2RING SCRIPTS & FORMS

USER GUIDE (DOCUMENT VERSION 3.2)

2Ring .. AMERICAS

Address 2Ring America, Inc.
8880 Cal Center Dr. Ste.400
Sacramento, CA 95826
USA
(916) 426-3790
(916) 426-3791
info-na@2Ring.com
www.2Ring.com

2Ring .. EMEAR

Address. 2 Ring, spol. s r.o.
Plynárenská 5
821 09 Bratislava 2
Slovak *Republic*
+421 2 5822 4550
+421 2 5822 4199
info@2Ring.com
www.2Ring.com

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

This document serves as a user guide for configuring the 2Ring SCRIPTS & FORMS application. This guide describes all available UI controls, as well as, how to use them to define the structure of the script or form screens.

Notice: One of the abbreviations 2Ring SF or SF can be used instead of the full name of 2Ring SCRIPTS & FORMS application throughout this document.

CHAPTER 2
BASIC TERMINOLOGY

In this chapter, we will introduce the following terms:

- Configuration File
- Client Application
- Reporting Application

2.1. Configuration File

Displaying wizards and forms requires creating JSON configuration files. The client application is used to display the content within the configuration file. The client application references the configuration file through a URL parameter.

2.2. Client Application

The client application is a web-based application that displays wizards and forms. Agents use the application to collect information from a caller, or as caller guiding script. The client application can be integrated within 2Ring Gadgets in Cisco Finesse or be used as standalone application.

There are two modes, which present information as a set of question cards.

- **Wizard** – a user is displayed information through a set of one or more cards in a sequence (each card is a page). Only one card is displayed at a time. A user will fill out the card and click *Next* to move to the next card. The user can click *Back* to navigate to previous card. A user can also display the wizard timeline, which shows all cards submitted as the user moves through the sequence.
- **Form (Hierarchy)** – all cards are shown on the same page. When a user selects an answer, the next card is displayed below the answer.

2.3. Reporting Application

The reporting application is a web-based application used by team supervisors to run basic reports about the information collected through a wizard.

CHAPTER 3

CONFIGURATION FILE

This chapter describes the configuration of a wizard or form. A JSON configuration file is used to display content with the client application.

Notice: Basic knowledge of JSON (JavaScript Object Notation) is necessary to configure files.

3.1. File Location

Configuration files are stored in the *Client/Content/Configurations* folder under the root directory of 2Ring/SF (default path is C:\inetPub\wwwroot\2Ring\SF\Client\Content\Configurations).

ATTENTION! A configuration file must have a .json extension.

3.2. Wizard/Form(Hierarchy) Setup

Creating a wizard/form requires using JSON syntax to setup a sequence of cards as well as other settings to configure the appearance, such as, wizard/form name, etc.

A Form, also called a Hierarchy, shows all cards on one page (there is no *Next*/*Previous* buttons). A table is defined with a set of rows and columns. Each cell contains set of cards. When an answer on the first card is selected, the next card is shown right below the selected answer. The recommended depth of answers is three.

Use the following properties to configure a wizard:

- **Legend** – name of the wizard, it is located right below the menu.
- **FormType** – use value 0 to indicate a wizard script. Use value 1 to indicate a form (hierarchy).
- **AllowMultipleEntries** – allow multiple form submissions. *StartOver* button is placed on the last page of the wizard once the Finish button is clicked. Possible values: True/False.
- **PostFormFinishMessages** – post conclusion list to parent window after form is submitted. Possible Values: True/False.
- **EnableEdit** – parameter allows the following actions:
  - *BackToThis* button is displayed on the last page. User is navigated back to the current form. The user will be able to edit the filled-out cards.
  - Allow the user to delete the filled-out wizard. *Delete* button is displayed on the last page.

Notice: Only cards with type: "form" can be edited.

- **Columns (array)** – defines columns of a hierarchy form (*FormType* property set to 1). It is not recommended to use more than 5 columns.
    - **Name** – name of the column (not shown anywhere).
    - **Rows (array)** – rows in the column. It is recommended that all columns have the same row count.
    - **Cards (array)** – all cards of the table cell. See *Cards* property below.
- **GlobalConclusions (array)** – specification of global conclusions to be submitted with the wizard or form. Global conclusions are URL parameters passed to the client application.
    - **ConclusionText** – text displayed in the conclusion box.
    - **DisplayInList** – shown in box or not.
    - **UrlParamName** – name of the URL parameter. See chapter *3.5* Global conclusions.
- **Cards (array)** – set of cards to be displayed in a wizard or form. Each card contains the following properties:
    - **CardId** – unique id of the card.
    - **Question** – text shown as a question on the card.
    - **NextCardId** (form card only) - unique CardId of the next card. User is navigated to this card next. The same *CardId* can be used in *NextCardId* property for all answers within one card if you want to go to this card regardless of the selected answer.

