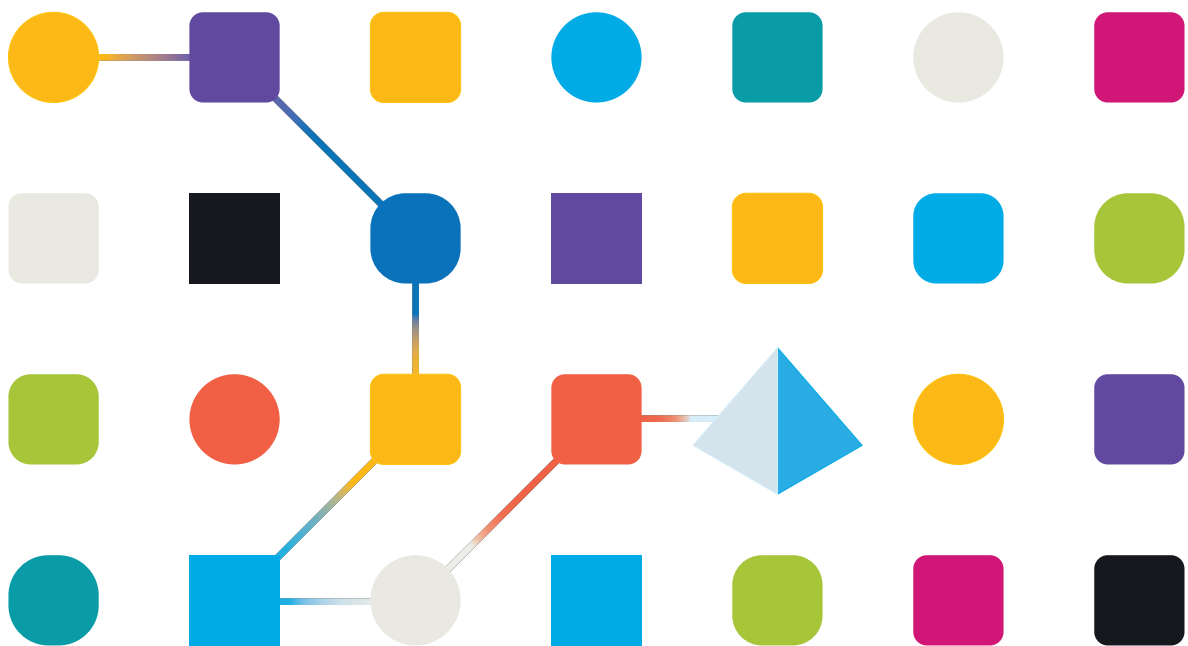




# Interact Web API Service 4.3

## User Guide

Document Revision: 2.0



## Trademarks and Copyright

The information contained in this document is the proprietary and confidential information of Blue Prism Limited and should not be disclosed to a third-party without the written consent of an authorized Blue Prism representative. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying without the written permission of Blue Prism Limited.

### © 2023 Blue Prism Limited

“Blue Prism”, the “Blue Prism” logo and Prism device are either trademarks or registered trademarks of Blue Prism Limited and its affiliates. All Rights Reserved.

All trademarks are hereby acknowledged and are used to the benefit of their respective owners. Blue Prism is not responsible for the content of external websites referenced by this document.

Blue Prism Limited, 2 Cinnamon Park, Crab Lane, Warrington, WA2 0XP, United Kingdom.  
Registered in England: Reg. No. 4260035. Tel: +44 370 879 3000. Web: [www.blueprism.com](http://www.blueprism.com)

# Contents

|  |           |
|--|-----------|
| <b>Interact Web API Service</b>                        | <b>4</b>  |
| Overview   | 4         |
| <b>Submission</b>                                      | <b>5</b>  |
| <b>Communicating back to Interact</b>                  | <b>7</b>  |
| Configure credentials for the Interact Web API service | 10        |
| <b>Actions</b>   | <b>12</b> |
| Get Submission   | 12        |
| Get Form Schema  | 15        |
| Get Users by Form Name                                 | 19        |
| Create Submission                                      | 21        |
| Raise Submission                                       | 22        |
| Upload File  | 23        |
| Edit Submission Field Value                            | 25        |
| Change Field State to Optional                         | 35        |
| Change Field State to Mandatory                        | 36        |
| Change Field State to Read Only                        | 37        |
| Change Field State to Hidden                           | 38        |
| Change Page State to Visible                           | 39        |
| Change Page State to Hidden                            | 40        |
| Move Submission to History Tab                         | 41        |
| Move Submission to Approved Tab                        | 42        |
| Move Submission to Review Tab                          | 43        |
| Move Submission to Declined Tab                        | 44        |
| Move Submission to Draft Tab                           | 45        |
| Move Submission to Inbox Tab                           | 46        |
| Move Submission to Archived Tab                        | 47        |
| <b>Human/Digital Worker collaboration</b>              | <b>48</b> |
| <b>Interact Web API Service object restrictions</b>    | <b>50</b> |

## Interact Web API Service

This user guide covers Blue Prism® Interact Web Application Programming Interface (API) Service.

Blue Prism® Interact enables developers to create web-based forms that will allow a user to interact with automations, not just at the point of initialization but also during the process if a human is required to review progress, enabling a “human/Digital Worker collaboration” approach.

Information from Interact is added into a Blue Prism queue for processing by the Digital Worker. Blue Prism can then send updated information, dynamically, back to Interact. This way a user can review and act upon the processed data and participate in the automation process.

It is assumed as part of this guide that the user is familiar with the Blue Prism digital workforce and has experience with related components such as Blue Prism® Hub and Blue Prism.

### Overview

As a digital workforce expands the addressable use cases of an enterprise, there is a further need for Digital Workers and humans to collaborate in the end-to-end execution of a business process. In addition, innovative ways of assigning work to a digital workforce are required to provide flexibility on who can interact with the resource. Blue Prism Interact provides new and existing Blue Prism users with a collaboration interface for end users to interact with their digital workforce within a business process. Initiate, verify, receive and authorize varied work related to your business processes. Blue Prism Interact reduces the skill criteria required by allowing users to create dynamic web interfaces either by using the no-code form designer or using a Blue Prism Visual Business Object (VBO) to create forms based on the business process requirements.

Blue Prism Interact functionality is split across two Blue Prism components:

- The Interact plugin within Blue Prism® Hub – Enables a developer utilizing Hub and the Forms plugin to create and publish Forms for an end user to use.
- The Interact web-based application – Provides the user interface where end users can, by utilizing published Forms, interact with the digital workforce.

The Interact end user interface is accessed via a web browser, negating the need for any locally installed software. Interact can be accessed as an internal web application or published externally.

This user guide covers the gathering of information from a Blue Prism queue and how the information can be updated and returned back to the Interact user interface after processing.

As well as detailing how to utilize the features of the Interact Web API Service, we will also demonstrate the functionality by stepping through an example process illustrating how Forms can be updated dynamically.

## Submission

Submitting a Form is detailed in the [Interact user guide](#). All submissions are processed through the IADA parser and are added to the Blue Prism queue defined in the Form. This information can then be retrieved from the queue using the Get Next Item action from the Blue Prism Internal Business Objects, Work Queues.

Using the Get Next Item action retrieves a collection into your automation ready for processing. It is recommended that the collection is not defined with fields. This is so all the information can be pulled from Interact. Defining fields in the collection will cause the automation to fail if the Form is updated with a new field or a field is deleted.

A typical collection is displayed below.

Collection Properties

Name: Data

Description:

Fields Initial Values Current Values

| first-name (Text) | last-name (Text) | date-change-request (Text) | name-of-new-bank (Text) | sort-code (Text) |
|-------------------|------------------|----------------------------|-------------------------|------------------|
| Alfred            | Jones            | 30/11/2020                 | MoneySafe Ltd           | 123456           |

Rows:

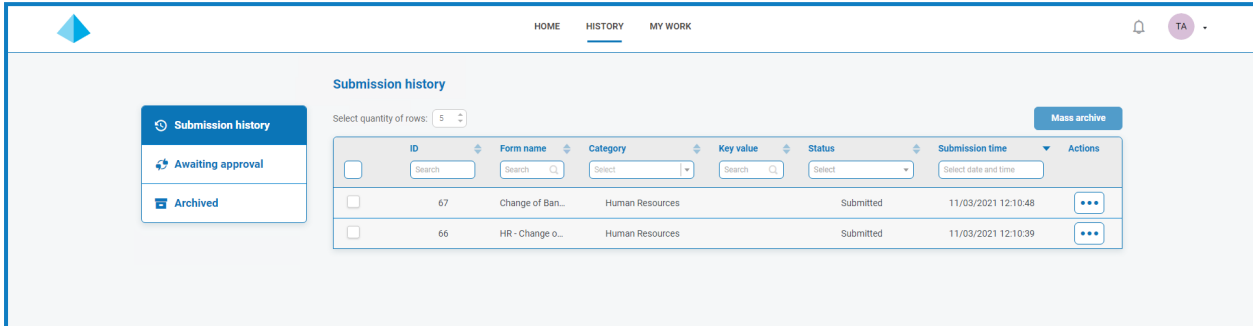
Add Remove

☒ Reset to Initial Value whenever this page runs ☒ Hide from other pages in the process ☐ Single Row

OK Cancel

It is important to note one data item that is automatically added to the collection, this is the '\_requestId' data item.

This data item is the submission ID of the Form submitted by the user. This ID is the number showing in the Submission History page as illustrated below.



Submission history


Select quantity of rows: 5

Mass archive

|                          | ID                                  | Form name                           | Category                            | Key value                           | Status                              | Submission time                                   | Actions |
|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|---|---------|
| <input type="checkbox"/> | <input type="text" value="Search"/> | <input type="text" value="Search"/> | <input type="text" value="Select"/> | <input type="text" value="Search"/> | <input type="text" value="Select"/> | <input type="text" value="Select date and time"/> |         |
| <input type="checkbox"/> | 67                                  | Change of Ban...                    | Human Resources                     |                                     | Submitted                           | 11/03/2021 12:10:48                               | ...     |
| <input type="checkbox"/> | 66                                  | HR - Change o...                    | Human Resources                     |                                     | Submitted                           | 11/03/2021 12:10:39                               | ...     |


This is the first key item of the linkage between Interact and Blue Prism. The second part of the linkage is the fields in the Form that contain the data items.

