

The MITRATECH logo consists of the word "MITRATECH" in a white, uppercase, sans-serif font, centered within a solid teal rectangular background.

MITRATECH

## **TeamConnect® Screen Designer 5.0**

© 2017 Mitrtech

User Guide

## TeamConnect® Screen Designer 5.0 User Guide

Document ID: tc\_sd\_5\_0\_user\_1, published on 9/11/2017

Copyright © 2017, Mitrtech Holdings, Inc. All rights reserved.

### Disclaimer of Warranty

Mitrtech Holdings, Inc. (Mitrtech) makes no representations or warranties, either expressed or implied, by or with respect to anything in this document, and shall not be liable for any implied warranties of merchantability or fitness for a particular purpose or for any indirect, special or consequential damages.

Mitrtech reserves the right to not support non-standard or non-default functionality and extended functionality available in third-party software, unless specifically documented as supported or certified in the Mitrtech product documentation. For further information regarding third-party non-standard or non-default functionality, please contact Mitrtech Support.

This document, along with the software that it describes, is furnished under license and may be used or copied only in accordance with the terms of such license. The content of this document is furnished for informational use only, is subject to change without notice, and should not be construed as commitment by Mitrtech.

The following document is for the TeamConnect™ Screen Designer 1.0 Update 1 release only. Though every effort was made to ensure that the information in this document is correct and reliable, Mitrtech does not assume any liability for any errors encountered in this document.

If you need support for TeamConnect™ Screen Designer 1.0 Update 1, please contact the Mitrtech support team by sending an email to: [support@mitrtech.com](mailto:support@mitrtech.com). For more information about Mitrtech, visit our web site: <http://www.mitrtech.com>.

"Mitrtech", TeamConnect™ Enterprise, TeamConnect™ Legal, TeamConnect™ Legal Matter Management, Collaborati®, TeamConnect™ Collaborati Spend Management®, TeamConnect™ Deadlines, TeamConnect™ AP Link, TeamConnect™ Office Suite, TeamConnect™ Legal Reports, TeamConnect™ SOP Manager, TeamConnect™ Financial Management, TeamConnect™ Screen Designer, and TeamConnect™ Upgrade Toolkit are trademarks and products of Mitrtech Holdings, Inc. All other products or services mentioned in this book are the trademarks or service marks of their respective companies or organizations.

### GOVERNMENT RIGHTS LEGEND:

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the applicable Mitrtech license agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013 (Feb 2012) FAR 12.212(a) (1995), FAR 52.227-19 (Dec 2007), or FAR 52.227-14, as applicable.

### CONTACT US:

Mitrtech Holdings, Inc.  
5001 Plaza on the Lake Suite 111, Austin, TX 78746  
Phone: (512) 382-7322

NOTE: Throughout Mitrtech product publications, in addition to using full product names where necessary, we also use familiar and shorter terms to increase your ease of reading. You may find the following aliases for our product names:

TeamConnect for TeamConnect Enterprise  
Matter Management for TeamConnect Legal Matter Management  
TeamConnect Legal for TeamConnect Legal Matter Management  
CSM for TeamConnect Collaborati Spend Management  
Collaborati Spend Management for TeamConnect Collaborati Spend Management  
SOP or SOP Manager for TeamConnect SOP Manager  
Legal Hold for TeamConnect Legal Hold  
Legal Reports for TeamConnect Legal Reports  
Deadlines for TeamConnect Deadlines  
AP Link for TeamConnect AP Link  
Office Suite for TeamConnect Office Suite  
Financial Management for TeamConnect Financial Management  
Screen Designer for TeamConnect Screen Designer  
Upgrade Toolkit for TeamConnect Upgrade Toolkit

# Acknowledgements

This product includes software developed by the following organizations:

Apache Software Foundation (<http://www.apache.org/>)

OpenSymphony Group (<http://www.opensymphony.com/>).

The license agreements for these and other supplemental software packages can be found in your installation media in subfolder Supplemental\_Software\_Licenses. That subfolder also contains Open Source Components.pdf, which lists the locations, license types, and specific versions of components that are available on the web.

# Table of Contents

<b>Part I Screen Designer Help</b>	<b>5</b>
1 Logging In .....	5
Using Profiles .....	5
2 Using Screen Designer .....	7
User Interface .....	8
System Fields and Custom Fields .....	10
Elements and Reports .....	11
List of Elements.....	13
Java Classes .....	17
Editing Properties .....	18
List of Properties.....	19
Interactions .....	40
Designing Drill-Down Interactions.....	42
Saving and Publishing Custom Blocks .....	42
Using Rows in Screen Designer .....	43
3 Screen Designer Walkthrough .....	43
Getting Started with Screen Designer .....	44
Creating a New Block .....	45
<b>Index</b>	<b>46</b>

# 1 Screen Designer Help

Welcome to the *TeamConnect® Screen Designer User Help*.

This documentation is meant for people who need an overview of how to use TeamConnect Screen Designer. It describes the product requirements and the installation and setup process, as well as how to use the tool to build custom blocks.

The topics are organized by the major tasks that you must perform when setting up and using TeamConnect Screen Designer.

## 1.1 Logging In

When launched, Screen Designer opens to a Login Screen. You must enter your TeamConnect details to log in and use Screen Designer.

### To log in to Screen Designer

1. Launch the Screen Designer application. The Login Screen is displayed.
2. Enter your TeamConnect details in the **User Name**, **Password**, and **Instance URL** fields.
  - To save your entered details as a Profile, select the **Save Profile** check-box. Selecting this check-box makes the **Profile Name** field appear.
    - Enter a name in the **Profile Name** field.
  - If a Profile already exists in Screen Designer, you may select it from the **Profile** drop-down list. This will automatically populate the **User Name** and **Server** fields. The **Password** field will need to be entered manually for security purposes. See [Using Profiles](#) for more information.
3. Click the **OK** button.
  - a. If any of your login details is incorrect, an error message will be displayed and you will be prompted to correct the details and log in again.

**Important:** Screen Designer needs `/webservice/rest` appended to the end of the TeamConnect Instance URL to log in. For example, you would enter `http://<YourDomain.com>/<path to TeamConnect>/webservice/rest` in the Instance URL field on the Login Screen.

### 1.1.1 Using Profiles

Profiles save the User Name and URL of TeamConnect instances, making it easier to quickly log in to different user accounts or TeamConnect instances when working with Screen Designer.

You can create a Profile when logging in to TeamConnect Screen Designer. See [Logging In](#) for more information.

**Important:** Screen Designer needs `/webservice/rest` appended to the end of the TeamConnect Instance URL to log in. For example, you would enter <http://YourDomain.com/TeamConnect/webservice/rest> in the Instance URL field on the Saved Profiles window.

### To create Profiles in Screen Designer

1. After logging into Screen Designer, click the **Profile** button in the top-right corner of the interface. By default, this button will display the URL of your TeamConnect instance.
2. Click **Manage Profiles**. The Saved Profiles window is displayed.
3. Click the **New** button.
4. Enter a name for your **Profile Name**. Enter the **User Name** and **Instance URL** fields with your TeamConnect login information.
5. Click the **Save** button.

### To manage Profiles in Screen Designer

1. After logging into Screen Designer, click the **Profile** button in the top-right corner of the interface.
2. Click **Manage Profiles**. The Saved Profiles window is displayed.
  - To edit a Profile, click its name in the **Profiles** list. Click the **Clear** button if you want to reset all the fields in the Profile. Edit the fields in the **Profile Info** section and click **Save**.
  - To delete a Profile, click its name in the **Profiles** list. Click the **Delete** button. Click **OK** on the confirmation prompt.

### To switch Profiles in Screen Designer

1. After logging into Screen Designer, click the **Profile** button in the top-right corner of the interface.
  - Make sure you have saved your work before switching Profiles.
2. Click **Switch Profiles**. The Login Screen is displayed.
3. Select a Profile from the drop-down list as you would when logging in to Screen Designer.
4. Enter your TeamConnect password in the **Password** field.
5. Click the **OK** button.

## 1.2 Using Screen Designer

The available fields and elements in Screen Designer are contained in panes located by default on the right side of the Screen Designer interface. Depending on the project and object type, the following types of items will be available for your custom block:

- System Fields
- Custom Fields
- Elements, Advanced Elements, HTML Elements, and JSP Elements
- Reports
- Wizard Parameters
- Java Classes

Click on the corresponding pane tab to see the list of available fields or elements of that type.

The general process for adding, moving, and deleting fields or elements of any type is described below. For more information on specific types of elements or fields, see [System Fields and Custom Fields](#) and [Elements and Reports](#). For more information on properties and adding further functionality to your custom blocks, see [Editing Properties](#) and [Interactions](#).

### To add fields to a custom screen

1. Click on the tab of the field type you wish to use, **System Fields** or **Custom Fields**.
2. For each field that you wish to use, drag it from the **System Fields** or **Custom Fields** pane to your desired position in the custom screen canvas.
  - The container you drag your field into will be highlighted in green if it is a valid location for your field. If it is not a valid location for your field, the container will be highlighted in red.
3. Edit the field's properties in the **Properties** pane.

### To edit fields on a custom screen

1. Click on a field or element in the custom screen canvas to select it.
2. Edit the field's properties in the **Properties** pane.