ATTENTION! Cards cannot have the same *NextCardId*.

    - **IsConclusion [true/false]** – card answer is saved to the database. Possible values: True/False.
    - **ConclusionText** – text of the conclusion.
    - **PostCardNavigationMessages** - post conclusion list to parent window when going to the next card. Only applicable to form type: Wizard. Possible values: True/False.
    - **Type [SingleChoice/MultiChoice/Form/Text/DropDownList]** – see section *3.3* - Cards.

Notice: In Form/Hierarchy mode, card type: Form must be at the end of the form. *NextCardId* of form cards are ignored in Form/Hierarchy mode.

    - **Answers (array)** – type of answers, see section *3.3.5*.
        - **Text** – text of the answer.
        - **NextCardId** (single and multi-choice card only) – Unique *CardId* of the next card. The next card the user is navigated to when an answer is selected.

ATTENTION! Same limitations are applied on answers NextCardId as cards one. See above.

        - **AnswerType** [Text/Number/Float/Date/YesNo] (form card only) – see section *3.3.5*.
        - **Required [true/false]** (form card only) – form answer required. Possible values: True/False.
        - **ValidationPattern** (form card only, text answer only) – regular expression pattern to validate input by user.

- **Min** (form card only, number/float/date answer only) – minimum value.
- **Max** (form card only, number/float/date answer only) – maximum value.
- **SetCurrentDateTime** (form card only, date answer only) – set current date as default value

Notice: For date answers, Min and Max can be set to value: *NOW*. The current date is added to the field.

Wizard Configuration Example

```
{
    "Legend": "Customer service",
    "FormType": "0",
    "AllowMultipleEntries" : true,
    "PostCardNavigationMessages": true,
    "EnableEdit": true,
    "GlobalConclusions" : [
        {
            "ConclusionText" : "Agent",
            "DisplayInList" : false,
            "UrlParamName" : "AgentID"
        }, ...
    ],
    "Cards": [
        {
            "CardId": "h1",
            "Question":"Are you happy?",
            "IsConclusion": true,
            "ConclusionText": "happiness",
            "Type" : "SingleChoice",
            "Answers": [
                {"Text":"Yes", "NextCardId": "h2" },
                {"Text":"No", "NextCardId": "be_happy" }
            ]
        }, ...
    ]
}
```

Form/Hierarchy Configuration Example

```
{
    "Legend": "IT Helpdesk",
    "FormType": "1",
    "GlobalConclusions" : [
        {
            "ConclusionText" : "CallId",
            "DisplayInList" : true,
            "UrlParamName" : "CallId"
        },
        {
            "ConclusionText" : "Agent",
            "DisplayInList" : false,
            "UrlParamName" : "AgentID"
        },
        ...
    ],
    "Columns": [
     {
      "Name":"Caller",
            "Rows": [
                {
                    "Cards": [
                        {
                            "CardId": "CallerInfo",
                            "Question":"Caller Information",
                            "ConclusionText": "CallerType",
                           "PostCardNavigationMessages": true,
                            "Type" : "SingleChoice",
                            "Answers": [
                   {"Text":"Internal Employe", "NextCardId": "internal" },
                   {"Text":"Customer", "NextCardId": "customer" },
                   {"Text":"Other", "NextCardId": "" }
                            ]
                        },
                        {
                            "CardId": "internal",
                            "Question":"Enter personal information",
                            "ConclusionText": "EmployeInfo",
                            "Type" : "Form",
                            "Answers": [
                                {
                                    "Text":"Name",
                                    "AnswerType": "Text",
                                    "Required": true
                                },
                                {
                                    "Text":"Surname",
                                    "AnswerType": "Text",
                                    "Required": true
                                }
                            ]
                        },
                        ...
                    ]
                }
            ]
     },
```

```
        {
          "Name":"Purpose of the call",
          "Rows":[
              {
                              "Cards": [
                                    ...
                              ]
                        }
          ]
        }
    ]
}
```

3.3. Cards

Each wizard or form is set of question cards. This section describes card types that can be configured. There are four types of cards.