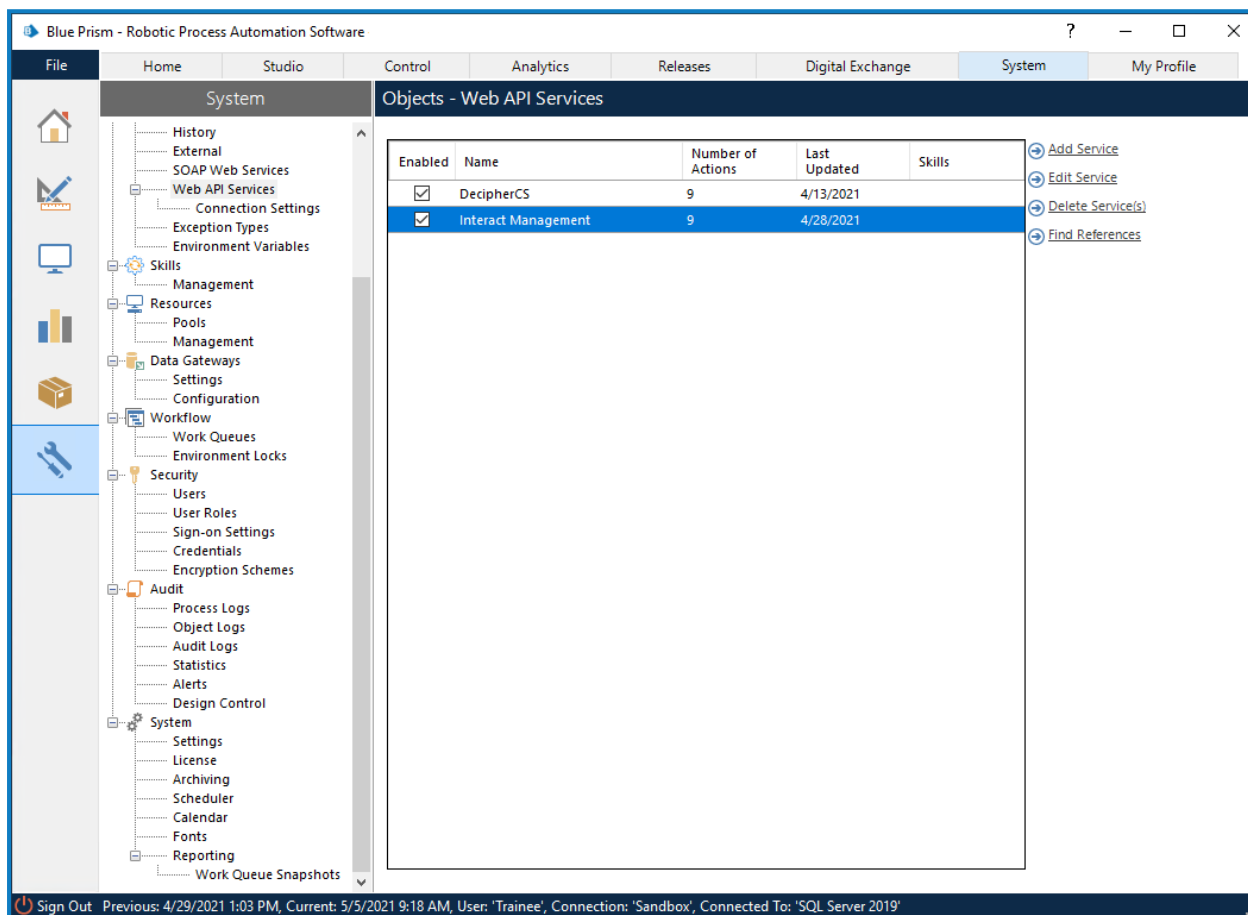
You can see in the collection displayed above, the Automation ID for the data items, for example first-name, last-name, date-of-employment, and so on. These are the 'names' that were defined for the fields when the Form was created in Hub. Hub will suggest an Automation ID for the field as you type the Label, you can choose to use the one supplied or change it for something else to suit your particular need.

 Automation IDs in Form fields must be in lowercase and the use of special characters is not allowed. If a space is entered, then the character is replaced by a hyphen '-' character. This is to aid readability.

## Communicating back to Interact

The communication channel back to Interact from Blue Prism is performed by a Blue Prism Web Service. The web service is available as a release that can be imported into Blue Prism. Once imported it will need to be updated with the appropriate base URL and authorization codes to enable secure communication.

 If you upgrade Interact to a new release, you must ensure you import the latest release file into Blue Prism.



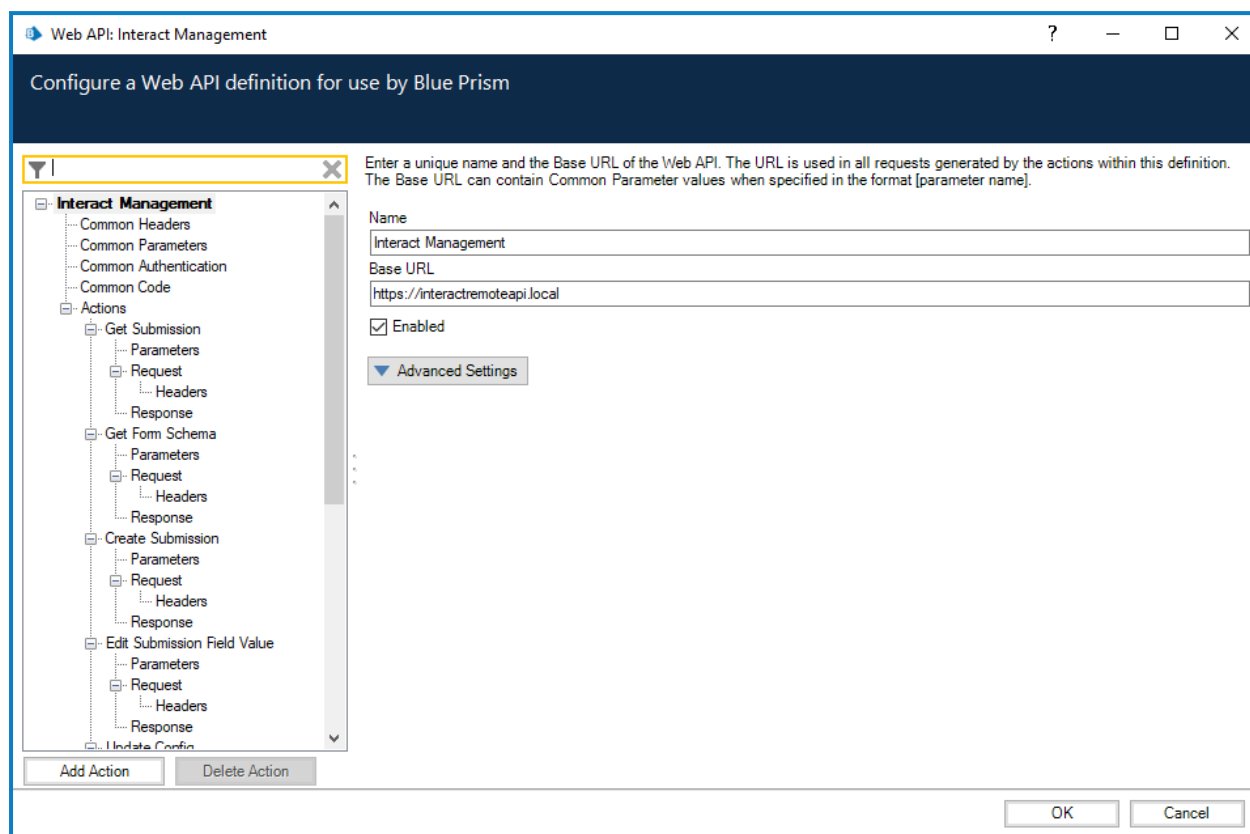
The screenshot shows the Blue Prism - Robotic Process Automation Software interface. The 'System' tab is selected, and the 'Objects - Web API Services' pane is open. The pane displays a table with the following data:

| Enabled                             | Name                | Number of Actions | Last Updated | Skills |
|-------------------------------------|---------------------|-------------------|--------------|--------|
| <input checked="" type="checkbox"/> | DecipherCS          | 9                 | 4/13/2021    |        |
| <input checked="" type="checkbox"/> | Interact Management | 9                 | 4/28/2021    |        |

On the right side of the table, there are four links: [Add Service](#), [Edit Service](#), [Delete Service\(s\)](#), and [Find References](#). The left sidebar shows a tree view of the system structure, including History, External, SOAP Web Services, Web API Services, Connection Settings, Exception Types, Environment Variables, Skills, Management, Resources, Pools, Data Gateways, Settings, Configuration, Workflow, Work Queues, Environment Locks, Security, Users, User Roles, Sign-on Settings, Credentials, Encryption Schemes, Audit, Process Logs, Object Logs, Audit Logs, Statistics, Alerts, Design Control, System, Settings, License, Archiving, Scheduler, Calendar, Fonts, Reporting, and Work Queue Snapshots.

At the bottom of the window, a status bar displays: **Sign Out** Previous: 4/29/2021 1:03 PM, Current: 5/5/2021 9:18 AM, User: 'Trainee', Connection: 'Sandbox', Connected To: 'SQL Server 2019'

The image below shows the web API service.



In the main window of the web API service we can see the Base URL defined, this will need to be updated to reflect the configuration of your platform setup.



The Common Parameters section is left blank and the Common Authentication section is used. This requests a Bearer Token using stored credentials from the Credential store.

Web API: New Web API Service

Configure a Web API definition for use by Blue Prism

This authentication type uses a client identifier and client secret stored in a credential to retrieve an OAuth 2.0 access token from an authorization server. The access token is then used to authenticate Web API requests.

Authentication Type  
OAuth 2.0 (Client Credentials)

Authorization URI  
https://authentication.local/connect/token

Credential  
Bearer Token

☒ Expose to process

Parameter Name  
OAuth 2 (Client Credentials) Authentication Credential Name

Scope  
interact-remote-api

Add Action Delete Action

OK Cancel

To set up the credentials, a 'client\_secret' must be configured. This is generated and displayed during the installation process and is stored in the Interact Remote API Secret Key file in C:\Program Files (x86)\Blue Prism\Interact Remote API secret key.txt. See the [Interact Installation guide](#) for more details.

## Configure credentials for the Interact Web API service

When configuring the credentials for the OAuth 2.0 (Client Credentials) the following four values must be configured in the Blue Prism System Settings ([Security > Credentials](#)):

- **Client ID** – Enter *InteractRemoteClient*.
- **Client Secret** – Enter the code displayed and copied during installation or, if not copied, from the secret key file contents found in: C:\Program Files (x86)\Blue Prism\Interact Remote API secret key.txt.
- **grant\_type** – Set the value to *client\_credentials*.
- **scope** – Set the value to *interact-remote-api*.

**Credential Details**

Name: Interact Credentials

Description: Credentials for the Interact Remote API

Type: OAuth 2.0 (Client Credentials)

**Application Credentials** | Access Rights

Use this credential type for OAuth 2.0 web authentication using client credentials.

Client ID: InteractRemoteClient

Expires: ☒ 2/10/2099

Client Secret: .....

☐ Marked as invalid

**Additional Properties**

| Name       | Value |
|------------|-------|
| grant_type | ..... |
| scope      | ..... |
|            |       |

OK Cancel

Remember to set up the Access Rights tab in the Credentials Details to ensure it can access the processes for the allowed users.