### To edit multiple fields on a custom screen

1. Hold down the **Ctrl** key and click on fields or elements to select multiple at the same time. Hold down the **Ctrl** key and click on selected fields or elements to deselect them.
  - a. Alternatively, click and drag a box around all the fields and elements you want to edit at the same time. Note that the box you drag must fully encompass the fields and elements you want to select.
2. Common **Properties** fields of the selected fields will appear in the **Properties** pane. Make modifications in the **Properties** pane to choose display formats and other attributes.

### To move fields on a custom screen

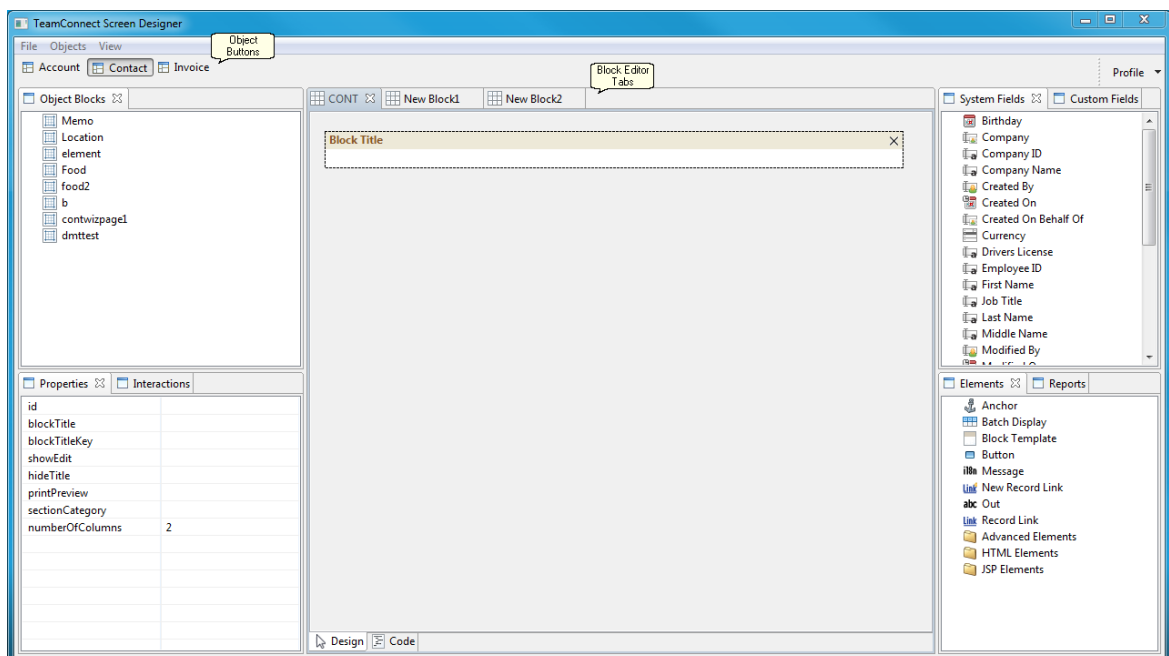
1. Move your mouse cursor over the label text of the field in the canvas you want to move. Your cursor icon will change to the move icon.
2. Click and drag the field to your new desired location.
  - The container you drag your field into will be highlighted in green if it is a valid location for your field. If it is not a valid location for your field, the container will be highlighted in red.

### To delete fields on a custom screen

1. Click the **X** icon in the top-right corner of the field you want to delete from the canvas.

## 1.2.1 User Interface

The sections below detail the different components that make up the Screen Designer interface.



The Screen Designer interface.

## Menus

- **File**—This menu contains commands for creating, opening, saving, and publishing custom pages in Screen Designer.
  - **New Block**—Creates a new custom block of the same current object type.
  - **Open Block**—Opens a saved custom block.
  - **Save Block**—Saves the current custom block as a TeamConnect Screen (\*.scr.xml) file. See [Saving and Publishing Custom Blocks](#) for more information.



- **Publish Block**—Publishes the current custom block to the TeamConnect instance. See [Saving and Publishing Custom Blocks](#) for more information.
- **Refresh**—Refreshes the Screen Designer with the most up-to-date custom fields, reports, java classes, and wizard parameters in TeamConnect.
- **Exit**—Exits Screen Designer.
- **Objects**—This menu contains a list of all object types in TeamConnect, including custom object types. Clicking on an object type from this menu will open a new custom block of that type, or switch focus to the custom block of that object type if one is already open in Screen Designer.
- **View**—This menu contains a list of all the Screen Designer panes. If a pane is closed, you can select it from this menu to open it again.

## Object Buttons

These buttons switch between the different object types you currently have open in Screen Designer. You can rearrange the order of the Object buttons by dragging and dropping them.

## Profile Button

This button opens a menu with the **Switch Profiles** and **Manage Profiles** options. See [Using Profiles](#) for more information.

## Panes

The panes contain elements and fields and their associated properties, interactions, and more. They are located by default on the left and right sides of Screen Designer. You can drag them around to your preferred location in the Screen Designer interface. You can also close any panes, then re-open them from the **View** menu.

- **Object Blocks Pane**—Located by default in the top-left corner of Screen Designer. Contains all the custom blocks of the current object type that are published to the TeamConnect instance. Double-click an item in this list to open the published custom block in Screen Designer.
- **Properties Pane**—Located by default in the bottom-left corner of Screen Designer. Contains all the properties of the selected field in the canvas. See [Editing Properties](#) for more information.
- **Interactions Pane**—Located by default in the bottom-left corner of Screen Designer. Contains the interface for adding interactions to fields. See [Interactions](#) for more information.
- **System Fields Pane**—Located by default in the upper-right corner of Screen Designer. Contains the default system fields associated with the current object type. See [System Fields and Custom Fields](#) for more information.
- **Custom Fields Pane**—Located by default in the upper-right corner of Screen Designer. Contains the custom fields associated with the current object type. See [System Fields and Custom Fields](#) for more information.
- **Wizard Parameters Pane**—Not displayed by default. Select **Wizard Parameters** from the **View** menu to open this pane. Contains all the wizard parameters that are available for the current object type.

- **Elements Pane**—Located by default in the bottom-right corner of Screen Designer. Contains page elements such as anchors, block templates, buttons, and more. See [Elements and Reports](#) for more information.
- **Reports Pane**—Located by default in the bottom-right corner of Screen Designer. Contains reports that are saved in TeamConnect. See [Reports](#) for more information.
- **Java Classes Pane**—Not displayed by default. Select **Java Classes** from the **View** menu to open this pane. Contains all the Java classes that are available for the current object type.

## Canvas

The canvas is the main workspace of Screen Designer. Drag fields and elements into this area to place them into your custom block. See [Using Screen Designer](#) for more information on adding, moving, and deleting fields or elements to the canvas.

Above the canvas are Block Editor Tabs, which represent each of your open custom blocks for the object type you are currently working on. You can rearrange the order of the Block Editor Tabs by clicking and dragging them to their new position. You can also view two or more custom blocks at a time by clicking and dragging a Block Editor Tab to the left, right, top, or bottom of the canvas to "split" the canvas view.

***Important:** The tool opens with the **Design** tab visible (located below the Screen Designer canvas for each Block Editor Tab). There is also a **Code** tab, which displays a read-only list of the XML code that is being generated while you use the tool. You may find the code listing helpful for verification purposes. All of your design actions will occur in the **Design** tab, not the **Code** tab.*

### 1.2.2 System Fields and Custom Fields

System Fields link to data fields in the TeamConnect database, allowing users of your custom screen to edit and update the values in these fields. The System Fields available in Screen Designer vary based on which Object Type the custom screen is being made for.

#### To add System Fields to a custom screen

1. Click on the **System Fields** tab. A list of system field names available for use in the block is displayed in the **System Fields** pane.
2. For each field that you want to use, drag it from the **System Fields** pane to your desired position in the canvas.

When a field is being worked on in the layout, the fields in the **Properties** pane can be edited to choose display formats and other attributes.

***Note:** The tool enforces only the most basic validation, such as not allowing characters in numeric fields. As a result, the tool may accept property edits which might not work at runtime, or might cause an error in more sophisticated validations. For example, if you edit the value property of a field to have a default value at runtime, but the field is one which will always be populated by a non-default value automatically, your edit will be accepted, yet will have no effect at runtime.*

3. If you want a field to be in a new position in the layout, simply drag it to the new position. If you don't want a field to appear in the layout, click the small **X** icon next to the field to delete it from the layout.

**Note:** Fields “collapse upward” to fill empty spaces in the layout. If you leave empty fields in a column, the populated fields below the empty ones will move up so that there are no empty positions in the layout at runtime.

4. If the **Properties** pane is visible, it will always show the properties of whichever field has focus. Properties vary depending upon the datatype of the field that has focus.

Custom Fields are created within object definitions under various categories, and are intended for storing your organization's custom information in TeamConnect. Using Screen Designer, you can add Custom Fields to your custom screens. For more information on creating Custom Fields in TeamConnect, see Creating Custom Fields.

#### To add Custom Fields to a custom screen

1. Click on the **Custom Fields** tab. A list of custom field names available for use in the block is displayed in the **Custom Fields** pane.
  - The category of the block template is set by the first Custom Field added to the block.
2. For each field that you wish to use, drag it from the **Custom Fields** pane to your desired position in the workspace area.