3.3.1. Single Choice

This card shows answers as radio buttons, of which only one can be selected. Each answer leads to the next card or finishes the path/wizard.

3.3.2. Drop Down List

This card shows answers as a drop-down list. Only one value can be selected. Each answer leads to the next card or finishes the path/wizard.

3.3.3. Multi Choice

Answers are presented as checkboxes, of which multiple answers can be selected. When more than one answer is selected the path diverts to multiple paths. When user finishes one path, the wizard shows the beginning of the next path.

3.3.4. Text

This card shows only text. It is useful with text variables.

3.3.5. Form

Answers are in the form of text fields, yes/no, etc. The following answer types are available:

- **Text** – text field displayed. User enters text.
- **Number** - only numbers are accepted.
- **Float** - floating numbers can be entered.
- **Date** - allows user to fill out date and time. Picker is attached to the text field.
- **YesNo** - Yes or No can be selected.

```
{
    "CardId": "1",
    "Question": "Please provide some data about you",
    "IsConclusion": true,
    "ConclusionText": "personal_data",
    "Type": "Form",
    "Answers": [
        {
            "Text": "Name",
            "AnswerType": "Text",
            "ValidationPattern": "(.*){1,32}"
        },
        {
            "Text": "Age",
            "AnswerType": "Number",
            "Min": 18,
            "Max": 99
        },
        {
            "Text": "Day of birth",
            "AnswerType": "Date",
            "Min": "1/1/1900",
            "Max": "NOW",
            "Required": true
        },
        {
            "Text": "Are you Slovak",
            "AnswerType": "YesNo",
            "Required": true
        }
    ],
    "NextCardId": "3"
}
```

3.4. Local Conclusions

3.4.1. Wizard

Every card can be marked as a conclusion card by the *IsConclusion* parameter. The result of the card's (selected/filled answers) is sent to database and is used for reports.

Each card and answer have a *ConclusionText* parameter.

For single and multi-choice cards, the resulting conclusion is in the form of a colon separated combination of *ConclusionTexts*. E.g.: happiness/Yes

For form cards, a conclusion saved to the database consists of [Card_ConclusionText] / [Answer_ConclusionText]:[Answer_Value].

Notice: When *ConclusionText* is not filled, the card question answer *Text* is used instead.

3.4.2. Form (Hierarchy)

Form/Hierarchy conclusions work the same way as Wizard conclusions. However, the *IsConclusion* parameter is ignored; all cards are saved as conclusion cards.

3.5. Global Conclusions

Global conclusions are URL parameters taken from the URL address of the Client application (or identity token received from 2Ring Gadgets). They are saved to the database when a wizard or form is submitted. And can be used to filter data in reports. The following parameter names below are set in the *UrlParamName* property.

Required Parameters

CallID (taken from URL)

AgentID (taken from Token)

AgentName (taken from Token)

CustomerNumber (taken from URL)

CustomerID (taken from URL)

Optional Parameters

CampaignID (taken from URL)

CampaignName (taken from URL)

QueueID (taken from URL)

QueueName (taken from URL)

TeamID (taken from Token)

TeamName (taken from Token)

ATTENTION! Parameters are case sensitive. Parameter taken from Token must be specified in URL when 2Ring SF runs as a standalone application.

For more information on URL construction, see *Chapter 4.*

3.6. Text Variables

A Card can contain text variables which are enclosed in brackets and will be replaced by URL parameters

Example

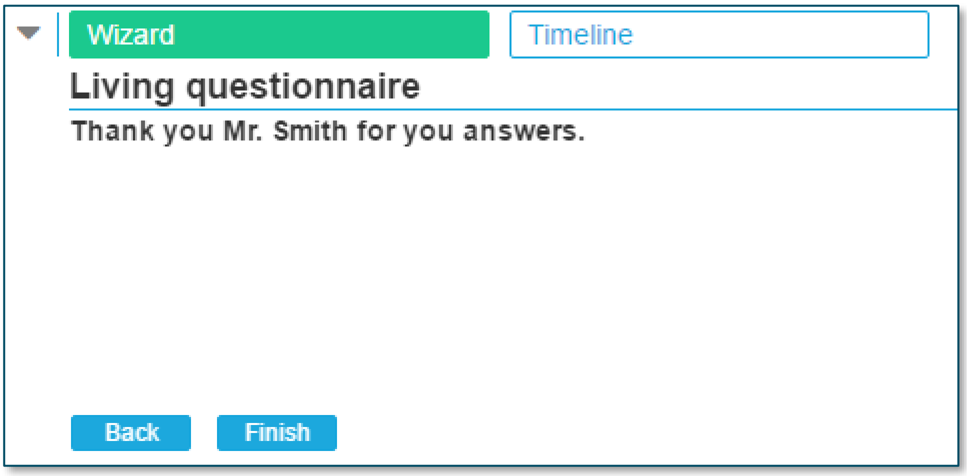
Configuration:

```
{
   "CardId": "end",
   "Question" :"Thank you {param1} for you answers." ,
   "Type" : "Text",
   "NextCardId": 0
}
```

