit properties of the Web service itself.

Faults Tab of the Web Service Operation Property Editor

Element	Description
Fault name	Lists the name of the fault.

Element	Description
Fault message	Lists the message defined in the WSDL file for this fault.

Tip: Click the banner (bearing the Web service's name) at the top of the Web service editor to display General Tab of the Web Service Property Editor [page 457], where you can view and edit properties of the Web service itself.

Service Mappings Dialog Box

You can use this dialog box to do one of the following:

- Set up the exchange of data between a Web service (invoked by a state or transition action) and your application.

- Map application data into an event (invoked by a transition action).

Element	Description
Service data	This column lists the Web service operations and their parameters, or the event data that is sent to an asynchronous orchestration workflow.
Application data	Use this column to map a field from the application's primary table to the needed parameter for the Web service or external event. To select a field from the primary table, click in a cell to display a list of values. For example, you might map the *Owner* field from the primary table to the *UserID* field required by the Web service operation. **Note:** If you change the name of a primary table field after mapping it here, be sure to review this mapping to determine whether it needs to be reestablished using the new name.
Constant value	Use this column to "hard code" a value as the input to the corresponding parameter, rather than mapping it to an application data field. **Tip:** Specifying a constant value clears the corresponding **Application data** selection, and selecting an application data field clears the corresponding **Constant value** entry.

Web Service Configuration Dialog Box

The following table describes the configuration settings for a new Web service.

Element	Description
WSDL	The name of the Web Service Description Language (WSDL) file that defines the Web service. Browse for a file with a `.wsdl` extension (or enter a URL), and then press the Tab key or click in another field to read the file.

Element	Description
Service	Lists the services defined in the WSDL file. If only one service is defined, this field is read-only.
Port	Lists the unique names of ports for the selected service, as defined in the WSDL file. If only one port is defined, this field is read-only.
Documentation	Optional information about the services, as provided by the creator of the WSDL file. This field is read-only.
Operations	Lists the individual operations available for the selected service, as defined in the WSDL file.

Chapter 28: Using Custom Endpoints

This section contains the following information:

About Custom Endpoints

Endpoints designed in SBM Composer are *process app endpoints*, which define a place in the design that a Web service is called. This contrasts with *environment endpoints*, which represent the actual server locations and authentication information used for calling the service when the process app is used in an environment. Although process app endpoints do include server paths and authentication information, these values are used only as default values for environments where the corresponding environment endpoint does not exist. For more information about endpoints, refer to About Endpoints [page 41].

There are two types of process app endpoints, *automatic* and *custom*.

- *Automatic endpoints* are created for every SOAP-based Web service, whose Web services are associated with a WSDL file, and when you configure a REST Grid widget using an URL.

- *Custom endpoints* are process app endpoints you define explicitly by adding an item to a Custom Endpoint Library in an application or orchestration.

Custom endpoints can be used in the following ways:

- As an alternative way of configuring a REST Grid widget.

- As a source for mapping endpoint information that you want to provide to the inputs of a Web service (for example, the RESTCaller service in orchestrations).

By using a custom endpoint to configure a REST Grid, you can control which endpoints are created and how they are reused between REST Grids on different forms. You can also name them to better describe their purpose.

Each application and orchestration has its own Custom Endpoint Library in the Application Explorer. If you want to share a custom endpoint across applications and/or orchestrations, create them with the same name, default URL and authentication properties in each of the Custom Endpoint Libraries. If the name matches and other properties do not, a validation error will occur.

Some examples of how these custom endpoints can be used in SBM Composer are provided in the following topics:

Custom Endpoints for REST Calls in Forms

After defining a custom endpoint, you can use it in the REST Grid widget configuration dialog by selecting it from the list. If you have already specified a URL for the REST Service, you can convert this into a custom endpoint by choosing **<Convert to Endpoint...>**. If the URL matches an existing custom endpoint, you can select it and it will be converted to use it, preserving all trailing path and parameter information.

If you are calling a REST service from an HTML/JavaScript widget using the JavaScript API `RESTServiceWrapper` object, you can use the string builder to reference the Endpoint ID, URL, Authentication Type, User Name or Password using the `{#EndpointName.PropertyName}` syntax. For example, you can type "`{WeatherMap.URL}`" to obtain the environment endpoint URL for the runtime environment. A `SecureRESTServiceWrapper` object is also available that takes an endpoint ID for specification of authentication information.

For an example of using custom endpoints in the string builder tool, refer to Using the String Builder Tool [page 277].

For information on using custom endpoints in REST Grid widgets, refer to Using the REST Grid Widget [page 260].

For information on `RESTServiceWrapper` and `SecureRESTServiceWrapper`, refer to *SBM JavaScript Library Guide*.

Custom Endpoints in Orchestrations

In orchestrations, custom endpoints are used for mapping to the inputs of SOAP-based service steps (not for the SOAP service itself). They are important when the SOAP service implementation calls another Web service that needs to use different endpoint information in different environments. RESTCaller is one such service that calls a REST service specified by its **restUrl** argument with authentication also specified in its inputs. You can create a custom endpoint for the called REST service in the Custom Endpoint Library for the orchestration, and then map its URL and authentication properties into the RESTCaller service inputs. At runtime, the endpoint values specified for the current environment will be passed to the RESTCaller service step.

For more information, refer to *SBM Orchestration Guide*.

Adding Custom Endpoints

To add a custom endpoint:

1. In SBM Composer App Explorer, select **Extensions**.

2. Under the application or orchestration for which you want to create the endpoint, select **Custom Endpoint Library**.

3. In the list area, right-click and select **Add New Endpoint**.

4. In the **Endpoint details** section, fill out the fields as described in the following section.

   Your changes are automatically saved.

Custom Endpoint Library Dialog Box

When you select the **Custom Endpoint Library** in App Explorer, any existing custom endpoints are listed followed by the endpoint details for the currently-selected endpoint.

The following describes the **Endpoint details** section of the dialog box.

Element	Description
Name	A descriptive name for the endpoint.
Description	A description for the endpoint.
URL	The default address of the endpoint. This value will be used to create a new environment endpoint when the process app is deployed if no matching environment endpoint can be found.

Element	Description
Authentication Type	The default authentication type used for this endpoint. This value will be used to create a new environment endpoint when the process app is deployed if no matching environment endpoint can be found. Options are as follows: • **None** Select **None** if the endpoint does not require authentication. • **HTTP Basic** Select **HTTP Basic** if the endpoint points to a Web service that requires Basic Access Authentication and you want to provide the credentials (user name and password). **Important:** The **HTTP Basic** credentials set in Application Repository for an endpoint override Basic credentials that are set in orchestration workflows. For example, if username `joe` and password `pwd` are specified in the orchestration, but Application Repository specifies **HTTP Basic** with different credentials, the Application Repository credentials are used instead. If you do not want to override the credentials specified in the orchestration workflow, select **None** in Application Repository. • **NTLM** Select **NTLM** if the endpoint points to a Web service that requires Windows domain credentials. Enter the username, password, and domain name. • **Security Token** Select **Security Token** if the endpoint points to an SBM Web service or other internal endpoint. • **Oauth 2** Select OAuth 2 for RESTful calls that use OAuth 2 security. For details on obtaining and specifying the OAuth 2 token that is required, refer to the *SBM Application Repository Guide*.

Important: Process app endpoint settings are used only as defaults for new environment endpoints created during the first deployment to an environment. Once the environment endpoint has been created, changing the URL or authentication settings in SBM Composer will not overwrite values specified in the environment endpoint. Use SBM Application Repository to change environment endpoint settings.

Chapter 29: Working with Orchestrations

Orchestrations are created in SBM Composer. They are containers for design elements such as orchestration workflows.

The primary purpose of an orchestration workflow is to enable the use of Web services for coordinating the interaction between an application workflow and one or more external systems. This lets the application workflow present data that can be exchanged and modified in these external systems. Orchestration workflows can also perform modifications on the data that flows within the application workflow.

Note: Orchestration workflows are used to automate processes, while application workflows are generally manual processes for users. Typically, users never see a process controlled by an orchestration workflow, but they must interact with an application workflow.

For more information about orchestrations, see the *SBM Orchestration Guide* or the SBM Composer online help.

Chapter 30: Using REST Data Sources

This section contains the following information:

About REST Data Sources

Use REST data sources to connect to any Web service that supports the REST format.

After you configure a REST data source and deploy your process app, the data source is available for users to select when creating an external data source for use with Kanban boards in SBM Work Center. Users can reconfigure parameters or add new parameters and values as necessary as part of the feed definition process.

Follow these steps for basic REST data source configuration:

1. In SBM Composer App Explorer, select **Extensions**.

2. Under the application for which you want to create the REST data source, right-click **REST Data Sources** and select **Add New Rest Data Source**.

3. Next to the **REST URL** field, click the **Configure** button.

4. Configure the REST service, using the information in REST Service Configuration Dialog Box [page 468] for guidance.

    Your changes are automatically saved.

REST Service Configuration Dialog Box

Element	Description
URL selection / Address	In the *URL selection* field, select from available options and fill in the adjoining *Address* field as follows: • **URL:** Type or paste the URL of the REST service endpoint into the adjoining *Address* field. • *Custom Endpoint*: The pre-defined URL for the selected custom endpoint is automatically pre-filled into the adjoining *Address* field. Refer to About Custom Endpoints [page 461]. • **<Convert to Endpoint...>:** If you have selected **URL** and entered an URL, or if you have selected a custom endpoint, you can then select the **<Convert to Endpoint...>** option to create a new endpoint based on the current contents of the adjoining *Address* field. Overtype the default name, Endpoint, to give the custom endpoint a meaningful name. Any input parameters are immediately added to the **Inputs** section, and the sample data and outputs columns are automatically populated.
Add input	Adds add a new input parameter. When you add a new parameter, the URL is immediately updated.
Remove input	Removes the selected input parameter.
Get Sample Data	Populates the sample data column based on the REST service URL and any URL parameters. Use the button to initially populate the column or to refresh the sample data from the URL after you edit the data.
using security token	This option is enabled only if **Security Token** is selected in **Authentication** and relates to the **Get Sample Data** option. If selected, this passes your security token, which is retrieved based on your SBM Composer Repository credentials, to the REST service to obtain sample data.
Edit Sample Data	Use to modify sample data. For example, you may want to duplicate repeating data. You can also import data. For details, refer to Edit Sample REST Data Dialog Box [page 471].

Element	Description
Inputs	Displays input parameters that are sent to a REST service. The parameters are populated when you provide a URL.
	You can set new default values for input parameters that override ones set in a provided URL. The URL is updated after you change a parameter name or value, and then click somewhere else or press the Enter key.
	(REST Grid widget only) Runtime values for parameters are specified on the **Query** tab.
	(REST data source only) You can parameterize the URL by using square brackets `[ ]` inside the URL portion before the ?. The format is `[`*name=value*`]`. After you create an input parameter, you can add options in the Inputs table.
	For example, if you want to provide different values for "geocode" in the following URL:
	`http://maps.googleapis.com/maps/api/geocode/` `json?address=chicago&sensor=false`
	modify the URL as follows:
	`http://maps.googleapis.com/maps/api/[apitype=geocode]/` `json?address=chicago&sensor=false`
Sample data	Displays sample data from the REST service.
	Select the **Wrap** check box to wrap the sample data so it is visible without scrolling.
	Select the **Format** check box to properly format the data.
Outputs	Displays output structure received from a REST service. Formatting hints include:
	• Bold nodes can be displayed.
	• Disabled nodes cannot be displayed.
	• Repeating nodes are shown as arrays (with [] after the name).
	• The array that is selected to be displayed is shown in bold italic text.

Element	Description
Authentication	Authentication can be specified to initialize the REST service endpoint and to retrieve sample data. Options available are: • **None** • **HTTP Basic:** Provide the user name and password information to authenticate the REST service call. • **Security Token:** Pass the security token from SBM Application Repository to the REST service for authentication. This setting does not apply to sample data retrieval. Refer to the **using security token** option for information on using Security Token for sample data retrieval. • **Oauth 2:** Use Oauth 2 to authenticate the REST service call. An administrator must obtain the required access and refresh tokens for Oauth 2 and enter them in the environment endpoint definition in SBM Application Repository. For details, refer to the *SBM Application Repository Guide*. Oauth 2 cannot be used for sample data retrieval. • **NTLM:** Use Windows domain credentials to authenticate the REST service call. Provide the user name, password, and domain in SBM Application Repository. If you do not provide credentials for REST services that require authentication, you are prompted to enter this information before you can continue. **Note:** This information is for initialization of the environment endpoints to which you map this process app endpoint during deployment. At runtime, the authentication method is always retrieved from the environment endpoint settings in SBM Application Repository. Changing it here in SBM Composer has no effect on the mapped environment endpoint after the initial deployment to an environment.
Bypass browser cache	Always goes directly to the server to get results, rather than relying on the browser cache. Select this check box if your REST results change over time and you want to have the latest data, not the cached data.

Element	Description
Ignore XML prefixes	Specify whether XML namespace prefixes should be ignored. Namespace prefixes are arbitrary, can be automatically generated, and can change. The REST data source and the REST Grid widget require a constant namespace prefix, so if the prefix changes, they do not work. If you select this check box, the namespace prefix that is returned in the data from the REST service is not used, so the widget works consistently. This checkbox appears only for REST services that return XML. It does not appear for REST services that return JSON.
Rows data	Defines the array in the data set to be used for rows in the grid. The first available array is selected by default, but the list can be edited if more than one array is present. If no arrays are detected, the list includes all potential elements, and at runtime, the user must select the element that will contain repeating data.

Edit Sample REST Data Dialog Box

Use this dialog box to view and modify sample data for the REST service.

This is especially useful if SBM Composer cannot connect to the data source. You can type or paste the data into this box so you can configure the data source or widget. The data source must be available at runtime, however.