When a field is being worked on in the layout, the fields in the **Properties** pane can be edited to choose display formats and other attributes.

**Note:** A block can only be associated with one category. As soon as you drag a custom field from the Custom Fields pane into the layout, the block becomes linked to that custom field's category. From then on, only custom fields from that category can be used in that block.

*If you want to change the category that a block is associated with, you must first delete all custom fields from the layout before choosing the new category. Except for these considerations, editing the layout and properties of Custom Fields works just as it does for System Fields.*

3. If you want a field to be in a new position in the layout, simply drag it to the new position. If you don't want a field to appear in the layout, click the small X icon next to the field to delete it from the layout.

### 1.2.3 Elements and Reports

Elements can be added to the Screen Designer canvas. Elements are restricted to just UI elements that look like System Fields and Custom Fields, but are not based on the object type. They are work fields that you create and control. Usually several properties, including **label** and **name**, will require editing after adding an element to the canvas. Some properties, such as **onClick**, have text values that are intended to contain JavaScript.

- **Anchor**—A link to an internal or external address. Can contain the Message and Out elements.
- **Batch Display**—The main use case for Batch Displays is adding a table in which you can create, view, and update child or embedded objects. Additionally, it can pull tabular data connected by a one-to-many relationship with the associated object, for creating, viewing, and updating the tabular data.
- **Block Template**—A container to group any amount of fields in a custom block.
- **Button**—An interactive element that will trigger an event based on the Screen Designer user specification.
- **Message**—Outputs a localized message given an i18n key. See Creating Custom Messages.
- **Message Parameter**—Inserts dynamic text into Messages. Message Parameters can only be placed in a Message element.
- **New Record Link**—Creates a new record of the specified object type.
- **Out**—Outputs a message in the block.
- **Record Link**—Links to a record of a specified object type.

## Advanced Elements

Advanced Elements allow additional functionality to TeamConnect fields. For example, if you need a relationship between one field and another field, you can connect them with Java Class files. However, you cannot create a brand new field from the Advanced Elements.

*Note: Advanced Elements must be attached to a Java Class to function.*

- **Category**—Displays a list of CategoryItemTreeItem objects returned from a Java class method. Displays the default category by default.
- **Checkbox**—A box that users can select or de-select to produce a Boolean value.
- **Date & Time Field**—A field for entering date and time information.
- **Date Field**—A field for entering date information.
- **Lookup**—A drop-down menu that lists the contents of a lookup table or a list method in a custom Java class.
- **Memo Text**—A text field that can hold more text than the Text Field Advanced Element. The limit for Memo Text fields is more than 200,000 characters.
- **Multi-Value List**—A multi-select box that lists the contents of a lookup table or a list method in a custom Java class.
- **Number Field**—A field that stores a numerical value.
- **Radio Button**—A list of radio buttons, wherein users can select only one radio button at a time.
- **Select**—A drop-down menu that allows the user to use a custom list force selection.
- **Text Field**—A text field that can hold up to 2,000 characters. The character limit of the input field can be defined in Screen Designer.

## HTML Elements

HTML Elements allow you to insert HTML and JavaScript code directly into your custom block.

- **Html**—Allows the user to add HTML elements to the custom block.
- **Script**—Loads an embedded script onto the page.

## JSP Elements

JSP Elements allow you to add logic to custom blocks utilizing a Java class. They require either a system Java class or custom Java class to work. System Java classes can be found in the Java docs.

- **If Statement**—Attaches a Boolean condition through a Java class. If that condition evaluates to **true**, then the contents of the tag are shown/loaded.
- **If-Else Statement**—Attaches a Boolean condition through a Java class, along with content to load when the condition is **true** and different content to load when the condition is **false**.
- **For Each**—Attaches to a List field in a Java class. Iterates through each item in that list, repeating some given content for each item (i.e. creating a label and a value for each item).

## Reports

Reports are placed in the canvas in the same manner as elements. They can be used to display report output, or links to reports, within a block. The **Reports** pane contains a hierarchical list of reports within folders. The report's properties will appear in the **Properties** pane after dragging a report into the canvas. The value of the link property determines whether the report's output, or a link to the report instead, appears in the block at runtime.

You can also drag a report folder into the canvas. In this case, only the report link is available in the block, not the report output. At runtime, when the link is clicked, that folder's contents (report names and subfolders) will appear in a Reports page. Parameters are not used with links to report folders.

### 1.2.3.1 List of Elements

Below are tables containing all the elements, advanced elements, HTML elements, and JSP elements in Screen Designer, the requirements for each element type, and example uses for each element.

#### Elements

Element	Description	Required Properties	Example Use
<b>Anchor</b>	A link to an internal or external address. Must contain either a <b>Message</b> or <b>Out</b> element.	<b>Message</b> or <b>Out</b> element	Linking to an external page or to a different section of the current page.

<b>Block Template</b>	A container to group any amount of fields in a custom block.	<b>numberOfColumns</b>	Delineating different sections of a screen from each other. Most screens will have one block template. Multiple ones can be added to a single screen if you want to make two sections look like two separate blocks to the user, or if you want to have different versions of a single block that are dependent upon the category of the record (for example, showing different custom fields that are relevant only to a particular category of records).
<b>Button</b>	An interactive element that will trigger an event based on the Screen Designer user specification.	N/A	Allowing the user to trigger some action.
<b>Empty</b>	Enters an empty <div> element to add spacing to the screen.	N/A	Adding more space between rows for better UI.
<b>Message</b>	Outputs a localized message given an i18n key. See Creating Custom Messages.	<b>key</b>	Displaying localized informational or cautionary messages on the screen. Identical to <b>out</b> tags except the message will be localized. Generally, this is preferred and should be used in place of <b>out</b> tags.
<b>Message Parameter</b>	Inserts dynamic text into Messages. Message Parameters can only be placed in a Message element.	<b>value</b>	Creating error messages that contain dynamic values that are defined by the Message Parameter element.
<b>New Record Link</b>	Creates a new record of the specified object type.	<b>entityCode</b>	Adding a "Create <Object>" link to a screen.
<b>Out</b>	Outputs a message in the block.	<b>value</b>	Displaying informational or cautionary messages on the screen. Identical to <b>message</b>

			tags except that the message will not be localized. Generally, <b>message</b> tags should be preferred over <b>out</b> tags.
<b>Record Link</b>	Links to a record of a specified object type.	<b>eoObject</b>	Linking to some other object.
<b>Row</b>	Adds a row for housing fields to your screen. Fields require a row to be placed on the screen.	<b>numberOfColumns</b>	Added to a screen for fields to be placed on the canvas.

**Advanced Elements**

<b>Element</b>	<b>Description</b>	<b>Required Properties</b>	<b>Example Use</b>
<b>Batch Display</b>	The main use case for Batch Displays is adding a table in which you can create, view, and update child or embedded objects. Additionally, it can pull tabular data connected by a one-to-many relationship with the associated object, for creating, viewing, and updating the tabular data.	<b>batchDisplayObject</b>	Displaying embedded objects in a single table. Displaying arbitrary tabular data pulled from a custom Java class.
<b>Category</b>	Displays a list of CategoryItemTreeItem objects returned from a Java class method. Displays the default category by default.	<b>name</b>	Displaying a list of categories from a custom Java class method.
<b>Checkbox</b>	A box that users can select or de-select to produce a Boolean value.	<b>name</b>	Linking to a field on a related object via its path, or via a custom Java class method.
<b>Date &amp; Time Field</b>	A field for entering date and time information.	<b>name</b>	Linking to a field on a related object via its path, or via a custom Java class method.

<b>Date Field</b>	A field for entering date information.	<b>name</b>	Linking to a field on a related object via its path, or via a custom Java class method.
<b>Lookup</b>	A drop-down menu that lists the contents of a lookup table or a list method in a custom Java class.	<b>name</b>	Linking to a field on a related object via its path, or via a custom Java class method.
<b>Memo Text</b>	A text field that can hold more text than the <b>Text Field</b> Advanced Element. The limit for <b>Memo Text</b> fields is more than 200,000 characters.	<b>name</b>	Linking to a field on a related object via its path, or via a custom Java class method.
<b>Multi-Value List</b>	A multi-select box that lists the contents of a lookup table or a list method in a custom Java class.	<b>name</b>	Linking to a field on a related object via its path, or via a custom Java class method.
<b>Number Field</b>	A field that stores a numerical value.	<b>name</b>	Linking to a field on a related object via its path, or via a custom Java class method.
<b>Radio Button</b>	A list of radio buttons, wherein users can select only one radio button at a time.	<b>name</b>	Linking to a field on a related object via its path, or via a custom Java class method.
<b>Select</b>	A drop-down menu that allows the user to use a custom list force selection.	<b>name</b>	Linking to a field on a related object via its path, or via a custom Java class method.
<b>Text Field</b>	A text field that can hold up to 2,000 characters. The character limit of the input field can be defined in Screen Designer.	<b>name</b>	Linking to a field on a related object via its path, or via a custom Java class method.

**HTML Elements**

<b>Element</b>	<b>Description</b>	<b>Required</b>	<b>Example Use</b>
----------------	--------------------	-----------------	--------------------



		Properties	
<b>Html</b>	Allows the user to add HTML elements to the custom block.	N/A	Inserting raw HTML into the screen in order.
<b>Script</b>	Loads an embedded script onto the page.	N/A	Inserting custom JavaScript into the screen, or linking to an external JavaScript file.

