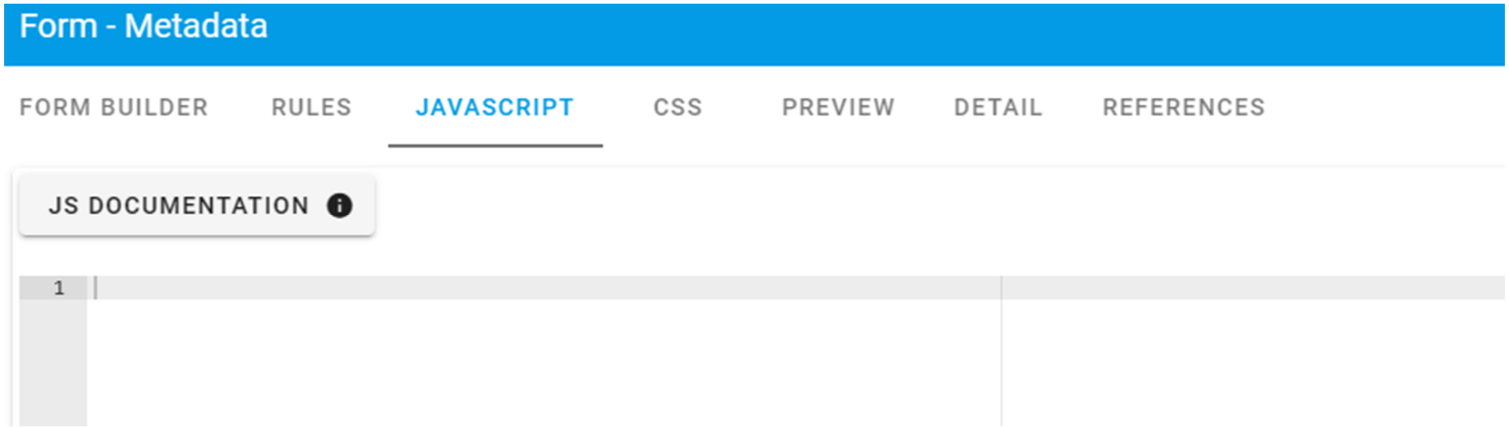
Form Scripting Reference Manual

©2024 Integrify, Inc.

Integrify Forms support custom scripting (ECMA Script/JavaScript) which provides a very flexible solution in creating a rich user experience. Each question type, as well as the various layout elements (Sections, Containers, etc.) have specific properties and methods available to use within the Form's script. This Reference Manual provides the information for the supported properties and methods of each element and Question Type/Field Type.

FORM SCRIPT

To add custom script to a form, from the Form Builder click on the JAVASCRIPT tab. This will display the Form's script editor.



Standard ECMA Script (JavaScript) is supported. You should have a basic understanding of ECMA Script/JavaScript before attempting to add custom script to an Integrify Form.

The custom script is executed when the Form is loaded in Runtime mode, Preview mode, or View mode (formState). You can create global variables within the script as well as bind to events. Note that there is a limited number of event types available for the Integrify form and Questions compared to the number available when using script on a standard HTML page.

SPECIAL NOTE ON FORM RULES: As general rule, and best practice, you should not mix Form Rules and JavaScript on the same form. If you have need for Script on the form, you should perform all functionality from the Form Script. Note that if you bind an event to a Question's onChange event, that binding will disable that Question triggering the evaluation of the Form Rules.

The intForm Object

The Integrify form object is named ***intForm***. All interactions within the script should be done using this intForm object. You should never (with 1 or 2 exceptions) address HTML objects directly, as they are created dynamically by the Integrify framework, and the HTML structures and references may change from release to release.

The formState property contains one of three values: preview, runtime, completed. Sometimes you want the Form's script to perform differently, depending on what mode (formState) it's in. You can incorporate the formState value in your script to accomplish this.

Example:

```
if(intForm.formState!==’completed’) {
    intForm.getElementByClientID(‘stxtRequestNumber’).Answer = intForm.recipientTask.InstanceID;
}
```

Form Events

The intForm object has an onSubmit event and an onSaveDraft event. You can assign either of these to perform custom functions when the SUBMIT button or SAVE DRAFT button is clicked by the user.

intForm.events.onSubmit = () => submitTheForm() will bind the custom function submitTheForm to the Form's onSubmit event. The onSubmit event occurs when the user clicks the SUBMIT button. It's important to remember that this will *replace* the built-in form submit functionality, so the custom script must also handle actually submitting the form. This is done by calling the Form's submit() method.

Example:

```
intForm.events.onSubmit() = () => validateForm()
function validateForm() {
    if(intForm.getElementByClientID(‘stxtEmployeeNumber’) {
        intForm.submit();
    }else{
        alert(‘Employee Number is a Required Field!’);
        //the Form will not be submitted
    }
}
```

intForm.events.onSaveDraft = () => saveFormDraft() will bind the custom function saveFormDraft to the Form's saveDraft event. The saveDraft event occurs when the user click SAVE DRAFT button. The custom function saveFormDraft must contain a call to the Form's saveDraft() method, to actually save the Form Draft.