The web service has nine defined actions. These actions are available within the Utility - Interact API Blue Prism VBO along with a number of additional actions which map to the web service. The actions are:

| Interact Web API            | Blue Prism VBO: Utility - Interact API   |
|-----------------------------|--|
| Get Submission              | Get Submission   |
| Get Form Schema             | Get Form Schema  |
| Create Submission           | Create Submission  |
| Edit Submission Field Value | Edit <Type> Field Value, where the type is: <ul style="list-style-type: none"> <li>• <a href="#">Checkbox Single</a></li> <li>• <a href="#">Checkbox Multiple</a></li> <li>• <a href="#">Date</a></li> <li>• <a href="#">Dropdown Single</a></li> <li>• <a href="#">Dropdown Multiple</a></li> <li>• <a href="#">Number</a></li> <li>• <a href="#">Radio</a></li> <li>• <a href="#">Table</a></li> <li>• <a href="#">Text</a></li> <li>• <a href="#">Time</a></li> </ul> |
| Update Config               | <a href="#">Change Field State to Optional</a><br><a href="#">Change Field State to Mandatory</a><br><a href="#">Change Field State to Read Only</a><br><a href="#">Change Field State to Hidden</a><br><a href="#">Change Page State to Visible</a><br><a href="#">Change Page State to Hidden</a>  |
| Update Status               | <a href="#">Move Submission to History Tab</a><br><a href="#">Move Submission to Approved Tab</a><br><a href="#">Move Submission to Review Tab</a><br><a href="#">Move Submission to Declined Tab</a><br><a href="#">Move Submission to Draft Tab</a><br><a href="#">Move Submission to Inbox Tab</a><br><a href="#">Move Submission to Archived Tab</a>   |
| Raise Submission            | Raise Submission   |
| Get Users by Form Name      | Get Users by Form Name   |
| Upload File                 | Upload File  |

The actions are called from the Utility - Interact API VBO so, after the initial configuration of the URL and authorization codes, the Web Service does not need to be amended. Each of the actions within the Utility - Interact API VBO are explained in the next section.

## Actions

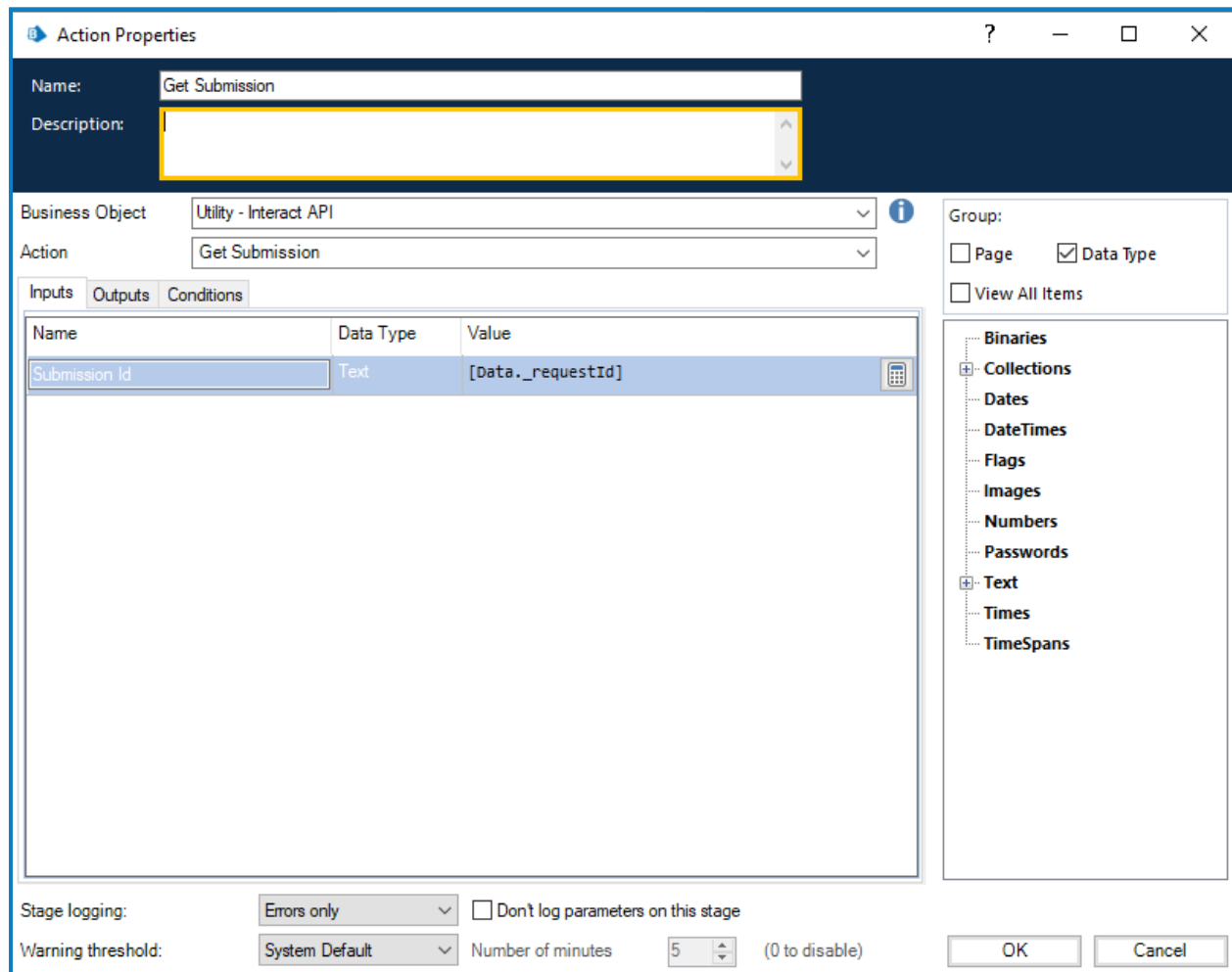
There are a number of actions within the Interact Web API Service, which are called using the Blue Prism Object (Utility - Interact API). Going through each of the actions in turn, the sections below explain how to use them in your automations.

 All of the web API service actions should be called from the Utility - Interact API VBO in Blue Prism, and not directly from the web service.

### Get Submission

The Get Submission action does a very similar action to the Get Next Item action when it retrieves the information from a queue.

To utilize the action, an action is added to your process and the Business Object selected, with the action set to Get Submission. There is only one input parameter which is the Submission ID as illustrated below.



The screenshot shows the 'Action Properties' dialog box for the 'Get Submission' action. The 'Name' field is set to 'Get Submission'. The 'Description' field is empty. The 'Business Object' is set to 'Utility - Interact API'. The 'Action' is set to 'Get Submission'. The 'Inputs' tab is selected, showing a table with one input parameter:

| Name          | Data Type | Value             |
|---------------|-----------|-------------------|
| Submission Id | Text      | [Data._requestId] |

The 'Outputs' and 'Conditions' tabs are also visible. On the right side, the 'Group' section has checkboxes for 'Page', 'Data Type' (checked), and 'View All Items'. Below this is a tree view showing various data types: Binaries, Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text (selected), Times, and TimeSpans. At the bottom, there are settings for 'Stage logging' (Errors only) and 'Warning threshold' (System Default, 5 minutes). The 'OK' and 'Cancel' buttons are at the bottom right.

The output parameters retrieves the contents of the Interact Form, using the submission ID into a collection.

?

—

□

×

Action Properties

Name:

Get Submission

Description:

Business Object

Utility - Interact API

Action

Get Submission

Inputs

Outputs

Conditions

| Name            | Data Type  | Store In  |
|-----------------|------------|---|
| Submission Data | Collection | <input checked="" type="checkbox"/> Submission Data |

Group:

☐ Page
 ☒ Data Type

☐ View All Items

Binaries

+

 Collections

Dates

DateTimes

Flags

Images

Numbers

Passwords

+

 Text

Times

TimeSpans

Stage logging:

Errors only

☐ Don't log parameters on this stage

Warning threshold:

System Default

Number of minutes

5

(0 to disable)


OK

Cancel

One difference when the submission is retrieved using this methodology rather than the standard Get Next Item action is the formatting of Date elements. Using the Get Next Item technique the Date is returned in a Text format, whereas the Get Submission returns it as a DateTime format as below.

| first-name (Text) | last-name (Text) | date-change-request (DateTime) | name-of-new-bank (Text) | sort-code (Text) |
|-------------------|------------------|--------------------------------|-------------------------|------------------|
| Alfred            | Jones            | 11/30/2020 12:00:00 AM         | MoneySafe Ltd           | 123456           |

For a complete list of restrictions, see [Interact Web API Service object restrictions on page 50](#).

 You cannot use the Get Submission action unless you know the submission ID, as this is unique per submission it cannot be hardcoded into your automation. This must be retrieved using the Get Next Item action.

## Get Form Schema

The Get Form Schema action gets a structure of an Interact Form in a collection allowing you to manipulate and then using the Create Submission action create an Interact Form either empty or partially created for a user to complete and submit.

There is only one input parameter which is the name of the Form within Interact that you are retrieving the structure for, as illustrated below.

The screenshot shows the 'Action Properties' dialog box for the 'Get Form Schema' action. The 'Name' field is set to 'Get Form Schema'. The 'Description' field is empty. The 'Business Object' is 'Utility - Interact API' and the 'Action' is 'Get Form Schema'. The 'Inputs' tab is selected, showing a table with one input parameter:

| Name      | Data Type | Value                    |
|-----------|-----------|--------------------------|
| Form Name | Text      | "Change of Bank Details" |

The 'Outputs' and 'Conditions' tabs are also visible. On the right, the 'Group' section has checkboxes for 'Page', 'Data Type' (checked), and 'View All Items'. Below this is a tree view of data types: Binaries, Collections, Data, Form Schema, Submission Data, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text (selected), Times, and TimeSpans. At the bottom, the 'Stage logging' section has a dropdown set to 'Errors only' and a checkbox for 'Don't log parameters on this stage'. The 'Warning threshold' is set to 'System Default' with a dropdown, and a 'Number of minutes' field set to '5' (0 to disable). 'OK' and 'Cancel' buttons are at the bottom right.

The outputs parameters are similar to the 'Get Submission' action where the request status is returned along with a collection containing the fields.

?

—

□

×

Action Properties

Name:

Get Form Schema

Description:

Business Object

Utility - Interact API

Action

Get Form Schema

Inputs

Outputs

Conditions

| Name        | Data Type  | Store In  |
|-------------|------------|---|
| Form Schema | Collection | <input checked="" type="checkbox"/> Form Schema |

Group:

☐ Page
☒ Data Type

☐ View All Items

Binaries

Collections

Data

Form Schema

Submission Data

Dates

DateTimes

Flags

Images

Numbers

Passwords

Text

Times

TimeSpans

Stage logging:

Errors only

☐ Don't log parameters on this stage

Warning threshold:

System Default

Number of minutes

5

(0 to disable)

OK

Cancel



The retrieved collection is a blank collection containing the structure of the Form.

Collection Properties

Name: Form Schema

Description:

Fields Initial Values Current Values

| first-name (Text) | last-name (Text) | date-change-request (Text) | name-of-new-bank (Text) | sort-code (Text) |
|-------------------|------------------|----------------------------|-------------------------|------------------|
|                   |                  |                            |                         |                  |

Rows:

Add Remove

☒ Reset to Initial Value whenever this page runs ☒ Hide from other pages in the process ☐ Single Row

OK Cancel

The structure of the fields presented in a similar format as the Get Next Item format and therefore the Date element is requesting a Text data type to be submitted. The reason that Text type is used, is to support the multiple format Date element types within Interact, where the Date can be supplied in UK, US or International format.