**JSP Elements**

Element	Description	Required Properties	Example Use
<b>If Statement</b>	Attaches a Boolean condition through a Java class. If that condition evaluates to true, then the contents of the tag are shown/loaded.	<b>test</b>	Showing a group of fields only if the record has a particular flag set, or showing a field only if the field already has a value.
<b>If-Else Statement</b>	Attaches a Boolean condition through a Java class, along with content to load when the condition is true and different content to load when the condition is false.	<b>test</b>	Making a group of fields non-editable if a record is past a certain date. Otherwise, allowing the fields to be edited.
<b>For Each</b>	Attaches to a List field in a Java class. Iterates through each item in that list, repeating some given content for each item (i.e. creating a label and a value for each item).	<b>var, items</b>	Showing a repeated set of data for a list of related objects

**1.2.4 Java Classes**

Java classes are custom files that are designed to customize functionality. Some examples of their use are generating conditions for if statements, generating batch displays, and populating advanced elements

For more information on creating custom Java classes, see *Creating Custom Pages* in the TeamConnect Enterprise Customization Help, and the Developer's Help for information on the TeamConnect API.

#### To attach a Java Class to a screen

1. Click on the **Java Classes** tab. A list of system field names available for use in the block is displayed in the **Java Classes** pane.

This tab is not displayed by default. Select **Java Classes** from the **View** drop-down menu to make this tab appear.

2. For each class that you want to use, drag it from the **Java Classes** pane to your desired position in the canvas. Java classes can be added to the following screen elements:
  - Block Templates
  - Screens (the canvas outside of Block Templates) when there is more than one Block Template
    - In this case, the Java class will be available to all Block Templates on the screen.
  - If Statements
    - In this case, the Java class will only be available under the conditions of the If Statement.
    - Another Java class outside the If Statement is required for the If Statement to function.
3. In the **Properties** pane, enter an **id** for the Java Class.
4. Add a reference to your added Java Class in properties of other desired elements.
  - Any other element that references the added Java Class must reference it by the **id** you enter in step 3. This will be of the form `${JavaClassID.methodName}`.

## 1.2.5 Editing Properties

Most fields and elements in Screen Designer can be customized further by editing their properties. Properties vary depending upon the object type of the field or element that has focus.

You can edit properties of elements and fields by clicking on them in the canvas. The **Properties** pane will always display the properties of whichever field or element has focus in Screen Designer. If more than one field or element is selected at a time, the **Properties** pane will only display the properties that are shared between the selected items.

As of Screen Designer 5.1, field label text can be changed from within the application (rather than solely via XML edits). Changes made in this way will not however be localized for other languages.

#### To edit properties on a field or element

1. Click the field or element in the canvas that you want to edit the properties of. The properties of the field or element appear in the **Properties** pane.

- a. You can edit the properties of multiple fields or elements by selecting more than one at a time, as described in [To edit multiple fields on a screen](#).
2. Adjust the desired properties.
  - a. Properties that have a Boolean datatype present a text box in which you type the value. Valid choices for this text box are shown below.
    - For 'true' the acceptable values are: "true", "yes" (case-insensitive).
    - For 'false' the acceptable values are: "false", "no" (case-insensitive).
    - If a JEXL expression is used, it may return any of the above strings, or a Java Boolean.

You can edit font properties on elements or fields that display some kind of text in view mode. For example, you can modify the properties of text fields, number fields, Memo fields, drop-down lists and more to alter the font type, font size, and font color of the displayed text in your custom screen.

#### To edit font properties on an element or field

1. Click the field or element in the canvas that you want to edit the properties of. The properties of the field or element appear in the **Properties** pane.
  - a. You can edit the properties of multiple fields or elements by selecting more than one at a time, as described in [To edit multiple fields on a screen](#).
2. Adjust the following font properties as desired:
  - a. **fontFamily**—A drop-down list of common font types.
  - b. **fontSize**—A number field that defines the size of the affected text. The default `fontSize` value in TeamConnect is 11.
  - c. **fontColor**—A pop-up window with a selection of common colors, as well as an advanced interface to select colors from the full RGB spectrum.

#### 1.2.5.1 List of Properties

Below is a table containing all the element properties in Screen Designer, the elements associated with each property, the data type of each property, and an explanation of each property.

Property	Elements	Data Type	Tooltip	Example Data
<b>aggressiveCacheMultiplier</b>	<ul style="list-style-type: none"> <li>• Batch Display</li> </ul>	Text	This value is multiplied by the number of rows originally requested from the database, and defines how many rows will be requested and cached	1, 10, or 50

			through the batch display. Defaults to '1'.	
<b>allowNullValue</b>	<ul style="list-style-type: none"> <li>• Category</li> <li>• Lookup</li> <li>• Multi-Value List</li> <li>• Select</li> <li>• Wizard Parameter (Dropdown)</li> </ul>	Boolean	Allows a null value. Set to True to enable.	True, False
<b>batchDisplayObject</b>	<ul style="list-style-type: none"> <li>• Batch Display</li> </ul>	Text	This property references the id of the useClass tag that should be associated with this batchDisplay.	useclassId
<b>blockTitle</b>	<ul style="list-style-type: none"> <li>• Block Template</li> </ul>	Text	Title of block template.	Welcome to the Block
<b>blockTitleKey</b>	<ul style="list-style-type: none"> <li>• Block Template</li> </ul>	Text	The i18nKey for block title.	some.i18n.key
<b>category</b>	<ul style="list-style-type: none"> <li>• Multi-Value List</li> </ul>	N/A	Non-editable property associated with custom fields. This is the category which the custom field is under.	INVC
<b>checked</b>	<ul style="list-style-type: none"> <li>• Checkbox</li> </ul>	Text	Defines whether or not field defaults to selected.	True, False
<b>colon</b>	<ul style="list-style-type: none"> <li>• True/False</li> </ul>	Text drop-down	Determines whether a colon is present after the label. Defaults to true.	True, False
<b>cols</b>	<ul style="list-style-type: none"> <li>• Memo Text</li> </ul>	Text	Defines the visible width of a text area (memo text).	1, 5, or 10
<b>containerID</b>	<ul style="list-style-type: none"> <li>• Number Aggregate Field</li> </ul>	Text	References the ID of the number column to be aggregated by the BatchDisplayNumberAggregateField.	numberColumnId
<b>cssClass</b>	<ul style="list-style-type: none"> <li>• Button</li> </ul>	Text	CSS class to use for field.	some_css_class

	<ul style="list-style-type: none"> <li>• Message</li> <li>• Out</li> </ul>			
<b>defaultBlankRows</b>	<ul style="list-style-type: none"> <li>• Batch Display</li> </ul>	Text	Defines the number of blank rows that appear for entry into batch display.	1, 5, or 10
<b>disabled</b>	<ul style="list-style-type: none"> <li>• Button</li> <li>• Select</li> </ul>	Boolean	Disables field if set to True.	True, False
<b>displayNamePath</b>	<ul style="list-style-type: none"> <li>• Select</li> </ul>	Text	This is the path of the value to display in the select field. Defaults to 'displayString'.	enterpriseEntity.fieldName
<b>editorAllowEmptyDisplay</b>	<ul style="list-style-type: none"> <li>• Category Column</li> <li>• Custom Lookup Column</li> <li>• Select Column</li> <li>• System Lookup Column</li> </ul>	Boolean	Defines whether to display an empty field at the top of a drop-down list. Set to True to enable.	True, False
<b>editorDefault</b>	<ul style="list-style-type: none"> <li>• Category Column</li> <li>• Checkbox Column</li> <li>• Contact Column</li> <li>• Custom Lookup Column</li> <li>• Date Column</li> <li>• Number Column</li> <li>• Project Column</li> <li>• Select Column</li> <li>• System Lookup Column</li> <li>• Text Column</li> </ul>	Text	The default value shown in a column editor within a batch display. For example: For a check-box column within a batch display, this could be set to True, False, or an expression to handle via JavaScript.	Depends on the specific column. For example, True or False for a check-box, but a tree position for a lookup column or category column, and any text for a text column.