Clicking the SUBMIT or SAVE DRAFT buttons disables both buttons prior to calling the onSubmit or onSaveDraft event, so you must re-enable the buttons if your code's logic does not end in execution of intForm.submit() or intForm.saveDraft() (ex. custom validation fails). If you do not enable the buttons, the user will not be able to submit or save the draft after making the necessary changes to their form data.

<u>Submit and Save Draft Buttons</u>

You may want to disable the Submit and/or Save Draft buttons from within your form script. If you have custom onSubmit or onSaveDraft function, it's best practice to disable the buttons at the very start of the function. This will prevent the user from clicking the button again while the script is executing. Each button has two methods available to do this:

```
intForm.submitButton.disable()
intForm.submitButton.enable()
intForm.saveDraftButton.disable()
intForm.saveDraftButton.enable()
```

Methods for Accessing the Form's Questions/Fields:

intForm.getElementByClientID(*c*), where *c* is the clientID of the Question. This method gives you access to the individual Questions/Fields on the form. Each Question Type has its own set of properties and methods, however many are common between the different types. See the detailed tables below, for a complete list of properties and methods available for each Question type.

intForm.getSectionByClientID(c), where *c* is the clientID of the Section.

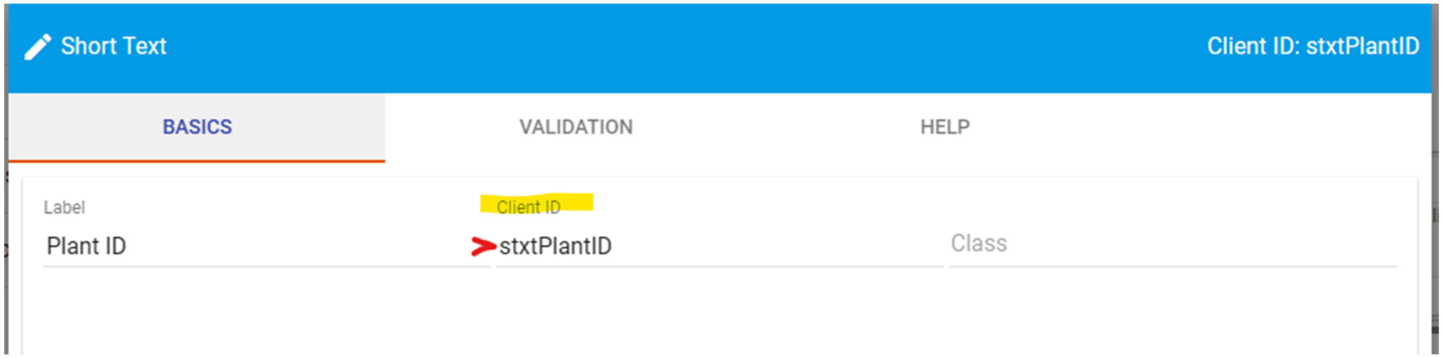
The recipientTask Object

intForm.recipientTask is present if the Form is opened from a Form Task. The object contains information specific to the current user's *Recipient Task*, but also contains basic information about the Request (Instance) and the Form Task itself. All values in the recipientTask object are considered READ ONLY – they should never be changed by the Form Script.

- TaskName
- InstanceSID
- InstanceID
- Instance
  - CreatedDate
  - LastMilestone

Standardizing Client IDs

When you are creating new Questions on a Form, it is recommended and a best practice to always change the generated Client ID with a more meaningful Client ID. This will allow you to more easily identify the Question references within the Script. It is also best practice to use standard prefixes for the Client IDs that indicate the Question Type. For example, “stxt” is used for Short Text Questions. In the script, you are then able to tell Question Type without having to open the configuration of the Question itself.



