

MITRATECH

TeamConnect® Screen Designer 5.2

© 2018 Mitratech

User Guide and Release Notes

TeamConnect® Screen Designer 5.2 User Guide

Document ID: tc_sd_52_user_1, published on 3/2/2018

Copyright © 2018, Mitratech Holdings, Inc. All rights reserved.

Disclaimer of Warranty

Mitratech Holdings, Inc. (Mitratech) 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.

Mitratech 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 Mitratech product documentation. For further information regarding third-party non-standard or non-default functionality, please contact Mitratech 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 Mitratech.

The following document is for the TeamConnect™ Screen Designer 5.2 release only. Though every effort was made to ensure that the information in this document is correct and reliable, Mitratech does not assume any liability for any errors encountered in this document.

If you need support for TeamConnect™ Screen Designer 5.2, please contact the Mitratech support team by sending an email to: support@mitratech.com. For more information about Mitratech, visit our web site: <http://www.mitratech.com>.

"Mitratech", 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 Mitratech 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 Mitratech 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:

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

NOTE: Throughout Mitratech 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

Part I Screen Designer Help	5
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.....	14
Java Classes	18
Editing Properties	19
List of Properties.....	20
Interactions	47
Saving and Publishing Custom Blocks	49
3 Screen Designer Walkthrough	50
Getting Started with Screen Designer	51
Setting Up TeamConnect for Custom Screens	52
Custom Field Creation Example.....	54
Creating a New Block	55
Designing Interactions	57
Publishing Your Screen	59
Part II Screen Designer 5.2 Release Notes	60
Index	69

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. Users must have administrator rights to install Screen Designer.

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 (including 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 outlined 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 outlined 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. It will then be highlighted in yellow.
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.
- 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.
 - o The container you drag your field into will be outlined 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 outlined 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.

Menus

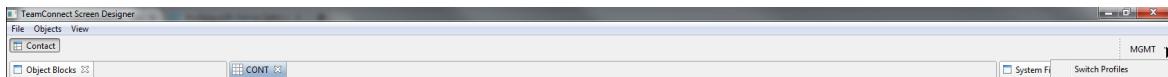
- **File**—This menu contains commands for creating, opening, saving, and publishing custom pages in Screen Designer.
 - **New Block**—Creates a new custom block of the same current object type.
 - **Open Block**—Opens a saved custom block.
 - **Save Block**—Saves the current custom block as a TeamConnect 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). Likewise, there is 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. The **Interactions** tab will display a read-only list of all interactions present within the given block.

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 that may 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 that 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. Designers can utilize the new **Empty** element to add additional spacing into their blocks. This element, added in Screen Designer 5.2, inserts an empty <div> element to create the desired spacing effect.

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 in the Enterprise Customization Help Guide.

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 will require editing after adding an element to the canvas. Some properties, such as **onChange**, 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.

- **Empty**—Enters an empty <div> element to add spacing to the screen. This ensures that rows do not "collapse upward" to fill empty spaces in the layout as per default behavior.
- **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.
- **Row**—Adds a row for housing fields to your screen. Fields require a row to be placed on the screen. Users can utilize rows to create disproportionate column widths and offset values applied to the entire block.

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 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.

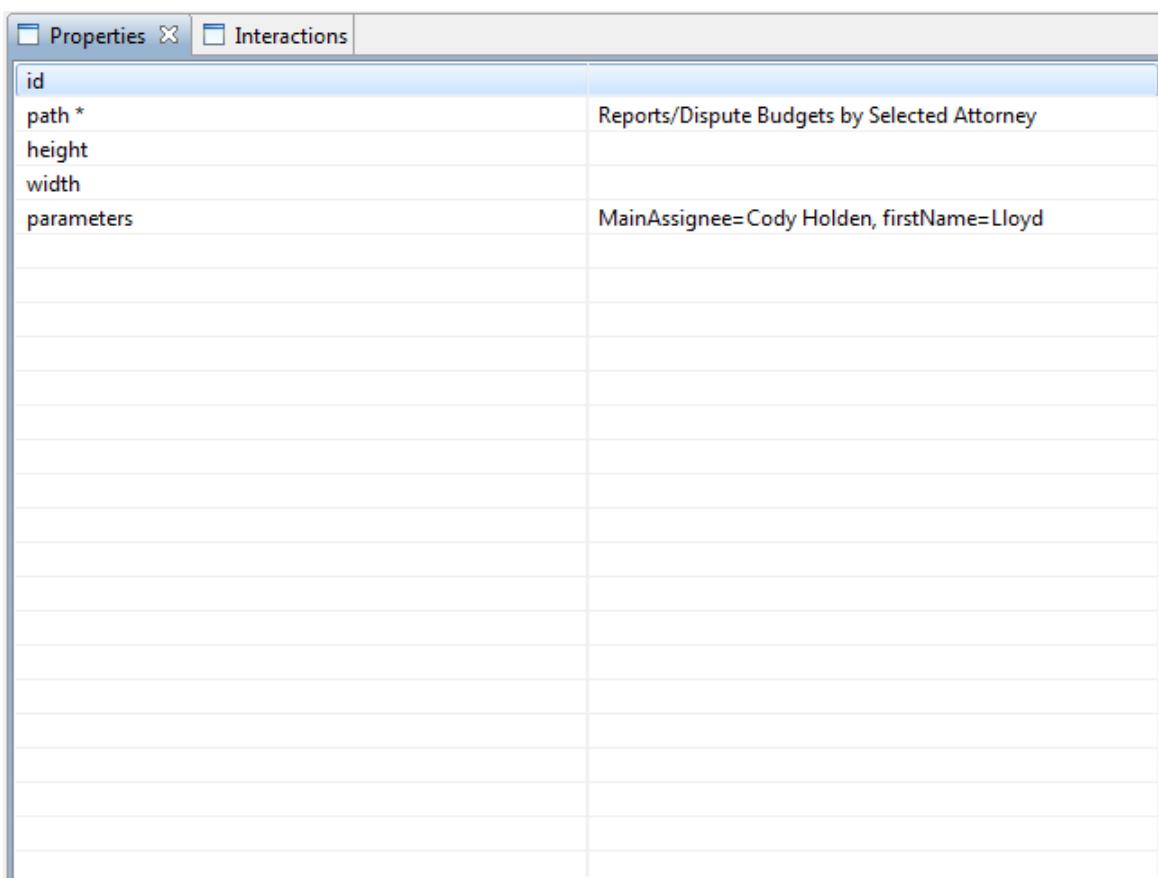
- **If Statement**—Attaches a Boolean condition through a Java class. If that condition evaluates to **true**, then the contents of the tag are shown/loaded.
- **If-Else Statement**—Attaches a Boolean condition through a Java class, along with content to load when the condition is **true** and different 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 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.

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.

As of Screen Designer 5.2, parameters can be edited from the Properties pane on reports. The **parameters** property will list all parameters on the selected report as a comma delimited list. Users can adjust the value of these fields directly by typing within the property, e.g., "MainAssignee=Cody Holden". If the parameter is erased, users can retype the **parameter name** followed by an equals sign to restore the value (as demonstrated by the image below).