Element	Description
Sample data	Initially shows data from the REST service based on the URL and any URL parameters. You can manually modify the data as needed. If you receive an error indicating that the remote service cannot be found, you can manually paste or import sample data into this box. Do this only if you are sure that your REST URL and parameters are correct.
Import	Click to import data from an external file. Contents must be in either XML or JSON format.
Format	Select to properly format data you add or import. This option is useful for troubleshooting validation errors since the problem is typically shown where formatting is incorrect.
Line/ Column Information	Shows line and column information for the cursor location.

REST Data Sources Editor

All fields are read-only, except for the following:

- **REST URL:** Click the **Configure** button to edit the details of the REST data source, or click the **Clear** button to remove the URL and start over. See the field descriptions in the REST Service Configuration Dialog Box [page 468] for guidance.

- **Documentation URL:** Enter the URL to the documentation about the Web service, if any. Click the **View documentation** link to open the URL.

Part 5: SBM Composer Reference

This section contains the following information:

Chapter 31: Interface Reference

The following topics describe options in these areas:

File Options

Use the **File** options to manage process app files, set options, and exit SBM Composer. Depending on your theme settings, these options are available in the upper left corner from the **File** tab, the Composer Start button, or a drop-down menu. For details, refer to Theme Options [page 482].

Command	Description
New	Opens the **Create New Process App** dialog box, in which you create a process app based on a template. Templates are grouped into these categories: • Process app with basic application or orchestration elements • Process app with basic application elements and custom forms with auto-sections • Empty process app • Templates stored on your file system See Create New Process App Dialog Box [page 67].
Open	Opens the **Open Process App** dialog box, in which you can open a process app that is stored on your computer (the Local Cache) or a process app that was checked into the repository. See Open Process App Dialog Box [page 78].

Command	Description
Import and Export | Import Import and Export | Import as New	Use to import a process app blueprint. Refer to Importing a Process App [page 48].
Import and Export | Export	Validates the process app that is open, and opens the **Save As** dialog box, in which you navigate to the directory where you want to store the process app. The export feature lets you send process app blueprints to others who do not have access to your repository. It also lets you export a process app and send the blueprint file to someone else to load into Application Repository, effectively publishing the process app. See Exporting a Process App [page 54].
Import and Export | Upgrade Snapshot	You cannot promote a process app snapshot if it was created in an earlier version of SBM Composer. SBM Composer has a feature that upgrades older process app snapshots to the latest version. This command opens the **Upgrade Process App Snapshot** dialog box, which lets you do this. After the process app is upgraded and saved to a file, the upgraded snapshot file must be loaded into Application Repository (using the **Load from File** command) and then promoted from Application Repository into the SBM Server (using the **Promote** command). See Upgrading a Snapshot [page 60].
Compare | With Local File	Opens the **Compare Process App** dialog box, in which you select a process app stored on the file system of your computer to compare to the open process app. Use the comparison to determine the modifications that would be required to change from the open process app to the process app you selected for comparison. See Comparing and Merging Process Apps [page 57].

Command	Description
Compare | With Published Process App	Opens the **Select Published Process App** dialog box, in which you select a process app stored in the repository to compare to the open process app. Use the comparison to determine the modifications that would be required to change from the open process app to the process app you selected for comparison. See Comparing and Merging Process Apps [page 57].
Save	Saves a process app to the Local Cache. This lets you save your work without having to check the process app in to the repository. For detailed information about this feature, see Saving a Process App [page 50].
Repository | Get Latest of All	Opens the **Get Latest Design Elements** dialog box, with all design elements selected. Use this dialog box to get the latest version of the design elements from the repository, overwriting any changes you made since checking in the design elements. See Concurrent Development [page 44] and Get Latest Design Elements Dialog Box [page 76].
Repository | Check Out All	Opens the **Check Out Design Elements** dialog box, with all design elements selected. Design elements must be checked out of the repository before you can edit them. See Concurrent Development [page 44] and Check Out Design Elements Dialog Box [page 70].
Repository | Check In All	Opens the **Check In Design Elements** dialog box, with all design elements selected. All design elements must be checked in to the repository before you can publish a process app. See Concurrent Development [page 44] and Check In Design Elements Dialog Box [page 69].
Repository | Undo Check Out All	Opens the **Undo Check Out for Design Elements** dialog box, with all design elements selected. Use this dialog box to discard any changes you made to the selected design elements since you checked them out, and leave the elements checked in and unchanged in the repository. See Concurrent Development [page 44] and Undo Check Out Design Elements Dialog Box [page 81].

Command	Description
Repository | Refresh Status All	Refreshes the status of the selected design elements from the repository. For example, this command refreshes the repository status of design elements that were recently checked out by someone else.
Validate	Validates the open process app. A process app must be validated before it can be published.
Publish	Opens the **Publish Process App** dialog box, in which you identify the version of the process app and specify whether the process app can be deployed by others. A process app must be published before it can be deployed. **Note:** Process apps are automatically published as part of the deployment process. Use the **Publish** command if you have privileges to publish a process app, but do not have privileges to deploy a process app. See Publishing a Process App [page 51] and Publish Process App Dialog Box [page 79].
Deploy | Deploy	Opens the **Deploy Process App** dialog box, in which you select the environment to which you want to deploy, identify the version of the process app, and specify whether the process app can be deployed by others. You access the **Deploy Options** dialog box from the **Deploy Process App** dialog box. Use the **Deploy Options** dialog box to configure e-mail notifications that are sent when a process app is deployed, and map endpoints. See Deploying a Process App [page 52], Deploy Process App Dialog Box [page 74], and Deploy Options Dialog Box [page 74].
Deploy | Quick Deploy	Deploys a process app without prompting you to provide information in the **Deploy Process App** dialog box. See Deploying a Process App [page 52].
Delete Process App | From Local Cache Only	Deletes the open process app from the Local Cache.
Delete Process App | From Local Cache and Remote Repository	Deletes the open process app from both the Local Cache and the repository. Deleting a process app from the repository does not affect any environments to which it has been deployed. See Deleting a Process App [page 60].

Command	Description
Close	Closes the open process app.
SBM Composer Options	Opens the **SBM Composer Options** dialog box. See SBM Composer Options [page 479].

SBM Composer Options

Open the dialog box by clicking **SBM Composer Options** at the bottom of the **File** menu. Click a link on the left side of the dialog box to view and change the corresponding settings:

- General Options [page 479]

- Theme Options [page 482]

- Repository Options [page 483]

- Comparison Options [page 484]

- Common Log Viewer Options [page 485]

- Application Options [page 486]

- Orchestration Workflow Options [page 488]

- Security Options [page 488]

- Resources Options [page 489]

> **Tip:** You can use shortcut keys instead of the mouse to switch links in the dialog box. For details, see Options Dialog Box Shortcut Keys [page 509].

General Options

The following options are available on the **General** tab of the **Composer Options** dialog box.

Element	Description
Startup	

Element	Description
Startup action	Specify what should open when you start a new SBM Composer session. • When **Show Start Page** is selected, SBM Composer opens the Start Page. The Start Page includes links to downloadable process apps, training, support, and the community of designers. If you do not have outside Internet access, consider selecting another option to enable the session to start more quickly. • When **Open most recent process app** is selected, the process app you were working on when you closed your last SBM Composer session opens. • When **(None)** is selected, nothing opens. You can open a process app from the **Open** command or **Recent Process Apps** list on the **File** menu, or open the Start Page from the **Home** tab of the Ribbon.
Recovery	
Auto-save process app every *n* minutes	Specify whether and how often to create a record of your changes to the open process app so that it can be recovered after a power failure or other inadvertent termination of SBM Composer. When this check box is selected, SBM Composer saves your process app automatically at the specified interval.
Create backups when deploying a process app	Selected by default, this option stores the specified number of backups on your local machine. A new backup is added each time you deploy or publish a process app. Click the link to open the folder where backups are stored.
Behavior	
Mode	Using SBM Composer in SBM On-Premise/PaaS mode exposes certain features that are incompatible with deployment in an SBM On-Demand environment. Select **On-Demand** only if you need to work on a process app intended for deployment to an SBM On-Demand environment. **Tip:** Because the operating mode affects (among other things) how design elements are validated, you cannot change this setting while a process app is open. Close the process app, change the setting, and then open the process app again.

Element	Description
Include validation for On-Demand compatibility	You can also choose whether SBM Composer should warn you when validation detects any features in your process app that are incompatible with SBM On-Demand deployment. This helps you confirm that a process app you are designing (while in SBM On-Premise/PaaS mode) can eventually be deployed to an SBM On-Demand environment.
Synchronize App Explorer with current editor	Specify whether SBM Composer should update App Explorer to select the design element that is open in the main editor pane.
Auto open item when switching filters in App Explorer	Specify whether SBM Composer should expand headings in App Explorer when you click on the corresponding filter buttons at the bottom of App Explorer.
Highlight validation Errors in App Explorer	Select this check box and click the color picker to change the way validation errors are highlighted.
Highlight validation Warnings in App Explorer	Select this check box and click the color picker to change the way validation warnings are highlighted.
Open recently checked-out item in an editor	Specify whether SBM Composer automatically opens an item on check out of design elements from App Explorer.
Provide introductory warnings	SBM Composer typically offers informational messages when you perform certain operations. For example, when you save a process app, SBM Composer reminds you that your changes were not checked into the repository. Clear this check box to avoid seeing these messages. **Note:** You can dismiss an individual category of messages by selecting the **Do not show me this message again** check box in any message box where it appears.
Warn on data constraint violations	Specify whether SBM Composer warns you when you enter a default value (on the **Data Mapping** tab of the Property Editor) that does not conform to the requirements of the specified data type of the item (such as a default value of 0.5 for an integer field).

Element	Description
Select 'All Items' filter on opening process app	Specify whether SBM Composer shows all design elements in App Explorer when you open an existing process app or when you create a new process app from the **Create New Process App** dialog box.
	If this check box is cleared, the **Workflow Design** filter is shown when you open the process app using **File>Open**. If you open a process app from the **Recent Process Apps** list, the last filter you selected in that process app is shown.
Warn about mode mismatches with the repository	Specify whether SBM Composer should warn you when you open an SBM On-Premise/PaaS process app while connected to an SBM On-Demand repository, and provide the opportunity to switch modes. If you switch modes, and if the process app contains features that are not available in SBM On-Demand, the process app cannot be exported or published.
Validate Scripts during process app validation	Specify whether scripts should be validated during process app validation. To optimize performance, only those scripts that were added or modified since the last validation are validated. (Errors will continue to be displayed in the Validation Results, even if the script was previously validated.) For information about handling errors, see Chapter 37: Troubleshooting Scripts [page 551].

Theme Options

Use **Theme** options to change the appearance of SBM Composer.

Option	Description
Primary theme	The Office 2013 theme is selected by default. As you change themes, the appearance of SBM Composer changes, enabling you to preview the theme before you save your changes.
Show borders	Determines if borders are visible on various panes in the interface.
Color options	Depending on the theme you choose, you can specify canvas color, base color, or tint color.
	Tip: To restore default colors, add **Theme** options to the ribbon bar, and then select **Reset Colors**.
Show theme option on the ribbon	Select to add **Theme** options to the upper right corner of the ribbon bar. You can then click the **Theme** icon to scroll through available themes or select options from the drop-down list.

Repository Options

Connection Options

Element	Description
Local Cache Path	Specify a different location for the Local Cache, or reset it to the default location. You could specify a different location if, for example, the drive containing the default location (typically the C: drive) is low on storage space. **Tip:** You cannot change the location of the Local Cache while a process app is open.
Repository Connection Settings	Specify whether you want to work online or offline. If you select **Work Online**, enter the computer name and port for the SBM Server, and the user name and password required to access it. The default installation port is 8085. By default, a deployment from SBM Composer times out if it takes more than two minutes. If you want to change the number of minutes, type or select a number from the **Timeout (in minutes)** combo box. The maximum number of minutes is 99. **Note:** Your current connection information is displayed at the bottom right corner of the SBM Composer window. **Tip:** You cannot connect to a different repository while a process app is open.
Use Secure Connection	Specify whether a Secure Socket Layer (SSL) connection to the repository should be used. **Important:** The **Client certificate** drop-down list appears if SBM Composer is required to use client certificate authentication with SBM Application Repository. Select the client certificate that your administrator has prepared for you.
Advanced Security Setup	Use the options in the **Advanced Security Setup** dialog box to set up client certificate authentication between SBM Composer and SBM or to set up Smart Card authentication for SBM Composer. You must work with your SBM Composer to complete either setup. Administrators can refer to "Configuring Client Certificate Authentication" or "Using Smart Card Authentication with SBM Composer" in the *SBM Installation and Configuration Guide* for help.

Element	Description
Use Internal URL	Select this check box if SBM Composer is on the same side of the firewall as SBM Application Repository.
Test Connection	Confirm that the repository can be reached with the information you provided.

Behavior Options

Element	Description
Behavior	Set preferences for warnings about common actions: • Warn when deleting checked out design elements • Warn when closing a process app with new design elements • Warn when closing a process app with checked out design elements • Don't show **Check Out** dialog box for automatic check-out Specify what should happen if when you publish, the repository has newer versions of design elements than those in your Local Cache. This happens when other designers have checked in design elements since the last time you performed the **Get Latest** command. • **Automatically get latest** publishes the latest version of all design elements from the repository and overwrites the versions in your Local Cache. • **Ignore newer versions** publishes the older versions of the design elements that are in your Local Cache. • **Prompt** warns you each time you publish if newer versions are in the repository and gives you the option to publish with the latest versions or ignore them.

Comparison Options

The **Comparison** tab of the **Composer Options** dialog box lets you change the colors in the Compare Process Apps Mode [page 70] that identify the differences between the two process apps that are being compared.

Element	Description
The items are identical	The color that indicates the design elements that are the same. By default, **No color** is selected.

Element	Description
The items are different	The color that indicates the design elements that are different. The default color is orange.
The item exists only in the open Process App	The color that identifies design elements that exist only in the process app that is open. The default color is yellow.
The item exists only in the compared Process App	The color that identifies design elements that are only in the process app that is being compared to the open process app. The default color is green.