The difference between the Get Next Item and Get Form Schema formats is the upload element field. The Get Next Item action returns a link to where the file is stored, however, Get Form Schema action returns a collection which includes not just a link but details of the file uploaded. This is also required when using the Upload File action.

Collection Properties

Name: Submission Data

Description:

Submission Data attach-cv

| Fields      | Initial Values | Current Values |                  |                               |                 |
|-------------|----------------|----------------|------------------|-------------------------------|-----------------|
| name (Text) |                | size (Number)  | extension (Text) | link (Text)                   | type (Text)     |
| Invoice.pdf |                | 32808          | pdf              | https://file.local/Files/self | application/pdf |

☒ Reset to Initial Value whenever this page runs ☒ Hide from other pages in the process ☐ Single Row

Add Row Remove Row

OK Cancel

The format of the collection containing the details of the upload file must be set correctly, as illustrated above:

- **Name** – The name of the file including the extension.
- **Size** – The size of the file in bytes.
- **Extension** – The file extension, without the leading period or full stop.
- **Link** – The link to the uploaded file.
- **Type** – The type of file for example 'application/pdf' or 'image/png'.

## Get Users by Form Name

The Get Users by Form Name action allows you to collect a list of users associated with a particular Form Name.

The input for the action is the Form Name, as illustrated below.

The screenshot shows the 'Action Properties' dialog box for the 'Get Users by Form Name' action. The dialog is titled 'Action Properties' and has a dark blue header. The 'Name' field is set to 'Get Users by Form Name'. The 'Description' field is empty. The 'Business Object' is set to 'Utility - Interact API'. The 'Action' is set to 'Get Users by Form Name'. The 'Inputs' tab is selected, showing a table with columns 'Name', 'Data Type', and 'Value'. The table contains one row: 'Form Name' (Text) with the value 'New Starter'. The 'Outputs' and 'Conditions' tabs are also visible. On the right side, there is a 'Group' section with checkboxes for 'Page' (unchecked) and 'Data Type' (checked). Below this is a 'View All Items' checkbox (unchecked). A list of data types is shown: Binaries, Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text, Times, and TimeSpans. At the bottom, there is a 'Stage logging' section with a dropdown set to 'Errors only' and a checkbox for 'Don't log parameters on this stage' (unchecked). The 'Warning threshold' is set to 'System Default' with a dropdown, and a 'Number of minutes' field set to '5' (0 to disable). 'OK' and 'Cancel' buttons are at the bottom right.

| Name      | Data Type | Value         |
|-----------|-----------|---------------|
| Form Name | Text      | "New Starter" |

The output is a collection that lists the users associated with that Form Name.

**Action Properties**

Name:

Description:

Business Object:

Action:

Group: ☐ Page ☒ Data Type ☐ View All Items

Inputs Outputs Conditions

| Name  | Data Type  | Store In                                  |
|-------|------------|---|
| Users | Collection | <input checked="" type="checkbox"/> Users |

Binaries

- Collections
- Dates
- DateTimes
- Flags
- Images
- Numbers
- Passwords
- Text
- Times
- TimeSpans

Stage logging:  ☐ Don't log parameters on this stage

Warning threshold:  Number of minutes  (0 to disable)

OK Cancel

## Create Submission

There are four parameters that need to be defined within the Create Submission action.

**Action Properties**

Name: Create Submission

Description:

Business Object: Utility - Interact API

Action: Create Submission

Inputs | Outputs | Conditions

| Name      | Data Type  | Value                    |
|-----------|------------|--------------------------|
| Form Name | Text       | "Change of Bank Details" |
| Username  | Text       | [Username]               |
| Password  | Password   | [Password]               |
| Fields    | Collection | [Form Schema]            |

Group: ☐ Page ☒ Data Type ☐ View All Items

Binaries  
Collections  
Dates  
DateTimes  
Flags  
Images  
Numbers  
Passwords  
Text  
Times  
TimeSpans

Stage logging: Errors only ☐ Don't log parameters on this stage

Warning threshold: System Default Number of minutes 5 (0 to disable)

OK Cancel

These four parameters are as follows:

- **Username** – This is the username of the Interact user that the submission will be created for.
- **Password** – The password for this Interact user.
- **Form Name** – The Form that will be created within Interact.
- **Fields** – The values that will be set within the created submission.

To assist in creating a new submission, [Get Form Schema on page 15](#) can be used to retrieve the basic structure of the Form. This could then be populated with new data items and then created as a new submission for a user.



To create a submission, the fields and form schema must match that of the Form within Interact. Missing fields or an incorrect structure will end up creating a corrupt submission.


Once initiated the created submission will appear in the specified users Submission History as a submitted Form and will be added to the specified Blue Prism queue detailed in the Form construct.

The user whose credentials are used to create the new submission must be allowed to access the Form. So, the Interact Role which contains the Form must have the user specified in the user list.

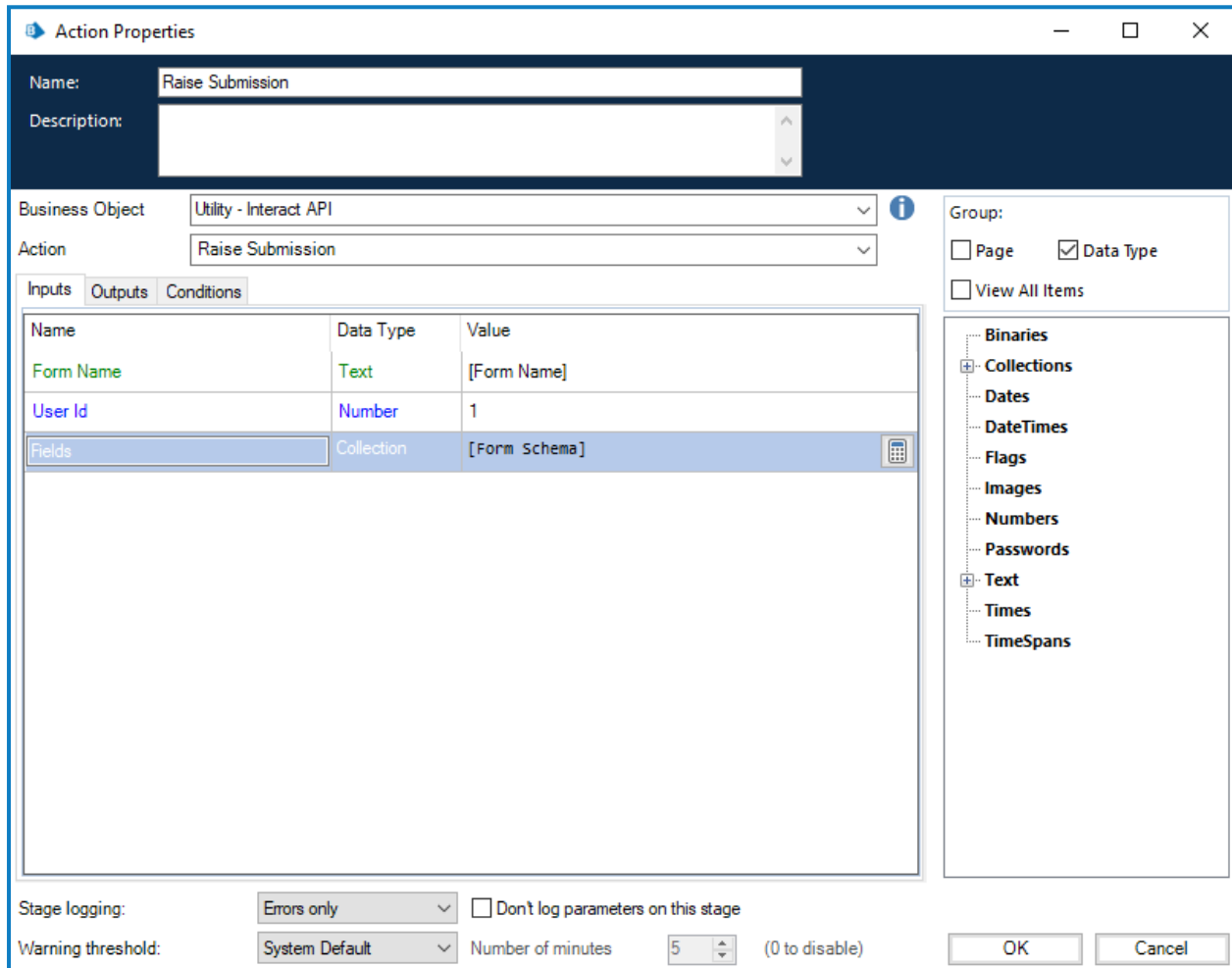
There are no outputs parameters for the Create Submission action.

## Raise Submission

The Raise Submission action allows you to create a submission in a user's particular Inbox folder. The submission raised can be fully or partially completed, or even completely empty depending on your requirements.

 If the form that your submission is based on contains the radio group capture type, you will be unable to raise an empty submission. The radio group requires a value to be set. For more information on form capture types, see the [Interact Plugin User Guide](#).


The inputs for the action are the Form Name, the User ID (which can be obtained by running [Get Users by Form Name](#)) and the data to be loaded in the fields, as illustrated below. The data file loaded can be created by running the [Get Form Schema](#) command.



The screenshot shows the 'Action Properties' dialog box for the 'Raise Submission' action. The 'Name' field is set to 'Raise Submission'. The 'Business Object' is 'Utility - Interact API' and the 'Action' is 'Raise Submission'. The 'Inputs' tab is selected, showing a table with three input fields: 'Form Name' (Text, Value: [Form Name]), 'User Id' (Number, Value: 1), and 'Fields' (Collection, Value: [Form Schema]). The 'Outputs' and 'Conditions' tabs are empty. On the right, the 'Group' section has 'Data Type' checked. Below it is a tree view of data types: Binaries, Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text, Times, and TimeSpans. At the bottom, 'Stage logging' is set to 'Errors only' and 'Warning threshold' is 'System Default'.

| Name      | Data Type  | Value         |
|-----------|------------|---------------|
| Form Name | Text       | [Form Name]   |
| User Id   | Number     | 1             |
| Fields    | Collection | [Form Schema] |

There are no outputs for the Raise Submission action.

 When using Raise Submission with [Upload File on the next page](#), ensure you correctly format the collection containing the details of the file being attached. See [Get Form Schema on page 15](#) for more details about the format of the collection.

## Upload File

The Upload File action allows you to upload a file to local storage, for example, an image file or a document. It can be used in conjunction with:

- [Create Submission on page 21](#)
- [Raise Submission on the previous page](#)

The Upload file action only uploads one file at a time. When uploading multiple files to a form, you must run the upload file action for each file then store the output links separately.

There are four inputs for the action, as illustrated below:

- **Form Name** – The Form within Interact that the uploaded file will be associated with.
- **Field Name** – The automation ID of the field on the Form.
- **File Name** – The file that will be uploaded into the local storage. This must include the full name of the file including the extension.
- **File** – The contents of the file attached using a Data Item with the type set to Binary.