Client ID: stxtPlantID

Label: Plant ID

Client ID: stxtPlantID

Class

Property and Method Detail of the intForm Field Types

Field Types: Short Text / Long Text / Number / Link / Email

Property/Method	Description/Use	Read/Write
Answer	The value of the field (e.g. the answer to the question). For these field types the value is always a simple text value. Note that the empty value for these fields types is null when the field has never had a value. If a previous answer value has been cleared/removed, then the value is an empty string, except for Number with the Currency option used where it is 0 (zero).	RW
class	The CSS style class(es) applied to the field on the form (See section on Defining and Using Classes)	RW
ClientID	The ID of the html element on the form. This is the attribute that is used to reference any element on the form. The ClientID for a field can be changed on the configuration screen. It is recommended that the default long numeric ID is changed to something more meaningful, so it is easier to identify when scripting. The best time to do this is when the field is initially created. (See also: id)	R
disabled	Set to either true or false. If true, the field is disabled and cannot be selected or changed.	RW
events	Contains any script bound to the onFocus, onBlur, and onChange events of the field on the form. These events are emitted similar to the corresponding standard HTML events.	RW
flex	The width % applied to the field on the form. This value is in relation to the container that contains the field on the form. Not available for the calendar field type.	RW
id	The system-assigned field ID linked to the field (see also: ClientID)	R
isdirty	Indicates whether the value in the field has been changed since the form was loaded. Returns true if the value has been changed, false if it has not.	R
Label	The label text shown on the form for the field.	RW
originalAnswer	The value of the field when the form initially loads.	R
QuestionType	ShortText, LongText, Number, Hyperlink, EmailAddress, Calendar	R
readonly	Set to either true or false. If true, the field can be selected but the value cannot be changed.	RW
setAnswer()	Sets the date for a Calendar type question. Date must be in YYYY-MM-DD format.	
show	Indicates whether the question field should be shown on the form (true/false)	RW
todaysDate	Available only for Calendar fields. Contains the current *server* date in the format MM/DD/YYYY. It can be useful for obtaining a date that is not dependent on a user's local system date.	R

validation	The validation values and messages used when a form submitted. They correspond to the Validation tab on the question's configuration screen. regExMessage – The message displayed when the regEx validation fails. regEx – The regEx expression to use validate the field value. Max – A maximum length value that is used to compare the field value against. maxMessage – The message displayed when the Max validation fails. Min – A minimum length value that is used to compare the field value against. minMessage – The message displayed when the Min validation fails. required (true/false) – Indicates whether the field is required. Setting this to true will also trigger the display of the required field indicator (*). requiredMessage – The message displayed when the required validation fails.	RW

Field Types: Calendar

Property/Method	Description/Use	Read/Write
Answer	The value of the field (e.g. the answer to the question). For the Calendar type the value can be represented via "moment object" or a date object. Answer should be considered Read-Only and only updated by using the setAnswer method.	R
answerUI	The date representation displayed in the Calendar question (i.e. 2022-09-14)	R
class	The CSS style class(es) applied to the field on the form (See section on Defining and Using Classes)	RW
ClientID	The ID of the element on the form. This is the attribute that is used to reference any element on the form. The ClientID for a field can be changed on the configuration screen. It is recommended that the default long numeric ID is changed to something more meaningful, so it is easier to identify when scripting. The best time to do this is when the field is initially created. A given ClientID should be used only one time on the form (it should be unique within a given form). (See also: id)	R
disabled	Set to either true or false. If true, the field is disabled and cannot be selected or changed.	RW
events	Contains any script bound to the onFocus, onBlur, and onChange events of the field on the form. These events are emitted similar to the corresponding standard HTML events.	RW
id	The system-assigned field ID linked to the field (see also: ClientID)	R
isdirty	Indicates whether the value in the field has been changed since the form was loaded. Returns true if the value has been changed, false if it has not.	R
Label	The label text shown on the form for the field.	RW
originalAnswer	The value of the field when the form initially loads.	R

QuestionType	ShortText, LongText, Number, Hyperlink, EmailAddress, Calendar	R
readonly	Set to either true or false. If true, the field can be selected but the value cannot be changed.	RW
setAnswer()	Method that sets the date for a Calendar type question. Date should be in YYYY-MM-DD format or can be a Date object.	
show	Indicates whether the question field should be shown on the form (true/false)	RW
todaysDate	Contains the current *server* date in the format MM/DD/YYYY. It can be useful for obtaining a date that is not dependent on a user's local system date.	R
validation	The validation values and messages used when a form submitted. They correspond to the Validation tab on the question's configuration screen. futureDatesOnly -- true/false value regExMessage – The message displayed when the regEx validation fails. regEx – The regEx expression to use validate the field value. maxDate – A maximum date value that is used to compare the field value against. maxMessage – The message displayed when the naxDate validation fails. minDate – A minimum value that is used to compare the field value against. minMessage – The message displayed when the Min validation fails. required (true/false) – Indicates whether the field is required. Setting this to true will also trigger the display of the required field indicator (*). requiredMessage – The message displayed when the required validation fails.	RW