	<ul style="list-style-type: none"> <li>• User Column</li> </ul>			
<b>editorDependentItemsMap</b>	<ul style="list-style-type: none"> <li>• Select Column</li> </ul>	Text	Allows the use of run-time created drop-down lists that are not existing system or custom lookup tables via the expression passed into this property. The value for this property should be in expression form, '\${}', and reference a method in the related Java class that returns a map of lists. If this property has a non-null value, then editorItems must be null, and vice-versa.	<code>#{cjb.items}</code>
<b>editorDependentItemsParentFieldId</b>	<ul style="list-style-type: none"> <li>• Select Column</li> </ul>	Text	String ID of the parent field within the batch display. Required when editorDependentItemsMap is used.	parentField
<b>editorEmptyDisplay</b>	<ul style="list-style-type: none"> <li>• Category Column</li> <li>• Custom Lookup Column</li> <li>• Select Column</li> <li>• System Lookup Column</li> </ul>	Text	String to display in an editor drop-down list within a batch display column when no value is currently selected.	(Select)
<b>editorEmptyDisplay18nKey</b>	<ul style="list-style-type: none"> <li>• Category Column</li> <li>• Custom Lookup Column</li> <li>• Select Column</li> <li>• System Lookup Column</li> </ul>	Text	The 18nKey for the string to display in an editor drop-down list within a batch display column when no value is currently selected. Defaults to 'common.select', which displays "Select".	common.select
<b>editorEntityCode</b>	<ul style="list-style-type: none"> <li>• Category Column</li> </ul>	Text	Unique object code corresponding to desired categories.	INVC

<b>editorFilterEntityCode</b>	<ul style="list-style-type: none"> <li>Project Column</li> </ul>	Text	Unique code of the project, only used for project column, batch display.	DISP
<b>editorFilterParentItemUniqueCode</b>	<ul style="list-style-type: none"> <li>Category</li> <li>Category Column</li> </ul>	Text	Unique code of a parent. Will only display children of this parent.	DISP
<b>editorFilterSearchDefinitionKey</b>	<ul style="list-style-type: none"> <li>User Column</li> </ul>	Text	Specific to searches for batch display user columns. Sets the search module to be used.	searchKey
<b>editorItems</b>	<ul style="list-style-type: none"> <li>Select Column</li> </ul>	Text	Allows the use of run-time created drop-down lists that are not existing system or custom lookup tables via the expression passed into this property. The value for this property should be in expression form, '\${}', and reference a method in the related Java class that returns the desired list. If this property has a non-null value, then editorDependentItemsMap must be null, and vice-versa.	\${cjb.items}
<b>editorMaxLength</b>	<ul style="list-style-type: none"> <li>Text Column</li> </ul>	Text	Defines the maximum length to accept in text editor.	10, 50, or 250
<b>editorOnChangeRefreshFormValues</b>	<ul style="list-style-type: none"> <li>Category Column</li> <li>Checkbox Column</li> <li>Contact Column</li> <li>Custom Lookup Column</li> <li>Date Column</li> <li>Number Column</li> </ul>	Boolean	Defines whether the page refreshes after a change to the field. Set to True to enable.	True, False

	<ul style="list-style-type: none"> <li>• Project Column</li> <li>• Select Column</li> <li>• System Lookup Column</li> <li>• Text Column</li> <li>• User Column</li> </ul>			
<b>editorSelectEmptyDisplay</b>	<ul style="list-style-type: none"> <li>• Contact Column</li> <li>• Project Column</li> <li>• User Column</li> </ul>	Text	String to display in an application entity-based column when the field is empty or null. If editorUseSelect is set to 'True', then this will be used to determine the value of the drop-down list when no item is selected. This attribute cannot be used when the required attribute is 'True' on the column.	(none)
<b>editorSelectEmptyDisplayI18nKey</b>	<ul style="list-style-type: none"> <li>• Contact Column</li> <li>• Project Column</li> <li>• User Column</li> </ul>	Text	The i18nKey for the string to display in an application entity-based column when the field is empty or null. If editorUseSelect is set to 'True', then this will be used to determine the value of the drop-down list when no item is selected. This attribute cannot be used when the required attribute is 'True' on the column.	common.none
<b>editorSelectItems</b>	<ul style="list-style-type: none"> <li>• Contact Column</li> <li>• Project Column</li> <li>• User Column</li> </ul>	Text	List of entities to use for the selection.	<code>#{cjb.items}</code>
<b>editorSize</b>	<ul style="list-style-type: none"> <li>• Contact Column</li> <li>• Project Column</li> <li>• Text Column</li> </ul>	Text	Text editor size. Defaults to browser value if not set.	10, 50, or 250



	<ul style="list-style-type: none"> <li>User Column</li> </ul>			
<b>editorTableCode</b>	<ul style="list-style-type: none"> <li>Custom Lookup Column</li> <li>System Lookup Column</li> </ul>	Text	Path to custom lookup table (unique code).	EXMP_FRUI_BAN A
<b>editorTextAreaCols</b>	<ul style="list-style-type: none"> <li>Text Column</li> </ul>	Text	Defines the visible width of a text area of a text field within a batch.	10, 50, or 250
<b>editorTextAreaRows</b>	<ul style="list-style-type: none"> <li>Text Column</li> </ul>	Text	Defines the number of visible rows in the text area of a text field within a batch.	1, 5, or 10
<b>editorUseSelect</b>	<ul style="list-style-type: none"> <li>Contact Column</li> <li>Project Column</li> <li>User Column</li> </ul>	Boolean	Defines whether to use select(dropdown) when dealing with an entity column. If set to True, editorSelectItems is required.	True, False
<b>editorUseTextArea</b>	<ul style="list-style-type: none"> <li>Text Column</li> </ul>	Text	Allows text area to be set.	True, False
<b>entityCode</b>	<ul style="list-style-type: none"> <li>Category</li> <li>New Record Link</li> </ul>	Text	The unique code of the category to map a New Record Link to. It is the unique code of the entity you want to create.	DISP
<b>entityToLink</b>	<ul style="list-style-type: none"> <li>New Record Link</li> </ul>	Text	Specifies if entity linked is parent or related object.	<code>#{cjb.linkRecord}</code> or <code>enterpriseEntity.pr oject</code>
<b>eoObject</b>	<ul style="list-style-type: none"> <li>Record Link</li> </ul>	Text	Refers to a getter method in a Java file that returns a TN object. (For example, <code>cjb.vendor</code> )	<code>#{cjb.myObject}</code>
<b>forceEditable</b>	<ul style="list-style-type: none"> <li>Batch Display</li> <li>Category</li> <li>Checkbox</li> </ul>	Boolean	Defines field to be editable at all times. Set to True to enable.	True, False

	<ul style="list-style-type: none"> <li>• Date &amp; Time Field</li> <li>• Date Field</li> <li>• Lookup</li> <li>• Memo Text</li> <li>• Multi-Value List</li> <li>• Number Field</li> <li>• Select</li> <li>• Text Field</li> </ul>			
<b>forceNotEditable</b>	<ul style="list-style-type: none"> <li>• Batch Display</li> <li>• Category</li> <li>• Checkbox</li> <li>• Date &amp; Time Field</li> <li>• Date Field</li> <li>• Lookup</li> <li>• Memo Text</li> <li>• Multi-Value List</li> <li>• Number Field</li> <li>• Select</li> <li>• Text Field</li> </ul>	Boolean	Defines field to be uneditable at all times. Set to True to enable.	True, False
<b>forceSearch</b>	<ul style="list-style-type: none"> <li>• Lookup</li> <li>• Multi-Value List</li> </ul>	Boolean	If 'True', the Lookup element will utilize a search field instead of populating a drop-down list.	True, False
<b>forceSelect</b>	<ul style="list-style-type: none"> <li>• Lookup</li> <li>• Multi-Value List</li> </ul>	Boolean	If 'True', the Lookup element will have its values displayed in a drop-down list.	True, False
<b>format</b>	<ul style="list-style-type: none"> <li>• Number</li> </ul>	Text	Users are able to format number fields using the following options: <ul style="list-style-type: none"> <li>• none</li> </ul>	.504

			<ul style="list-style-type: none"> <li>• %</li> <li>• currency</li> <li>• \$</li> <li>• plain</li> <li>• decimal</li> </ul>	
<b>fractionDigits</b>	<ul style="list-style-type: none"> <li>• Number Aggregate Field</li> </ul>	Text	This sets both minimumFractionDigits and maximumFractionDigits to the same value.	4
<b>hiddenInForms</b>	<ul style="list-style-type: none"> <li>• Category Column</li> <li>• Checkbox Column</li> <li>• Contact Column</li> <li>• Custom Lookup Column</li> <li>• Date Column</li> <li>• HTML Column</li> <li>• Number Column</li> <li>• Project Column</li> <li>• Select Column</li> <li>• System Lookup Column</li> <li>• Text Column</li> <li>• User Column</li> </ul>	Text	If set to True, the field referenced by the batchDisplayColumn tags will appear in the "read-only" table at the bottom, but will not appear as an editable field in the edit row.	True, False
<b>hideAddFunction</b>	<ul style="list-style-type: none"> <li>• Batch Display</li> </ul>	Boolean	Hides the Add button on batch display. Set to True to enable.	True, False
<b>hideRemoveFunction</b>	<ul style="list-style-type: none"> <li>• Batch Display</li> </ul>	Boolean	Hides the Remove button on batch display. Set to True to enable.	True, False

<b>hideTitle</b>	<ul style="list-style-type: none"> <li>• Block Template</li> </ul>	Boolean	Hides the Title from displaying at the top of the block. Set to True to enable.	True, False
<b>hideUpdateFunction</b>	<ul style="list-style-type: none"> <li>• Batch Display</li> </ul>	Boolean	Hides the Update button on batch display. Set to True to enable.	True, False
<b>horizontalTabbing</b>	<ul style="list-style-type: none"> <li>• Batch Display</li> </ul>	Boolean	Defines whether tabbing between editor fields is allowed. Set to True to enable.	True, False
<b>href</b>	<ul style="list-style-type: none"> <li>• Anchor</li> </ul>	Text	Specifies the URL the link refers to.	<a href="https://www.google.com/">https://www.google.com/</a> or #blockHTMLID
<b>id</b>	<ul style="list-style-type: none"> <li>• Anchor</li> <li>• Batch Display</li> <li>• Block Template</li> <li>• Button</li> <li>• Category</li> <li>• Checkbox</li> <li>• Date &amp; Time Field</li> <li>• Date Field</li> <li>• Lookup</li> <li>• Memo Text</li> <li>• Multi-Value List</li> <li>• New Record Link</li> <li>• Number Field</li> <li>• Select</li> <li>• Text Field</li> <li>• Wizard Parameter (Textbox)</li> </ul>	Text	Defines the HTML ID.	my_id