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
Anchor	A link to an internal or external address. Must contain either a Message or Out element.	Message or Out element	Linking to an external page or to a different section of the current page.
Block Template	A container to group any amount of fields in a custom block.	numberOfColumns	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).
Button	An interactive element that will trigger an event based on the Screen Designer user specification.	N/A	Allowing the user to trigger some action.
Empty	Enters an empty <div> element to add spacing to the screen.	N/A	Adding more space between rows for better UI. This ensures that rows do not "collapse upward" to fill empty spaces in the layout as per default behavior.
Message	Outputs a localized message given an i18n key. See Creating Custom Messages.	key	Displaying localized informational or cautionary messages on the screen. Identical to out tags except the message will be localized. Generally, this is preferred and should be used in place of out tags.
Message Parameter	Inserts dynamic text into Messages. Message Parameters can only be placed in a Message element.	value	Creating error messages that contain dynamic values that are defined by the Message Parameter element.
New Record Link	Creates a new record of the specified object type.	entityCode	Adding a "Create <Object>" link to a screen.
Out	Outputs a message in the block.	value	Displaying informational or cautionary messages on the screen. Identical to message tags except that the message will not be localized. Generally,

			message tags should be preferred over out tags.
Record Link	Links to a record of a specified object type.	eoObject	Linking to some other object.
Row	Adds a row for housing fields to your screen. Fields require a row to be placed on the screen. Users can utilize rows to create disproportionate column widths and offset values applied to the entire block.	numberOfColumns	Added to a screen for fields to be placed on the canvas. You can adjust the column width with the columnWidth properties. Adjusting these widths can allow designers to set up disproportionate width columns if desired.

Advanced Elements

Element	Description	Required Properties	Example Use
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.	batchDisplayObject	Displaying embedded objects in a single table. Displaying arbitrary tabular data pulled from a custom Java class.
Category	Displays a list of CategoryItemTreeItem objects returned from a Java class method. Displays the default category by default.	name	Displaying a list of categories from a custom Java class method.
Checkbox	A box that users can select or de-select to produce a Boolean value.	name	Linking to a field on a related object via its path, or via a custom Java class method.

Date & Time Field	A field for entering date and time information.	name	Linking to a field on a related object via its path, or via a custom Java class method.
Date Field	A field for entering date information.	name	Linking to a field on a related object via its path, or via a custom Java class method.
Lookup	A drop-down menu that lists the contents of a lookup table or a list method in a custom Java class.	name	Linking to a field on a related object via its path, or via a custom Java class method.
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.	name	Linking to a field on a related object via its path, or via a custom Java class method.
Multi-Value List	A multi-select box that lists the contents of a lookup table or a list method in a custom Java class.	name	Linking to a field on a related object via its path, or via a custom Java class method.
Number Field	A field that stores a numerical value.	name	Linking to a field on a related object via its path, or via a custom Java class method.
Radio Button	A list of radio buttons, wherein users can select only one radio button at a time.	name	Linking to a field on a related object via its path, or via a custom Java class method.
Select	A drop-down menu that allows the user to use a custom list for selection.	name	Linking to a field on a related object via its path, or via a custom Java class method.
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.	name	Linking to a field on a related object via its path, or via a custom Java class method.

HTML Elements

Element	Description	Required Properties	Example Use
Html	Allows the user to add HTML elements to the custom block.	N/A	Inserting raw HTML into the screen in order.
Script	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
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.	test	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.
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.	test	Making a group of fields non-editable if a record is past a certain date. Otherwise, allowing the fields to be edited.
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).	var, items	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 5.2, field label text can be changed from within the application (rather than solely via XML edits). However, changes made in this way will not 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.

- 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.
- 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 (including `labelFontFamily`, `labelFontSize`, `labelFontColor`) 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.
- 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:
 - **fontFamily**—A drop-down list of common font types.
 - **fontSize**—A number field that defines the size of the affected text. The default `fontSize` value in TeamConnect is 14px.
 - **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	Description	Example Data
aggressiveCacheMultiplier	• Batch Display	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'.	
allowNullValue	<ul style="list-style-type: none"> • Category • Lookup • Multi-Value List • Select • Wizard Parameter (Dropdown) 	Boolean	Allows a null value. Set to True to enable.	True, False
batchDisplayObject	<ul style="list-style-type: none"> • Batch Display 	Text	This property references the id of the useClass tag that should be associated with this batchDisplay.	useclassId
blockTitle	<ul style="list-style-type: none"> • Block Template 	Text	Title of block template.	Welcome to the Block
blockTitleKey	<ul style="list-style-type: none"> • Block Template 	Text	The i18nKey for block title.	some.i18n.key
category	<ul style="list-style-type: none"> • Multi-Value List 	N/A	Non-editable property associated with custom fields. This is the category which the custom field is under.	INV.C
checked	<ul style="list-style-type: none"> • Checkbox 	Text	Defines whether or not field defaults to selected.	True, False
colon	<ul style="list-style-type: none"> • True/False 	Text drop-down	Determines whether a colon is present after the label. Defaults to true.	True, False
cols	<ul style="list-style-type: none"> • Memo Text 	Text	Defines the visible width of a text area (memo text).	1, 5, or 10
columnWidth	<ul style="list-style-type: none"> • Row • Block template 	Text	Defines the width of a given column within a row.	20, 25, 33, 50, 75, 100

			(Property will be titled column1Width, column2Width....)	
cssClass	<ul style="list-style-type: none"> • Button • Message • Out • Current Phase 	Text	CSS class to use for field.	some_css_class
defaultBlankRows	<ul style="list-style-type: none"> • Batch Display 	Text	Defines the number of blank rows that appear for entry into batch display.	1, 5, or 10
disabled	<ul style="list-style-type: none"> • Button • Select 	Boolean	Disables field if set to True.	True, False
displayNamePath	<ul style="list-style-type: none"> • Select 	Text	This is the path of the value to display in the select field. Defaults to 'displayString'.	enterpriseEntity.fieldName
editorAllowEmptyDisplay	<ul style="list-style-type: none"> • Category Column • Custom Lookup Column • Select Column • System Lookup Column 	Boolean	Defines whether to display an empty field at the top of a drop-down list. Set to True to enable.	True, False
editorDefault	<ul style="list-style-type: none"> • Category Column • Checkbox Column • Contact Column • Custom Lookup Column 	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"> • Date Column • Number Column • Project Column • Select Column • System Lookup Column • Text Column • User Column 			
editorDependentItemsMap	<ul style="list-style-type: none"> • Select Column 	Text	<p>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.</p>	\${cjb.items}
editorDependentItemsParentFieldId	<ul style="list-style-type: none"> • Select Column 	Text	<p>String ID of the parent field within the batch display. Required when editorDependentItemsMap is used.</p>	parentField
editorEmptyDisplay	<ul style="list-style-type: none"> • Category Column • Custom Lookup Column 	Text	<p>String to display in an editor drop-down list within a batch display column when no value is currently selected.</p>	(Select)