Question Type: Select List

Answer	The *value* of the item selected in the list. Comma separated values if multiple answers are selected. If the field has no value it can be either null or an empty string. Note that this is a comma delimited list, so Answer values cannot contain commas.	RW
Choices	Array of objects containing the Label/Value pairs contained in the drop-down list. Note that when updating this list via script you need to only populate the Label and Value properties, the other properties are only used in Form Design.	RW
class	The style class(es) applied to the field on the form (See section on Defining and Using Classes)	RW
ClientID	The ID of the html element on the form	R
disabled	Set to either true or false. If true, the field is disabled and cannot be selected or changed.	RW
events	Contains any script bound to the onFocus, onBlur, and onChange events of the field on the form. These events are emitted similar to the corresponding standard HTML events.	RW
flex	The width % applied to the question on the form. This value is in relation to the container that contains the question field on the form.	RW
id	The system-assigned field ID linked to the field (see also: ClientID)	R

isdirty	Indicates whether the value in the field has been changed since the form was loaded. Returns true if the value has been changed, false if it has not.	R
Label	The label shown on the form for the question	RW
multiple	Indicates whether multiple selections are allowed (true/false)	RW
multiChoiceAnswer	Array of value selected when multiple=true	
originalAnswer	The value of the field when the form initially loads.	R
QuestionType	DbSelectList	R
readonly	Set to either true or false. If true, the field can be selected but the value cannot be changed.	RW
show	Indicates whether the question field should be shown on the form (true/false)	RW
validation	The validation values and messages used when a form submitted. They correspond to the Validation tab on the question's configuration screen. required (true/false) – Indicates whether the field is required. Setting this to true will also trigger the display of the required field indicator (*). requiredMessage – The message displayed when the required validation fails.	RW

Question Type: Checkboxes

Answer	Comma-space separated list of the values of the selected checkboxes. If the field has no value it can be either null or an empty string. If you are setting the value of a checkbox within a script, then you must set both the Answer value and set the Selected attribute of the corresponding Choices. Note that this is a comma delimited list, so Answer values cannot contain commas	RW
Choices	An array of objects corresponding to the checkboxes. Each object contains the following: Label – the label of the individual checkbox Selected (true/false) – Indicates whether the checkbox is displayed as checked (true) or not checked (false). If the checkbox has never been checked then this attribute is undefined. Value – the value of the individual checkbox that is stored in Answer if the checkbox is selected Note that when updating this list via script you need to only populate the Label and Value properties, the other properties are only used in Form Design.	RW
class	The style class(es) applied to the field on the form (See section on Defining and Using Classes)	RW
ClientID	The ID of the html element on the form	R
columnOrRow	Indicates horizontal (row) or vertical (column) checkbox orientation	RW
disabled	Set to either true or false. If true, the field is disabled and cannot be selected or changed.	RW
events	Contains any script bound to the onFocus, onBlur, and onChange events of the field on the form. These events are emitted similar to the corresponding standard HTML events.	RW

flex	The width % applied to the question on the form. This value is in relation to the container that contains the question field on the form.	RW
id	The system-assigned field ID linked to the field (see also: ClientID)	R
isDirty	Indicates whether the value in the field has been changed since the form was loaded. Returns true if the value has been changed, false if it has not.	R
Label	The label shown on the form for the question	RW
originalAnswer	The value of the field when the form initially loads.	R
QuestionType	DbCheckbox	R
readonly	Set to either true or false. If true, the field can be selected but the value cannot be changed.	RW
show	Indicates whether the question field should be shown on the form (true/false)	RW
validation	The validation values and messages used when a form submitted. They correspond to the Validation tab on the question's configuration screen. required (true/false) – Indicates whether the field is required. Setting this to true will also trigger the display of the required field indicator (*). requiredMessage – The message displayed when the required validation fails.	RW

Question Type: Radio Buttons

Answer	The value of the selected radio button. If no radio button is selected the value will be null.	RW
Choices	An array of objects corresponding to the radio buttons. Each object contains the following: Label – the label of the individual checkbox Value – the value of the individual checkbox that is stored in Answer if the checkbox is selected Note that when updating this list via script you need to only populate the Label and Value properties, the other properties are only used in Form Design.	RW
class	The style class(es) applied to the question on the form (See section on Defining and Using Classes)	RW
ClientID	The ID of the html element on the form	R
disabled	Set to either true or false. If true, the field is disabled and cannot be selected or changed.	RW
events	Contains any script bound to the onFocus, onBlur, and onChange events of the field on the form. These events are emitted similar to the corresponding standard HTML events.	RW
flex	The width % applied to the question on the form. This value is in relation to the container that contains the question field on the form.	RW
id	The system-assigned field ID linked to the field (see also: ClientID)	R
isDirty	Indicates whether the value in the field has been changed since the form was loaded. Returns true if the value has been changed, false if it has not.	R
Label	The label shown on the form for the question	RW

originalAnswer	The value of the field when the form initially loads.	R
QuestionType	DbRadioButton	R
readonly	Set to either true or false. If true, the field can be selected but the value cannot be changed.	RW
show	Indicates whether the question field should be shown on the form (true/false)	RW
validation	The validation values and messages used when a form submitted. They correspond to the Validation tab on the question's configuration screen. required (true/false) – Indicates whether the field is required. Setting this to true will also trigger the display of the required field indicator (*). requiredMessage – The message displayed when the required validation fails.	RW