URL:

```
…&param1=Mr. Smith&…
```

Figure 1: Text Variable



CHAPTER 4
CLIENT APPLICATION

4.1. Deployment

The client application has two modes of operation:

- Standalone web application.
- Embedded in the 2Ring Gadget Browser.

4.1.1. Basics

The format of client application URL is:

```
[SCHEME]://[SF_SERVER_ADDRESS]/[APPLICATION_PATH]/Client/?form=[CONFIGURATION_FILE_NAME]&[GLOBAL_CONCLUSIONS]
```

Replace the following placeholders in the URL:

Required:

- **[SCHEME]** – http or https.
- **[SF_SERVER_ADDRESS]** – IP address or hostname of server where 2Ring SF is installed.
- **[APPLICATION_PATH]** – virtual path for 2Ring SF in IIS. By default, the application path is 2Ring/SF.
- **[CONFIGURATION_FILE_NAME]** – name of JSON configuration file without extension.
- **[GLOBAL_CONCLUSIONS]** – global conclusions are additional parameters passed to the client application through the client URL address. Every global conclusion that is added to the URL address must be specified in the JSON configuration file. Global conclusions are delimited by an ampersand sign. Each parameter consists of a KEY and VALUE which are separated by an equal sign. For a list of global conclusions, see section *3.5* Global Conclusions.

  Example of client URL (Absolute path):

  ```
  [SCHEME]://[SF_SERVER_ADDRESS]/[APPLICATION_PATH]/Client/?form=[CONFIGURATION_FILE_NAME]&CallID=1&CustomerID=555&QueueName=Queue_EN
  ```

  Example of client URL (Relative path):

  ```
  ../../SF/Client/?form=[CONFIGURATION_FILE_NAME]&CallID=1&CustomerID=555&QueueName=Queue_EN
  ```

ATTENTION! [SCHEME] and [SF_SERVER_ADDRESS] must be the same as used for 2Ring GADGETS. Better way is to use relative path.

4.1.2. 2Ring BROWSER Gadget

To embed the client application into the 2Ring BROWSER gadget, the following code must be added to the Browser gadget's *browser.json* file and Dialog gadget's *workflow.js* file located in *Managed* or *Custom* folder.

Follow these steps:

- Open the Dialog gadget *workflow.js* file and add similar code below. Below is a sample code snippet which opens a new form when a new call is presented to the agent.
- Replace [SCHEME] with http/https. Replace [SF_SERVER_ADDRESS] with the IP address or hostname of server where 2Ring SF is installed. Replace [APPLICATION_PATH] with virtual path for 2Ring SF in IIS. By default, the application path is 2Ring/SF.

ATTENTION! [SCHEME] and [SF_SERVER_ADDRESS] must be the same as used for 2Ring GADGETS. Better way is to use relative path.

2Ring DIALOG gadget – *workflow.js*

```
workflow.dialog.onNew = function (user, dialog, gettingCallData) {
   gettingCallData.done(function (callData) {
      var formName = 'example';
      var callId = dialog.getId();
      var customerId = dialog.getFromAddress();
      var queueName = workflow.dialog.getCallVariable(dialog,
         'callVariable1');

      var url = '[SCHEME]://[SF_SERVER_ADDRESS]/
[APPLICATION_PATH]/Client/?form=' + formName +
         '&CallID=' + callId + '&CustomerID=' + customerId +
         '&QueueName=' + queueName + '#';

      workflow.browser.addTab('browserGadget',"tab_SF", "Form", url,
"orange", false, true, true, false,
["getAccessToken","getCredentials","getFinesseMetadata","getIdentityToken
"]);

   }).fail(function (error) {
      workflow.messaging.addMessage("Error receiving call data.
         Error: " + error);
   });
};

// Called when a dialog ends.
workflow.dialog.onEnd = function (user, dialog) {
    workflow.browser.removeTab('browserGadget', 'tab_SF');
};
```

Notice: To receive the conclusions list in the workflow, the onMessageReceived event must be configured.