	<ul style="list-style-type: none"> • Select Column • System Lookup Column 			
editorEmptyDisplayI18nKey	<ul style="list-style-type: none"> • Category Column • Custom Lookup Column • Select Column • System Lookup Column 	Text	The i18nKey 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
editorEntityCode	<ul style="list-style-type: none"> • Category Column 	Text	Unique object code corresponding to desired categories.	INV
editorFilterEntityCode	<ul style="list-style-type: none"> • Project Column 	Text	Unique code of the project, only used for project column, batch display.	DISP
editorFilterParentItemUniqueCode	<ul style="list-style-type: none"> • Category • Category Column 	Text	Unique code of a parent. Will only display children of this parent.	DISP
editorFilterSearchDefinitionKey	<ul style="list-style-type: none"> • User Column 	Text	Specific to searches for batch display user columns. Sets the search module to be used.	searchKey
editorItems	<ul style="list-style-type: none"> • Select Column 	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	\${cjb.items}

			desired list. If this property has a non-null value, then editorDependentItemsMap must be null, and vice-versa.	
editorMaxLength	• Text Column	Text	Defines the maximum length to accept in text editor.	10, 50, or 250
editorOnChangeRefreshFormValues	<ul style="list-style-type: none"> • Category Column • Checkbox Column • Contact Column • Custom Lookup Column • Date Column • Number Column • Project Column • Select Column • System Lookup Column • Text Column • User Column 	Boolean	Defines whether the page refreshes after a change to the field. Set to True to enable.	True, False
editorSelectEmptyDisplay	<ul style="list-style-type: none"> • Contact Column • Project Column • User Column 	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	(none)

			when no item is selected. This attribute cannot be used when the required attribute is 'True' on the column.	
editorSelectEmptyDisplayI18nKey	<ul style="list-style-type: none"> • Contact Column • Project Column • User Column 	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
editorSelectItems	<ul style="list-style-type: none"> • Contact Column • Project Column • User Column 	Text	List of entities to use for the selection.	\${cjb.items}
editorSize	<ul style="list-style-type: none"> • Contact Column • Project Column • Text Column • User Column 	Text	Text editor size. Defaults to browser value if not set.	10, 50, or 250
editorTableCode	<ul style="list-style-type: none"> • Custom Lookup Column • System Lookup Column 	Text	Path to custom lookup table (unique code).	EXMP_FRUI_BANA
editorTextAreaCols	• Text Column	Text	Defines the visible width of a text area of a text field	10, 50, or 250

			within a batch.	
editorTextAreaRows	<ul style="list-style-type: none"> Text Column 	Text	Defines the number of visible rows in the text area of a text field within a batch.	1, 5, or 10
editorUseSelect	<ul style="list-style-type: none"> Contact Column Project Column User Column 	Boolean	Defines whether to use select(dropdown) when dealing with an entity column. If set to True, editorSelectItems is required.	True, False
editorUseTextArea	<ul style="list-style-type: none"> Text Column 	Text	Allows text area to be set.	True, False
entityCode	<ul style="list-style-type: none"> Category New Record Link 	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
entityToLink	<ul style="list-style-type: none"> New Record Link 	Text	Specifies if entity linked is parent or related object.	\${cjb.linkRecord} or enterpriseEntity.project
eoObject	<ul style="list-style-type: none"> Record Link 	Text	Refers to a getter method in a Java file that returns a TN object. (For example, cjb.vendor)	\${cjb.myObject}
forceEditable	<ul style="list-style-type: none"> Batch Display Category Checkbox Date & Time Field Date Field Lookup Memo Text 	Boolean	Defines field to be editable at all times. Set to True to enable.	True, False

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

			<ul style="list-style-type: none"> • currency • \$ • plain • decimal 	
fractionDigits	<ul style="list-style-type: none"> • Number Aggregate Field 	Text	This sets both minimumFractionDigits and maximumFractionDigits to the same value.	4
hiddenInForms	<ul style="list-style-type: none"> • Category Column • Checkbox Column • Contact Column • Custom Lookup Column • Date Column • HTML Column • Number Column • Project Column • Select Column • System Lookup Column • Text Column • User Column 	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
hideAddFunction	<ul style="list-style-type: none"> • Batch Display 	Boolean	Hides the Add button on batch display. Set to True to enable.	True, False

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

	<ul style="list-style-type: none"> • New Record Link • Number Field • Select • Text Field • Wizard Parameter (Textbox) • Wizard Parameter (Dropdown) • Project • Report • User 			
items	<ul style="list-style-type: none"> • For Each 	Text	<p>Input should be a JSP expression, formatted '\${id.method}', 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.</p>	\${cjb.items}
key	<ul style="list-style-type: none"> • Category • Checkbox • Date & Time Field • Date Field • Lookup • Memo Text • Message • Multi-Value List 	Text	<p>The i18nKey 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.</p>	invoice.numberString

	<ul style="list-style-type: none"> • Number Field • Select • Text Field • External DMS • Radio button • User 			
label	<ul style="list-style-type: none"> • Action • Button • Category • Category Column • Checkbox • Checkbox Column • Contact Column • Current Phase • Custom Lookup Column • Date & Time Field • Date Column • Date Field • External DMS • HTML Column • Lookup • Memo Text 	Text Action Column	<p>Defines the value of the label for the field. This field can be changed within the application (rather than just by XML edits).</p> <p>However, changing this field in the application will not localize for other languages.</p>	Invoice Number

	<ul style="list-style-type: none"> • Multi-Value List • Number Column • Number Field • Project Column • Project • Select • Select Column • System Lookup Column • Text Column • Text Field • User Column • User 			
labelFontFamily	<ul style="list-style-type: none"> • All elements with labels (as noted above) 	Text Drop-down	Defines the Font Family style for the label.	Arial
labelFontSize	<ul style="list-style-type: none"> • All elements with labels (as noted above) 	Numeric	Defines the size of the font in the label.	18
labelFontColor	<ul style="list-style-type: none"> • All elements with labels (as noted above) 	Numeric (hex-code entry) or select from palette	Defines the color of the label. This does not affect the color of the font within the actual field.	RGB {128, 255, 0}

labelVerticalAlignment	<ul style="list-style-type: none"> All elements with labels (as noted above) 	Text drop-down	Allows users to position the label above, below, or at the same level as the field.	Above
labelWidth	<ul style="list-style-type: none"> All elements with text, numeric entry, or calendar items. 	Numeric	Allows users to adjust the width of the label itself.	10
labelI18nKey	<ul style="list-style-type: none"> Action Column Category Column Checkbox Column Contact Column Custom Lookup Column Date Column HTML Column Number Column Project Column Select Column System Lookup Column Text Column User Column 	Text	The i18nKey of the label. This is used for the columns within a batch display	custom.labelXKey

labelKey	• Button	Text	i18nKey for the button label.	custom.labelX
list	• Lookup • Multi-Value List • Select	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
maximumFractionDigits	• Number Aggregate Field • Number Column • Number Field	Text	Maximum decimal places to support.	2, 3, or 4
maxLength	• Number Field • Text Field	Text	Defines the maximum number of characters that can be entered in a field.	100
maxLevel	• Multi-Value List	Integer	Defines the number of levels to show for a hierarchical lookup table. 0 will show all levels.	0, 1, 3, or 5
minimumFractionDigits	• Number Aggregate Field • Number Column • Number Field	Text	Minimum decimal places to support.	2
multiple	• Category • Lookup • Multi-Value List • Select	Boolean	Allows the selection of multiple items at once. Set to True to enable.	True, False

name	<ul style="list-style-type: none"> • Action Column • Category • Category Column • Checkbox • Checkbox Column • Contact Column • Custom Lookup Column • Date & Time Field • Date Column • Date Field • HTML Column • Lookup • Memo Text • Multi-Value List • Number Aggregate Field • Number Column • Number Field • Project Column • Select • Select Column 	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"> • System Lookup Column • Text Column • Text Field • User Column • Wizard Parameter (Textbox) • Wizard Parameter (Dropdown) 			
negate	• If Statement	Text	Negates the condition received by the element.	True, False
noResize	• Memo Text	Boolean	Prevents the user from resizing the memo text area in internet explorer. Defaults to False.	True, False
notEditable	<ul style="list-style-type: none"> • Category Column • Checkbox Column • Contact Column • Custom Lookup Column • Date Column • Number Column • Project Column • Select Column 	Boolean	Defines a column within the batch display as uneditable. Set to True to enable.	True, False