The screenshot shows the 'Action Properties' dialog for the 'Upload File' action. The 'Name' field is set to 'Upload File'. The 'Business Object' is 'Utility - Interact API' and the 'Action' is 'Upload File'. The 'Inputs' tab is selected, showing a table with four input fields: 'Field Name' (Text, [Field Name]), 'Form Name' (Text, [Form Name]), 'File Name' (Text, [File Name]), and 'File' (Binary, [Alt File]). The 'File' input has a file icon button. On the right, the 'Group' section has 'Page' unchecked and 'Data Type' checked. Below it, 'View All Items' is unchecked. A list of data types is shown: Binaries, Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text, Times, and TimeSpans. At the bottom, 'Stage logging' is 'Enabled' and 'Warning threshold' is 'System Default'. The 'Number of minutes' is set to 5, with a note '(0 to disable)'. 'OK' and 'Cancel' buttons are at the bottom right.

| Name       | Data Type | Value        |
|------------|-----------|--------------|
| Field Name | Text      | [Field Name] |
| Form Name  | Text      | [Form Name]  |
| File Name  | Text      | [File Name]  |
| File       | Binary    | [Alt File]   |

When uploading a file, you are provided with a link which can then be used in [Create Submission on page 21](#) or [Raise Submission on the previous page](#).

**Action Properties**

Name:

Description:

Business Object:

Action:

Group: ☐ Page ☒ Data Type  
☐ View All Items

Inputs Outputs Conditions

| Name | Data Type | Store In                                 |
|------|-----------|--|
| Link | Text      | <input checked="" type="checkbox"/> Link |

Stage logging:  ☐ Don't log parameters on this stage

Warning threshold:  Number of minutes  (0 to disable)

OK Cancel

**Binaries**

**Collections**

Dates

DateTimes

Flags

Images

Numbers

Passwords

**Text**

Times

TimeSpans

Once the file has been uploaded, you can attach the contents of the file to [Create Submission on page 21](#) or [Raise Submission on page 22](#). To do this, you need to:

1. **Upload File** – Upload the file to the local storage.
2. **Get Form Schema** – Obtain the structure of the form where the file will be attached to.
3. **Create Submission or Raise Submission** – Which references the modified Form structure and attaches the uploaded file to the new submission.



When using Create Submission, the attached file can be downloaded and viewed from the Submission History tab of Interact. If Raise Submission is used, the user can only edit the link replacing the file, the user cannot view the attached file, unless it is an image, nor can they download the file.



## Edit Submission Field Value

The Edit Submission Field Value action, like all the other actions, is called from the object. There are several actions within the object which allow you to update specific field elements, these actions link directly to the element type that is used in the Interact Form.

The Field Name in the sections below is the Automation ID that was created within the Form designer. Details for the different element types are described in the following sections.

### Checkbox element (Single)

The Checkbox Edit Value is two different actions supporting use of both Single and Multiple Checkboxes. To update the values for a Single Checkbox, it is a simple text string that is submitted.

The screenshot shows the 'Action Properties' dialog box for the 'Edit Checkbox Single Field Value' action. The dialog is divided into several sections:

- Name:** Edit Checkbox Single Field Value
- Description:** (Empty text box)
- Business Object:** Utility - Interact API
- Action:** Edit Checkbox Single Field Value
- Inputs/Outputs/Conditions:** A table with three columns: Name, Data Type, and Value.

| Name          | Data Type | Value                   |
|---------------|-----------|-------------------------|
| Submission Id | Text      | [Data._requestId]       |
| Field Name    | Text      | "what-pets-do-you-have" |
| Field Value   | Text      | "dog"                   |
- Group:** ☐ Page ☒ Data Type ☐ View All Items
- Binaries:** A list of categories: Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text, Times, TimeSpans.
- Stage logging:** Errors only (dropdown) ☐ Don't log parameters on this stage
- Warning threshold:** System Default (dropdown) Number of minutes: 5 (0 to disable)
- Buttons:** OK, Cancel

## Checkbox element (Multiple)

For the Checkbox Edit Value (Multiple) again it is a simple text string that is submitted but using a comma separated string for the different values.

The screenshot shows the 'Action Properties' dialog box for the 'Edit Checkbox Multiple Field Value' action. The dialog is titled 'Action Properties' and has a dark blue header bar. Below the header, there are fields for 'Name' (set to 'Edit Checkbox Multiple Field Value') and 'Description' (empty). The 'Business Object' is set to 'Utility - Interact API' and the 'Action' is set to 'Edit Checkbox Multiple Field Value'. The 'Inputs' tab is selected, showing a table with three columns: 'Name', 'Data Type', and 'Value'. The table contains three rows: 'Submission Id' (Text, [Data\_requestId]), 'Field Name' (Text, "what-pets-do-you-have"), and 'Field Value' (Text, "dog, cat, rabbit"). The 'Field Value' row is highlighted. To the right of the table is a 'Group' section with checkboxes for 'Page', 'Data Type' (checked), and 'View All Items'. Below the 'Group' section is a tree view showing various data types: Binaries, Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text (selected), Times, and TimeSpans. At the bottom of the dialog, there are settings for 'Stage logging' (Errors only) and 'Warning threshold' (System Default). The 'Number of minutes' is set to 5, and there is a checkbox for 'Don't log parameters on this stage'. The dialog has 'OK' and 'Cancel' buttons at the bottom right.

| Name          | Data Type | Value                   |
|---------------|-----------|-------------------------|
| Submission Id | Text      | [Data_requestId]        |
| Field Name    | Text      | "what-pets-do-you-have" |
| Field Value   | Text      | "dog, cat, rabbit"      |

## Date element

The Date element uses Text format to edit values in an Interact Form, this is to support the three different formats within the Interact Form. The Form will allow you to use these formats for dates:

- DD/MM/YYYY – UK format;
- MM/DD/YYYY – US format;
- YYYY/MM/DD – International format.

If you have set the Interact Form to accept the International format for dates and you send the date back in either the UK or US format, the Interact Form will display an invalid data message.

The screenshot shows the 'Action Properties' dialog box for the 'Edit Date Field Value' action. The dialog is divided into several sections:

- Name:** Edit Date Field Value
- Description:** (Empty text box)
- Business Object:** Utility - Interact API
- Action:** Edit Date Field Value
- Group:**
  - ☐ Page
  - ☒ Data Type
  - ☐ View All Items
- Inputs/Outputs/Conditions:**

| Name          | Data Type | Value            |
|---------------|-----------|------------------|
| Submission Id | Text      | [Data_requestId] |
| Field Name    | Text      | "date-of-birth"  |
| Field Value   | Text      | "22/06/1972"     |
- Stage logging:**
  - Errors only (selected)
  - ☐ Don't log parameters on this stage
- Warning threshold:**
  - System Default (selected)
  - Number of minutes: 5 (0 to disable)
- OK** and **Cancel** buttons.

On the right side of the dialog, there is a tree view showing the following categories:

- Binaries
- Collections
- Dates
- Datetimes
- Flags
- Images
- Numbers
- Passwords
- Text
- Times
- TimeSpans

## Dropdown element (Single)

The Dropdown element, like the Checkbox element, supports two actions for the single and multiple entries. To update the values for a Single Dropdown, it is a simple text string that is submitted.

?

—

□

×

Action Properties

Name:

Edit Dropdown Single Field Value

Description:

Business Object

Utility - Interact API

Action

Edit Dropdown Single Field Value

Group:

☐ Page
☒ Data Type
☐ View All Items

Binaries

+

 Collections

Dates

DateTimes

Flags

Images

Numbers

Passwords

+

 Text

Times

TimeSpans

Inputs

Outputs

Conditions

| Name          | Data Type | Value                   |
|---------------|-----------|-------------------------|
| Submission Id | Text      | [Data_requestId]        |
| Field Name    | Text      | "what-pets-do-you-have" |
| Field Value   | Text      | "snake"                 |

Stage logging:

Errors only

☐ Don't log parameters on this stage

Warning threshold:

System Default

Number of minutes

5

(0 to disable)

OK

Cancel

## Dropdown element (Multiple)

For the Dropdown Edit Value (Multiple) again it is a simple text string that is submitted but using a comma separated string for the different values.

The screenshot shows the 'Action Properties' dialog box for the 'Edit Dropdown Multiple Field Value' action. The dialog is titled 'Action Properties' and has a dark blue header bar. Below the header, there are fields for 'Name' (set to 'Edit Dropdown Multiple Field Value') and 'Description' (empty). The 'Business Object' is set to 'Utility - Interact API' and the 'Action' is set to 'Edit Dropdown Multiple Field Value'. The 'Inputs' tab is selected, showing a table with three columns: 'Name', 'Data Type', and 'Value'. The table contains three rows: 'Submission Id' (Text, [Data.\_requestId]), 'Field Name' (Text, "what-pets-do-you-have"), and 'Field Value' (Text, "fish, snake"). The 'Field Value' row is highlighted. To the right of the table is a 'Group' section with checkboxes for 'Page', 'Data Type' (checked), and 'View All Items'. Below this is a tree view showing various data types: Binaries, Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text, Times, and TimeSpans. At the bottom, there are settings for 'Stage logging' (Errors only) and 'Warning threshold' (System Default, 5 minutes). The 'OK' and 'Cancel' buttons are at the bottom right.

| Name          | Data Type | Value                   |
|---------------|-----------|-------------------------|
| Submission Id | Text      | [Data._requestId]       |
| Field Name    | Text      | "what-pets-do-you-have" |
| Field Value   | Text      | "fish, snake"           |

## Number element

The Number element will accept either a number or a text being passed to it. Though the field type is a Text type you must supply a number using only the acceptable delimiters such as commas and decimal points. If text is sent back, the Interact Form will display a message regarding invalid data.

The screenshot shows the 'Action Properties' dialog box for the 'Edit Number Field Value' action. The dialog is titled 'Action Properties' and has a dark blue header bar. Below the header, there are fields for 'Name' (set to 'Edit Number Field Value') and 'Description' (empty). The 'Business Object' is set to 'Utility - Interact API' and the 'Action' is set to 'Edit Number Field Value'. The 'Inputs' tab is selected, showing a table with three rows: 'Submission Id' (Text, [Data.\_requestId]), 'Field Name' (Text, "salary"), and 'Field Value' (Text, "54,107.67"). The 'Field Value' row is highlighted. To the right of the table is a 'Group' section with checkboxes for 'Page', 'Data Type' (checked), and 'View All Items'. Below this is a list of data types: Binaries, Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text (selected), Times, and TimeSpans. At the bottom, there are settings for 'Stage logging' (Errors only) and 'Warning threshold' (System Default, 5 minutes). The 'OK' and 'Cancel' buttons are at the bottom right.

| Name          | Data Type | Value             |
|---------------|-----------|-------------------|
| Submission Id | Text      | [Data._requestId] |
| Field Name    | Text      | "salary"          |
| Field Value   | Text      | "54,107.67"       |

## Radio group element

The Radio group element is like a single Checkbox or Dropdown element. The edited value is a single text field which includes the required updated value.