Question Type: File Attachment

Answer	An array of objects, one for each file attachment, with the following elements: file_key – a unique value assigned to the file attachment lastModifiedDate – the last modified date and time of the file name – the file name size – the size of the file text – the file name text displayed on the form type – the file type designator (e.g. image/jpeg)	RW
class	The style class(es) applied to the question on the form (See section on Defining and Using Classes)	RW
ClientID	The ID of the html element on the form (see also id)	R
disabled	Set to either true or false. If true, the field is disabled and cannot be selected or changed.	RW
events	Contains any script bound to the onFocus, onBlur, and onChange events of the field on the form. These events are emitted similar to the corresponding standard HTML events.	RW
flex	The width % applied to the question on the form. This value is in relation to the container that contains the question field on the form.	RW
id	The system-assigned field ID linked to the field (see also: ClientID)	R
isDirty	Indicates whether the value in the field has been changed since the form was loaded. Returns true if the value has been changed, false if it has not.	R
Label	The label shown on the form for the question	RW
multiple	If true, selection of multiple file attachments is allowed, else false	RW
QuestionType	FileAttachment or MultiFileAttachment	R
readonly	Set to either true or false. If true, the field can be selected/file opened but the value cannot be changed.	RW
show	Indicates whether the question field should be shown on the form (true/false)	RW

validation	The validation values and messages used when a form submitted. They correspond to the Validation tab on the question's configuration screen. requiredMessage – The message displayed when the Required validation fails.	RW

Question Type: Contact Search

Answer	An array of objects, one for each contact selected. Each object contains: Email – contacts email address ID – contact ID Name – the name from the contact profile SID – contact SID Title – the title of the user UserName – the Integrify user name text – the contact name displayed on the form	RW
class	The style class(es) applied to the question on the form (See section on Defining and Using Classes)	RW
ClientID	The ID of the html element on the form	R
disabled	Indicates whether the question field on the form is set to disabled (true/false)	RW
events	Contains any script for the onFocus, onBlur, and onChange events bound to the question field on the form.	RW
flex	The width % applied to the question on the form. This value is in relation to the container that contains the question field on the form.	RW
id	The system-assigned field ID linked to the field (see also: ClientID)	R
isDirty	Indicates whether the value in the field has been changed since the form was loaded. Returns true if the value has been changed, false if it has not.	R
Label	The label shown on the form for the question	RW
originalAnswer	The value of the field when the form initially loads.	R
QuestionType	ContactSearch, MultiContactSearch	RW
searchFilter	An array containing the SID (0) and name (1) of the contact search group filter	RW
show	Indicates whether the question field should be shown on the form (true/false)	RW
validation	The validation values and messages used when a form submitted. They correspond to the Validation tab on the question's configuration screen. requiredMessage – The message displayed when the Required validation fails.	RW

Question Type: Search Box

Answer	The value of the selected item (see also setAnswer, clearAnswer)	R
Choices	An array of Label/Value pairs that are available in the Search Box list (for non-db-derived results)	RW
class	The style class(es) applied to the question on the form (See section on Defining and Using Classes)	RW
clearAnswer	Method to clear the label/value settings and update the field display (see also setAnswer)	
ClientID	The ID of the html element on the form	R
dbSettings	An object containing the data source, query and mapping information. DbSourceLabelColumn—The column name from the db source to use as the Label DbSourceValueColumn—The column name from the db source to use as the Value hideSearchResultFields—List of column names returned by the db source to exclude from the displayed list. SID—The SID of the db source used Mappings—An array containing ClientID and ColumnName pairs for mapping other columns in the query results to fields on the form	RW
disabled	Set to either true or false. If true, the field is disabled and cannot be selected or changed.	RW
events	Contains any script for the onFocus, onBlur, and onChange events bound to the question field on the form.	RW
flex	The width % applied to the question on the form. This value is in relation to the container that contains the question field on the form.	RW
friendlyAnswer	The LABEL that is displayed in the field when a selection is made.	R
id	The system-assigned field ID linked to the field (see also: ClientID)	R
isDirty	Indicates whether the value in the field has been changed since the form was loaded. Returns true if the value has been changed, false if it has not.	R
Label	The label shown on the form for the question	RW
originalAnswer	The value of the field when the form initially loads.	R
QuestionType	SearchBox	R
readonly	Set to either true or false. If true, the field can be selected but the value cannot be changed.	RW
setAnswer	Method to set the values of the Answer and Label and related attributes. You must supply a json object containing the keys Label and Value. (e.g. setAnswer({“Value”: 712, Label: “HR”})	
show	Indicates whether the question field should be shown on the form (true/false)	RW
validation	The validation values and messages used when a form submitted. They correspond to the Validation tab on the question’s configuration screen. requiredMessage – The message displayed when the Required validation fails.	RW