2Ring BROWSER gadget – *workflow.js*

```
workflow.browser.onMessageReceived = function (browserData, message) {
  // Handle messages received only from specific tab
  if (tabID == "tab_SF") {
      console.log("The message was delivered to Workflow");
      // Handle message
  }
};
```

4.2. Usage

This section describes the client application from a user's perspective. It is the primary tool that users use to collect information from callers.

4.2.1. Application Menu

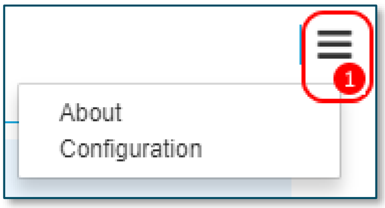
The client menu displays 3 sections in all contexts of the application.

- **Current Form** – loaded form applicable to the customer. First card in the form is displayed.
- **Incomplete Forms** – saved forms for the current customer.
- **Completed Forms** – list of last X forms completed for the customer (default is 10).

4.2.2. Client Settings

To access client application settings, click on the 3-line icon in the top right corner.

Figure 2: Client Settings



1 Client Settings

Client Settings Menu

- **About** – version of 2Ring SF and 2Ring contact information.
- **Configuration**
  - **Top History Forms** – number of completed forms displayed for the customer on the completed forms page. Default is 10.

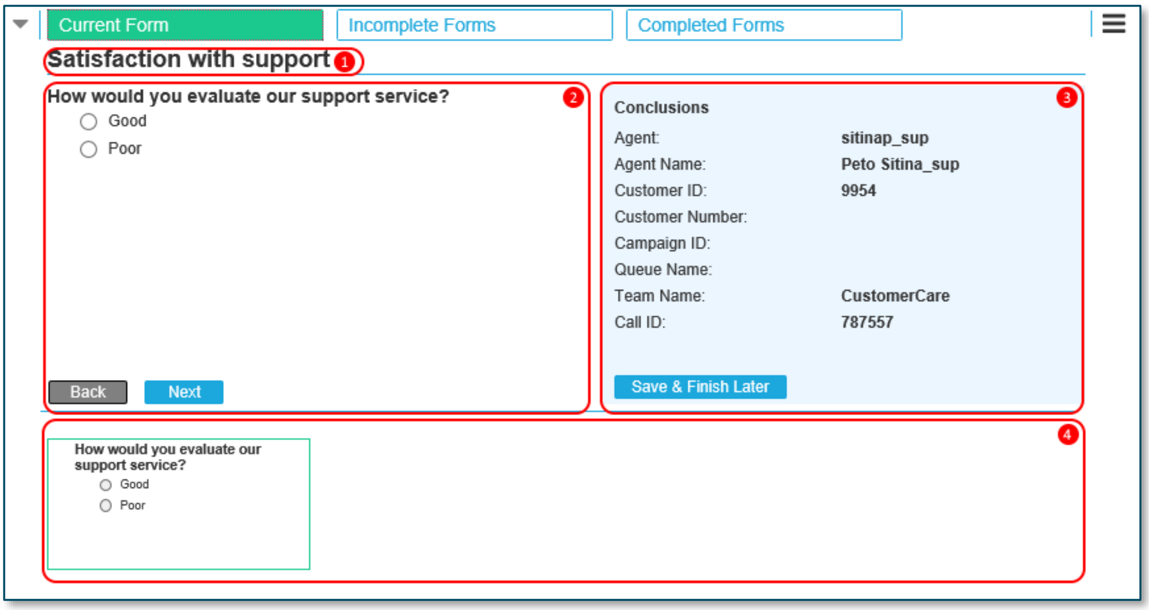
4.2.3. Wizard View

Current Form

The current loaded form for the customer. The current form card is located on the left side of the form and the current conclusions (global and local) are on the right side. The form timeline is located at the bottom of the page.

Use the **Back** and **Next** buttons to navigate between next and previous cards. A user is prevented from moving on to the next card if there are validation errors. The **Save & Finish Later** button allows the user to save the form and complete it later if the customer returns. The timeline can be used to quickly navigate between cards. Just click on the card thumbnail to navigate to that card.