	<ul style="list-style-type: none"> • System Lookup Column • Text Column • User Column 			
nullValueDisplay String	<ul style="list-style-type: none"> • Category • Lookup • Multi-Value List • Select • Wizard Parameter (Dropdown) 	Text	String to display in a drop-down list if the value is unselected or null.	(Select)
nullValueKey	<ul style="list-style-type: none"> • Category • Lookup • Multi-Value List • Select • Wizard Parameter (Dropdown) 	Text	The i18nKey for empty value in a drop-down list. Defaults to common.select, which displays "Select".	(custom.choose)
numberOfColumns	<ul style="list-style-type: none"> • Block Template • Row 	Integer	The number of columns to include in the block template.	2
onBlur	<ul style="list-style-type: none"> • Checkbox • Memo Text • Text Field 	JavaScript	Runs the inputted JavaScript when a field loses focus.	function() { alert("field lost focus"); }
onChange	<ul style="list-style-type: none"> • Category • Date & Time Field 	JavaScript	Runs the inputted JavaScript when a field has been changed.	function(this) { alert(this.name +

	<ul style="list-style-type: none"> • Date Field • External DMS • Lookup • Memo Text • Multi-Value List • Number Field • Select • Text Field 			"has been changed"); }
onClick	<ul style="list-style-type: none"> • Anchor • Button • Checkbox 	JavaScript	Runs the inputted JavaScript on a field on click.	function() { alert("clicked a thing"); }
pageArgs	<ul style="list-style-type: none"> • New Record Link 	Text	Passes in a parameter to newRecordLink.	? name=bob&color=red
pageSize	<ul style="list-style-type: none"> • Batch Display 	Text	Defines rows per page for batch display. Defaults to system setting for search rows per page.	10, 50, or 200
preserveValueAfterAdd	<ul style="list-style-type: none"> • Category Column • Checkbox Column • Contact Column • Custom Lookup Column • Date Column • HTML Column 	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

	<ul style="list-style-type: none"> • Number Column • Project Column • Select Column • System Lookup Column • Text Column • User Column 			
printPreview	<ul style="list-style-type: none"> • Block Template 	Boolean	Defines the section as a default for the user.	True, False
radioButtonList	<ul style="list-style-type: none"> • Lookup 	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
required	<ul style="list-style-type: none"> • Category • Category Column • Checkbox • Checkbox Column • Contact Column • Custom Lookup Column • Date & Time Field • Date Column • Date Field 	Boolean	Defines whether the field is required. Set to True to enable.	True, False

	<ul style="list-style-type: none"> • External DMS • Lookup • Memo Text • Number Column • Number Field • Project Column • Select • Select Column • System Lookup Column • Text Column • Text Field • User Column 			
richText	<ul style="list-style-type: none"> • Memo Text 	Boolean	Adds a rich text control to the memo field. Set to True to enable.	True, False
rootItem	<ul style="list-style-type: none"> • Multi-Value List 	Text	Defines which node should be displayed as the root item in a hierarchical table. Defaults to ROOT.	BANK, FRUI_APPL
rows	<ul style="list-style-type: none"> • Memo Text 	Text	Defines the number of visible rows in the text area of the memo field.	1, 3, 10, or 50
sectionCategory	<ul style="list-style-type: none"> • Block Template 	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	N/A

			belong to this category will be rendered. Screen Designer automatically sets this property the first 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.	
showCopyAction	• Batch Display	Boolean	Defines whether the Copy button is active in in-line editing of batch display. Set to True to enable.	True, False
showEdit	• Block Template	Boolean	Defines whether the edit link on block in view mode is shown. Set to True to enable.	True, False
showRoot	• Category • Multi-Value List	Boolean	Defines whether the root category is shown. Set to True to enable.	True, False
showTime	• Date & Time Field	Boolean	Required to show the time, otherwise tag functions as Date. Set to True to enable.	True, False
size	• Category • Lookup • Multi-Value List • Number Field • Select • Text Field • User	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

sortPath	• Select	Text	Expression path to sort the list associated with the drop-down list. Defaults to 'displayString'.	name
src	• Script	Text	A reference to an external JavaScript file to be loaded into the screen.	customUtilities.js
style	• Button	Text	CSS style to apply to field.	width: 90px; color: red;
tableCode	• Lookup • Multi-Value List	Text	Link to specific table via unique code. For Custom field.	CURR
tableName	• Lookup • Multi-Value List	Text	Link to specific table via unique code. For System field.	Currency
target	• Anchor	Text	Where to open the linked URL. Similar to the HTML <a> target attribute.	http://www.google.com, #section
test	• If Statement	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.	\${cjb.myBooleanMethod}
title	• Anchor • Button • Category • Check-box • Date • Date Tile • External DMS • Lookup	Memo Field	Mouse-over description text.	Click this to make something happen

	<ul style="list-style-type: none"> • Multi-value list • Text Box • Number • Project • Radio Button • Select • Text • User 			
type	<ul style="list-style-type: none"> • Script 	Text	Specifies the mime type of a script.	text/JavaScript
value	<ul style="list-style-type: none"> • Category • Date & Time Field • Date Field • HTML • Lookup • Memo Text • Message Parameter • Multi-Value List • Number Field • Out • Project • Select • Text Field • User 	Text	Represents the default display value for Advanced Elements and the display value for the Out field.	Hello, I am a field. Change me.
var	<ul style="list-style-type: none"> • For Each 	Text	Name of the current object, which should have a setter and getter method	matter, dispute, relatedObject

			of the same name in the associated class, and be kept as a global variable in the associated class.	
width	<ul style="list-style-type: none"> • Category Column • Checkbox Column • Contact Column • Custom Lookup Column • Date Column • HTML Column • Number Column • Project Column • Select Column • System Lookup Column • Text Column • User Column 	Text	Defines the width of a field by percentage of table.	1, 50, or 99
wizardUniqueKey	<ul style="list-style-type: none"> • New Record Link • Wizard Parameter (Textbox) • Wizard Parameter (Dropdown) 	Text	Unique key of the wizard.	uniqueWizardKey

wrap	• Memo Text	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
-------------	----------------	------	---	------------

Report-Specific Properties

The following properties apply to reports added into your screen.

Property	Data Type	Description	Example Data
height	Numeric	Sets the height of the report element on your screen.	50
parameter	Text	Allows users to edit parameters from within Screen Designer. Each parameter will be listed by its parameter name. If a user deletes a parameter, retype the parameter name in a comma delimited list, e.g., "firstName=Cody, middleName=Lloyd".	firstName=Cody
path (required)	Text	Sets the location of the report within your TeamConnect instance.	Reports/All Dispute Budgets by All Attorneys
width	Numeric	Sets the width of the report element on your screen.	50

1.2.6 Interactions

Using interactions, fields can have the ability to affect the visibility of another field, block 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 precedence.

To add an interaction to an element

1. Click the field, block 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 or block you click in the canvas.
2. Click **New** to start designing your interaction.
3. 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**.
4. 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.
5. Click **Save**. If you edit an interaction, you must save prior to editing a new interaction or the first changes will not be recorded.

Important: After creating an interaction, the inverse condition is automatically created. Users do not need to manually define the opposite of their action. This applies to both show/hide functionality as well as required/not required and will only apply to newly created interactions. As an example, if a user set a given field to "show" if the Current Phase is "Closed," the user does not need to manually create a "hide" condition for all phases besides "Closed."