?

—

□

×

Action Properties

Name:

Edit Radio Field Value

Description:

Business Object

Utility - Interact API

Action

Edit Radio Field Value

Inputs

Outputs

Conditions

| Name          | Data Type | Value              |
|---------------|-----------|--------------------|
| Submission Id | Text      | [Data_requestId]   |
| Field Name    | Text      | "gender-selection" |
| Field Value   | Text      | "female"           |

Group:

☐ Page
☒ Data Type

☐ View All Items

Binaries

Collections

Dates

DatesTimes

Flags

Images

Numbers

Passwords

Text

Times

TimeSpans

Stage logging:

Errors only

☐ Don't log parameters on this stage

Warning threshold:

System Default

Number of minutes

5

(0 to disable)

OK

Cancel

## Table element

The Table element uses a collection to enable you to update the values. The collection must contain the same number of columns as the form field within the targeted form. Automation IDs must be used for the field names of the collection. The number of rows can be increased or decreased to accommodate the data being updated.

The screenshot shows the 'Action Properties' dialog box for the 'Edit Table Field Value' action. The dialog is titled 'Action Properties' and has a dark blue header bar. Below the header, there are fields for 'Name' (set to 'Edit Table Field Value') and 'Description' (empty). The 'Business Object' is set to 'Utility - Interact API' and the 'Action' is set to 'Edit Table Field Value'. The 'Inputs' tab is selected, showing a table with three columns: 'Name', 'Data Type', and 'Value'. The table contains three rows: 'Submission Id' (Text, [Data\_requestId]), 'Field Name' (Text, "sales-orders"), and 'Field Value' (Collection, [Sales Orders]). The 'Field Value' row is highlighted. To the right of the table, there is a 'Group' section with checkboxes for 'Page', 'Data Type' (checked), and 'View All Items'. Below the 'Group' section is a list of data types: Binaries, Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text, Times, and TimeSpans. At the bottom of the dialog, there are settings for 'Stage logging' (Errors only) and 'Warning threshold' (System Default). The 'Don't log parameters on this stage' checkbox is unchecked. The 'Number of minutes' is set to 5, with a note '(0 to disable)'. There are 'OK' and 'Cancel' buttons at the bottom right.

| Name          | Data Type  | Value            |
|---------------|------------|------------------|
| Submission Id | Text       | [Data_requestId] |
| Field Name    | Text       | "sales-orders"   |
| Field Value   | Collection | [Sales Orders]   |

The Edit Table action overwrites the whole table as opposed to appending rows to an existing table.



The Text element requires a simple Text field being passed to it.

## Time element

The Time element uses Text format to edit values in an Interact Form. You can supply the Time in one of two formats these being:

- 03:00:00 PM; or
- 15:00:00.

Both will return 3 o'clock in the afternoon in Interact.

The screenshot shows the 'Action Properties' dialog box for the 'Edit Time Field Value' action. The 'Name' field is 'Edit Time Field Value' and the 'Description' field is empty. The 'Business Object' is 'Utility - Interact API' and the 'Action' is 'Edit Time Field Value'. The 'Inputs' tab is selected, showing a table with three rows: 'Submission Id' (Text, [Data.\_requestId]), 'Field Name' (Text, "current-time"), and 'Field Value' (Text, "15:00:00"). The 'Field Value' row is highlighted. The 'Outputs' and 'Conditions' tabs are also visible. On the right, the 'Group' section has 'Page' unchecked and 'Data Type' checked. Below this is a list of categories: Binaries, Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text, Times, and TimeSpans. The 'Text' category is expanded. At the bottom, 'Stage logging' is set to 'Errors only' and 'Don't log parameters on this stage' is unchecked. 'Warning threshold' is set to 'System Default' and 'Number of minutes' is set to 5 (0 to disable). 'OK' and 'Cancel' buttons are at the bottom right.

| Name          | Data Type | Value             |
|---------------|-----------|-------------------|
| Submission Id | Text      | [Data._requestId] |
| Field Name    | Text      | "current-time"    |
| Field Value   | Text      | "15:00:00"        |

## Change Field State to Optional

The Change Field State to Optional action allows you to change the state of a field within a submission, moving it to an Optional condition.

The inputs for the action are the Submission ID and the Field Name you are changing the state to optional, as illustrated below. This example shows a field with the name of 'security' being set to a 'optional' state.

The screenshot shows the 'Action Properties' dialog box for the 'Change Field State to Optional' action. The 'Name' field is set to 'Change Field State to Optional'. The 'Description' field is empty. The 'Business Object' is 'Utility - Interact API'. The 'Action' is 'Change Field State to Optional'. The 'Inputs' tab is selected, showing a table with two input fields: 'Submission Id' (Text, Value: [Data\_requestId]) and 'Field Name' (Text, Value: "security"). The 'Outputs' tab is empty. The 'Conditions' tab is empty. The 'Group' section has 'Page' unchecked and 'Data Type' checked. The 'View All Items' checkbox is unchecked. The 'Binaries' list is expanded, showing 'Collections', 'Dates', 'DateTimes', 'Flags', 'Images', 'Numbers', 'Passwords', 'Text', 'Times', and 'TimeSpans'. The 'Stage logging' section has 'Errors only' selected and 'Don't log parameters on this stage' unchecked. The 'Warning threshold' is 'System Default' and the 'Number of minutes' is 5 (0 to disable). The 'OK' and 'Cancel' buttons are at the bottom right.

| Name          | Data Type | Value            |
|---------------|-----------|------------------|
| Submission Id | Text      | [Data_requestId] |
| Field Name    | Text      | "security"       |

There are no outputs for the Change Field State to Optional.

## Change Field State to Mandatory

The Change Field State to Mandatory action allows you to change the state of a field within a submission, moving it to a Mandatory condition.

The inputs for the action are the Submission ID and the Field Name you are changing the state to mandatory, as illustrated below. This example shows a field with the name of 'security' being set to a 'mandatory' state.

The screenshot shows the 'Action Properties' dialog box for the 'Change Field State to Mandatory' action. The 'Name' field is set to 'Change Field State to Mandatory'. The 'Description' field is empty. The 'Business Object' is 'Utility - Interact API'. The 'Action' is 'Change Field State to Mandatory'. The 'Inputs' tab is selected, showing a table with two input fields: 'Submission Id' (Text, Value: [Data.\_requestId]) and 'Field Name' (Text, Value: "security"). The 'Outputs' and 'Conditions' tabs are empty. The 'Group' section has 'Page' unchecked and 'Data Type' checked. The 'View All Items' checkbox is also unchecked. A list of data types is shown on the right: Binaries, Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text, Times, and TimeSpans. The 'Stage logging' section has 'Errors only' selected and 'Don't log parameters on this stage' unchecked. The 'Warning threshold' is 'System Default' and the 'Number of minutes' is 5 (0 to disable). The 'OK' and 'Cancel' buttons are at the bottom right.

| Name          | Data Type | Value             |
|---------------|-----------|-------------------|
| Submission Id | Text      | [Data._requestId] |
| Field Name    | Text      | "security"        |

There are no outputs for the Change Field State to Mandatory.

## Change Field State to Read Only

The Change Field State to Read Only action allows you to change the state of a field within a submission, moving it to a Read Only condition.

The inputs for the action are the Submission ID and the Field Name you are changing the state to read only, as illustrated below. This example shows a field with the name of 'security' being set to a 'read only' state.

The screenshot shows the 'Action Properties' dialog box for the 'Change Field State to Read Only' action. The 'Name' field is set to 'Change Field State to Read Only'. The 'Business Object' is 'Utility - Interact API' and the 'Action' is 'Change Field State to Read Only'. The 'Inputs' tab is selected, showing a table with two input fields: 'Submission Id' (Text, Value: [Data\_requestId]) and 'Field Name' (Text, Value: "security"). The 'Field Name' field is highlighted. On the right, the 'Group' section has 'Data Type' checked. Below it, a list of data types is shown: Binaries, Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text, Times, and TimeSpans. At the bottom, 'Stage logging' is set to 'Errors only' and 'Warning threshold' is set to 'System Default' with a value of 5 minutes.

| Name          | Data Type | Value            |
|---------------|-----------|------------------|
| Submission Id | Text      | [Data_requestId] |
| Field Name    | Text      | "security"       |

There are no outputs for the Change Field State to Read Only.

## Change Field State to Hidden

The Change Field State to Hidden action allows you to change the state of a field within a submission, moving it to a Hidden condition.

The inputs for the action are the Submission ID and the Field Name you are changing the state to hidden, as illustrated below. This example shows a field with the name of 'security' being set to a 'hidden' state.

The screenshot shows the 'Action Properties' dialog box for the 'Change Field State to Hidden' action. The 'Name' field is 'Change Field State to Hidden' and the 'Description' field is empty. The 'Business Object' is 'Utility - Interact API' and the 'Action' is 'Change Field State to Hidden'. The 'Inputs' tab is selected, showing a table with two input fields: 'Submission Id' (Text, Value: [Data\_requestId]) and 'Field Name' (Text, Value: "security"). The 'Outputs' tab is empty. The 'Conditions' tab is empty. The 'Group' section has 'Page' unchecked and 'Data Type' checked. The 'View All Items' checkbox is unchecked. The 'Stage logging' section has 'Errors only' selected and 'Don't log parameters on this stage' unchecked. The 'Warning threshold' is 'System Default' and the 'Number of minutes' is 5 (0 to disable). The 'OK' and 'Cancel' buttons are at the bottom right.

| Name          | Data Type | Value            |
|---------------|-----------|------------------|
| Submission Id | Text      | [Data_requestId] |
| Field Name    | Text      | "security"       |

There are no outputs for the Change Field State to Hidden.

## Change Page State to Visible

The Change Page State to Visible action allows you to make a page become visible within a submission.

The inputs for the action are the Submission ID and the Page Name you are changing to become visible, as illustrated below. This example shows a page with the name of 'inputs' being set to a 'visible' state.

The screenshot shows the 'Action Properties' dialog box for the 'Change Page State to Visible' action. The dialog has a dark blue header with the title 'Action Properties'. Below the header, there are fields for 'Name' (set to 'Change Page State to Visible') and 'Description' (empty). The 'Business Object' is set to 'Utility - Interact API' and the 'Action' is set to 'Change Page State to Visible'. The 'Inputs' tab is selected, showing a table with two input fields: 'Submission Id' (Text, Value: [Data.\_requestId]) and 'Page Name' (Text, Value: "Inputs"). The 'Outputs' and 'Conditions' tabs are also visible. On the right side, there is a 'Group' section with checkboxes for 'Page' (unchecked) and 'Data Type' (checked), and a 'View All Items' checkbox (unchecked). Below this is a list of categories: Binaries, Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text, Times, and TimeSpans. At the bottom, there are settings for 'Stage logging' (Errors only) and 'Warning threshold' (System Default, 5 minutes). The 'OK' and 'Cancel' buttons are at the bottom right.

| Name          | Data Type | Value             |
|---------------|-----------|-------------------|
| Submission Id | Text      | [Data._requestId] |
| Page Name     | Text      | "Inputs"          |

There are no outputs for the Change Page State to Visible.