	<ul style="list-style-type: none"> <li>• Wizard Parameter (Dropdown)</li> </ul>			
<b>items</b>	<ul style="list-style-type: none"> <li>• For Each</li> </ul>	Text	Input should be a JSP expression, formatted <code>'\${id.method}'</code> , where 'id' is the ID assigned to a Java class within the block template, and 'method' is the name of a method within the Java class that returns a list of objects.	<code>#{cjb.items}</code>
<b>key</b>	<ul style="list-style-type: none"> <li>• Category</li> <li>• Checkbox</li> <li>• Date &amp; Time Field</li> <li>• Date Field</li> <li>• Lookup</li> <li>• Memo Text</li> <li>• Message</li> <li>• Multi-Value List</li> <li>• Number Field</li> <li>• Select</li> <li>• Text Field</li> </ul>	Text	The <code>i18nKey</code> for the label tag. The label, key, and required tags are applied to the label tag that is added when you add one of these fields.	invoice.numberString
<b>label</b>	<ul style="list-style-type: none"> <li>• Action</li> <li>• Button</li> <li>• Category</li> <li>• Category Column</li> <li>• Checkbox</li> <li>• Checkbox Column</li> <li>• Contact Column</li> <li>• Custom Lookup Column</li> </ul>	Text	Defines the value of the label for the field. This field can be changed within the application (rather than just by XML edits). <b>However, changing this field in the application will not localize for other languages.</b>	Invoice Number

	<ul style="list-style-type: none"> <li>• Date &amp; Time Field</li> <li>• Date Column</li> <li>• Date Field</li> <li>• HTML Column</li> <li>• Lookup</li> <li>• Memo Text</li> <li>• Multi-Value List</li> <li>• Number Column</li> <li>• Number Field</li> <li>• Project Column</li> <li>• Select</li> <li>• Select Column</li> <li>• System Lookup Column</li> <li>• Text Column</li> <li>• Text Field</li> <li>• User Column</li> </ul>			
<b>labelI18nKey</b>	<ul style="list-style-type: none"> <li>• Action</li> <li>• Category Column</li> <li>• Checkbox Column</li> <li>• Contact Column</li> <li>• Custom Lookup Column</li> <li>• Date Column</li> <li>• HTML Column</li> <li>• Number Column</li> <li>• Project Column</li> <li>• Select Column</li> </ul>	Text	The i18nKey of the label. This is used for the columns within a batch display	custom.labelKey

	<ul style="list-style-type: none"> <li>• System Lookup Column</li> <li>• Text Column</li> <li>• User Column</li> </ul>			
<b>labelKey</b>	<ul style="list-style-type: none"> <li>• Button</li> </ul>	Text	i18nKey for the button label.	custom.labelKey
<b>list</b>	<ul style="list-style-type: none"> <li>• Lookup</li> <li>• Multi-Value List</li> <li>• Select</li> </ul>	Text	Parameter that provides the list of elements to populate a drop-down list.	Name of a list field to pull from, either a table name (for system tables), table code (for custom tables), or cjb method
<b>maximumFractionDigits</b>	<ul style="list-style-type: none"> <li>• Number Aggregate Field</li> <li>• Number Column</li> <li>• Number Field</li> </ul>	Text	Maximum decimal places to support.	2, 3, or 4
<b>maxLength</b>	<ul style="list-style-type: none"> <li>• Number Field</li> <li>• Text Field</li> </ul>	Text	Defines the maximum number of characters that can be entered in a field.	100
<b>maxLevel</b>	<ul style="list-style-type: none"> <li>• Multi-Value List</li> </ul>	Integer	Defines the number of levels to show for a hierarchical lookup table. 0 will show all levels.	0, 1, 3, or 5
<b>minimumFractionDigits</b>	<ul style="list-style-type: none"> <li>• Number Aggregate Field</li> <li>• Number Column</li> <li>• Number Field</li> </ul>	Text	Minimum decimal places to support.	2
<b>multiple</b>	<ul style="list-style-type: none"> <li>• Category</li> <li>• Lookup</li> <li>• Multi-Value List</li> <li>• Select</li> </ul>	Boolean	Allows the selection of multiple items at once. Set to True to enable.	True, False

name		Text	Internal name of field. Not visible on screen.	command.categories["CONT"].customFields["Matter"].name or \${cjb.matter.name}
	<ul style="list-style-type: none"> <li>• Action</li> <li>• Category</li> <li>• Category Column</li> <li>• Checkbox</li> <li>• Checkbox Column</li> <li>• Contact Column</li> <li>• Custom Lookup Column</li> <li>• Date &amp; Time Field</li> <li>• Date Column</li> <li>• Date Field</li> <li>• HTML Column</li> <li>• Lookup</li> <li>• Memo Text</li> <li>• Multi-Value List</li> <li>• Number Aggregate Field</li> <li>• Number Column</li> <li>• Number Field</li> <li>• Project Column</li> <li>• Select</li> <li>• Select Column</li> <li>• System Lookup Column</li> <li>• Text Column</li> <li>• Text Field</li> <li>• User Column</li> <li>• Wizard Parameter (Textbox)</li> </ul>			



	<ul style="list-style-type: none"> <li>Wizard Parameter (Dropdown)</li> </ul>			
<b>negate</b>	<ul style="list-style-type: none"> <li>If Statement</li> </ul>	Text	Negates the condition received by the element.	True, False
<b>noResize</b>	<ul style="list-style-type: none"> <li>Memo Text</li> </ul>	Boolean	Prevents the user from resizing the memo text area in internet explorer. Defaults to False.	True, False
<b>notEditable</b>	<ul style="list-style-type: none"> <li>Category Column</li> <li>Checkbox Column</li> <li>Contact Column</li> <li>Custom Lookup Column</li> <li>Date Column</li> <li>Number Column</li> <li>Project Column</li> <li>Select Column</li> <li>System Lookup Column</li> <li>Text Column</li> <li>User Column</li> </ul>	Boolean	Defines a column within the batch display as uneditable. Set to True to enable.	True, False
<b>nullValueDisplayString</b>	<ul style="list-style-type: none"> <li>Category</li> <li>Lookup</li> <li>Multi-Value List</li> <li>Select</li> <li>Wizard Parameter (Dropdown)</li> </ul>	Text	String to display in a drop-down list if the value is unselected or null.	(Select)
<b>nullValueKey</b>	<ul style="list-style-type: none"> <li>Category</li> </ul>	Text	The i18nKey for empty value in a drop-down list.	(custom.choose)

	<ul style="list-style-type: none"> <li>• Lookup</li> <li>• Multi-Value List</li> <li>• Select</li> <li>• Wizard Parameter (Dropdown)</li> </ul>		Defaults to <code>common.select</code> , which displays "Select".	
<b>numberOfColumns</b>	<ul style="list-style-type: none"> <li>• Block Template</li> </ul>	Integer	The number of columns to include in the block template.	2
<b>onBlur</b>	<ul style="list-style-type: none"> <li>• Checkbox</li> <li>• Memo Text</li> <li>• Text Field</li> </ul>	JavaScript	Runs the inputted JavaScript when a field loses focus.	<code>function() { alert("field lost focus"); }</code>
<b>onChange</b>	<ul style="list-style-type: none"> <li>• Category</li> <li>• Date &amp; Time Field</li> <li>• Date Field</li> <li>• Lookup</li> <li>• Memo Text</li> <li>• Multi-Value List</li> <li>• Number Field</li> <li>• Select</li> <li>• Text Field</li> </ul>	JavaScript	Runs the inputted JavaScript when a field has been changed.	<code>function(this) { alert(this.name + "has been changed"); }</code>
<b>onClick</b>	<ul style="list-style-type: none"> <li>• Anchor</li> <li>• Button</li> <li>• Checkbox</li> </ul>	JavaScript	Runs the inputted JavaScript on a field on click.	<code>function() { alert("clicked a thing"); }</code>
<b>pageArgs</b>	<ul style="list-style-type: none"> <li>• New Record Link</li> </ul>	Text	Passes in a parameter to <code>newRecordLink</code> .	? <code>name=bob&amp;color=red</code>
<b>pageSize</b>	<ul style="list-style-type: none"> <li>• Batch Display</li> </ul>	Text	Defines rows per page for batch display. Defaults to system setting for search rows per page.	10, 50, or 200

<b>preserveValueAfterAdd</b>	<ul style="list-style-type: none"> <li>• Category Column</li> <li>• Checkbox Column</li> <li>• Contact Column</li> <li>• Custom Lookup Column</li> <li>• Date Column</li> <li>• HTML Column</li> <li>• Number Column</li> <li>• Project Column</li> <li>• Select Column</li> <li>• System Lookup Column</li> <li>• Text Column</li> <li>• User Column</li> </ul>	Boolean	Normally fields that are being entered into a batch display are cleared after you hit add. Setting this to true keeps the value in the field after clicking the Add button.	True, False
<b>printPreview</b>	<ul style="list-style-type: none"> <li>• Block Template</li> </ul>	Boolean	Defines the section as a default for the user.	True, False
<b>radioButtonList</b>	<ul style="list-style-type: none"> <li>• Lookup</li> </ul>	Boolean	Defines whether the lookup table will be displayed as a radio button list. Set to True to display as radio button list; False to display as drop down list.	True, False
<b>required</b>	<ul style="list-style-type: none"> <li>• Category</li> <li>• Category Column</li> <li>• Checkbox</li> <li>• Checkbox Column</li> <li>• Contact Column</li> <li>• Custom Lookup Column</li> </ul>	Boolean	Defines whether the field is required. Set to True to enable.	True, False