Users should thereby make sure when deleting interactions that they remove both interactions, as unintended duplicate interactions can result from creating new interactions after only deleting half of the pre-existing interaction. **This automatic inverse creation does not apply to numeric-type fields.**

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
Equals	Returns true if the Selected Field is equal to the entered Value.
Not Equals	Returns true if the Selected Field is not equal to the entered Value.
>	Returns true if the Selected Field is greater than the entered Value. This operator is only available for numeric fields.
=>	Returns true if the Selected Field is greater than or equal to the entered Value. This operator is only available for numeric fields.
=	Returns true if the Selected Field is equal to the entered Value. This operator is only available for numeric fields.
=<	Returns true if the Selected Field is less than or equal to the entered Value. This operator is only available for numeric fields.

<	Returns true if the Selected Field is less than the entered Value. This operator is only available for numeric fields.
Is Null	Returns true if the Selected Field contains a null value.
Is Not Null	Returns true if the Selected Field contains any value other than null.
Contains	<p>Only available for List fields.</p> <p>Returns true if the Selected Field contains any of the selected options. Users can use shift to select multiple values from the available list.</p> <p>Selecting multiple values for Contains 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.</p>
Does Not Contain	<p>Only available for List fields.</p> <p>Returns true if the Selected Field does not contain any of the selected options. Users can use shift to select multiple values from the available list.</p> <p>Selecting multiple values for Does Not Contain operates with "AND" 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 both values are not present. If either of the values match, the interaction will not be triggered.</p>

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**.
 - o 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**.
 - o 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**.
 - o 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**.
 - o 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.

1.3 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 5.2 are encouraged to follow the entire guide to ensure a complete overview of how Screen Designer works with TeamConnect.

1. [Getting Started with Screen Designer](#)
2. [Setting Up TeamConnect For Custom Screens](#)
3. [Custom Field Creation Example](#)
4. [Creating a New Block](#)
5. [Designing Interactions](#)
6. [Publishing Your Screen](#)

1.3.1 Getting Started with Screen Designer

Before installing Screen Designer 5.2, ensure that you meet the following requirements:

- TeamConnect Enterprise Version 5.2
- Java 1.8 or later
- Screen Designer runs on versions of Microsoft Windows that support Java 8.

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.

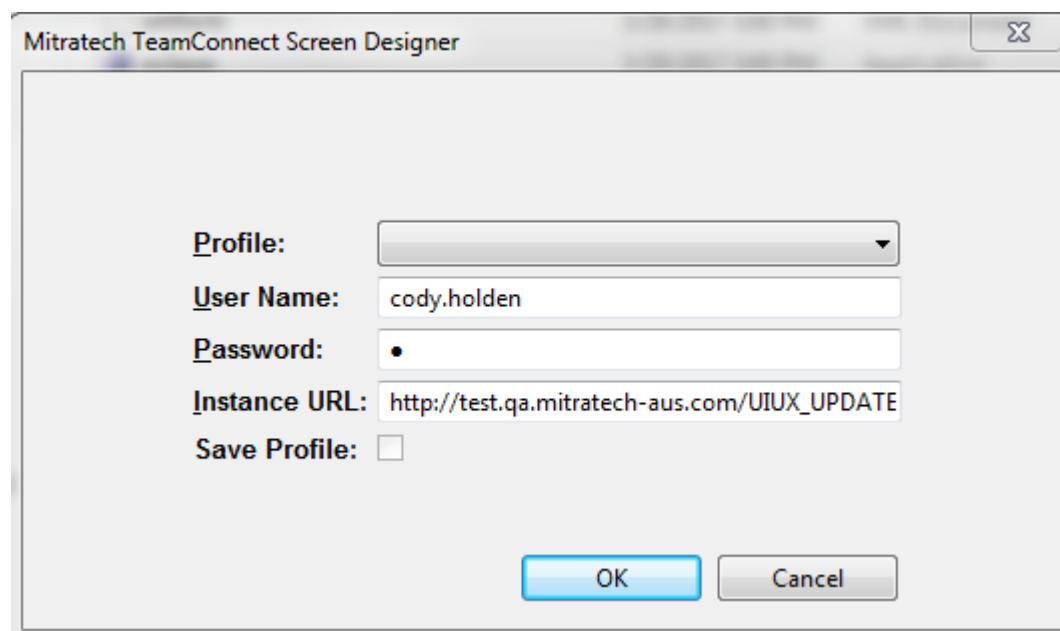
For example, if the URL for your TeamConnect is:

`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.



1.3.2 Setting Up TeamConnect for Custom Screens

Before creating your custom screens, you may want to design custom fields to be used for your teams. It is recommended to do a thorough analysis of what fields are needed for your use case prior to running the Screen Designer. Once all necessary custom fields are decided upon, administrators can save time by adding these fields at once via TeamConnect Setup.

To use blocks for your custom pages, you must first create the necessary custom fields.

To create desired custom fields

1. Analyze your organization's specific information that needs to be captured in the selected object, in addition to the default fields provided within TeamConnect.
2. Decide which custom fields you need to create to capture this information.

Tip: Make a list of the labels for these custom fields and their field types.

3. Make sure the custom lookup tables are created for the custom fields of the type List.
4. Because all custom fields are category-specific, decide to which categories in the object definition you want to add your custom fields.
5. Make sure the desired object categories are added to the object definition.
6. Add your custom fields to the appropriate category in the object definition. See Adding Custom Fields.
7. Include your custom fields into the blocks, depending on your needs and design.

To add a custom field to an object definition

1. In the **Setup** menu, from the **Go to** drop-down list select **Object Definitions**.
2. Select the appropriate object definition from the displayed list.
3. Open the **Custom Fields** tab of the displayed object definition.
4. Select the **Number of entries you would like to add** from the drop-down list.
5. For each data entry row, enter the values in the appropriate fields as described in the Custom Fields Tab table.
6. If you want to continue adding custom fields, click **add more**. Otherwise, click **Save** on the toolbar.
TeamConnect automatically saves the custom field information and adds the fields to the system block called **Details**.
7. Create the appropriate components for the object view—that is, custom blocks.

Points To Remember

Custom fields can be created and displayed by category only. Because of this category dependency, you must carefully plan for which categories to create the desired custom fields, taking into consideration the following:

- For all categories and custom fields you add in an object definition, associated access rights are automatically created. They must be assigned to the appropriate users and user groups in order for the users to be able to view the categories and their custom fields.
- The root-level category in both custom and system objects is always automatically added to the object records to provide access to other categories (unless you do not intend to use categories

and custom fields in your design at all). This means that all custom fields created for this category are automatically displayed in all records for all users.

- Always arrange your custom fields so that you do not have to replicate the same fields for different categories.

This improves performance and also prevents possible confusion if the custom fields of two categories are used in the same object record.

- A category name must be unique within its object if it has any custom fields included in Data Warehouse. For more information, see Data Warehouse Requirements for Custom Fields.

1.3.2.1 Custom Field Creation Example

The following example demonstrates creating a custom field to later be utilized by the Screen Designer module. Please see Custom Fields in the TeamConnect Setup and Development Guide for further information on custom fields in general as well as specific details on how they function in TeamConnect Enterprise.

For our example, we will assume that our Contacts records would benefit from having a check-box "Discount Eligible" custom field as well as a number-entry field for "Discount Eligible Amount." The steps below detail creating these fields. For more detail on creating interactions based on this field, please see [Designing Interactions](#) in the Screen Designer guide.



1. Navigate to **Setup** [Setup](#) in TeamConnect.
2. From the **Go to...** drop-down list, select **Object Definitions**. Click **Contact** from the resulting screen.
3. Click the **Custom Fields** tab.
4. Add the Field Name, Label, and Field Type as well as other desired fields for "Discount Eligible." For this use case, we will use "Check Box." Click **Add More** when done. *Note: The Field Name cannot include special characters or spaces. The Label field can include spaces.*
5. Add the Field Name, Label, and Field Type as well as other desired fields for "Discount Eligible Amount." For this use case, we will use "Number Field." Click **Add More** when done.
6. Click **Save and Close**.