Common Log Viewer Options

Element	Description
Delete local copies... after *N* days	When you select the Details tab of the log viewer [page 540] (or click the **Download messages** button on that tab), SBM Composer copies log messages for the workflow you are viewing (or for the entire process app) to your computer, so you can still see them when you work offline. To keep your computer from filling up with these copied messages, SBM Composer automatically deletes any local copies that have been stored for longer than what you specify here.
Clear local message cache now	Deletes the local copies of all the log messages currently stored on your computer. Message caching resumes immediately.
Maximum message count displayed in viewer	The maximum number of messages to be displayed on the **Details** tab of the Common Log Viewer. Valid values are 200 to 20000, inclusive.
Message retrieval batch size	Messages are retrieved from the server in batches of the size you specify until all messages are retrieved. Valid values are 10 to 500, inclusive.
Maximum message length	The maximum size of a message. If the message is larger than this, it is truncated and `[TRUNCATED]` is appended to it. The default length is 16,384 (16 KB of data). The maximum length is 1,048,576 (1 MB of data).
Defaults	Restores the message count to 1000, the batch size to 50, and the message length to 16384.

CAUTION:

⚠️   Increasing the message count, batch size, and message length could lower the performance of the Common Log Viewer.

Application Options

Form Options

Element	Description
Editor	Specify preferences for behavior in the form editor: • Show insertion point during drag-and-drop operation even when invalid • Show cell handles when control is selected • Preserve form zoom level and position
New form Size Guide default settings	Specify default settings for the size of new forms. The values you specify are displayed in the Property Editor for the new form.
Preview	Select the **Enable JavaScripts** check box to specify whether JavaScripts that are included in forms are executed or suppressed during form preview. By default, this check box is selected. 📝 **Note:** Some scripts are too complex to show during form preview; you can override this setting for individual forms on the Form Preview Dialog Box [page 317].
Enable HTML5 features	Select the **Enable HTML5 features** option to enable advanced HTML features. For example, this check box enables or disable the Rich Text Editor for formatting default values for *Text* fields. Choose to preview the form in the Classic or Work Center style. This setting also determines the appearance of preview mode for the "Classic" style. When this check box is selected, form preview uses a modern theme. When this check box is cleared, form preview uses a legacy theme.
External Forms	Specify whether new custom forms can be based on tables that are in another application in the same process app or in a referenced application in another process app.

HTML and Script Options

Element	Description
Font	Specify the name and size of the font used in the Javascript editor and Script editor, in the **Message Detail** dialog box in the Common Log Viewer, and in the **Content** tab of some form widgets (for example, the HTML/JavaScript widget).
Syntax Highlighting	The current color and font style of the syntax elements are shown in the **Preview** box. If these values have never been changed, or if default values were restored, the default color and font style are shown. • To change the syntax highlighting, select a syntax element and adjust the color and font style. The changes you make are immediately reflected in the **Preview** box. • If you change your mind, click **Restore selected style to default** to change the selected syntax element back to the default color and font style, or click **Restore all styles to default** to change all syntax elements back to the default color and font style.

Table Options

Element	Description
Editor	Specify preferences for behavior of the table editor: • Immediately rename new fields
Warnings	Specify preferences for warnings related to tables: • Warn when allowing selection by group for *Multi-User* fields

Workflow Options

Element	Description
Editor	Specify preferences for behavior of the workflow editor: • Immediately rename new states • Immediately rename new transitions • Preserve workflow zoom level and position. See Using the Workflow Editor [page 107] for more information. • Show warning when derived workflow will be affected. See Application Workflow Overview [page 99] for more information.

Event Options

Element	Description
Warn when exporting unlocked event	Specify whether a warning should be displayed when you export an event definition that is not locked. An event definition must be locked if it is exported and then used by an asynchronous orchestration workflow in another process app.

Orchestration Workflow Options

Element	Description
Editor	Specify preferences for behavior of the workflow editor: • Preserve workflow zoom level and position. By default, SBM Composer remembers the zoom level and position of the orchestration workflow within its editor after you close the process app. To override this, clear this check box.

Security Options

The following options are available on the **Security** tab of the **Composer Options** dialog box.

Element	Description
Enable FIPS 140-2 compliance	Select this option for machines that require FIPS 140-2 compliance. This ensures that only secure and approved cryptographic algorithms and data processing standards are used by the system.

> **Important:** After you select **Enable FIPS 140-2 compliance** and click **OK** in SBM Composer, the Windows **Local Policy** | **Security Option** for FIPS is enabled, which affects the entire Windows operating system. This means other applications that are running on this machine can be impacted by this change.
>
> Check with your system administrator before enabling FIPS 140-2 compliance in SBM Composer.

Resources Options

The following options are available on the **Resources** tab of the **Composer Options** dialog box.

Element	Description
About	Displays the software version, and provides access to computer system information. This information could be requested by Support.
Community	Provides access to SBM users.
Logging Level	Enables you to set the level of system logging: Off, Errors (default), Information, Exception details, or Trace.
Log file	Displays the log of system messages, which could be requested by Support.

Import and Export Menu

The following import and export commands are available from the **File** menu.

Import

The **Import** command opens the **Import Process App Blueprint** dialog box so you can navigate to a process app blueprint (.msd) file or template (.mst) file.

For details, refer to Importing a Process App [page 48].

Import as New

The **Import as New** command opens the **Create New Process App from File** dialog box so you can navigate to a process app blueprint (.msd) file, and then create a new process app based on an .msd file. The newly imported process app has a unique identity.

For details, refer to Importing a Process App [page 48].

Export

The **Export** command validates the process app that is open, and then opens the **Save As** dialog box. You can navigate to the directory where you want to store the process app.

The export feature lets you send process app blueprints to others who do not have access to your repository, and lets you export a process app and send the blueprint file to someone else to load into Application Repository, effectively publishing the process app.

For details, refer tp Exporting a Process App [page 54].

Upgrade Snapshot

If you attempt to load or promote a snapshot created in an earlier version SBM into Application Repository, you may receive a warning that the snapshot needs to be loaded and upgraded in SBM Composer. In SBM Composer, the **Upgrade Snapshot** command opens the **Upgrade Process App Snapshot** dialog box, which lets you upgrade the snapshot to the latest version.

For more information about upgrading snapshots, see Upgrading a Snapshot [page 60].

Compare Menu

The following comparison commands are available from the **File** menu.

With Local File

The **With Local File** command opens the **Compare Process App** dialog box, in which you select a process app stored in the file system of your computer to compare to the open process app. Use the comparison to determine the modifications that would be required to change from the open process app to the process app you selected for comparison.

With Published Process App

The **With Published Process App** command opens the **Select Published Process App** dialog box, in which you select a process app stored in the repository to compare to the open process app. Use the comparison to determine the modifications that would be required to change from the open process app to the process app you selected for comparison.

With Recently Compared Local File

The **With Recently Compared Local File** command compares with the last comparison process app that is stored on your computer. This is the last process app that was compared to any process app (not just to the open process app).

For more information about the comparison commands, see the following topics:

Comparing and Merging Process Apps [page 57]

Compare Process Apps Mode [page 70]

Select Published Process App Dialog Box [page 80]

Repository Menu

The following repository commands are available from the **File** menu.

Get Latest of All

The **Get Latest of All** command opens the **Get Latest Design Elements** dialog box, with all design elements selected. Use this dialog box to get the latest version of the design elements from the repository, overwriting any changes you made since checking in the design elements.

> **Tip:** There is an option on the Repository Options [page 483] tab of the **Composer Options** dialog box that automatically gets the latest version of design elements before you publish.

Check Out All

The **Check Out All** command opens the **Check Out Design Elements** dialog box, with all design elements selected. Design elements must be checked out of the repository before you can edit them.

Check In All

The **Check In All** command opens the **Check In Design Elements** dialog box, with all design elements selected. All design elements must be checked into the repository before a process app can be published.

Undo Check Out All

The **Undo Check Out All** command opens the **Undo Check Out for Design Elements** dialog box, with all design elements selected. Use this dialog box to discard any changes you made to the selected design elements since you checked them out, and leave the design elements checked in and unchanged in the repository.

Refresh Status All

The **Refresh Status All** command refreshes the status of the selected design elements from the repository. For example, this command refreshes the repository status of design elements that were recently checked out by someone else.

For more information about the repository commands, see the following topics:

- Concurrent Development [page 44]

- Get Latest Design Elements Dialog Box [page 76]

- Check Out Design Elements Dialog Box [page 70]

- Check In Design Elements Dialog Box [page 69]

- Undo Check Out Design Elements Dialog Box [page 81]

- Refresh Status of Design Elements Dialog Box [page 81]

Deploy Menu

The following deployment commands are available from the **File** menu.

Deploy

The **Deploy** command opens the **Deploy Process App** dialog box, in which you select the environment to which you want to deploy, identify the version of the process app, and specify whether the process app can be deployed by others.

You access the **Deploy Options** dialog box from the **Deploy Process App** dialog box. The **Deploy Options** dialog box lets you configure e-mail notifications that are sent when a process app is deployed, and map endpoints.

Quick Deploy

The **Quick Deploy** command lets you deploy a process app without being prompted to provide the information that is specified in the **Deploy Process App** dialog box. If this is the first time you are deploying a process app, it is recommended that you use the **Deploy** command, not the **Quick Deploy** command. The **Deploy** command lets you see the default settings and change them, if necessary.

The **Quick Deploy** command also lets you decide whether you want to create a new version of the process app and whether you want to leave design parts checked in after the deployment completes.

For more information about the deployment commands, see the following topics:

Delete Menu

The following deletion commands are available from the **File** menu.

From Local Cache Only

The **From Local Cache Only** command lets you delete the open process app from the Local Cache.

From Local Cache and Remote Repository

The **From Local Cache and Remote Repository** command lets you delete the open process app from both the Local Cache and the repository. Deleting a process app from the repository does not affect any environments to which it has been deployed.

For more information about the deletion commands, see Deleting a Process App [page 60].

Quick Access Toolbar

The quick access toolbar, at the top of the SBM Composer window, contains commonly used commands.

Command	Description
Save locally	Saves the process app to the Local Cache (on a local or network drive).
Undo	Returns the process app to its condition before the last operation (or multiple operations if you click the down arrow and select them).
Redo	Returns the process app to its condition after the last operation (or multiple operations if you click the down arrow and select them) that was undone. Each redo undoes the last undo.
Navigate Backward	Displays a previous view in the current editing session. You can browse the history of previous views by clicking the down arrow.
Navigate Forward	Displays a subsequent view in the current editing session. You can browse the history of subsequent views by clicking the down arrow. (This command is only available if you are viewing a previous view through the **Navigate Backward** command.)

Command	Description
Validate open Process App	Validates the open process app to determine whether it is ready to publish to Application Repository.
Publish open Process App	Publishes the open process app to Application Repository.
Deploy open Process App	Deploys the open process app (if you have an environment set up in Application Repository to allow deployment from SBM Composer).
Deploy open Process App using previous settings	Deploys the open process app (if you have an environment set up in Application Repository to allow deployment from SBM Composer) using settings previously specified in the **Deploy Process App** dialog box.
Find in open process app	Opens the **Find Items** dialog box to locate specific design elements in the open process app.

Ribbon

The commands in the Ribbon are grouped in tabs:

- Home Tab of the Ribbon [page 493]
- Deployment Tab of the Ribbon [page 496]
- Design Tab of the Ribbon [page 497]
- Appearance Tab of the Ribbon [page 502]
- Script Tab of the Ribbon [page 503]

Home Tab of the Ribbon

New

Command	Description
Component	Add a new application, orchestration, or application reference to the open process app.
Element	Add one of the listed elements to an application.

Edit

Command	Description
Paste	Place the content of the Windows Clipboard.
Cut	Move the selected item to the Clipboard.
Copy	Move a copy of the selected item to the Clipboard.
Duplicate	Move a copy of the selected item to the Clipboard, and place the content of the Clipboard.

Repository

Command	Description
Get Latest	Get a read-only copy of the selected design elements from the repository. This does not create a new version, and leaves the design elements available for check out.
Check Out	Retrieve an editable version of the selected design elements from the repository. This creates a new version, and locks the design elements so they cannot be checked out and changed by another user.
Check In	Store the selected design elements and any changes you made in the repository. This makes the new version available for check out.
Refresh Status	Update the repository status of the open process app. For example, checked out by you, checked out by another, and so on.
Undo Check Out	Cancel the check out of the selected design elements. This leaves them checked in, unchanged, and available for check out from the repository.
Version History	View a list of the versions of the selected design elements that were checked in to the repository.

Zoom

Command	Description
Zoom	Select **Fit to Window** or a percentage of the normal size to see more or less of the form or workflow in the editor pane.
100%	Display the form or workflow in the editor pane at its normal size.

Print

Command	Description
Print	Print the design element in the editor pane.
Save As Image	Save an image of the form or workflow in the editor pane as PNG, JPEG, or other common image format.

Common Views

Use the options in this area to toggle the display of various types of information at the center of the SBM Composer window, below the editor pane. When more than one option is selected, each view is displayed as a tab.

Command	Description
Property Editor	Toggle the display of the Property Editor. This option is selected by default. See Using the Property Editors [page 32] for more information.
Validation Results	Toggle the display of information generated when the process app is validated. See Validating a Process App [page 50] for more information.
Common Log Viewer	Toggle display of information generated in a log file while a process app is running. See About the Common Log Viewer [page 537] for more information.
Activity Log	Toggle the display of other information generated when major operations, such as deployment and other repository tasks are performed in SBM Composer.

Find

Command	Description
Find Items	Opens the Find Items Dialog Box [page 504].
End-User Help	Opens the Finding End-user Help Text [page 446].
Where Used	Opens the Where Used Dialog Box [page 81].

Resources

Command	Description
Company Logo	Opens the http://www.microfocus.com Web site.

Command	Description
Start Page	Opens the SBM Composer Start Page, from which you can see procedures to help you get started with SBM Composer. The Start Page also contains other useful resources that you can view or download.

Deployment Tab of the Ribbon

Deployment

Command	Description
Validate	Validate the open process app to determine whether it is ready to publish to Application Repository.
Publish	Package a process app and make it available in Application Repository. Publishing makes a process app available for deployment.
Deploy	Deploy the open process app (if you have an environment set up in Application Repository to allow deployment from SBM Composer). This command opens the **Deploy Process App** dialog box, in which you specify deployment settings.