## Change Page State to Hidden

The Change Page State to Hidden action allows you to make a page hidden from view within a submission.

The inputs for the action are the Submission ID and the Page Name you are changing to become hidden , as illustrated below. This example shows a page with the name of 'inputs' being set to a 'hidden' state.

The screenshot shows the 'Action Properties' dialog box for the 'Change Page State to Hidden' action. The 'Name' field is set to 'Change Page State to Hidden'. The 'Description' field is empty. The 'Business Object' is 'Utility - Interact API' and the 'Action' is 'Change Page State to Hidden'. The 'Inputs' tab is selected, showing a table with two inputs: 'Submission Id' (Text, Value: [Data\_requestId]) and 'Page Name' (Text, Value: "Inputs"). The 'Outputs' and 'Conditions' tabs are empty. The 'Group' section has 'Page' unchecked and 'Data Type' checked. The 'View All Items' checkbox is also unchecked. A list of data types is shown on the right: Binaries, Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text, Times, and TimeSpans. At the bottom, 'Stage logging' is set to 'Errors only' and 'Warning threshold' is set to 'System Default' (5 minutes). The 'OK' and 'Cancel' buttons are at the bottom right.

| Name          | Data Type | Value            |
|---------------|-----------|------------------|
| Submission Id | Text      | [Data_requestId] |
| Page Name     | Text      | "Inputs"         |

There are no outputs for the Change Page State to Hidden.



## Move Submission to History Tab

The Move Submission to History Tab action allows you to move a submission from its current folder to the **Submission history** folder under **History**. This action also updates the status of the submission to Submitted. The Status filter in Interact can be used to display just the submissions with this status.

The input for the action is the Submission ID of the submission being moved, as illustrated below.

The screenshot shows the 'Action Properties' dialog box for the 'Move Submission to Submitted History Tab' action. The dialog has a dark blue header with the title 'Action Properties'. Below the header, there are fields for 'Name' (set to 'Move Submission to Submitted History Tab') and 'Description' (empty). The 'Business Object' is set to 'Utility - Interact API' and the 'Action' is set to 'Move Submission to Submitted History Tab'. The 'Inputs' tab is selected, showing a table with one input: 'Submission Id' of type 'Text' with value '[Data.\_requestId]'. The 'Outputs' and 'Conditions' tabs are empty. On the right, the 'Group' section has checkboxes for 'Page' (unchecked), 'Data Type' (checked), and 'View All Items' (unchecked). Below this is a tree view of data types: Binaries, Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text, Times, and TimeSpans. At the bottom, there are settings for 'Stage logging' (set to 'Errors only') and 'Warning threshold' (set to 'System Default'). There are also checkboxes for 'Don't log parameters on this stage' and 'Number of minutes' (set to 5). The dialog ends with 'OK' and 'Cancel' buttons.

| Name          | Data Type | Value             |
|---------------|-----------|-------------------|
| Submission Id | Text      | [Data._requestId] |

There are no outputs for the Move Submission to History Tab.

## Move Submission to Approved Tab

The Move Submission to Approved Tab action allows you to move a submission from its current folder to the **Submission history** folder under **History**. This action also updates the status of the submission to Approved. The Status filter in Interact can be used to display just the submissions with this status.

The input for the action is the Submission ID of the submission being moved, as illustrated below.


The screenshot shows the 'Action Properties' dialog box for the 'Move Submission to Approved Tab' action. The dialog is titled 'Action Properties' and has a dark blue header. Below the header, there are fields for 'Name' (set to 'Move Submission to Approved Tab') and 'Description' (empty). The 'Business Object' is set to 'Utility - Interact API' and the 'Action' is set to 'Move Submission to Approved Tab'. The 'Inputs' tab is selected, showing a table with one input: 'Submission Id' of type 'Text' with value '[Data.\_requestId]'. The 'Outputs' and 'Conditions' tabs are empty. On the right, the 'Group' section has checkboxes for 'Page' (unchecked), 'Data Type' (checked), and 'View All Items' (unchecked). Below this is a tree view of data types: Binaries, Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text, Times, and TimeSpans. At the bottom, there are settings for 'Stage logging' (set to 'Errors only') and 'Warning threshold' (set to 'System Default'). There are also checkboxes for 'Don't log parameters on this stage' and 'Number of minutes' (set to 5). The dialog has 'OK' and 'Cancel' buttons at the bottom right.

| Name          | Data Type | Value             |
|---------------|-----------|-------------------|
| Submission Id | Text      | [Data._requestId] |

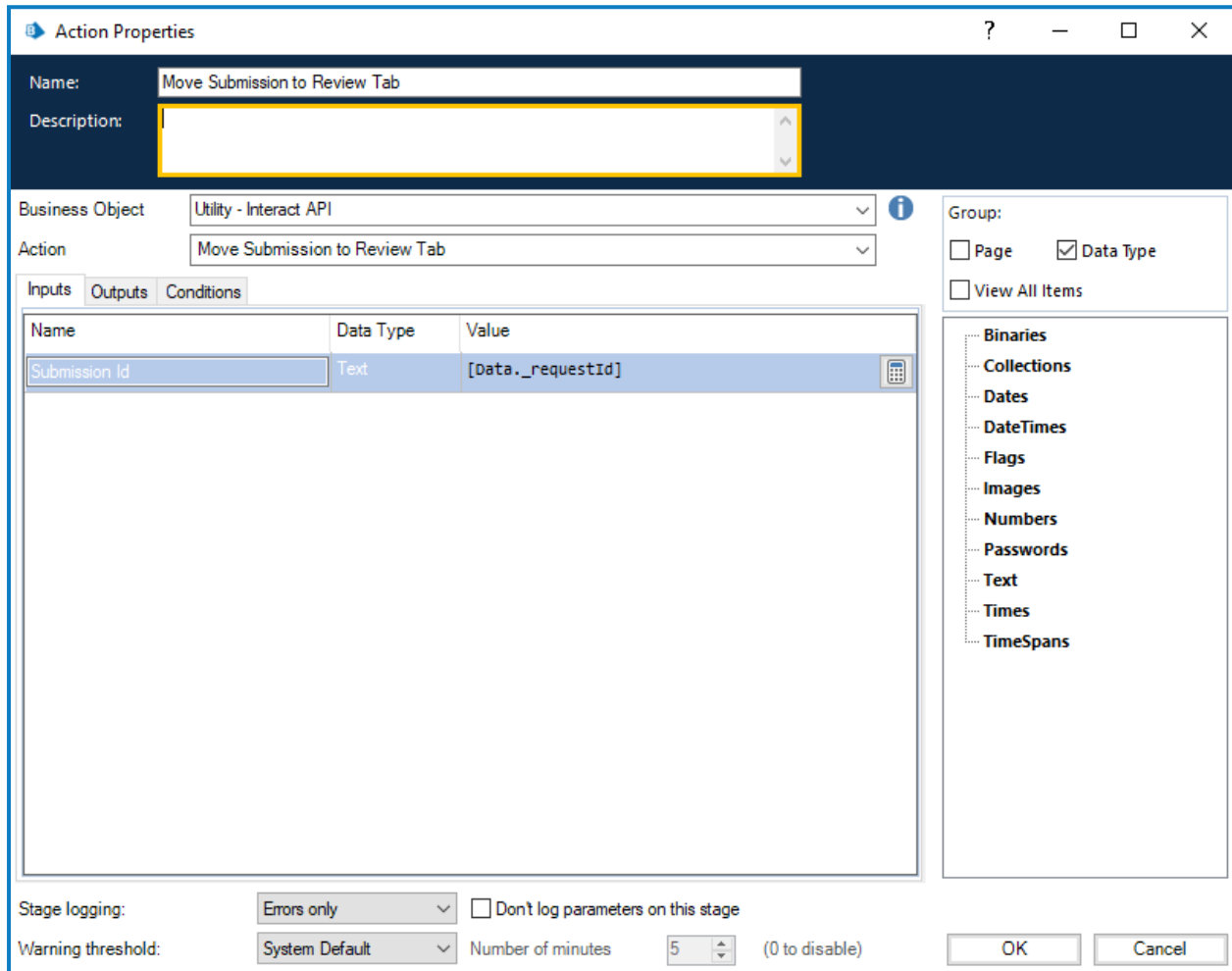
There are no outputs for the Move Submission to Approved.

## Move Submission to Review Tab

The Move Submission to Review Tab action allows you to move the submission to the **Awaiting approval** folder under **History** for the submitter, and into the **Approvals** folder under **My Work** for the approver.

 This requires a form to be configured with either **Single Approver** or **Any Approver**. If not, the form will move to the **Awaiting approval** folder but will not move to the **My Work** tab of an approver.

The input for the action is the Submission ID of the submission being moved, as illustrated below.



The screenshot shows the 'Action Properties' dialog box for the 'Move Submission to Review Tab' action. The dialog has a dark blue header with the title 'Action Properties'. Below the header, there are fields for 'Name' (set to 'Move Submission to Review Tab') and 'Description' (empty). The 'Business Object' is set to 'Utility - Interact API'. The 'Action' is set to 'Move Submission to Review Tab'. The 'Inputs' tab is selected, showing a table with one input: 'Submission Id' of type 'Text' with value '[Data.\_requestId]'. The 'Outputs' and 'Conditions' tabs are also visible. On the right, there are checkboxes for 'Page', 'Data Type' (checked), and 'View All Items'. Below these is a list of data types: Binaries, Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text, Times, and TimeSpans. At the bottom, there are settings for 'Stage logging' (set to 'Errors only') and 'Warning threshold' (set to 'System Default'). A 'Number of minutes' field is set to '5' with a note '(0 to disable)'. 'OK' and 'Cancel' buttons are at the bottom right.

| Name          | Data Type | Value             |
|---------------|-----------|-------------------|
| Submission Id | Text      | [Data._requestId] |

There are no outputs for the Move Submission to Review Tab.

## Move Submission to Declined Tab

The Move Submission to Declined Tab action allows you to move a submission from its current folder to the **Submission history** folder under **History**. This action also updates the status of the submission to Declined. The Status filter in Interact can be used to display just the submissions with this status.

The input for the action is the Submission ID of the submission being moved, as illustrated below.

The screenshot shows the 'Action Properties' dialog box for the 'Move Submission to Declined Tab' action. The dialog has a title bar with a question mark, minimize, maximize, and close buttons. The main area is divided into several sections:

- Name:** 'Move Submission to Declined Tab' (text field)
- Description:** (empty text field)
- Business Object:** 'Utility - Interact API' (dropdown menu)
- Action:** 'Move Submission to Declined Tab' (dropdown menu)
- Inputs:** A table with columns 'Name', 'Data Type', and 'Value'. It contains one row: 'Submission Id' (Name), 'Text' (Data Type), and '[Data.\_requestId]' (Value). There is a calculator icon next to the value field.
- Outputs:** (empty table)
- Conditions:** (empty table)
- Group:** A section with checkboxes for 'Page', 'Data Type' (checked), and 'View All Items'.
- Binaries:** A list of data types: Binaries, Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text, Times, and TimeSpans.
- Stage logging:** A dropdown menu set to 'Errors only' and a checkbox for 'Don't log parameters on this stage'.
- Warning threshold:** A dropdown menu set to 'System Default', a text field for 'Number of minutes' with the value '5', and a note '(0 to disable)'.
- Buttons:** 'OK' and 'Cancel' buttons.