General | Categories | Custom Fields | Forms | Blocks | Object Views | Search Views | Custom Messages | Conditions | Rules | Templates | Wizards | Notifications | History | Duplicate Cor

Custom Fields

For Category: Contact

Number of entries you would like to add: 1

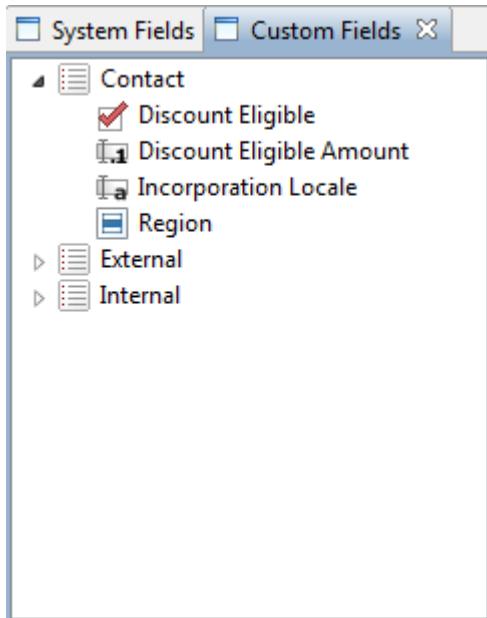
Field Name	Label	Field Type	Include in Data Warehouse	Is Required ?	Default Value	Exclude From Custom Search	Exclude From Global Search*
1		(Select)	(Select)	(Select)		NO	NO
+ add more							
Field Name	Label	Field Type	Include in Data Warehouse	Is Required ?	Default Value	Exclude From Custom Search	Exclude From Global Search*
1 <input type="checkbox"/> DiscountEligible	Discount Eligible	Check Box	NO	N/A	Unchecked	NO	NO
2 <input type="checkbox"/> DiscountEligibleAmt	Discount Eligible Amount	Number Field	NO	NO	No default value	NO	NO
3 <input type="checkbox"/> Incorporation	Incorporation Locale	Text	YES	NO	No default value	NO	NO
4 <input type="checkbox"/> Region	Region	Multi-Value List (Region)	NO	NO	(None)	NO	NO

[Check All](#) [Uncheck All](#) [edit](#) [delete](#)

*Any changes to the custom fields on a record type will require a re-index of this record type for the changes to reflect in the global search results.

Custom Fields Added

The fields will be available upon logging into Screen Designer.



1.3.3 Creating a New Block

Before designing a custom screen, users should select the **Object** for which they'd like to add new blocks. Select **Objects** from the tab at the top of the Screen Designer application. Object blocks will automatically populate in the left pane **Object Blocks** section. The following procedural outlines a typical process featuring the custom fields created in the [Custom Field Creation Example](#) of this section.

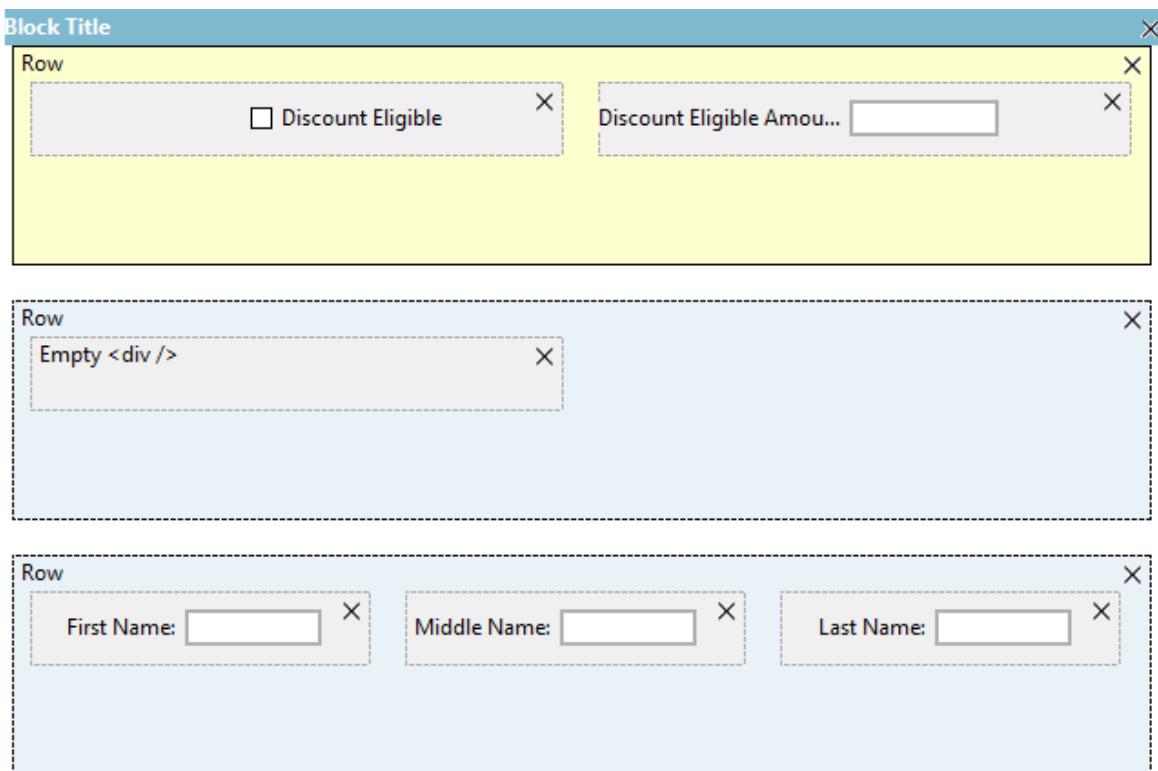
Create a New Block

1. Open and run the Screen Designer application.
2. Click the **Objects** tab and select the desired object to add blocks. For this example, we will be using **Contact**.
3. After opening up the Contact object, click **File>New Block** from the top menu.
4. After creating the block, click **File>Save Block**.

Adding Fields

Users are able to add system/custom fields from the right pane of the application. However, custom fields must be created in TeamConnect Setup before they can be used in screen designer. For more information on creating custom fields, please see Custom Fields in the *TeamConnect Setup and Development* guide.

1. Add a row from the **Elements** tab to the canvas. *Note: The canvas will begin with one row after creating a new block.*
2. Drag in the fields from the **System Fields** and **Custom Fields** tabs on the right of the Screen Designer application.
 - For this example we will add multiple rows containing: Discount Eligible, Discount Eligible Amount, an empty row for spacing, First Name, Last Name, and Driver's License fields. **Note:** All fields must be placed into **row** elements to fit onto the canvas.
3. Adjust the row properties. Adjusting the **numberOfColumns** property will allow users to further customize the appearance of these rows. Users can select the width of these columns to occupy either 20, 25, 33, 50, 75, 100 percent of the row. This value will default to the **Block-level setting for numberOfRows**.



Additional elements are available for users to design more complex, customized screens. For more detail on how these elements function within Screen Designer, please see [Elements and Reports](#) and the [List of Elements](#) pages within the Screen Designer help.

1.3.4 Designing Interactions

Using interactions, fields can have the ability to affect the visibility of another field, block or element on the canvas. Visibility will be determined by cases, which are listed in the **Interactions** pane. More information on Interactions as well as another example can be found on the [Interactions](#) section of the Screen Designer guide.