Quick Deploy

Command or Option	Description
Quick Deploy	Deploy a process app with the settings already specified in the **Deploy Process App** dialog box.
Create Versions of Process App Elements	If this option is selected, check in design elements during the deployment and create process app versions in Application Repository. **Note:** You cannot clear this check box unless **Enable Development Deployment** is selected in the **Composer** list in the **Edit Environment** dialog box in SBM Application Repository for the environment to which you are deploying.
Deploy To	Lets you select the environment to which you want to deploy.
Keep Design Elements Checked Out	If this option is selected, check out design elements after the deployment completes. This only applies to design elements that were checked out before the deployment began.

Launch

Command or Option	Description
Launch SBM Work Center When Deployed	If this option is selected, open SBM Work Center after the deployment completes.
Startup Application	Open SBM Work Center to the selected application in the process app.
SBM Work Center	Open SBM Work Center to preview your process app. (You must have deployed your process app from SBM Composer at least once.)
Application Repository	Open Application Repository.

Note: Internet Explorer presents a security warning when you click SBM Work Center or **Application Repository**. You can safely click the message and trust the blocked content.

Design Tab of the Ribbon

Form Tools

Preview

Command	Description
Preview	View a mockup of the form as would appear to someone using your application. Use the controls at the top of the preview window to view the form as it would appear in a different workflow, state, or transition, or role.

Form Layout

Command	Description
Expander Layout	Uses a single column with expandable and collapsible headings for privilege sections.
Tab Layout	Uses a separate tab for each privilege section.
Mixed Layout	Uses a both expanders and tabs.

Command	Description
Portal Layout	Uses a "best fit" of privilege sections into a compact area. This layout is suitable to use as a Web page.
Help Layout	Provides an area for each section in which you can type descriptive text to help the user with this form.

Note: The form layouts provide starting points for form designs that you can modify as you want.

Style

Command	Description
Print Form Style	Set the form to look like a form that is optimized for printing.
State Form Style	Set the form to look like a read-only state form.
Transition Form Style	Set the form to look like an editable transition form.
Control Style	Set the selected form component to look like a control (typically black).
HyperLink Style	Set the selected form component to look like a hyperlink (typically blue with underline).
Label Style	Set the selected form component to look like a label (typically blue).
Expander Style	Set the selected form component to be expandable and collapsible.
Print Section Style	Set the selected form component to look like it cannot be expanded or collapsed.
Tab Style	Set the selected form component to be a tab that users can click to bring it into focus.

Font

Command	Description
Font name, size, attributes	Set the text attributes of the selected form component. **Note:** Not all fonts are supported in each Web browser. For example, some symbolic fonts such as Wingdings and Symbol are not supported in all browsers. Be sure to verify that the font you select is supported in the browser that is used by the majority of your users.

Background

Command	Description
Color 1	Apply a background color to the selected components.
Color 2	Apply a second background color for these controls: • Tab containers • Expander containers • Section containers (print forms) • Detail controls For expander containers, detail controls, and print form sections, the second color is used in the header. For tabs, use the second color to differentiate between selected and unselected tabs. To use this option, the **Enable HTML5 features** check box must be selected. Refer to Form Options [page 486].
Image	Select an imported image to use as the background for the form or the selected form component.
Image Icons	Select to have the imported image (if any) repeat horizontally or vertically across the background on the form or container.
Corner Radius	Apply a corner radius to containers, detail controls, and print form sections. For print form sections, you must apply a background color for the radius to be visible on the form. For panels, the corner radius is visible when you apply a background image or the background color is different than the panel color. To use this option, the **Enable HTML5 features** check box must be selected. Refer to Form Options [page 486].

Alignment

Command	Description
(Tool Icons)	Align rows and columns vertically (top, middle, bottom, full-height) and horizontally (left, center, right, full-width). You also use these tools to modify the alignment of a selected form widget. See Using Form Widgets [page 255] for related information.

Size

Command	Description
Width, Height	Modify the size of rows and columns on the form you are editing. You also use these controls to modify the dimensions and characteristics of a selected form widget. See Using Form Widgets [page 255] for related information.

Row & Column

Command	Description
(Tool Icons)	Add a row above or below the selected row on the form being edited, add a column to the left or right of the selected column, or delete the selected row or column.

Workflow Tools

Layout

Command	Description
Auto Arrange	Let SBM Composer create a compact, linear arrangement of the states and transitions in the workflow.
Re-inherit from Parent	*(Sub-workflows only)* Restore the arrangement of states and transitions defined for the parent workflow, as closely as possible.

View Mode

Command	Description
Presentation	Shows icons on states and transitions that represent all design elements that have relationships with the states and transitions. Hover over the icon for a custom form or orchestration workflow to see a thumbnail image of the form or workflow. Click or double-click the icon to view or edit the properties of the state or transition related to the design element.
Relationships	Shows the Relationships bar. For more information, see About the Relationships Bar [page 101].
Properties	Shows icons on states and transitions to indicate that certain properties are set. Click or double-click the icons to view or edit the corresponding properties in the Property Editor.

Show/Hide

Command	Description
Unreachable Paths	Toggle the display of states and transitions that are not connected to the **Submit** state on the workflow being edited.
Disabled Transitions	Toggle the display of transitions with **Disabled** selected in the transition Property Editor.
Deleted States	Toggle the display of states you deleted from the workflow after the workflow was published at least once. **Note:** States you added and deleted since the last publication are not affected by this setting.
Inheritance	Toggle the display of double lines in a sub-workflow to indicate states and transitions that were inherited from the parent workflow.
Annotations	Toggle the display of text annotations you can add to states, transitions, and workflows.
Labels	Toggle the display of the text labels on transitions.
Grid	Toggle the display of the blue "graph paper" background.

Transition Style

Command	Description
Preferred, Normal, Optional, Exception	Apply a predefined color or line style to provide a visual cue about the nature of the selected transition. Your choice does not affect the appearance or behavior of the transition on the associated form.

Swimlanes

Command	Description
Horizontal, Vertical	Specify whether you want horizontal or vertical swimlanes. See About Swimlanes [page 102] for information about swimlanes.

Appearance Tab of the Ribbon

Style

Command	Description
Style	Set the style of the annotation or swimlane.

Line

Command	Description
Style, width	Set the style and width of the line around the annotation or swimlane.

Font

Command	Description
Font name, size, attributes	Set the text attributes of the selected annotation or swimlane.

Background

Command	Description
Color, image, image orientation	Use a single background color or an imported image to use as the background for the annotation or swimlane. Specify whether you want the image to be repeated vertically, horizontally, or both.

Alignment

Command	Description
(Tool icons)	Align the annotation text or the swimlane label vertically (top, middle, bottom, full-height) and horizontally (left, center, right, full-width).

Script Tab of the Ribbon

Validation

Command	Description
Validate Script	Click this icon to validate the selected script. Results appear in the Validation Output pane below the script editor.

Font

Command	Description
Font name, size	Use these controls to specify how you want to view the script in the editor.

Find Options

- Find Dialog Box [page 503]

- Find Items Dialog Box [page 504]

- Find Results Pane [page 504]

Find Dialog Box

This dialog box is used to find items in a specific area of SBM Composer. To extend your search to a broader area of SBM Composer, use the Find Items Dialog Box [page 504].

Element	Description
Find what	Type the name of the element you want to find, or use the drop-down list to reuse a previous search string.
Find options	**Match case.** Select this check box to restrict the search to items that match the case of the word or words in the **Find what** box.

Find Items Dialog Box

Use this dialog box to locate specific design elements in the process app context that you specify.

Element	Description
Find items containing	Type the name of the element you want to find, or use the drop-down list to reuse a previous search string.
Look in	Specify whether you want to limit your search to the current editor, the current application, the entire process app, or the entire process app and referenced applications.
Which types	The choices in this section depend on the scope of the search. Select the check box for the design element or design elements you want to search, or select the **Select all** check box. To start over, click the **Unselect all** check box.
Find options	Select any of the options to further restrict this search.

> **Note:** You can also use the Find Dialog Box [page 503] to locate design elements. It has less options than the **Find Items** dialog box.

Find Where Used Dialog Box

Use this dialog box to refine the search for items using the "where used" feature.

Element	Description
Search for	Specify whether you want to search for the selected item or all items of the same type as the selected item. The results are provided for each item in the Where Used dialog box, along with the number of times an item is used. This enables you to easily delete items that have zero occurrences.
Look in	Specify whether you want to limit your search to the current application or orchestration, the entire process app, or the entire process app and referenced applications.
Which types	The choices in this section depend on the scope of the search. Select the check box for the design element or design elements you want to search, or select the **Select all** check box. To start over, click the **Unselect all** check box.

Find Results Pane

This pane shows the results of a search from the Find Items Dialog Box [page 504] or Finding End-user Help Text [page 446] dialog box.

This pane occupies the same space as the Property Editor, below the editor pane. When multiple panes are open in this area, arrange them the same way as you arrange editor tabs in the editor pane.

If you double-click an item in this pane:

- The applicable editor opens, and the item is selected in the editor.

- The Property Editor for the item opens.

- The item is displayed in blue to indicate that it has been "visited."

If you right-click an item in this pane, the context menu includes the following commands:

- **Go to Location** is the same as double-clicking the item (see above).

- **Remove** deletes the item from the pane.

- **Clear** removes all items from the pane.

- **Show Details** expands the item and provides additional information. For example, when the search results are from the **Find End-User Help Text** dialog box, any existing help text for the item is displayed.

For a complete list of context menu commands, see Context Menu Options for Lists and Tables [page 512].

Shortcut Keys

You can use shortcut keys instead of the mouse when you work in SBM Composer. These shortcut keys work on standard keyboards only.

- Ribbon Operations [page 506]

- Global Shortcut Keys [page 506]

- Local Shortcut Keys [page 507]

- Auto-Hide and Focus Shortcut Keys [page 508]

- Options Dialog Box Shortcut Keys [page 509]

- Selection Field Shortcut Keys [page 509]

- Form Shortcut Keys [page 510]

- Palette Shortcut Keys [page 510]

Ribbon Operations

You do not have to remember shortcut keys that perform Ribbon operations, because when you press the Alt key, they are highlighted. For example, to open the Composer menu, press the Alt key to highlight the keys, release the Alt key, and then press the M key to open the menu.

> **Note:** If you already know a shortcut key, you can simply press and release Alt, and then press *key* to perform its associated operation. It is important to release the Alt key before pressing *key*, because there are shortcut keys in the editor pane and Property Editors that use the same letter as those in the Ribbon, but require you to press Alt and *key* at the same time. The editor pane and Property Editor shortcut keys take precedence over the Ribbon shortcut keys, so you could get unexpected results if you do not press the Ribbon keys sequentially.

Global Shortcut Keys

The following table lists shortcut keys you can use from anywhere in SBM Composer.

Operation	Shortcut
Open the Composer menu.	Alt, M
Open the **Create New Process App** dialog box.	Ctrl+N
Open the **Open Process App** dialog box.	Ctrl+O
Open the **Import Process App Blueprint** dialog box.	Ctrl+Shift+I
Open the **Create New Process App from File** dialog box.	Ctrl+Shift+I
Open a dialog box to export a process app to a file.	Ctrl+Shift+E
Save the open process app.	Alt, S *or* Ctrl+S
Undo the last operation.	Ctrl+Z
Redo the last operation you performed with the "Undo" operation.	Ctrl+Y
Navigate backward.	Alt, 8
Navigate forward.	Alt, 9
Validate process app.	Alt, V *or* Ctrl+Shift+V

Operation	Shortcut
Start a Publish operation.	Alt, P *or* Ctrl+Shift+P
Start a Deploy operation.	Alt, D *or* Ctrl+Shift+D
Start a Quick Deploy operation.	Alt, Q *or* Ctrl+Shift+Q
Open the **Find Items** dialog box. This dialog box lets you search for all matching items in an editor pane, application, or process app.	Alt, F *or* Ctrl+Shift+F
Zoom in.	Ctrl +
Zoom out.	Ctrl -
Open SBM Work Center.	Ctrl+Shift+W
Open Application Repository.	Ctrl+Shift+M
Open or close Validation Results.	Ctrl+Shift+H
Open or close Common Log Viewer.	Ctrl+Shift+L
Open or close Activity Log.	Ctrl+Shift+G
Open or close Property Editor.	Ctrl+Shift+R

Local Shortcut Keys

The following table lists shortcut keys that you can use only when you have certain parts of SBM Composer in focus. For example, the shortcut key for the "find" operation is available from App Explorer or an editor.

Operation	Shortcut
Open the **Find** dialog box. This dialog box lets you search for matching items one at a time in App Explorer or in a list in the editor pane.	Ctrl+F
Find the next occurrence of an item you specified in the **Find** dialog box.	F3

Operation	Shortcut
Toggle the presence of the **Look for** area in a list in the editor pane. This area lets you filter the results of a "find" operation.	Ctrl+I
Remove waypoints (small circles on a transition) so the transition follows the line from the center of one state to the center of another state. This gives the workflow a cleaner look.	Shift+Delete
Show the **Actions** tab on the Property Editor for a transition or state.	Alt+3, A *or* Ctrl+Shift+3, A
Show the **Forms** tab on the Property Editor for a workflow; show the Form tab on the Property Editor of a transition or state.	Alt+3, F *or* Ctrl+Shift+3, F
Show the **Field Privileges** tab on the Property Editor for a workflow, transition, or state.	Alt+3, R *or* Ctrl+Shift+3, R
Show the **Field Overrides** tab on the Property Editor for a workflow, transition, or state.	Alt+3, V *or* Ctrl+Shift+3, V

Auto-Hide and Focus Shortcut Keys

The following table lists shortcut keys you can use to bring focus to major areas of the SBM Composer user interface. If the area in focus is in the "Auto-Hide" mode (that is, the pushpin icon at the top right of the area is unpinned or pointing horizontally), you can use these shortcut keys to hide or show the areas.