Question Type: Rich Text

Answer	HTML tagged content that has been entered into the field. This can contain standard HTML elements, such as tables, links, images, etc.	RW
class	The style class(es) applied to the question on the form (See section on Defining and Using Classes)	RW
ClientID	The ID of the html element on the form	R
disabled	Indicates whether the question field on the form is set to disabled (true/false)	RW
events	Contains any script for the onBlur, and onChange events bound to the question field on the form.	RW
flex	The width % applied to the question on the form. This value is in relation to the container that contains the question field on the form.	RW
id	The system-assigned field ID linked to the field (see also: ClientID)	R
isDirty	Indicates whether the value in the field has been changed since the form was loaded. Returns true if the value has been changed, false if it has not.	R
Label	The label shown on the form for the question	RW
loaded	Indicates the Question has initialized and any prefill data has been set. true/false	R
QuestionType	RichText	R
show	Indicates whether the question field should be shown on the form (true/false)	RW
validation	The validation values and messages used when a form submitted. They correspond to the Validation tab on the question's configuration screen. requiredMessage – The message displayed when the Required validation fails.	RW

Question Type: Time Zone

Answer	Contains the name of the selected time zone.	RW
class	The style class(es) applied to the question on the form (See section on Defining and Using Classes)	RW
ClientID	The ID of the html element on the form	R
disabled	Indicates whether the question field on the form is set to disabled (true/false)	RW
events	Contains any script for the onFocus, onBlur, and onChange events bound to the question field on the form.	
flex	The width % applied to the question on the form. This value is in relation to the container that contains the question field on the form.	RW
id	The system-assigned field ID linked to the field (see also: ClientID)	R
isDirty	Indicates whether the value in the field has been changed since the form was loaded. Returns true if the value has been changed, false if it has not.	R

Label	The label shown on the form for the question	RW
originalAnswer	The value of the field when the form initially loads.	R
QuestionType	TimeZone	R
show	Indicates whether the question field should be shown on the form (true/false)	RW
validation	The validation values and messages used when a form submitted. They correspond to the Validation tab on the question’s configuration screen. requiredMessage – The message displayed when the Required validation fails.	RW

Question Type: Password

Answer	Contains the password value entered (see also ConfirmPassword)	RW
class	The style class(es) applied to the question on the form (See section on Defining and Using Classes)	R
clearAnswer	Method that clears both the Answer and ConfirmPassword attributes	R
ClientID	The ID of the html element on the form	R
ConfirmPassword	Contains the Confirm Password value entered (see also Answer)	R
disabled	Indicates whether the question field on the form is set to disabled (true/false)	RW
events	Contains any script for the onFocus, onBlur, and onChange events bound to the question field on the form.	
flex	The width % applied to the question on the form. This value is in relation to the container that contains the question field on the form.	RW
id	The system-assigned field ID linked to the field (see also: ClientID)	R
isDirty	Indicates whether the value in the field has been changed since the form was loaded. Returns true if the value has been changed, false if it has not.	R
Label	The label shown on the form for the question	RW
QuestionType	Password	R
show	Indicates whether the question field should be shown on the form (true/false)	RW
validation	The validation values and messages used when a form submitted. They correspond to the Validation tab on the question’s configuration screen. regExMessage – The message displayed when the regEx validation fails. regEx – The regEx expression to use validate the field value. Max – A maximum length value that is used to compare the field value against. maxMessage – The message displayed when the Max validation fails. Min – A minimum length value that is used to compare the field value against. minMessage – The message displayed when the Min validation fails.	RW

	required (true/false) – Indicates whether the field is required. Setting this to true will also trigger the display of the required field indicator (*). requiredMessage – The message displayed when the required validation fails.	

Question Type: Grid

addRow()	Method that adds a new row object to the grid (see getRowObejct())	
Answer	An array of JSON objects. Contains the data entered for the rows in the grid. The JSON object label/key is the name given to the column in the grid. The JSON object value is either a discrete data value or an object (e.g. MultiChoiceSelectList column). Each answer also has a delete property (true/false) which indicates the status of the checkbox for each row/answer in the grid.	R
class	The style class(es) applied to the question on the form (See section on Defining and Using Classes)	RW
ClientID	The ID of the HTML element on the form. Used to reference the question with script.	R
events	onChange (onBlur and onFocus not yet implemented) The change event for the grid; returns an object containing data about the change made to the row.	RW
flex	The width % applied to the question on the form. This value is in relation to the container that contains the question field on the form.	RW
getFooterValues()	Returns an array of JSON objects containing the grid footer columns, or if supplied a column key returns the value of that column's footer. Each array object contains: name, value, and aggregationType.	
getRowObject()	Method that returns an empty row object that matches the column definition. It's used in conjunction with the addRow method to insert new rows into the grid.	
gridOptions.data	Contains an array of row objects for the row data contained in the grid	R
isdirty	true/false	
Label	The label shown on the form for the question	RW
loaded		
show	Indicates whether the question field should be shown on the form (true/false)	RW
showDeleteButton(bool)	Hides/Shows the Delete button for the grid (true/false)	
showDeleteColumn(bool)	Hides/Shows the selection checkboxes for each row in the grid (true/false)	
showLabel	Hides/Shows the label for the grid (true/false)	RW

The Integrify form object, intForm, is the supported scripting object for accessing and working with an Integrify form created in the Form Designer. You can access all the form fields (aka Questions) and layout objects via the intForm object. This Reference Manual provides a guide to all supported properties and methods available to a form developer for making truly powerful and intelligent Integrify forms.