	<ul style="list-style-type: none"> <li>• Date &amp; Time Field</li> <li>• Date Column</li> <li>• Date Field</li> <li>• Lookup</li> <li>• Memo Text</li> <li>• Number Column</li> <li>• Number Field</li> <li>• Project Column</li> <li>• Select</li> <li>• Select Column</li> <li>• System Lookup Column</li> <li>• Text Column</li> <li>• Text Field</li> <li>• User Column</li> </ul>			
<b>richText</b>	<ul style="list-style-type: none"> <li>• Memo Text</li> </ul>	Boolean	Adds a rich text control to the memo field. Set to True to enable.	True, False
<b>rootItem</b>	<ul style="list-style-type: none"> <li>• Multi-Value List</li> </ul>	Text	Defines which node should be displayed as the root item in a hierarchical table. Defaults to ROOT.	BANK, FRUI_APPL
<b>rows</b>	<ul style="list-style-type: none"> <li>• Memo Text</li> </ul>	Text	Defines the number of visible rows in the text area of the memo field.	1, 3, 10, or 50
<b>sectionCategory</b>	<ul style="list-style-type: none"> <li>• Block Template</li> </ul>	N/A	A non-editable field that indicates which category the block template belongs to for the purpose of rendering custom fields. Only custom fields which belong to this category will be rendered. Screen Designer automatically sets this property the first	N/A

			time a custom field is dropped onto the block template, and will thereafter prevent you from adding custom fields which do not belong to the defined category. Removing all custom fields will reset the property.	
<b>showCopyAction</b>	<ul style="list-style-type: none"> <li>• Batch Display</li> </ul>	Boolean	Defines whether the Copy button is active in in-line editing of batch display. Set to True to enable.	True, False
<b>showEdit</b>	<ul style="list-style-type: none"> <li>• Block Template</li> </ul>	Boolean	Defines whether the edit link on block in view mode is shown. Set to True to enable.	True, False
<b>showRoot</b>	<ul style="list-style-type: none"> <li>• Category</li> <li>• Multi-Value List</li> </ul>	Boolean	Defines whether the root category is shown. Set to True to enable.	True, False
<b>showTime</b>	<ul style="list-style-type: none"> <li>• Date &amp; Time Field</li> </ul>	Boolean	Required to show the time, otherwise tag functions as Date. Set to True to enable.	True, False
<b>size</b>	<ul style="list-style-type: none"> <li>• Category</li> <li>• Lookup</li> <li>• Multi-Value List</li> <li>• Number Field</li> <li>• Select</li> <li>• Text Field</li> </ul>	Text	The number of characters shown in the field, or the number of items shown in a drop-down list.	1, 3, 7, 11, or 40
<b>sortPath</b>	<ul style="list-style-type: none"> <li>• Select</li> </ul>	Text	Expression path to sort the list associated with the drop-down list. Defaults to 'displayString'.	name
<b>src</b>	<ul style="list-style-type: none"> <li>• Script</li> </ul>	Text	A reference to an external JavaScript file to be loaded into the screen.	customUtilities.js

<b>style</b>	<ul style="list-style-type: none"> <li>• Button</li> </ul>	Text	CSS style to apply to field.	width: 90px; color: red;
<b>tableCode</b>	<ul style="list-style-type: none"> <li>• Lookup</li> <li>• Multi-Value List</li> </ul>	Text	Link to specific table via unique code. For Custom field.	CURR
<b>tableName</b>	<ul style="list-style-type: none"> <li>• Lookup</li> <li>• Multi-Value List</li> </ul>	Text	Link to specific table via unique code. For System field.	Currency
<b>target</b>	<ul style="list-style-type: none"> <li>• Anchor</li> </ul>	Text	Where to open the linked URL. Similar to the HTML <a> target attribute.	http:// www.google.com, #section
<b>test</b>	<ul style="list-style-type: none"> <li>• If Statement</li> </ul>	Text	Input should be a JSP expression, formatted '{id.method}'. The 'method' should refer to a method within an associated Java class and should return true or false to be used by an if statement.	<code>#{cjb.myBoolean Method}</code>
<b>title</b>	<ul style="list-style-type: none"> <li>• Anchor</li> <li>• Button</li> </ul>	Memo Field	Mouse-over description text.	Click this to make something happen
<b>type</b>	<ul style="list-style-type: none"> <li>• Script</li> </ul>	Text	Specifies the mime type of a script.	text/JavaScript
<b>value</b>	<ul style="list-style-type: none"> <li>• Category</li> <li>• Date &amp; Time Field</li> <li>• Date Field</li> <li>• HTML</li> <li>• Lookup</li> <li>• Memo Text</li> <li>• Message Parameter</li> <li>• Multi-Value List</li> <li>• Number Field</li> </ul>	Text	Represents the default display value for Advanced Elements and the display value for the Out field.	Hello, I am a field. Change me.

	<ul style="list-style-type: none"> <li>• Out</li> <li>• Select</li> <li>• Text Field</li> </ul>			
<b>var</b>	<ul style="list-style-type: none"> <li>• For Each</li> </ul>	Text	Name of the current object, which should have a setter and getter method of the same name in the associated class, and be kept as a global variable in the associated class.	matter, dispute, relatedObject
<b>width</b>	<ul style="list-style-type: none"> <li>• Category Column</li> <li>• Checkbox Column</li> <li>• Contact Column</li> <li>• Custom Lookup Column</li> <li>• Date Column</li> <li>• HTML Column</li> <li>• Number Column</li> <li>• Project Column</li> <li>• Select Column</li> <li>• System Lookup Column</li> <li>• Text Column</li> <li>• User Column</li> </ul>	Text	Defines the width of a field by percentage of table.	1, 50, or 99
<b>wizardUniqueKey</b>	<ul style="list-style-type: none"> <li>• New Record Link</li> <li>• Wizard Parameter (Textbox)</li> <li>• Wizard Parameter (Dropdown)</li> </ul>	Text	Unique key of the wizard.	uniqueWizardKey

wrap	<ul style="list-style-type: none"> <li>• Memo Text</li> </ul>	Text	The wrap property of the html text area tag (used to render memo text fields). If set to hard, then the browser will insert line breaks on form submission to ensure that no line is longer than the width of the element (specified via the cols property, which is required if this is set to hard). If soft, the browser will not insert any additional line breaks. Defaults to soft.	Hard, Soft
------	---	------	---	------------

### 1.2.6 Interactions

Using interactions, fields can have the ability to affect the visibility of another field or element on the canvas. Visibility will be determined by cases, which are listed in the **Interactions** pane.

**Note:** *The custom screen executes each case from top to bottom, so if two cases contradict each other, the latter case will take precedent.*

#### To add an interaction to an element

1. Click the field or element in the canvas to be affected by the interaction. The **Affected Field** in the **Interactions** pane (located by default in the lower-right corner of the Screen Designer interface) is defined by the field you click in the canvas.
2. In the **If** section of the **Interactions** pane, formulate a logic condition using the following three fields: **Selected Field**, **Operator**, and **Value**.
  - a. In the **Selected Field** drop-down list, select the field in the canvas that will affect the defined **Affected Field**.
  - b. Select an **Operator** from the drop-down list. The available operators will depend on the field type of the **Selected Field**.
  - c. In the **Value** field, enter a value to compare to the **Selected Field** with the selected **Operator**. The **Value** field can be a text field or a drop-down list depending on the field type of the **Selected Field**.
3. In the **Then** section of the **Interactions** pane, define how the **Affected Field** will be affected if the **If** condition returns a true value.
  - a. Select a **Visibility** value.
    - Selecting **Show** will force the **Affected Field** to display on the custom block if the **If** condition is true.



- Selecting **Hide** will force the **Affected Field** to not display on the custom block if the **If** condition is true.
  - Selecting **N/A** will not affect the visibility of the **Affected Field**, regardless of the value of the **If** condition.
- b. Select the **Required** check-box to make the **Affected Field** required on the custom block if the **If** condition is true.
  - c. Alter the appearance of the label font with the following fields: **Label Font-Family**, **Label Font-Size**, **Label Font-Color**.
    - Select the label font from the **Label Font-Family** drop-down list.
    - Enter the label font size in the **Label Font-Size** field.
    - Click the **Color** button to define the label font color.
  - d. Alter the appearance of the value font with the following fields: **Value Font-Family**, **Value Font-Size**, **Value Font-Color**.
    - Select the value font from the **Value Font-Family** drop-down list.
    - Enter the value font size in the **Value Font-Size** field.
    - Click the **Color** button to define the value font color.
4. Click **Save**.

After saving an interaction on a field or element in the canvas, you can add more interactions to the same field or element by clicking the **New** button in the **Interactions** pane.

**Note:** A list field that has been turned into a radio button list using the **radioButtonList** property cannot be made required for interactions.