Operation	Shortcut
Activate and bring focus to hidden App Explorer.	Ctrl+1
Bring focus to editor pane.	Ctrl+2
Activate and bring focus to hidden Property Editor.	Ctrl+3
Activate and bring focus to hidden palette.	Ctrl+4
Activate and bring focus to hidden Message List.	Ctrl+5
Hide App Explorer, Property Editor, or palette.	Esc

Options Dialog Box Shortcut Keys

The following table lists shortcut keys you can use to switch tabs in the **Options** dialog box. When you hover over the tab name, the shortcut key for the associated tab is shown in a tooltip.

Operation	Shortcut
Switch to **General** tab.	Ctrl+G
Switch to **Theme** tab.	Ctrl+M
Switch to repository **Connection** tab.	Ctrl+R
Switch to repository **Behavior** tab.	Ctrl+B
Switch to **Comparison** tab.	Ctrl+N
Switch to **Common Log Viewer** tab.	Ctrl+L
Switch to application **Form** tab.	Ctrl+F
Switch to application **HTML and Script** tab.	Ctrl+H
Switch to application **Table** tab.	Ctrl+T
Switch to application **Workflow** tab.	Ctrl+W
Switch to application **Event** tab.	Ctrl+E
Switch to orchestration **Workflow** tab.	Ctrl+O
Switch to **Resources** tab.	Ctrl+S

Selection Field Shortcut Keys

The following table lists shortcut keys you can use to edit and navigate selections in *Single Selection* and *Multi-Selection* fields.

Operation	Shortcut
Edit a selected value in the **Value** column.	Enter *or* F2
Apply changes to a selected value.	Enter
Discard changes to a selected value.	Esc

Operation	Shortcut
Add a new value.	Insert
Delete a selected value.	Delete
Open the drop-down list in the **Status** column.	Alt+Down
Select a value in the **Status** column.	Up, Down, Left, Right
Close the drop-down list in the **Status** column.	Esc
Change the value of the **Weight** column.	Ctrl+Up, Ctrl+Down

Form Shortcut Keys

The following table lists shortcut keys you can use in the form preview and form editor.

Operation	Shortcut
Open form preview mode.	F5
Navigate between cells in the form editor (the first cell must be selected).	Up, Down, Left, Right
Select tab on Tab container control.	Left, Right
Select top left cell of a focused form or container control.	Enter
Select parent control, tab, or cell.	Esc
Expand a container control.	+ (in Num Lock mode)
Collapse a container control.	- (in Num Lock mode)

Palette Shortcut Keys

The following table lists shortcut keys you can use from any palette (**Workflow Palette**, **Form Palette**, and so on).

Operation	Shortcut
Display shortcut keys for the selected palette.	Alt+4 *or* Shift+Ctrl+4

Operation	Shortcut
Select an item in the palette.	Select the palette section and then press Up or Down. (To select the palette, press Ctrl+4.)
Add item from a palette to the editor pane.	Select an appropriate cell in the editor (for forms), navigate to the item in the palette, and then press Enter.
Expand a palette section.	+ (in Num Lock mode)
Collapse a palette section.	- (in Num Lock mode)
Move to the top of the palette.	Home
Move to the bottom of the palette.	End
Move to a specific item in the palette.	Select the palette section and then press the first letter of the item.
Move to the top of the palette.	PgUp (repeat as needed)
Move to the bottom of the palette.	PgDn (repeat as needed)

Context Menu Options for Columns

In an editor that displays a list or table, right-click column headings to use the commands described in the following table.

Command	Description
Hide *Column Name* Column	Hide the selected column. Use the **Columns** command to show hidden columns.
Columns	Show or hide columns you select from the submenu that opens.
Fit Available Width	Extend the columns to the width of the editor pane.
Best Fit	Set the width of this column to accommodate the longest entry in this column.
Best Fit (All Columns)	Optimize the width of all columns to accommodate all entries.

Command	Description
Add Sort Ascending	Sort the list in ascending order.
Add Sort Descending	Sort the list in descending order.
Reset Sort	Restore the default sort order of design elements in App Explorer.
Sort By	Sort by up to three columns.

Context Menu Options for Lists and Tables

A list of entities such as references and application workflows opens in the editor pane when you click one of those headings in App Explorer. Lists include the process app editor, the table editor, a collection of forms, and so on. Lists also include messages displayed in the Validation Results, Common Log Viewer, and Activity Log.

By default, the list includes the following columns:

- Name

- Type

- Version

- Created by

- Created on

- Updated by

- Updated on

Right-click a column header to modify the list of columns shown. For example, you can add the **Guid** column, which enables you to see the unique identifier for each item in the list. This is useful for troubleshooting purposes.

Common Table and List Options

Right-click a list or table or an item in one of them, a menu opens. The following table describes the commands that are common to many lists and tables.

> **Note:** For message-specific commands, see Working with Validation Messages [page 535], Working with Common Log Messages [page 541], and Working with Activity Log Messages [page 549].

Command	Description
Collapse All	Collapses a group so only the group header is displayed.
Copy to Clipboard	Copies the entire list of messages to the Windows Clipboard so you can paste them into another program.

Command	Description
Copy Row to Clipboard	Copies the selected message to the Windows Clipboard so you can paste it into another program.
Expand All	Expands a group so the complete list of items is displayed under the group header.
Export	Opens the **Save As** dialog box so you can save the items in the list or table in a `.csv` file. Common Log Viewer messages can be saved in a `.csv` or `.xml` file.
Filter	Opens the **Look for** box, which removes messages that do not contain the text you type.
Find	Opens the **Find** dialog box, which selects each message that contains the text you type, one at a time.
Find Next	Selects the next message containing the text you typed in the **Find** dialog box.
Open	Opens the item in its editor, replacing the content of the selected tab.
Open in New Tab	Opens the item in its editor in a new tab.
Show Comments	Shows the text in the **Comment** field in a dialog box, such as the **Publish Process App** or **Check In Design Elements** dialog box.
Show Description	Shows the text in the **Description** field in a Property Editor.
Show Details	Expands each message so complete information about it can be viewed.
Show Documentation	Shows information about a Web service, as defined in the WSDL file.
Where Used	Opens the Where Used Dialog Box [page 81], which lets you see where the item is used in the process app.

About SBM Composer Dialog Box

Use this dialog box if you are asked by Support to provide information about your software version and computer system configuration.

There are two ways to open this dialog box:

- Click the **Help** icon in the upper right corner of the Quick Access Tool Bar, and then click **About**.

- Click the Composer button, and then click **SBM Composer Options**. In the **SBM Composer Options** dialog box, select **Resources**, and then click **About**.

Chapter 32: Application Field Reference

The following topics describe options in these areas:

- Field Reference [page 515]

Field Reference

This section discusses:

- System Fields for Primary Tables [page 515]

- System Fields for All Auxiliary Tables [page 518]

- System Behavior for Contacts and Companies Fields [page 524]

- Invalid and Restricted Field Names [page 525]

System Fields for Primary Tables

System fields for primary tables help ensure that the process you define is followed and that correct data is gathered as items move through that process. System fields also provide greater reporting and searching benefits than custom fields. For example, only system fields are available on Multi-table reports, which let users search for information across multiple primary and auxiliary tables.

Primary table fields are created in SBM Composer. However, you can configure specific field settings for projects in SBM Application Administrator.

On the **General** tab of the Property Editor for system fields, **[System]** is appended to the field type. In the table editor, you can show the optional **System** column, which displays **Yes** for system fields and **No** for custom fields.

Two types of system fields are provided for primary tables: those that are required and those that are optional.

For details, refer to:

- Required System Fields for All Primary Tables [page 515]

- Optional System Fields for All Primary Tables [page 517]

Required System Fields for All Primary Tables

Required system fields are automatically added to all primary tables. These fields cannot be deleted or moved into the Not Used section. You can, however, move them into the Hidden field section. You can edit these fields and change certain property settings and move them to different field sections.

> **Note:** To ensure data integrity for fields that are auto-populated by the system, such as the *Owner* field, consider setting them as read-only to prevent users from changing the values. You can also move these fields into the Hidden field section.

Icon	Field Name	Notes
	Active/ Inactive	A *Binary/Trinary* field that indicates a primary item's status. States use the *Active/Inactive* field to automatically determine the status of items in each state. In SBM Composer, you can change the active and inactive labels. The *Active/Inactive* field is always shown on forms as a drop-down list.
	Item ID	An auto-populated *Text* field that numbers primary items based on project settings defined in SBM Application Administrator. The *Item ID* is used to identify and locate items in the ID Search feature, SourceBridge, reports, and more. If your Item Type field uses prefixes, they are prefixed to the *Item ID*.
	Item Type	A *Single Selection* field populated with the types of items you wish to track, such as defects and enhancement requests. Specify a three-letter prefix that is prefixed to the Item ID. For example, a defect might have an Item ID of DEF00011 and an enhancement might have an Item ID of ENH00025. You can tailor your process to manage each item type differently, if needed. For example, you can modify a workflow by restricting the item types available for certain transitions. **Note** By default, the Item *Type* field is not set as required. For best results, set the field to be required as items are added to the system.
	Last Modified Date	An auto-populated *Date/Time* field that indicates the date and time an item was last updated. This information is used for auditing and establishing change history for items.
	Owner	An auto-populated *User* field indicating the current primary owner of an item. The value of the owner field is determined by the state in which an item resides and changes as items are transitioned through a workflow. In SBM Composer, you select a User field type as the value for the Owner field for each state in the General tab of the state Property Editor.
	Project	A system *Project* field populated with all projects in an application. If users change the value of the *Project* field for a specific item, they are also moving that item to the specified project. Depending on your workflow and project relationships, the item could follow a different process than you intended. To avoid this problem, set the field to read-only or move it to the Hidden section. **Note:** Projects are defined in SBM Application Administrator.

Icon	Field Name	Notes
	State	A system *State* field populated with all states for a specific workflow. If users change the value of the *State* field for a specific item, they are also moving that item to the specified state in the workflow and it could follow a different process than you intended. To avoid this problem, set the field to read-only or move it to the Hidden field section or restrict the section according to privileges.
	Submitter	An auto-populated *User* field indicating the user who submitted an item. Users whose privileges include submitting items into the system are available as values for this field.
	Title	An 80-character fixed-length *Text* field. The system *Title* field is displayed by default in many areas, such as built-in report and search results. This is the optimal display length for displaying item titles to users. You can increase the character length of the system *Title* field. However, doing so could return unexpected results.

Optional System Fields for All Primary Tables

Optional system fields for primary tables let you collect additional data to enhance your tracking system. Add these fields when setting up the primary table for your application in SBM Composer.

Field Name	Notes
Close Date	An auto-populated *Date/Time* field that records the date and time items are transitioned to an inactive state. If the item is later moved to an active state, the *Close Date* value is cleared. If you do not add the *Close Date* field to your application, the close date and time are still captured in various areas, such as the state change history and the timestamp for *Text/Journal* fields. Adding the *Close Date* field to your system lets users query for this information in search features and reports, particularly State Change reports.
Description	By default, the system *Description* field is a *Text Memo* field that is set to allow keyword searching.

Field Name	Notes
Last Modifier	An auto-populated *User* field that records which user last changed an item. The value of the *Last Modifier* field changes as items are updated and transitioned through a workflow. All users who can update and transition items into the system are available as values for this field, but you can control who updates and transitions items through privileges.
	If you do not add the *Last Modifier* field to your primary table, the name of the user who last modified an item is still recorded in the change history. Adding the *Last Modifier* field to your primary table lets users query for this information in search features and reports.
	Note that the **Last Updated by** value in Work Center activity views defaults to *Submitter* if you do not add the *Last Modifier* field to the primary table.
Last State Change Date	An auto-populated *Date/Time* field that records the date and time an item last moved to another state. The value of the *Last State Change Date* field changes as items are transitioned to different states. The *Last State Change Date* field is useful for searching and reporting on items based on the last date and time they were last moved. To view a complete history of state changes for particular items, users can create State change reports.
Last State Changer	An auto-populated *User* field that records the user who last moved an item to a new state. The value of the *Last State Changer* field changes as items are transitioned through a workflow. All users who can transition items into the system are available as values for this field, but you can control who transitions items through privileges and transition restrictions.
	If you do not add the *Last State Changer* field to your application, the Change History section shows who moved items from state to state. Adding the *Last State Changer* field to your application lets users query for this information in search features and reports.
	Note that the **Transitioned to** value in Work Center activity views defaults to *Submitter* if you do not add the *Last State Changer* field to the primary table.
Pause Status	An auto-populated *Pause Status* field that shows why an item is paused. A paused item cannot use transitions that move it to another state.
Submitting Agent	An auto-populated *User* field that records the user who submits an item on behalf of another user.

System Fields for All Auxiliary Tables

System fields for auxiliary tables help ensure that correct data is gathered for items that may support a workflow process. System fields also provide greater reporting and

searching benefits than custom fields. For example, only system fields are available on Multi-Table reports, which let users search for information across applications and auxiliary tables.

At a minimum, auxiliary tables must contain the *Title* system field. This field, which is automatically added to all auxiliary tables, typically serves as the value display format setting for the table, but you can use a different field for this setting as needed.

Auxiliary table fields are created in SBM Composer. You can configure specific field settings for auxiliary tables in SBM System Administrator, however.

The following table lists the optional system fields available for auxiliary tables.