For users with Screen Designer 5.2 and later, creating a conditional interaction will automatically result in the inverse condition being created. Users do not need to manually define the opposite of their action. This applies to both show/hide functionality as well as required/not required and will only apply to newly created interactions. As an example, if a user set a given field to "show" if the Current Phase is "Closed," the user does not need to manually create a "hide" condition for all phases besides "Closed."

Users should make sure when deleting interactions that they remove both interactions, as unintended duplicate interactions can result from creating new interactions after only deleting half of

the pre-existing interaction. This automatic inverse creation does not apply to numeric-type fields. Users will need to refresh after creating a new interaction to see the inverse condition listed.

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

Adding an interaction for sample fields

In this example, we will design an interaction for the **Discount Eligible** and **Discount Eligible Amount** custom fields created in the [Custom Field Creation Example](#) of the Screen Designer Walkthrough guide. With this interaction, we will ensure that the Discount Eligible Amount field only shows if the Discount Eligible field is checked.

1. Click the **Discount Eligible Amount** field. Select this field because it will be the field shown/hidden based on a different field.
2. Click **New** to start designing your interaction.
3. From the **Interactions** panel, adjust the **If** section to the following:
 - Selected Field: Discount Eligible
 - Operator: Equals
 - Value: True
4. From the **Interactions** panel, ensure the **Then** section shows the following for **Discount Eligible Amount**:
 - Affected Field: Discount Eligible Amount
 - Visibility: Show
 - Required: Required
5. Click **Save**.

1.3.5 Publishing Your Screen

After your block is designed with all requisite rows, fields and interactions, you will need to save and publish the block for use in the TeamConnect application. Once published, the block will be applied to the selected object for your teams. The following example is built on the [Screen Designer Walkthrough](#) fields. For general details on this process, please see [Saving and Publishing Custom Blocks](#).

Saving and Publishing a screen to TeamConnect

1. Once the block is fully designed, click **File> Save Block**.
2. After saving, click **File>Publish Block**. Enter the desired file name and click **OK**. *Note: You do not have to save prior to publishing, though it is a best practice recommendation.*

Adding a published screen to your instance

1. Navigate to **Setup** in the top-right corner of your TeamConnect instance.
2. From the **Go to...** drop-down menu, select **Object Definitions**. Select the object under which you created your custom screen. For our example, this is the Contact object.
3. Click the **Blocks** tab.
4. Select the .scr file you published in the Screen Designer application and add a Friendly Name (label) for it. Save.
5. Navigate to the **Object Views** tab and select the view you are using for the object. *If you are unsure, you can click the System Settings... drop-down menu and click Default Object Views.*
6. Using the **Add the block** field, select your field to add to an existing block. Users can also add a new tab for their block if they so choose. Click add to finalize upload of the block.
7. Save. You may need to log out and log back into the application to view this new field.

Object View |

General

This view is defined for Contact

* View Name: Streamlined Conta
* Unique Key: StreamlinedContact

Preview

* Nav tab name:

Reposition highlighted tab

Profile	Notes & Docs	Rates	Employees	Involvement	Relations	Discount Eligible
---------	--------------	-------	-----------	-------------	-----------	-------------------

Add the block

Discount Contact

* Required fields are indicated by an asterisk.

Newly Created Tab for Custom Block

New Person

Discount Contact

Discount Eligible: <input checked="" type="checkbox"/>	*Discount Eligible Amount: <input type="text" value="100.00"/>	
*First Name: <input type="text"/>	Middle Name: <input type="text"/>	*Last Name: <input type="text"/>

*Required fields are noted by an asterisk

Successfully Added Custom Field

2 Screen Designer 5.2 Release Notes

This document details the upgrade information, new features, resolved issues, and known issues that are included in TeamConnect Screen Designer 5.2.

1 Installation and System Requirements

System Requirements

To install TeamConnect Screen Designer 5.2, you must be running the following versions:

- TeamConnect Enterprise 5.2 or later

Java 8 (JDK 8) or later is required to run TeamConnect Screen Designer. Users must have administrator rights to install Screen Designer.

Screen Designer runs on versions of Microsoft Windows that support Java 8. For further information on Java 8'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. 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.

2 Changes and Enhancements

Screen Designer 5.2 contains enhancements designed to improve the functionality and user experience in the four key areas. These enhancements are organized below in the following format:

- A description of the issue
- Tracking code
- Additional detail or documentation, if available

Ease of Use Enhancements

Enhancement: Users are able to drag-and-drop onto the canvas while scrolling. This allows users designing with smaller windows or larger screens to quickly add elements in one motion.

Tracking code: TC-5685, TC-7296

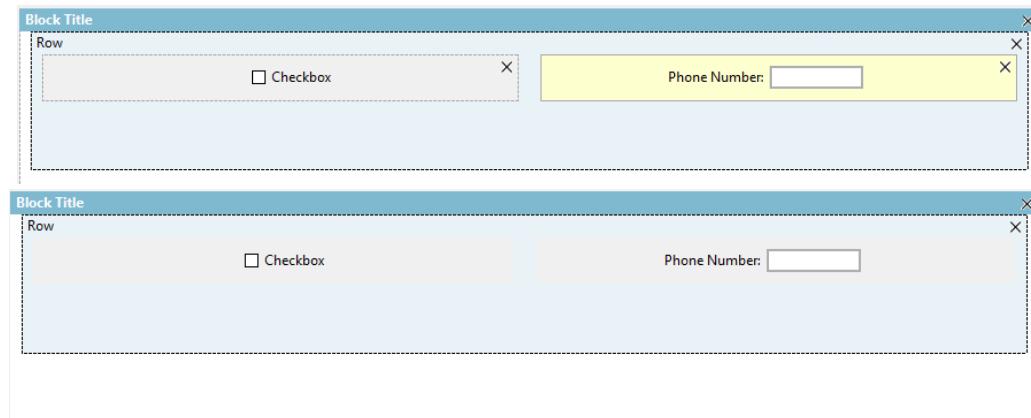
Enhancement: When a given field or object is selected, it will be highlighted so that users can more easily see which part of the screen they are designing.

Tracking code: TC-8707

Enhancement: From the **View** tab at the top of the screen, users can toggle grid visibility. This will adjust whether or not borders are present on elements on the screen while designing.

Tracking code: TC-7309

Additional detail: The image above is Grid Visibility on, while the lower graphic is Grid Visibility off.



Publishing Enhancements

Enhancement: After initially publishing a block, the publishing dialog box will preserve your settings to save clicks when publishing multiple updates for the same screen.

Tracking code: TC-5742, TC-7337

Enhancement: The "Title" property has been added for designating the hover-over text on a given field.

Tracking code: TC-5643

Additional detail:

Properties	
label	Phone Number
key	
id	
name *	
category	
forceEditable	
forceNotEditable	true
checked	
onClick	
labelFontFamily	
labelFontSize	
labelFontColor	
colon	true
title	Primary Phone Number

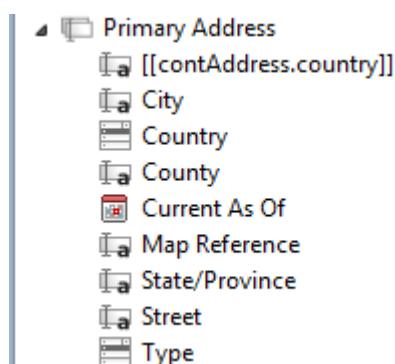
Enhancement: Blocks are alphabetically sorted in the publishing drop-down list.

Tracking code: TC-7591

Enhancement: When items within a Category are children of a parent category/object, they will now appear as nested, indented items underneath the parent object.