NOTE: Integrify form fields have other properties and methods which may be discovered by a form developer while developing a form. Properties and methods not included in this Reference Manual ***are not supported*** for form scripting, and their functionality may change, or be eliminated entirely, as part of any Integrify software release.

The same properties and methods are available for most Integrify form fields. Some field types may have unique properties, due to the functionality provided by that field type. Some properties or methods may function slightly differently for some field types, again due to that field's specific functionality. Table 1 summarizes the availability and usage exceptions for each of the Integrify form field types.

TABLE 1. Property/Method Usage Summary

	Short Text	**Long Text**	**Select List**	**Checkboxes**	**Radio Buttons**	**Calendar**	**File Attachment**	**Contact Search**	**Search Box**	**Email**	**Number**	**Rich Text**	**Link**	**Time Zone**	**Password**	**Grid**	**Signature**
PROPERTY/METHOD	FIELD/QUESTION TYPE Y indicates the property is valid for the field/question type, N indicates the property is not valid for the field/question type.																
addRow	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N
Answer	Y	8	5	Y	Y	4	2	1	Y	Y	9	Y	Y	Y	Y	7	Y
Answer initializes to/clears to: (⌀ indicates empty string)	null/⌀	null/⌀	null/null (5)	null/⌀	null/null	null/null	[ ]/[ ]	[ ]/[ ]	⌀ / ⌀	null/⌀	null/null (9)	null/⌀	null/⌀	null/und	null/⌀	[ ]/[ ]	null/⌀
Choices (array of objects with Label and Value properties, also Selected property (t/f) for Checkbox)	N	N	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N
class	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
clearAnswer	N	N	N	N	N	Y	N	N	Y	N	N	N	N	N	Y	N	N
ClientID	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ConfirmPassword	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N
dbSettings (object with useDB (true/false); DbSourceLabelColumn, DbSourceValueColumn, searchTestValue) (Searchbox also has Name, mappings, propertyNameList)	N	N	N	Y	Y	N	N	N	Y	N	N	N	N	N	N	N	N
disabled (true/false) (initially undefined if not set; Password initializes to false)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
events (.onChange, .onBlur, .onFocus) **NOTE:** Any scripted onChange event will over-ride any Form Rule based on that question.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	10	Y	Y	Y	Y	Y

	Short Text	Long Text	Select List	Checkboxes	Radio Buttons	Calendar	File Attachment	Contact Search	Search Box	Email	Number	Rich Text	Link	Time Zone	Password	Grid	Signature
flex (Width %)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N
getFooterValues	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N
getRowObject	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N
id	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
isdirty (true/false) (true if value of Answer is different than when the form was initialized)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Label	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
multiple (true/false)	N	N	Y	N	N	N	6	6	N	N	N	N	N	N	N	N	N
multiChoiceAnswer	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N
originalAnswer	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N
QuestionType	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
readonly (true/false) (for most fields, initially undefined if not set to true) isReadOnly() => returns true/false	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
refreshGrid	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N
searchFilter (index 0 is guid, 1 is label/name)	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N
selectedValue (comma delimited)	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N
setAnswer	N	N	N	N	N	Y	N	N	Y	N	N	N	N	N	N	N	N
setReadonly																Y	
show (true/false)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

	Short Text	Long Text	Select List	Checkboxes	Radio Buttons	Calendar	File Attachment	Contact Search	Search Box	Email	Number	Rich Text	Link	Time Zone	Password	Grid	Signature
showDeleteButton (true/false)	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N
showDeleteColumn (true/false)	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N
showLabel (true/false)	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y	N
todaysDate (the current date from the server)	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N
validation (object) (not all validation for all types) max; maxMessage; min; minMessage; regEx; regExMessage; required; requiredMessage isValid() => returns true/false	Y	Y	Y	Y	Y	3	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y