Field Name	Description
Active/ Inactive	The *Active/Inactive Binary/Trinary* field for auxiliary items indicates the status of the item and lets users enable and disable items in auxiliary tables. You can change the active and inactive labels for this field, but it always appears on forms as a drop-down list.
	You can use the *Active/Inactive* field in auxiliary tables to limit the selections available in *Relational* field types without deleting items in auxiliary tables. For example, you can add the *Active/Inactive* field to a *Companies* table, letting users disable companies that are no longer needed without deleting the data associated with those companies. The inactive companies are not available as new values for the *Company* field, but they are available as selections made when they were active. In addition, users can still access them in reports and searches.
Description	A *Text/Memo* field that is set to allow keyword searching.
Item ID	An auto-populated *Text* field that numbers auxiliary items based on the order in which items are added to the table. The *Item ID* is used to identify and locate items.
Last Modified Date	An auto-populated *Date/Time* field that indicates the date and time an auxiliary item was last updated. This information is used for auditing and establishing change history for items.
Last Modifier	An auto-populated *User* field that records which user last changed an auxiliary item. The value of the *Last Modifier* field changes as items are updated. All users who can update items into the system are available as values for this field, but you can control who updates auxiliary items through privileges.
Submit Date	An auto-populated *Date/Time* field that records the date and time auxiliary items were submitted.
Submitter	An auto-populated *User* field indicating the user who submitted an auxiliary item. All users who can submit items into the system are available as values for this field, but you can control who submits items through privileges.

Field Name	Description
Title	A *Text* field that serves as the system title field and the default for the value display format for this table. By default, a *Title* field is automatically added to custom auxiliary tables. This field cannot be deleted.

System Fields for System Auxiliary Tables

Seven system auxiliary tables are provided with SBM, and system fields listed in the following tables are provided with each of these tables.

Companies Table

Field Name	Notes
Company Name	A *Text* field that serves as the system title field for this table and the default for the value display format for this table.
Customer ID	A *Text* field that can store additional identifying information for a company record.
Primary Contact and Secondary Contact	*Single Relational* fields that let users associate primary and secondary contacts for companies. These fields provide special system behavior.

Contacts Table

Field Name	Notes
Company	A *Single Relational* field based on the *Companies* table that lets users select a company to associate with a contact. This field provides special system behavior.
E-mail	A *Text* field that lets users provide an e-mail address for contacts. When you create *Contact* records from user accounts, the *E-mail* field in the *Contacts* table is populated with the e-mail address provided with the user account.
First Name	A *Text* field that by default is used as part of the value display format for this table. When you create *Contact* records from user accounts, the *First Name* field in the *Contacts* table is populated with the first word provided in the Name box for the user account.

Field Name	Notes
Last Name	A *Text* field that serves as the system title field for this table and by default, is used as part of the value display format. When you create *Contact* records from user accounts, the *Last Name* field in the *Contacts* table is populated with the last word provided for the user account.
Middle Name	A *Text* field that by default is used as part of the value display format for this table. When you create *Contact* records from user accounts, the *Middle Name* field in the *Contacts* table is populated with the words between the first and last words provided for the user account.
Phone Number	A *Text* field that lets users provide phone numbers for contacts. When you create *Contact* records from user accounts, the *Phone Number* field in the *Contacts* table is populated with the phone number provided for the user account. The *Phone Number* field is optional.
User Name	A system *User* field that is automatically populated with a SBM user name when a *Contact* record is created for that user. The option to create a *Contact* record for users is on the General tab of the Add/Edit User dialog box. This field is set to read only by the system and this property cannot be modified.

Languages Table

Field Name	Notes
Active/ Inactive	A *Binary/Trinary* field that indicates the active or inactive status of the language.
Language	A *Text* field that stores the names of languages the user interface has been translated into. The *Language* field is the system Title field and the default for the value display format for this table.
Locale	A *Text* field that stores the ISO 639-1 locale string for the language.

Problems Table

Field Name	Notes
Description	A *Text* field that lets users provide a description for problems. The *Description* field is provided with the *Problems* table by default, but is optional.

Field Name	Notes
Folder	A *Folder* field that lets users create links to problems in folders. Knowledge Base folders are designed to allow anonymous access to items, but users can also create Problem links in shared or private folders. The *Folder* field is optional, but using it can help users better organize records in the *Problems* table.
Title	A *Text* field that serves as the system title field and the default for the value display format for this table.
Visibility	A *Binary/Trinary* field that lets users set the visibility of a problem to internal or public. Users who are granted the "View Public Problems and Resolutions" privilege can view items with public visibility. Anonymous users can also view items with public visibility if they are stored in Knowledge Base folders that allow anonymous access. By default, all new problems have internal visibility.
Last Updated	An auto-populated *Date/Time* field that indicates the date and time a problem was last updated. This information is used for auditing and establishing change history for problems. The *Last Updated* field is provided with the *Problems* table by default, but is optional.

Resolutions Table

Field Name	Notes
Description	A *Text* field that lets users provide a description for resolutions. The *Description* field is provided with the *Resolutions* table by default, but is optional.
Last Updated	An auto-populated *Date/Time* field that indicates the date and time a resolution was last updated. This information is used for auditing and establishing change history for resolutions. The *Last Updated* field is provided with the *Resolutions* table by default, but is optional.
Problem	A *Single Relational* field based on the *Problems* table and lets users select a problem to associate with a resolution.
Title	A *Text* field that serves as the system title field and the default for the value display format for this table.
Visibility	A *Binary/Trinary* field that lets users set the visibility of a resolution to internal or public. Users who are granted the "View Public Problems and Resolutions" privilege can view items with public visibility. Anonymous users can also view resolutions with public visibility if they are associated with problems that are stored in Knowledge Base folders that allow anonymous access. By default, all new resolutions have internal visibility.

SharePoint Project Servers Table

This table contains relationship information between a SharePoint® site and an SBM project. It is populated automatically and must not be populated manually.

SharePoint Servers Table

Field Name	Notes
Title	Identifies what this SharePoint® server is used for, such as "Sales" or "Human Resources."
Site Url	Contains the Site Collection root or high-level SharePoint folder under which SharePoint sites will be created. For example, `http://SharePointServerName:port/sites/ImageBuilder`
User Name	The account that will be used to create the folders in SharePoint. This account needs permissions in SharePoint to add folders and add, edit, and delete documents. This account is also used when attachment actions are performed in SBM by an anonymous user.
Password	The password used to access the account. This password is stored in the database, so it should be configured to not expire.

String IDs Table

Field Name	Notes
Name	A *Text* field that stores string representations of unique resource identifiers associated with strings. The *Name* field serves as the system title field and the default for the value display format for this table.
Root Value	Stores the user-visible value for the string. Root values should not be modified.

Strings Table

Field Name	Notes
Create Date	An auto-populated *Date/Time* field that records the date and time strings were submitted. The *Create Date* field is provided with the *Strings* table by default, but is optional.

Field Name	Notes
Creator	An auto-populated *User* field indicating the user who submitted a string. All users who can submit items into the system are available as values for this field, but you can control who submits strings through privileges. To ensure data integrity for *Creator* field values, set this field to read-only or move it into the Hidden Fields section.
Description	A *Text* field that lets users provide a description for strings.
Language ID	A *Single Relational* field associated with the *Languages* table that lets you associate a translated string with a language.
Last Modified Date	An auto-populated *Date/Time* field that indicates the date and time a string was last updated. This information is used for auditing and establishing change history for items. The *Last Modified Date* field is provided with the *Strings* table by default, but is optional.
Last Modifier	An auto-populated *User* field that records which user last changed a string. The value of the *Last Modifier* field changes as strings are updated. All users who can update items into the system are available as values for this field, but you can control who updates strings through privileges. To ensure data integrity for *Last Modifier* field values, set this field to read-only or move it into the **Hidden Fields** section. The *Last Modifier* field is provided with the *Strings* table by default, but is optional.
String	A *Text* field that stores strings that are displayed in SBM elements, such as labels, button names, and error messages.
String ID	A *Single Relational* field associated with the *String IDs* table that lets you associate a unique resource identifier, referred to as an internal ID, with a string. The *String ID* field is the default for the value display format for this table.

System Behavior for Contacts and Companies Fields

The *Companies* and *Contacts* system tables are included in every environment. These system tables let you store information about companies you work with and contacts that are associated with those companies.

In addition, system *Contacts* and *Companies* fields can be defined in each application. Special behavior is provided with these system fields. For example, you can set up your system so that an existing user can grant External access to a contact. You can also grant privileges that let contacts view items submitted by other contacts within in the same company.

This behavior is only available for the system *Contacts* or *Companies* fields. For example, if you create a *Single Relational Contacts* field in an application, the system behavior does not apply.

Invalid and Restricted Field Names

This topic describes field names that cannot be used in SBM Composer.

Invalid Fields

The following fields are considered invalid database names, so they cannot be used for non-system fields or non-*Sub-Relational* fields:

ACCESS, ADD, ALL, ALTER, AND, ANY, ARRAYLEN, AS, ASC, AUDIT, BETWEEN, BY, CHAR, CHECK, CLUSTER, COLUMN, COMMENT, COMPRESS, CONNECT, CREATE, CURRENT, DATE, DECIMAL, DEFAULT, DELETE, DESC, DISTINCT, DROP, ELSE, EXCLUSIVE, EXISTS, FILE, FLOAT, FOR, FROM, GRANT, GROUP, HAVING, IDENTIFIED, IMMEDIATE, IN, INCREMENT, INDEX, INITIAL, INSERT, INTEGER, INTERSECT, INTO, IS, LEVEL, LIKE, LOCK, LONG, MAXEXTENTS, MINUS, MODE, MODIFY, NOAUDIT, NOCOMPRESS, NOT, NOTFOUND, NOWAIT, NULL, NUMBER, OF, OFFLINE, ON, ONLINE, OPTION, OR, ORDER, PCTFREE, PRIOR, PRIVILEGES, PUBLIC, RAW, RENAME, RESOURCE, REVOKE, ROW, ROWID, ROWLABEL, ROWNUM, ROWS, START, SELECT, SESSION, SET, SHARE, SIZE, SMALLINT, SQLBUF, SUCCESSFUL, SYNONYM, SYSDATE, TABLE, THEN, TO, TRIGGER, UID, UNION, UNIQUE, UNUSED, UPDATE, USER, VALIDATE, VALUES, VARCHAR, VARCHAR2, VIEW, WHENEVER, WHERE, WITH

Restricted Fields

SBM Composer checks for "reserved" database names for non-system fields that have never been published. If any are found, validation fails.

The following field names are restricted in both primary and auxiliary tables:

- ISSUEID

- ACTIVEINACTIVE

- TITLE

- DESCRIPTION

- LASTMODIFIER

- LASTMODIFIEDDATE

- SUBMITTER

- SUBMITDATE

- ID

- UUID

The following field names are restricted in primary tables, but not in auxiliary tables:

- PROJECTID

- STATE

- LASTSTATECHANGEDATE

- CLOSEDATE

- OWNER

- LASTSTATECHANGER

- ISSUETYPE

- SECONDARYOWNER

- RESOLUTIONSUMMARY

- RESOLUTIONDESC

- COMPANY

- CONTACT

- INCIDENT_ID

Chapter 33: Transition Reference

This section discusses:

- Transition Types [page 527]

Transition Types

This topic describes the transition types you can add to an application workflow or sub-workflow.

Regular

Regular transitions let users transition items from one state to another.

Quick

Quick transitions transition items from one state to another without showing a form to users. Outgoing transitions from decisions are automatically configured as quick transitions, because the workflow logic, not the user, determines the next state in the workflow.

Post

Post transitions lets users submit new items from the item they are currently working with. Posted items receive a new Item ID and follow the workflow of the project they are posted to. A link is created between the original item and the new item.

Post transitions can perform the following actions:

- Post a new item to another project using the same primary table.

- Post a new item to a project using a different primary table.

- Post a new item to an auxiliary table.

For field mapping details, refer to Mapping Fields for Post, Publish, Copy, or Subtask Transitions [page 150].

Post transitions can be executed manually by a user clicking a transition button or automatically, by Web service calls in an orchestration workflow. For information about using Web service calls to execute a Post transition, see the orchestration use cases in the SBM Composer online help or the *SBM Orchestration Guide*.

> **Note:** You can use an action on a **Post** transition action to cause a second transition to execute after the workflow reaches the transition on which the action is defined. For details, see Tutorial: Submitting Multiple Primary Items [page 430].

Subtask

A subtask is a primary item that is associated with a principal task and that is displayed differently from the principal task. Subtasks are typically used when a set of smaller tasks needs to be worked on before a larger task, or principal task, can continue its process. A

Subtask transition lets users create a new primary item in a specified project. You can specify that the new item be linked to the source (original), or principal, item and that subtasks are transitioned according to values in a specific *Binary/Trinary* or *Single Selection* field. For example, when all subtasks are completed, the principal item could be transitioned to a completed state.

Subtask transitions are useful for setting up an entire set of new tasks at once. For example, at the beginning of a large project, a user can create a principal task and use a **Subtask** transition to create subtasks from that larger task.

After adding a **Subtask** transition to your workflow, you can define actions that transition subtasks and principal items.

For field mapping details, refer to Mapping Fields for Post, Publish, Copy, or Subtask Transitions [page 150].

Viewing Subtasks

Users who are granted the **View Principal and Subtasks** privilege and who selected the **Subtasks** display preference can view the **Subtasks** section if it is included on a form. The **Subtasks** section is visible only if an item is a subtask of another item or has one or more subtasks associated with it. In addition, users can access subtasks only in projects for which they have "view" privileges.

> **Tip:** Subtasks appear as links in the **Subtasks** section. By default, these links contain the Item ID and Title of the item. You can modify this display using the value display format setting using the Property Editor for the primary table used by the subtask workflow.

Copy

The **Copy** transition lets users copy primary items and place them in another location in the project hierarchy within the same table.

Considerations

- When users click the **Copy** transition button, instead of the transition moving the item to the next state in the current project, a copy of the item is created in the specified project.

- The project to which items will be copied must be set up to allow new items to be submitted. Verify that **Allow New Items to be Submitted** project setting is selected in SBM Application Administrator. Then make sure that users who will execute the transition have privileges to submit items into the project to which items will be copied.

- The project hierarchy does not open for users for those transitions that are configured to copy to a specific project. If you select the **Select at Runtime** option for the **Copy** transition in Application Administrator, the project hierarchy is displayed for users to select the project to which to copy an item.

- By default, the submitter for the copied item is the same user who submitted the original item. To set the submitter for the copied item as the user who executed the **Copy** transition, edit the copy transition, click the **Options** tab, and then select **Reset submitter field to current user**.

  For additional field mapping details, refer to Mapping Fields for Post, Publish, Copy, or Subtask Transitions [page 150].