There are no outputs for the Move Submission to Declined Tab.

## Move Submission to Draft Tab

The Move Submission to Draft Tab action allows you to move a submission from its current folder to the **Draft** folder under **My Work**.

The input for the action is the Submission ID of the submission being moved, as illustrated below.

The screenshot shows the 'Action Properties' dialog box for the 'Move Submission to Draft Tab' action. The dialog has a title bar with a question mark, minimize, maximize, and close buttons. The main area is divided into several sections:

- Name:** 'Move Submission to Draft Tab' (text field)
- Description:** (empty text field)
- Business Object:** 'Utility - Interact API' (dropdown menu)
- Action:** 'Move Submission to Draft Tab' (dropdown menu)
- Inputs/Outputs/Conditions:** A tabbed interface with 'Inputs' selected. It contains a table with the following data:

| Name          | Data Type | Value             |
|---------------|-----------|-------------------|
| Submission Id | Text      | [Data._requestId] |
- Group:** A section with checkboxes for 'Page' (unchecked), 'Data Type' (checked), and 'View All Items' (unchecked).
- Binaries/Collections/Dates/DateTimes/Flags/Images/Numbers/Passwords/Text/Times/TimeSpans:** A list of categories on the right side of the dialog.
- Stage logging:** A section with a dropdown menu set to 'Errors only' and a checkbox for 'Don't log parameters on this stage' (unchecked).
- Warning threshold:** A section with a dropdown menu set to 'System Default' and a numeric input field set to '5' (0 to disable).
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

There are no outputs for the Move Submission to Draft Tab.

## Move Submission to Inbox Tab

The Move Submission to Inbox Tab action allows you to move a submission from its current folder to the **Inbox** folder under **My Work**.

The input for the action is the Submission ID of the submission being moved, as illustrated below.

**Action Properties**

Name: Move Submission to Inbox Tab

Description:

Business Object: Utility - Interact API

Action: Move Submission to Inbox Tab

Inputs Outputs Conditions

| Name          | Data Type | Value             |
|---------------|-----------|-------------------|
| Submission Id | Text      | [Data._requestId] |

Group: ☐ Page ☒ Data Type ☐ View All Items

- Binaries
- Collections
- Dates
- DateTimes
- Flags
- Images
- Numbers
- Passwords
- Text
- Times
- TimeSpans

Stage logging: Errors only ☐ Don't log parameters on this stage

Warning threshold: System Default Number of minutes 5 (0 to disable)

OK Cancel

There are no outputs for the Move Submission to Inbox Tab.

## Move Submission to Archived Tab

The Move Submission to Archived Tab action allows you to move a submission from its current folder to the Archived folder under **History**.

The input for the action is the Submission ID of the submission being moved, as illustrated below.

The screenshot shows the 'Action Properties' dialog box for the 'Move Submission to Archived Tab' action. The dialog has a dark blue header with the title 'Action Properties'. Below the header, there are fields for 'Name' (set to 'Move Submission to Archived Tab') and 'Description' (empty). The 'Business Object' is set to 'Utility - Interact API' and the 'Action' is set to 'Move Submission to Archived Tab'. The 'Inputs' tab is selected, showing a table with one input: 'Submission Id' of type 'Text' with value '[Data.\_requestId]'. The 'Outputs' and 'Conditions' tabs are empty. On the right, the 'Group' section has checkboxes for 'Page' (unchecked) and 'Data Type' (checked), and a 'View All Items' checkbox (unchecked). Below this is a list of categories: Binaries, Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text, Times, and TimeSpans. At the bottom, there are settings for 'Stage logging' (set to 'Errors only') and 'Warning threshold' (set to 'System Default'). A checkbox 'Don't log parameters on this stage' is unchecked. The 'Number of minutes' is set to 5, with a note '(0 to disable)'. 'OK' and 'Cancel' buttons are at the bottom right.

| Name          | Data Type | Value             |
|---------------|-----------|-------------------|
| Submission Id | Text      | [Data._requestId] |

There are no outputs for the Move Submission to Archived Tab.

## Human/Digital Worker collaboration

To best show the way that human/Digital Worker collaboration operates let us consider a simple scenario. This scenario is a simple change of address request from a user.

The steps on a high-level are detailed below:

- The user will initiate the request through Interact;
- The Digital Worker will review the request and move the item to the user's inbox;
- The user will then review and update the request, providing their response to a security question, and submit the request;



At this point the user can leave the Form open in the inbox, see the Form being updated dynamically by the Digital Worker and provide additional information when requested.

- The Digital Worker processes the information provided and then requests additional information from the user, the new address;
- The user provides the information and the Digital Worker updates the record and then completes the task.

The Form created in Interact is displayed below. The Form initially captures the First and Last Names, as well as the current address. The other fields, as you can see in the testing area, are hidden. These are the security question and the new address fields.

The screenshot displays the Interact form editor for 'HR - Change of Address'. The top navigation bar includes icons for Search fields, Create page, Collapse all, Expand all, and statistics (1 Pages, 6 Fields, 0 Rules). A 'Save' button is also present.

**Design View (Left):** A vertical timeline of form fields with their status:

- Instructions:** Read only
- First Name:** Enter your First Name, Mandatory
- Last Name:** Enter your Last Name, Mandatory
- Current Address:** Enter your Current Address, Mandatory
- Security Question:** Enter your Mother's maiden name, Hidden
- New Address:** Enter your New Address, Hidden
- Submitter:** (Dashed box)

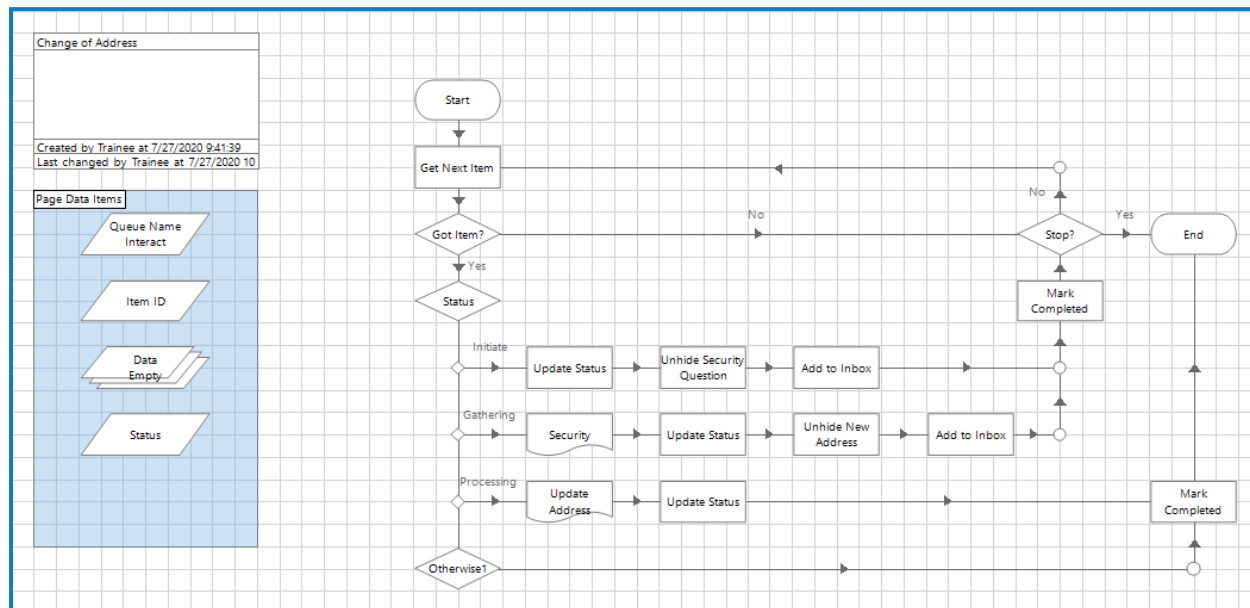
**Preview View (Right):** Shows the rendered form with the following fields:

- Initial Details:** Initiating request for Change of Address
- Instructions:** This form is an interactive form managing the Change of Address process, part of the Human Resources processes. Once submitted, this form will appear in the user's inbox where they can interact directly with the digital worker to complete the process.
- First Name \*:** Enter your First Name
- Last Name \*:** Enter your Last Name
- Current Address \*:** Enter your Current Address

The automation associated with the process is illustrated below. This is a process that loops through and allows to the user to work dynamically with the Digital Worker.




The process operates dependent on the value of the 'Status' flag. This is a read-only radio group across the top of the Form. As the process progresses through the different stages this flag is updated and the Status changes in the Form.



## Interact Web API Service object restrictions

The following restrictions apply to the Interact Web API Service object.

The table below lists the restrictions.

| Function          | Element Type     | Applicable Restrictions  |
|-------------------|------------------|--|
| Get Submission    | Number           | Number elements are delivered as 'Text' format when using the Get Submission Web API Service. Cast to Number once inside Blue Prism  |
| Get Submission    | Date             | Retrieving a Date element using the Get Submission Web API Service, returns the Date in 'DateTime' format. It is recommended that DateTime values are converted to Text once inside Blue Prism. The use of Text is recommended to support the different Date formats within Interact.  |
| IADA              | Number           | Number elements are delivered as 'Text' format into a Blue Prism queue. Cast to Number once inside Blue Prism  |
| Create Submission | Number           | When creating a new submission using the Web API Service, the submission will fail if a Number element is left blank. Submit Number element fields with a number set   |
| Create Submission | Radio Group      | When creating a new submission using the Web API Service, the submission will fail if a Radio Group element is left blank. Submit Radio Group element fields with a value set  |
| Create Submission | Upload           | When creating a new submission using the Web API Service, you cannot send anything back when using the Upload element. This is the expected functionality.   |
| Edit Submission   | Upload           | When editing a submission using the Web API Service you cannot send anything back when using the Upload element. This is the expected functionality.   |
| Interact Form     | Priority and SLA | Though the Priority and SLA can be set in the Interact Form creator, they have no significance in this release as IADA 'Get Next Prioritized Item' (GNPI) function is not available in an on-premise configuration. Expected functionality for this release.   |
| Interact Forms    | Rules            | Rules are not enforced when updating Interact Forms from Blue Prism. For example, a rule that reveals a hidden field when a flag is set, will not be initiated if the flag is set correctly when updated. If the field is not read-only, a user can click in the field and then click elsewhere in the Form to see the Rule applied.<br><br> You can use Update Config to replace some rules. |
| Interact Forms    | Verification     | Regex and other verification such as character string lengths are not enforced when updating Interact Forms from Blue Prism. If the field is not read-only a user can click in the field and then click elsewhere in the Form to see the Rule applied. No workaround available.  |