1. For ContactSearch *Answer* is an Array. Within each item in the array is an object for the selected user(s) with the following properties: Email, ID, Name, SID, Title, UserName, r, selected, text. Answer is empty array if there is no selection.
2. File Attachment *Answer* is an array of File objects, each with attributes of file_key, lastModified, lastModifiedDate, name, size, text, type, webkitRelativePath. Answer is an empty array if no file has been selected.
3. Calendar type question *validation* is an object containing the following: futureDatesOnly (true/false), maxDate, mdDateFilter, minDate, required (true/false), requiredMessage
4. For Calendar type questions, *Answer* is null if no date has been entered/selected. Answer is a date object and should not be changed directly. Use setAnswer(YYYY-MM-DD) to change the date value. You can also pass in a Date object that has been set to the desired date.
5. Select List fields that have no selection, or the selection has been cleared, can be either null or empty string, even for multi-selection type. For multiple selections the selected values in Answer are stored as a comma separated list.
6. For File Attachment and Contact Search types, checking the "Allow Multiples" checkbox changes the Question Type, but for Select List type a *multiple* attribute is added and set to **true**.
7. For Grid questions, Answer is an Array. If no rows have been added, then Answer is an empty array. The array will contain one json object for each row, the names being the defined column names. You should not directly modify the value of the Answer for Grid questions, use the getRow() and addRow() functions.
8. LongText is backslash encoded. Use standard backslash codes, such as \n for a carriage return, to include non-displayed control characters.

9. Number initializes and clears to null, unless a currency format is selected and then it initializes and clears to the value 0.
10. RichText field does not support the onFocus event.

CODE EXAMPLES

Hiding and Showing Questions

All Questions support the ***show*** attribute. This attribute's value is either true or false. Setting it to true will show the Question on the form, setting it to false will hide the question on the form. Note that even if a Question is hidden (show=false) it still exists on the form, so the Answer will still be saved when the form is submitted, and you can reference the Question in your scripts.

```
intForm.getElementByClientID(‘radQuarter’).show = false;
```

Read-Only and Disabling/Enabling

Most Questions support the ***disabled*** attribute. This attribute's value is either true or false. Setting it to true will disable the question on the form, preventing any changes to the Question by the user. If a Question is disabled the user is also prevented from selecting the text of the Answer.

Getting and Setting the Answer

Simple form type answers (Short Text, Long Text, Number, Link, SelectList, RadioButton, Email, TimeZone)

The Answer contains only the text of the answer. You can get the value of Answer directly or set the value directly.

```
let fName = intForm.getElementByClientID(‘stxtFirstName’).Answer
let lName = intForm.getElementByClientID(‘stxtLastName’).Answer
let fullName = fName + ‘ ‘ + lName
intForm.getElementByClientID(‘stxtFullName’).Answer = fullName;
```

Checkboxes

```
intForm.getElementByClientID(‘ckGroup’).Answer = ‘245’
```

```
intForm.getElementByClientID(‘ckGroup’).Choices[3].Selected=true;
```

Helper function for setting Checkboxes Answer:

```
function setCheckbox(q,a) {
  //q is the ClientID of the Question; a is the Answer value to set
  //if the value of a does not exist in the array of Choices, no action is taken
  if(q && a) {
```

```
                    let targ=intForm.getElementByClientID(q);
                    let i=targ.Choices.findIndex(el=>el.Value===a);
                    if(i  > -1){
                            targ.Choices[i].Selected=true;
                            targ.Answer=a;
                    }
             }
        }
```

Calendar

Always use the setAnswer method to set the value of Answer of a Calendar question. You can supply a static date string as the argument:

```
intForm.getElementByClientID('calStartDate').setAnswer('08-28-2022');
```

This isn't usually realistic, as the desired date is usually calculated. Create a standard Date object and set the date value as appropriate (refer to your ECMA documentation). Use the created Date object as the argument of the setAnswer method:

```
let sd = new Date('08-28-2022');

intForm.getElementByClientID('calStartDate').setAnswer(sd);
```

Grid

The value of Answer for a Grid Question is an array of objects representing each row in the grid. You must use the getRowObject() and addRow() methods to add rows to a Grid Question. The getRowObject method returns an empty json object with keys matching the column names of the Grid.

NOTE: In the 24 Feb 2023 release the grid data structure was changed to add a "name" key to each column definition. The "name" is a unique identifier for each column and must be used when referencing columns in your script. A new "displayName" key was also added, which is the text used as the column header. In grid columns added prior to this update the "name" will be the same as the original column name, on columns added after the release the column name will be a unique identifier (GUID) value.

```
let newRow = intForm.getElementByClientID('grdItems').getRowObject();

newRow["colItemCode"] = 'WD9930-33';

newRow["itemDesc"] = 'Advanced Widget';

newRow["itemPrice"] = '2430.12';
```

```
intForm.getElementByClientID(‘grdItems’).addRow(newRow);
```

Removing rows from a Grid Question:

Use the standard array methods to remove rows from the Answer array. For example, to remove the 4th row (arrays are 0 based):

```
intForm.getElementByClientID(‘grdItems’).splice(3,1);
```

Form State:

```
If (intForm.formState === 'completed') {
   doSomething();
} else if (intForm.formState === 'preview') {
   doSomething();
} else if (intForm.formState === 'runtime') {
   doSomething();
};
```