- **Copy** transitions always have a "To State" of "Same."

- Ensure that **Copy** transitions do not change the project, state, or both in such a way as to make the item fall outside the workflow of the target project. To avoid this, only allow copying of items between projects that share inherited workflow properties. This lets all states and transitions available to the source item's project to be available in the newly copied item's final project.

- You can configure a project to only allow items to be submitted from the **Copy** transition, instead of being submitted manually by users through a **Submit** transition. To do so, in SBM Composer, clear the **Default submit transition** check box on the **Options** tab of the Property Editor for the **Submit** transition.

Copy transitions can be executed manually by a user clicking a transition button; or automatically, by Web service calls in an orchestration workflow. For information about using Web service calls to execute a Copy transition, see the orchestration use cases in the SBM Composer online help or the *SBM Orchestration Guide*.

Update

Update transitions are unusual in that they do not move items from state to state. Instead, **Update** transitions let users with the appropriate privileges update data in an item at a particular state in the workflow. You can set the "from" state for **Update** transitions, but the "to" state is always the same state. This transition is automatically included on the **Any** item in new workflows, but you can disable that default **Update** transition and create others that are available only for selected states.

> **Note:** Update transitions are controlled by the "update" privileges in the Item privilege category. Users who are granted these privileges can update individual items using the **Update** transition, or modify multiple items using the Rich Editable Grid.

Delete

Delete transitions do not move items from state to state. They let users with the appropriate privileges delete an item at a particular state in the workflow. You can set the "from" state for **Delete** transitions, but the "to" state is always **Deleted**. This transition is automatically included on the **Any** state in new workflows, but you can disable that default **Delete** transition and create others that are available only for selected states or in particular sub-workflows.

> **Note:** Field overrides and triggers do not apply to custom **Delete** transitions.

Publish

The **Publish** transition lets users quickly publish problems and resolutions pertaining to primary items to the legacy Knowledge Base that is included with SBM. When users execute the **Publish** transition, Submit forms for the Problems and Resolutions tables automatically open. By default, the problem contains the **Title** and **Description** from the item. Users can modify this information and select other values for the problem and resolution, such as which folder the problem will be stored in and its visibility.

Considerations

- The **Publish** transition type is available for any workflow, but problems can be posted only to the Problems table, and resolutions can be posted only to the Resolutions table.

- Publish transitions are available only to users who have been granted privileges to submit items into the Problems table. In addition, users who have privileges to submit items into the Problems table but not the Resolutions table will not receive a Submit form for the Resolutions table after submitting a problem.

- Users who publish items to the Knowledge Base must be granted a role that allows them to add links to the folders to which you want them to publish items. Roles are granted on folders in SBM Work Center.

- You can choose to specify **Publish** transitions as "quick transitions," letting users bypass the Transition form for the workflow of the original item and to immediately submit a problem to the Problems table.

- You can map additional fields for a **Publish** transition.

External Post

External Post transitions let users transition items from one state to another, as well as generate and send an e-mail message to an external database requesting that the item be posted. You select the external database to which you want the item to be added.

> **Note:** *File* and *URL* fields cannot be used for **External Post** transitions.

Part 6: Troubleshooting

This section contains the following information:

Chapter 34: Using the Validation Results

This section contains the following information:

- About the Validation Results [page 533]
- Using the Validation Results [page 534]

About the Validation Results

Use the Validation Results to troubleshoot validation problems.

When you click the **Validate** button on the Quick Access Toolbar, SBM Composer attempts to validate the open process app to determine if it is ready to be published to SBM Composer. If you click the **Publish**, **Deploy**, or **Quick Deploy** buttons before you validate, SBM Composer tries to validate the process app first.

For example, if you did not connect a state with a transition in an application workflow, a warning message is displayed in the Validation Results telling you that the state is unreachable. Or, if you fail to provide a required value for a step input in an orchestration workflow, an error message is displayed.

The Validation Results tab displays "Succeeded" or "Failed" as appropriate. Results remain current until the user makes any change to the process app, including using Repository Menu [page 490] commands. When that happens, "Succeeded (out of date)" is displayed on the Validation Results tab. If validation results are current, validation is not performed with the next export, publish, or deploy operation.

Validation Results messages are categorized by logging level. The following table describes each level.

Logging Level	Icon	Description
Error	❌	An error message is displayed when something is wrong in the process app that you need to fix.
Warning	⚠️	A warning message is displayed to notify you of a potential problem or of a condition that could cause an undesirable or unexpected situation.
Info	ℹ️	An information message is logged to provide information about the validation process.

The following table describes the elements that appear in the Validation Results.

Element	Description
Description	Error message text

Element	Description
Severity	Severity of the message
Element name	Name of the element to which the message refers
Element type	Element type
Design element	Name of the design element that contains the element
Application/ Orchestration	Name of the application or orchestration that contains the design element
Show	Filter messages to be displayed. You can select logging levels, specify whether dismissed messages should be displayed, and display messages that pertain only to the current editor. (See Limiting Validation Results [page 534].)

Using the Validation Results

This section contains the following topics:

- Opening the Validation Results [page 534]

- Limiting Validation Results [page 534]

- Working with Validation Messages [page 535]

- Debugging With Error and Warning Validation Messages [page 535]

Opening the Validation Results

To open the Validation Results:

1. On the **Home** tab of the Ribbon, in the **Common Views** group, select the **Validation Results** check box.

   The Validation Results opens.

   > **Note:** The Validation Results automatically opens when you validate a process app. For details about its content, see About the Validation Results [page 533].

2. If the **Validation Results** check box is already selected and the Validation Results is not visible, click the **Validation Results** tab below the Property Editor.

Limiting Validation Results

You can remove messages you do not need to see from the Validation Results. The check boxes at the bottom let you toggle between showing or hiding messages based on logging level (described below), whether you previously dismissed the messages (see Working

with Validation Messages [page 535]), and whether the messages pertain to the current editor.

> **Note:** If no editor is open when you select the **Current editor only** check box, messages from the entire process app are displayed.

To specify which messages are displayed in the Validation Results:

- Select or clear the appropriate check boxes in the **Show** section.

Working with Validation Messages

The Validation Results provides commands to help you work with messages. To use the commands, right-click a message or the message list and then select a command from the menu that opens.

The following table describes commands that are specific to the Validation Results; see Context Menu Options for Lists and Tables [page 512] for a description of other commands in the menu.

Command	Description
Go to Location	Opens the editor (and in some cases, the Property Editor) for the applicable design element, and selects the design element. Alternatively, you can double-click the message to invoke this command.
Show Message Help	If available, opens a help topic that explains the message in more detail.
Dismiss	Strikes through and shades the message text so you can ignore the messages.
Dismiss All Like	Dismisses all messages that report the same basic problem.
Undismiss	Reverses the **Dismiss** command. The **Dismissed** check box must be selected to see dismissed messages and use this command.
Undismiss All Like	Reverses the **Dismiss All Like** command. The **Dismissed** check box must be selected to see dismissed messages and use this command.
Show Details	Expands each message so complete information about it can be viewed.
Clear	Removes all messages from the Validation Results.

Debugging With Error and Warning Validation Messages

This section shows you how to work with error and warning validation messages to debug process apps.

The following is a sample warning message. All validation messages appear in this format.

Description	Severity	Element Name	Element Type	Design Element	Application/ Orchestration
⚠️ State 'Closed' is unreachable	Low	Closed	Regular State	CreditApprovalWorkflow	CreditApprovalApp

In this message, `State 'Closed' is unreachable` indicates that there is a problem with the regular state named `Closed` that is part of the application workflow named `CreditApprovalWorkflow`, which is contained within the application named `CreditApprovalApp`.

To go to the source of the message:

- If the **Element type** column contains an entry other than `Workflow`, right-click anywhere in the message row, and then select **Go to Location** on the menu.

  SBM Composer opens the appropriate design element in the workflow editor and selects the element. The Property Editor for some of the design elements also opens.

In the example, SBM Composer opens `CreditApprovalWorkflow` (application workflow) in the workflow editor and selects the `Closed` state. In this case, the `Closed` state is not connected by a transition in `CreditApprovalWorkflow`. If the `Closed` state needs to be connected to another state, then you should fix the problem. If the `Closed` state does not need to be connected, then you can ignore the warning.

Chapter 35: Using the Common Log Viewer

This section contains the following information:

- About the Common Log Viewer [page 537]

- Using the Common Log Viewer [page 537]

- Using Advanced Options [page 548]

About the Common Log Viewer

The Common Log Viewer displays a list of messages that tell you what went wrong if a process app encountered an error or if it did not work as expected during runtime. The Common Log Viewer also contains general diagnostic messages, such as notifications of when a Web service is invoked or when the SBM Orchestration Engine is sending a message.

The Common Log Viewer generates messages only for workflow elements, including the workflows themselves (design elements), not for other elements such as tables or references.

Using the various features in the Common Log Viewer, you can specify which messages you want to see based on their type (user and technical) and logging level and when and where they occurred. You can also sort the messages and export them to a file.

User messages generally consist of SOAP messages and significant runtime events. Technical messages contain information about routine runtime activity and process flow.

> **Note:** For additional information about using the Common Log Viewer to troubleshoot orchestrations, refer to the *SBM Orchestration Guide* or the SBM Composer online help.

Using the Common Log Viewer

This section contains the following topics:

- Opening the Log Viewer [page 538]

- Using Debug Logging [page 538]

- Generating Common Log Messages [page 539]

- Viewing Common Log Messages [page 540]

- Working with Common Log Messages [page 541]

- Filtering Messages [page 541]

- Troubleshooting the Common Log Viewer [page 545]

- Sorting Messages [page 545]

- Viewing Messages in a Dialog Box [page 546]

- Exporting Messages [page 547]

- Exceeding the Message Limit [page 547]

- Using Application Repository to Filter and View Common Log Messages [page 548]

Note: The Common Log Viewer displays messages from the Common Log in SBM Application Repository.

Opening the Log Viewer

To open the Common Log Viewer:

1. On the **Home** tab of the Ribbon, in the **Common Views** group, select the **Common Log Viewer** check box. Alternatively, press Ctrl+Shift+L.

   The Common Log Viewer opens.

2. If the **Common Log Viewer** check box is already selected and the Common Log Viewer is not visible, click the **Common Log Viewer** tab.

For a description of the elements that appear in the Log Viewer, see Overview Tab [page 539] and Details Tab [page 540].

Using Debug Logging

To use the Common Log Viewer for debugging process apps, you must turn the Debug Logging feature on. You can only do this after you deployed the process app.

Note: If you do not turn on Debug Logging, only error messages are displayed in the Common Log Viewer. If you turn off Debug Logging, the only *new* messages that are displayed are error messages.

To turn Debug Logging on:

1. In the Common Log Viewer, select the **Overview** tab.

   A list of all the applications and orchestrations and the workflows contained in them is displayed.

2. Turn on Debug Logging for the applications and orchestrations in the process app as follows:

   a. Right-click the name of an application or orchestration.

   b. Select **Debug Logging**.

   [Debug Logging ON] appears to the right of the name of each application and orchestration. When Debug Logging is on, the option on the menu is checked.

   Important: You should turn on Debug Logging only while you are developing and testing a process app. Running a process app with Debug Logging turned on causes the process app to run more slowly, so be sure to turn this feature off in the production environment when it is ready to be used by users.

To turn Debug Logging off:

1. Right-click the name of an application or orchestration for which Debug Logging is turned on.

2. Select **Debug Logging**.

   When Debug Logging is off, the option on the menu is not checked.

Overview Tab

Element	Description
Hierarchical list	Displays a list of all the applications and orchestrations in the process app and the workflows contained in them. Subroutines appear as children of the orchestration workflows (or subroutine workflows, if they are nested) that they are called from, and can appear more than once in the list.
Set filter	Displays the **Message Filter** dialog box, which filters Common Log Viewer messages by environment, timeframe, and type. (See Message Filter Dialog Box [page 542].) The messages are displayed on the **Details** tab of the Log Viewer.
Refresh	Updates the Common Log Viewer with messages logged since the last time you ran the process app based on the filters that you set in the **Message Filter** dialog box and on the **Details** tab. This button is not available until you deploy the process app.
Export	Displays the **Export log messages** dialog box, which lets you save the messages to an XML file. (See Exporting Messages [page 547].)

Generating Common Log Messages

To generate messages for the Common Log Viewer:

1. Create and deploy a process app, and then run it.

2. In SBM Composer, in the Common Log Viewer, click **Refresh** to display any new messages.

   If there are messages for any workflows in the process app, numbers other than 0 (zero) appear in brackets to the right of the name. The number of error messages appears in the first position, the number of warning messages in the second, the number of info messages in the third, and the number of debug messages appears in the fourth position.

   For example, `AnOrchestrationWorkflow [1/1/0/15]` indicates that this orchestration workflow has 1 error message, 1 warning message, 0 info messages, and 15 debug messages. (This example assumes that you selected **User and Technical Messages** in the **Message Filter** dialog box. (See Filtering Messages [page 541].)

   The messages are displayed on the **Details** tab.

Viewing Common Log Messages

In the Common Log Viewer, you can view messages for the workflow that is selected on the **Overview** tab.

To view messages for a workflow:

1. In the Common Log Viewer, click the **Overview** tab.

2. Select a workflow in the list. The workflows with messages are displayed in bold.

3. Click the **Details** tab.

Details Tab

The **Details** tab displays messages for the workflow that is selected in the **Overview** tab. Use this tab to filter messages by run, by logging level, and by selected design element.

Click the vertical divider to the right of the list of messages to show or hide a preview of the selected message. To change the format of the message, right-click in the preview window and select the desired options. For details about the options, see Message Detail Dialog Box [page 546].

> **Note:** SBM Composer retains the settings on this tab after you close the process app.