Figure 3: Current Form



1 Form Name

2 Current Card

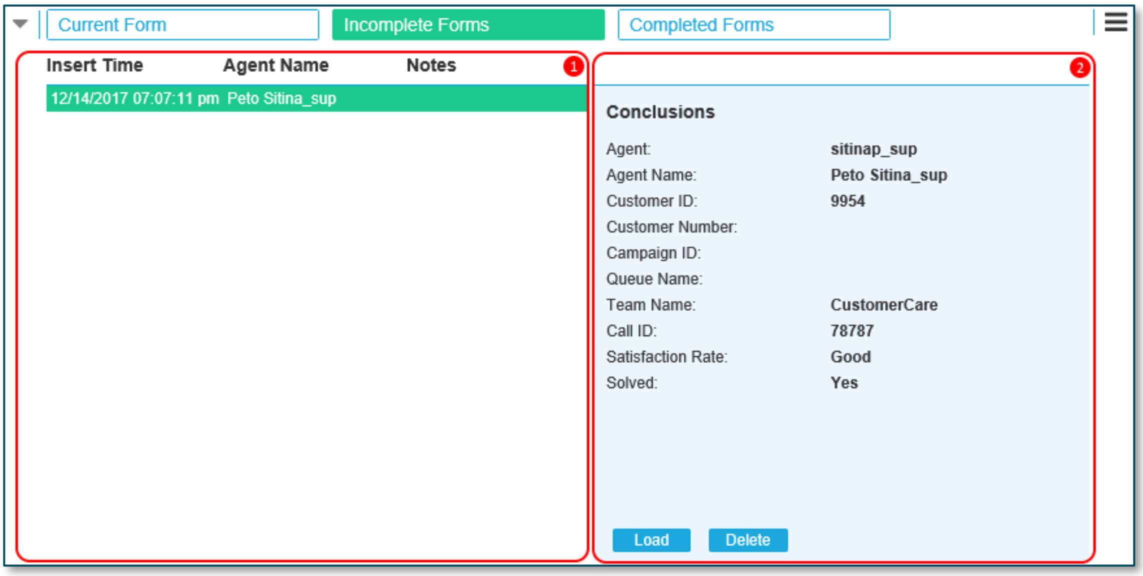
3 Conclusions – Both Global and Local

4 Form timeline

Incomplete Forms

All incomplete forms are displayed in this section. To view an incomplete form, click on a row in the list. The conclusions for the form are listed on the right side. To restore a saved form, click the Load button. To delete a saved form, click the Delete button.

Figure 4: Incomplete Forms



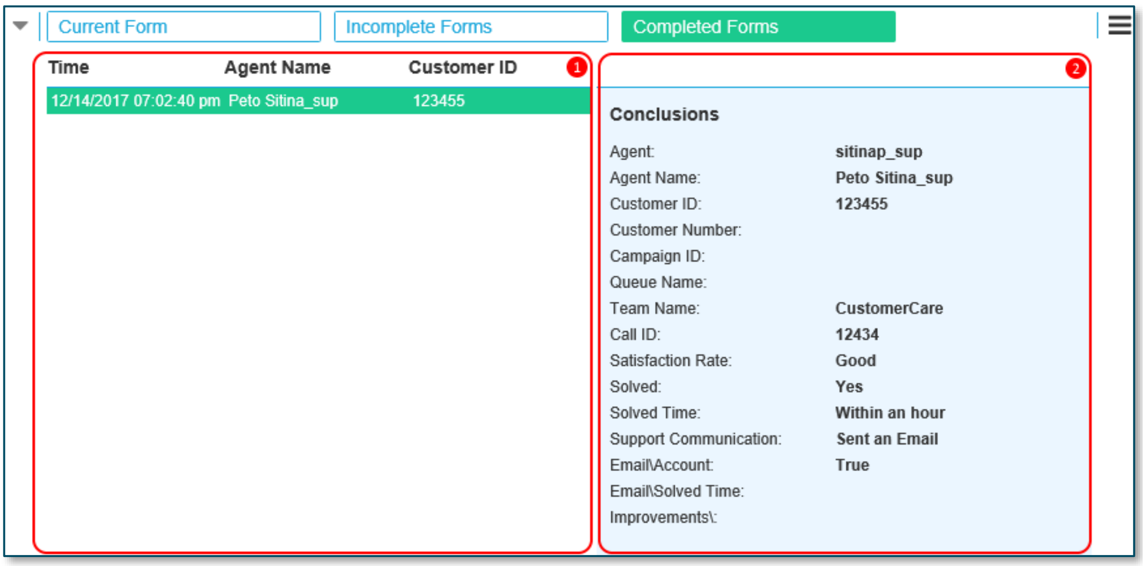
❶ Incomplete Forms

❷ Delete incomplete form

Completed Forms

All completed forms are displayed in this section. To view a complete form, click on the form name in the list. The conclusions for the form are listed on the right side.