## Operators

The table below displays all the operators that are available for interactions in Screen Designer.

Operators	Description
<b>Equals</b>	Returns true if the <b>Selected Field</b> is equal to the entered Value.
<b>Not Equals</b>	Returns true if the <b>Selected Field</b> is not equal to the entered Value.
<b>&gt;</b>	Returns true if the <b>Selected Field</b> is greater than the entered Value. This operator is only available for numeric fields.
<b>=&gt;</b>	Returns true if the <b>Selected Field</b> is greater than or equal to the entered Value. This operator is only available for numeric fields.

=	Returns true if the <b>Selected Field</b> is equal to the entered Value. This operator is only available for numeric fields.
=<	Returns true if the <b>Selected Field</b> is less than or equal to the entered Value. This operator is only available for numeric fields.
<	Returns true if the <b>Selected Field</b> is less than the entered Value. This operator is only available for numeric fields.
Is Null	Returns true if the <b>Selected Field</b> contains a null value.
Is Not Null	Returns true if the <b>Selected Field</b> contains any value other than null.
Contains	Selecting multiple values operates with "Or" logic. As an example, if the multi-select list has values "A, B, C, D, E," and a user selects B and D, the Interaction will be triggered if either value matches. If none of the values match, the interaction will not be triggered. If the interaction was previously triggered, the action will be reversed.

### 1.2.6.1 Designing Drill-Down Interactions

## 1.2.7 Saving and Publishing Custom Blocks

You can save your custom blocks to a file for importing into your instance of TeamConnect, or simply to save your work and continue working on it at a later time. You can also publish your custom block directly to TeamConnect from the Screen Designer application.

### To save a custom block

1. After editing your custom block, open the **File** menu and select **Save Block**.
  - If there are any missing required properties on your custom block, an error message will be displayed with a list of the affected elements or fields. Fill in the missing information and save the screen again.
2. Enter a file name and location for your custom screen file. Your custom screen file will be saved in XML format with the file extension *\*.scr.xml*.
3. Click **Save**.

### To open a custom block

1. Open the **File** menu and select **Open Block**.
2. Click **Browse....**
  - a. Locate your previously saved custom block file, which should have the file extension *\*.scr.xml*.

- b. Click **Open**.
3. Select the object type of the custom block file from the **Target Object** drop-down list.
4. Click **Open**.

#### To publish a custom block

1. After editing your custom block, open the **File** menu and select **Publish Block**.
  - If there are any missing required properties on your custom block, an error message will be displayed with a list of the affected elements or fields. Fill in the missing information and publish the screen again.
2. Select the **Publish Type: Enter New Name** or **Select Existing File**.
  - If you selected **Enter New Name**, enter a **File Name** for your custom block, then click **OK**. Your custom screen will be saved and uploaded to the TeamConnect instance with the file extension *\*.scr.xml*.
  - If you selected **Select Existing File**, choose an existing custom block file from the **Select File** drop-down list, then click **OK**. Your custom block will be saved and uploaded to the TeamConnect instance, replacing the past version of the selected file.

### 1.2.8 Using Rows in Screen Designer

## 1.3 Screen Designer Walkthrough

The following walkthrough provides examples of end-to-end custom screen design. This guide is split into multiple steps so that users can quickly navigate to a desired task. However, users new to Screen Designer 5.1 are encouraged to follow the entire guide to ensure a complete overview of how Screen Designer works with TeamConnect.

- [Getting Started with Screen Designer](#)
- 2
- 3
- 4
- 5
- 6
- 7
- 8

- 9
- 10

### 1.3.1 Getting Started with Screen Designer

Before installing Screen Designer, ensure that you meet the following requirements:

- TeamConnect Enterprise Version 5.1
- Java 1.7 or later
- Screen Designer runs on versions of Microsoft Windows that support Java 7. For further information on Java 7's system requirements, see the [Java website](#).

#### Installation

Screen Designer is available for download from the **Available Updates** section of the **About** page in TeamConnect Enterprise. You must have admin user rights to access this page.

#### To launch Screen Designer

1. Log in to TeamConnect Enterprise as a user with admin rights.
2. Click on the **All** button and select **Admin Settings**.
3. In the left pane, click the **About** link.

The **About** page opens.

4. Click the **Show updates available for installed products** button.

The page displays the modules and versions that are available for your TeamConnect instance.

5. Locate the **TeamConnect Screen Designer** line. Click the download link for your version of Windows (32-bit or 64-bit) to download the Screen Designer zip file.
6. After the Screen Designer zip file finishes downloading, extract it to your desired location.

You will need full user rights to the directory to which you extract Screen Designer. For example, if you extract Screen Designer to an administrator directory, then you will need to run Screen Designer as a Windows administrator user.

7. Open the contained **eclipse.exe** executable file. Screen Designer will launch.

## Logging in

After installing and running Screen Designer from the **eclipse.exe** file, you will need to enter the login and instance information.

**Profile:** If you have saved a login by clicking the Save Profile check-box before, you can select that named profile here. The User Name and Instance URL fields will autocomplete. The password will still need to be re-entered.

**User Name:** Enter your login name for your TeamConnect instance.

**Password:** Enter your password for your TeamConnect instance.

**Instance URL:** Enter the base URL for your TeamConnect instance. After entering in the instance URL, add `"/webservice/rest/"` to the end of the URL, not including the `/login` appender.

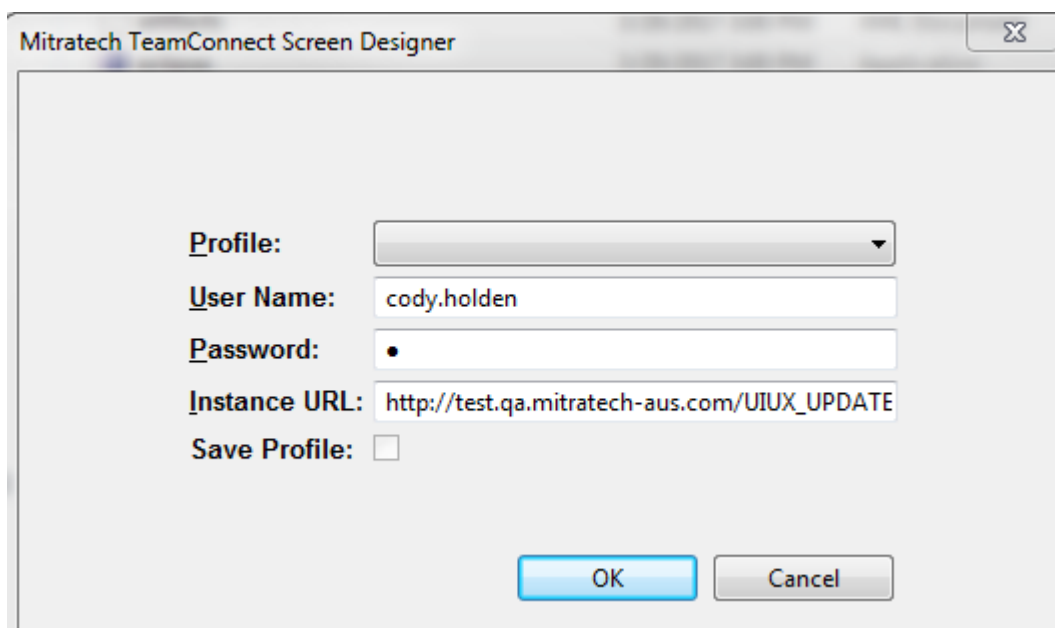
*For example, if the URL for your TeamConnect is:*

`https://companyprod.teamconnect.com/COMPANYPROD/standardlogin`

*Your TeamConnect URL for Screen Designer will be:*

`https://companyprod.teamconnect.com/COMPANYPROD/webservice/rest/`

**Save Profile:** Check this box to save the User Name and URL to a profile. You will be prompted to enter a unique profile name before saving the profile.



The screenshot shows a dialog box titled "Mitratech TeamConnect Screen Designer". It contains the following fields and controls:

- Profile:** A dropdown menu.
- User Name:** A text field containing "cody.holden".
- Password:** A text field with a single bullet point (masked password).
- Instance URL:** A text field containing "http://test.qa.mitratech-aus.com/UIUX\_UPDATE".
- Save Profile:** An unchecked checkbox.
- Buttons:** "OK" and "Cancel" buttons at the bottom.

### 1.3.2 Creating a New Block

# Index

## - C -

custom blocks  
    publishing 42  
    saving 42  
custom screen 43

## - D -

drill-down interactions 42

## - E -

elements  
    advanced 13  
    HTML 13  
    JSP 13

## - F -

fields  
    adding 7  
    Custom Fields 10  
    deleting 7  
    editing 7  
    editing multiple 7  
    moving 7  
    System Fields 10

## - G -

getting started screen 44

## - I -

interactions  
    adding 40  
    operators 40

## - J -

Java classes 17

## - L -

logging in  
    Screen Designer 5

## - M -

multi-value list 42

## - P -

profiles 5  
properties 19  
    editing 18  
    font properties 18

## - R -

reports  
    Screen Designer 11

## - S -

screen designer 42, 43, 44, 45

## - U -

user interface  
    canvas 8  
    menus 8  
    Object buttons 8  
    panes 8  
    Profile button 8

## - W -

walkthrough 43