Element	Description
Date	Displays the date and time that the message was logged.
Run ID (column)	Displays the run number for the workflow.
Associated element	Displays the design element that is associated with the message.
Workflow	Displays the workflow where the associated element is located.
Text	Displays the text of the message.
Run (subroutines only)	When a subroutine is selected, displays the orchestration workflow that executed the subroutine. You can switch to another orchestration workflow by selecting it from the list.
Run (list)	Displays the starting time, duration, and run number for each run. (See Filtering by Run [page 543].)
Show	Displays messages for the selected logging levels. (See Filtering by Logging Level [page 544].)
Autoselect latest run	Automatically selects the latest run in the **Run** list.

Element	Description
Restrict by selection	Displays only messages for the design element that is selected in the workflow editor. If no design element is selected, then messages for the entire workflow are displayed. (See Filtering by Selected Design Element [page 544].)
Set filter	Opens the **Message Filter** dialog box, which filters messages by environment, timeframe, and type. (See Message Filter Dialog Box [page 542].) The messages are displayed on the **Details** tab of the Common Log Viewer.
Refresh	Updates the Common Log Viewer with messages logged since the last time you ran the process app based on the filters that you set in the **Message Filter** dialog box and on the **Details** tab. This button is not available until you deploy the process app.
Export	Displays the **Export log messages** dialog box, which lets you save the messages to an XML file. (See Exporting Messages [page 547].)

Working with Common Log Messages

The Common Log Viewer provides commands to help you work with messages. To use the commands, right-click a message or the message list and then select a command from the menu that opens.

The following table describes commands that are specific to the Common Log Viewer; see Context Menu Options for Lists and Tables [page 512] for a description of other commands in the menu.

Command	Description
Show Associated Element	Opens the appropriate design element in the workflow editor and selected the design element associated with the message.
Show Message	Opens the Message Detail Dialog Box [page 546], which displays the entire contents of the text portion of the message.

Filtering Messages

Common Log Viewer filtering tools let you display only the messages that you want to see on the **Details** tab of the Common Log Viewer. You can filter messages based on the following criteria:

- Environment

- Type (user and technical messages)

- Timeframe

- Run

- Selected design element

- Logging levels (significance, or severity)

Use the **Message Filter** dialog box to filter messages by environment, type, and timeframe. Use the **Details** tab of the Common Log Viewer to filter messages by run, logging levels, and selected design element.

> **Note:** Any filters that you set in the **Message Filter** dialog box remain in effect until you change them, even after you close and then reopen a process app in SBM Composer.

Message Filter Dialog Box

The following table describes the elements that appear in the **Message Filter** dialog box.

Element	Description
Environment	Displays messages from the selected environment.
Since last publish	Displays messages since the process app was last published.
Range	Displays messages for a specified date and time range.
User messages only	Displays user messages only. User messages are most useful to business users for debugging process apps. Usually, you should be able to find the cause of a problem by reading user messages.

Element	Description
User and technical messages	Displays both user and technical messages. Technical messages are more appropriate for advanced users such as developers, integrators, and support personnel. When Debug Logging is enabled in SBM Composer, only user messages are displayed. To see technical messages, you must select this option and also do the following: 1. In SBM Application Repository, click **Advanced** and then click **Common Log**. 2. On the **Common Log** tab, click **Settings**. 3. In the **Common Log Settings** dialog box, select any logging level except **user**, and then click **OK**. (The **trace** logging level is typically used for this purpose.) **Note:** If you disable Debug Logging, the Common Logger logging level is reset to **error**, so only error messages are logged. If you enable Debug Logging, the logging level is reset to **user**, so only user messages are logged. This overrides any previous configuration in Application Repository, so perform the preceding steps if you need to see technical messages. (For instructions on enabling or disabling Debug Logging, see Using Debug Logging [page 538].)
Show messages from failed runs only	Select this check box if you want to see messages only about orchestration workflows whose execution failed. The reason for failure is typically an unhandled fault.

Filtering by Run

For application workflows, a *run* is logged each time an item is submitted into a project. The run is the Item ID assigned to the submitted item. For orchestration workflows, a *run* is logged each time an orchestration workflow is invoked by a process app. You can choose to display messages for a single run or for all runs for the workflow that is selected on the **Overview** tab. The most recent run is the one with the highest number.

To specify the messages that are displayed by run:

1. In the Common Log Viewer, select the **Overview** tab.

2. Select a workflow in the list.

3. Select the **Details** tab.

4. On the **Run** menu, select a single run or **All Runs**.

Filtering by Logging Level

Common Log messages are categorized by significance, or severity. The following table describes the four levels available in the Common Log Viewer. The levels are listed in descending order of severity. As the severity decreases, the number of messages increases.

> **Note:** The Common Log settings in Application Repository determine which levels of messages are generated for an application or orchestration. If the setting for a particular workflow is less than "Info," you will not be able to see all logging levels in the Common Log Viewer for that workflow. (See *SBM Application Repository Guide*.)

Logging Level	Icon	Description
Error	❌	An error message is logged when the execution of an application workflow or an orchestration workflow fails or when a process app fails or is stopped. The message explains the reason for the error and might suggest a possible solution, and an icon is displayed in the workflow editor next to the design element where the error occurred.
Warning	⚠️	A warning message is logged to notify you that an operation completed, but there were potential problems or unexpected results.
Info	ℹ️	An info message is logged to provide information about significant runtime events.
Debug	Ⓓ	A debug message is logged to provide detailed information about the flow through the system and about routine operations.

To specify which messages are displayed based on logging level:

1. In the Common Log Viewer, select the **Details** tab.

2. Select or clear the **Error**, **Warning**, **Info**, and **Debug** check boxes.

Filtering by Selected Design Element

You can choose to see messages for a single element only. For example, you could select a transition in an application workflow or a **Decision** step in an orchestration workflow. You can select an element using App Explorer or you use the Common Log Viewer.

To view messages for a single element using App Explorer:

1. In App Explorer, select an application workflow or an orchestration workflow.

2. In the workflow editor that opens, select an element.

3. In the Common Log Viewer, select the **Details** tab.

4. Select the **Restrict by selection** check box.

You can also select an element using the Common Log Viewer.

To view messages for a single element using the Common Log Viewer:

1. In the Common Log Viewer, locate a message associated with a specific element. (The **Associated elements** column should contain an entry other than `Workflow`.)

2. Right-click anywhere in the message row, and then select **Go to Location**.

   If the application workflow is not already visible in the workflow editor, SBM Composer opens the appropriate workflow in the editor and selects the associated element.

3. In the Common Log Viewer, select the **Restrict by selection** check box.

Troubleshooting the Common Log Viewer

If you do not see any messages in the Common Log Viewer, or if you see only error messages when you expect other levels as well, check for the following:

- Did you turn on Debug Logging for the applications and orchestrations that contain the workflows you want to debug?

- Have you run the process app since you turned on Debug Logging?

- Did you click the **Refresh** button after you ran the process app?

- Is the **Message timeframe** setting in the **Message Filter** dialog box **Since last publish**, or were the messages generated within the specified range?

- If the process app contains an asynchronous orchestration workflow, did you wait for it to run?

- If the process app contains an orchestration workflow that is invoked by external events, did you send the correct authentication credentials, such as user ID and password?

Sorting Messages

You can sort Common Log messages in a variety of ways. Following are five ways to change how messages are displayed.

To sort messages by date and time:

- Click the **Date** column heading to sort the messages in ascending or descending order by date and time, as indicated by the direction of the arrow.

To sort messages by Item ID or run number:

- Click the **Run** column to group the messages by run in ascending or descending numerical order, as indicated by the direction of the arrow.

To sort messages by associated element:

- Click the **Associated element** column to group the messages by the first letter of the name of the associated element in ascending or descending alphabetical order, as indicated by the direction of the arrow.

To sort message text in alphabetical order:

- Click the **Text** column to group the message text by the first letter of the message in ascending or ascending alphabetical order, as indicated by the direction of the arrow.

To display the message text in a separate row below the Date, Run, and Associated element columns:

- Right-click any of the messages, and then select **Show Details**.

   The second row can contain up to five lines of text.

Viewing Messages in a Dialog Box

Sometimes Common Log messages are too long to fit in the **Text** column. You can display the entire contents of the message in the **Message Detail** dialog box.

To display the contents of a message in a dialog box:

1. Right-click the message, and then select **Show Message**.

   The **Message Detail** dialog box opens.

2. Click the **Next** and **Previous** buttons to move up or down the list of messages according to the specified sort order. (See Sorting Messages [page 545].)

Message Detail Dialog Box

The following table describes the elements that appear in the **Message Detail** dialog box.

Element	Description
Previous	Displays the contents of the previous message on the list.
Next	Displays the contents of the next message on the list.
Display	**Plain text** displays the message with no formatting. **XML** uses colors to distinguish different parts of the message. The colors are specified on the HTML and Script Options [page 487] tab of the SBM Composer Options [page 479] dialog box. **Formatted XML** uses colors, indents, and separate lines for each element to make the message easier to read.
Word wrap	Wraps the lines of text to fit the width of the dialog box. If the message includes line breaks, clearing this check box enables you to easily scan multiple lines.
Close	Closes the dialog box.

To copy a message into a text editor or application:

1. Do one of the following:

   - Select the text, right-click, and then select **Copy**.

- Right-click in the message, select **Select All**, right-click again, and then select **Copy**.

2. Paste the message.

> **Note:** Colors and formatting are also copied if the target text editor or application supports them.

Exporting Messages

You can export and save the current Common Log messages to an XML document for later reference or for use by support personnel. All messages are saved, that is, any filters that you specified in the Common Log Viewer are not applied.

To export the messages for all of the application workflows and orchestration workflows:

1. In the Common Log Viewer, click the **Overview** tab.

2. Click **Export**.

   The **Export log messages** dialog box opens with a default file name of `DebugLogSnapshot`. You can change the name.

3. Using the **Saved in** menu, navigate to the location where you want to save the file, and then click **Save**.

   The file is saved in XML format along with a cascading style sheet (`.css`) and an XSL style sheet (`.xsl`) file. You can view the file in a Web browser or in any XML editor.

To export the messages for a particular application workflow or orchestration workflow:

- Follow the steps in the procedure above, except click the **Details** tab and select a message before you click **Export**.

Exceeding the Message Limit

The Common Log Viewer can display a maximum of 20,000 messages. If you exceed this limit, you will receive an error message.

To limit the number of messages to the allowable number after you receive an error message:

1. In the error message box, click **OK**.

2. Click **Set filter**.

3. Do one or both of the following:

   - If **User and technical messages** is selected, select **User messages only**.

   - Reduce the range in the **Timeframe** section.

4. Click **OK**.

Using Application Repository to Filter and View Common Log Messages

If you have access to the Common Log in Application Repository, you can see all of the messages that are logged for the applications and orchestrations in a process app. Using the Common Log settings, you can specify the types of messages, by logging level, that are generated for a particular application or orchestration. If Verbosity is set to all, the log shows fatal and trace messages in addition to error, warning, info, and debug. See *SBM Application Repository Guide* for additional information.

Using Advanced Options

Advanced options for debugging, such as setting the logging level to trace or fatal, are available in Application Repository. These options should be used by Development and Support organizations. Refer to the *SBM Application Repository Guide* for more information.

Chapter 36: Using the Activity Log

The Activity Log contains messages pertaining to major SBM Composer operations. It reports on deployment and repository operations, and activities such as opening, closing, saving, exporting, importing, validating, and deleting a process app.

- Opening the Activity Log [page 549]

- Working with Activity Log Messages [page 549]

- Determining the Cause of Failed Deployments [page 550]

Opening the Activity Log

To open the Activity Log:

1. On the **Home** tab of the Ribbon, in the **Common Views** group, select the **Activity Log** check box.

   The Activity Log opens.

   > **Note:** The Activity Log automatically opens when you deploy a process app.

2. If the **Activity Log** check box is already selected and the Activity Log is not visible, click the **Activity Log** tab below the Property Editor.

Working with Activity Log Messages

The Activity Log provides commands to help you work with messages. To use the commands, right-click a message or the message list and then select a command from the menu that opens.

The following table describes commands that are specific to the Activity Log; see Context Menu Options for Lists and Tables [page 512] for a description of other commands in the menu.

Command	Description
View	If the message has details (for example, about deployment), shows more information in a popup window.
Remove	Removes the selected message from the Activity Log.
Clear	Removes all messages from the Activity Log. **Note:** Activity Log messages automatically clear after you close the process app (either directly or by opening another process app), or delete the process app from the local cache or the repository.

Determining the Cause of Failed Deployments

If a process app fails to deploy, an info (information) message is displayed in the Activity Log. For example, if the deployment operation for a process app named `TestProcessApp` fails, you will see the following message: `Deployment of 'TestProcessApp' has aborted`.

To find the reason for the failure:

1. Right-click the message, and then select **Go to Location**.

    The **Deployment Log** dialog box opens.

2. Look for any `ERROR` entries in the log.

Following are some typical reasons that process apps fail to deploy:

*   An endpoint is missing or is not configured correctly in the runtime environment.

*   You do not have the proper permissions to deploy the process app.

*   You used a name for an application workflow or a primary table that already exists in the repository.

*   A BPEL compilation error occurred.

Chapter 37: Troubleshooting Scripts

Scripts are shown under the **Scripts** heading in App Explorer. After you add a script or change a script in the script editor, click the **Validate Script** button on the **Validation** tab of the Ribbon, or right-click in the script editor and then select **Validate**.

- If the script is valid, the Validation Output pane displays "This script is valid."

- If the script has errors, the Validation Output pane contains a row for each error that includes the error code, error type, error details, and the line number in the script where the error occurred. Double-click the error in the Validation Output pane to go to its location in the script editor.

For an included script, the script name is shown in front of the line number. When you double-click the error in the Validation Output pane, the included script opens in a separate tab and the error is highlighted. Another way to open an included script is to right-click the line with the include tag in the script editor and then select **Open scriptname**.

For a script in a referenced application, including the Global Application, the script opens in read-only mode. Click the yellow alert bar at the top of the script editor to open the related process app so you can edit the script.

> **Note:** The Validation Output pane also opens if you double-click a script error in the Validation Results. (Script errors are included in the Validation Results if you select the **Validate Scripts during Process App validation** check box on the General Options [page 479] tab of the SBM Composer Options [page 479] dialog box.)