Tracking code: TC-5680

Additional detail:



Layout and Design Enhancements

Enhancement: A row object has been created to house all fields within a block.

Tracking code: TC-7305, TC-7588

Additional detail: Rows can be configured to have disproportionate column widths. For more information, please see the *Creating a New Block* page in the Screen Designer 5.2 Help Guide.

Enhancement: Memo field labels can now be "top-aligned" or "middle-aligned."

Tracking code: TC-7745

Enhancement: Row functionality applies to JSP elements as well as standard/advanced elements.

Tracking code: TC-8347

Enhancement: A new element "Empty" is available and can be used to add spacing to your screens.

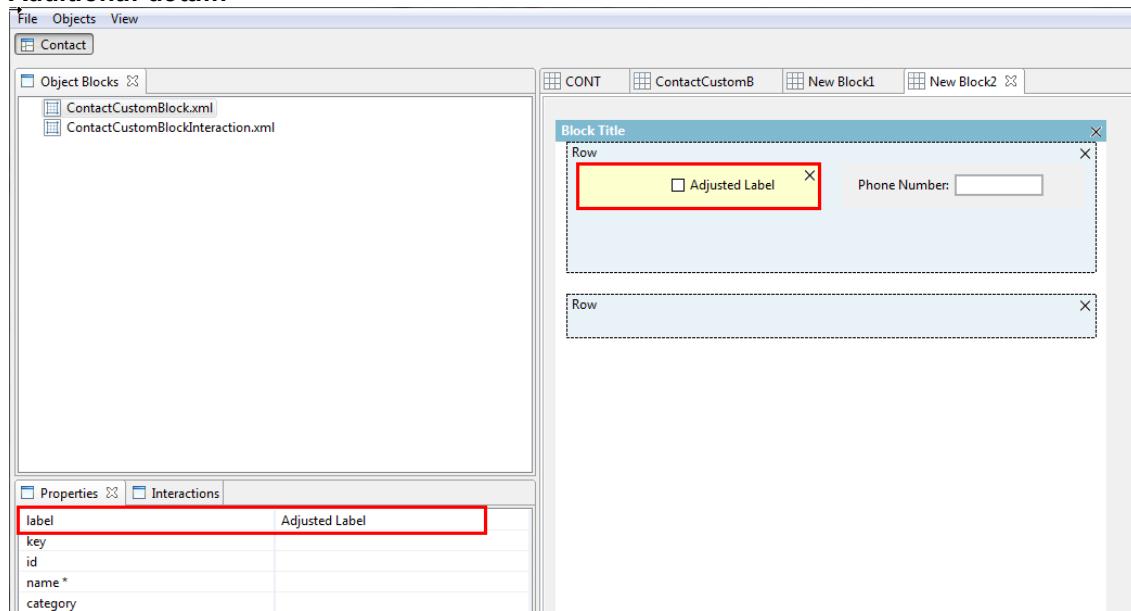
Tracking code: TC-7131

Additional detail: The element adds an empty <div> to the screen to adjust positioning of other items.

Changes to Properties and Elements

Enhancement: Users can change the field label text from the **Label** property without having to enter the XML editing view.

Tracking code: TC-5670

Additional detail:

Enhancement: The "colon" property has been added so that users can show or hide colons after labels. Setting the property to "true" will display the colon, while setting to false will remove the colon.

Tracking code: TC-5671, TC-7578

Enhancement: Users can drag in reports and report parameters to be used in Screen Designer. This functionality was present in the initial 1.x Screen Designer release.

Tracking code: TC-5682

Additional detail: For more information on using reports, please see *Reporting and Screen Designer* in the Screen Designer Help.

Enhancement: The "Current Phase" system field is available in Screen Designer and can be used in designing interactions. This is a read-only field in TCE.

Tracking code: TC-5673

Additional detail: The Current Phase field can be leveraged in Interactions to show or hide fields based on the matter phase. For example, users can designate that if the Current Phase field EQUALS "closed," a designated field can be automatically hidden.

Interactions Enhancements

Enhancement: After creating an interaction, the inverse condition is automatically created. Users do not need to manually define the opposite of their action. This applies to both show/hide functionality as well as required/not required and will only apply to newly created interactions. As an example, if a user set a given field to "show" if the Current Phase is "Closed," the user does not need to manually create a "hide" condition for all phases besides "Closed."

Tracking code: TC-5646

Additional detail: Users should make sure when deleting interactions that they remove both interactions, as unintended duplicate interactions can result from creating new interactions after only deleting half of the pre-existing interaction. This automatic inverse creation does not apply to numeric-type fields. Users will need to refresh after creating a new interaction to see the inverse condition listed.

Enhancement: Show/Hide functionality can be applied to entire blocks.

Tracking code: TC-5647

Additional detail: Click and highlight the block to begin designing block interactions. Hiding an entire block will override field settings, e.g., a field set to "show" will not be visible if it is within a hidden block.

Enhancement: The **Contains** and **Does Not Contain** operator has been added for use in interactions.

Tracking code: TC-7467, TC-90141

Additional detail: For more information on how the Contains operator functions, please see the [Interactions](#) page of the Screen Designer Help.

Enhancement: There is an Interactions tab at the bottom of the canvas that serves as a master list of all created interactions.

Tracking code: TC-9702

3 Known Issues

The following items are known Screen Designer 5.2 limitations. Each known issue is documented in the following format:

- A description of the issue

- Tracking code
- Workaround, if available

Issue: If you nest a JSP element within another JSP element (e.g., "If" element within another "If" element), the label will disappear.

Issue: Screen Designer does not let JSP elements be dropped outside of a block.

Issue: When designing an interaction with two lists where the selection in the first list determines the options in the second (via Root Item Code), users should not use the new Contains operator. Rather, they should leverage the Equals operator as done prior to the Screen Designer 5.2 release.

Tracking code: TC-8754

Additional detail: An example of this scenario would be two multi-value lists where list 'A' has Canada AND Mexico as selections and list 'B' has respective regions of these countries. The interaction will be designed so that "IF list A EQUALS Canada" THEN display "Alberta; Ontario" as list B items. Similarly, selecting "IF list A EQUALS Mexico" THEN the interaction will display "Oaxaca; Guerrero;" as list B items. For more information on interactions between two multi-value lists (where the options available are contingent on the first list), please see *Multi-Value List Interactions* in the online help.

Issue: After making changes to the font style or color of a list field value, Screen Designer may require a refresh to reflect these changes.

Tracking code: TC-10505

Issue: Interactions designed to trigger if a system numeric field IS NULL will not be triggered in TeamConnect Enterprise; system fields default to '0'. Numeric type custom fields should be set to '0' as their default value as a workaround.

Tracking code: TC-9631

Issue: If users decide to create a screen with the deadlineID field, they should make sure to not set this ID value; this value must not be adjusted to properly function as designed.

Tracking code: TC-6599

Issue: Users may encounter issues when scrolling to the top of a screen while designing.

Tracking code: TC-10771

Additional detail: Maximizing the window size will fix this issue.

Index

- C -

custom blocks
 publishing 49
 saving 49
custom screen 50

- E -

elements
 advanced 14
 HTML 14
 JSP 14

- F -

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

- G -

getting started screen 51

- I -

interactions
 adding 47
 operators 47

- J -

Java classes 18

- L -

logging in

Screen Designer 5

- P -

profiles 5
properties 20
 editing 19
 font properties 19

- R -

release notes 60
reports
 Screen Designer 11

- S -

screen designer 50, 51, 55, 60

- U -

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

- W -

walkthrough 50