Figure 5: Completed Forms

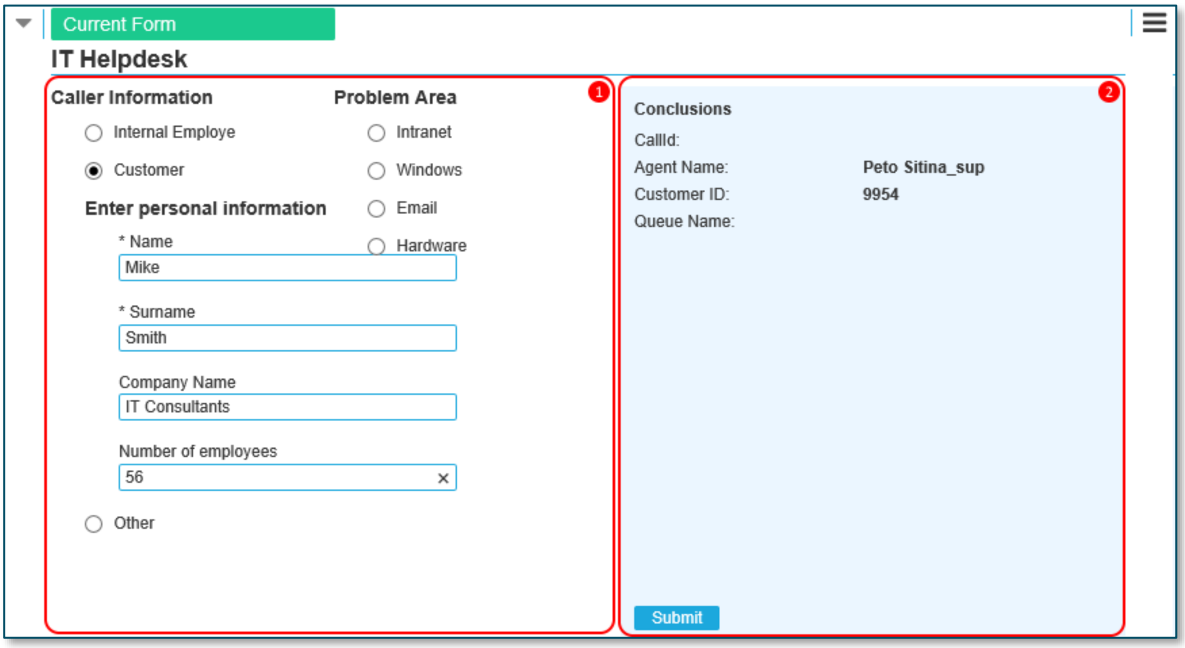


❶ Completed forms list

2 Form conclusions

4.2.4. Form (Hierarchy) View

A Form, also called Hierarchy, looks similar to a Wizard but all cards are shown on one page. When a user clicks the **Submit** button (form is without validation errors), the form is sent to database.



1 Form

2 Form conclusions

CHAPTER 5
REPORTING APPLICATION

The reporting application allows supervisors to see data gathered from customer.

ATTENTION! Reporting can be used for Wizard configurations only.

5.1. Access

Example of Reporting URL (Absolute path):

```
[SCHEME]://[SF_SERVER_ADDRESS]/[APPLICATION_PATH]/Reporting
```

Example of Reporting URL (Relative path):

```
../../SF/Reporting
```

Replace the following placeholders in the URL:

Required:

- **[SCHEME]** – http or https.
- **[SF_SERVER_ADDRESS]** – IP address or hostname of server where 2Ring SF is installed.
- **[APPLICATION_PATH]** – virtual path for 2Ring SF in IIS. By default, the application path is 2Ring/SF.

ATTENTION! [SCHEME] and [SF_SERVER_ADDRESS] must be the same as used for 2Ring GADGETS. Better way is to use relative path.

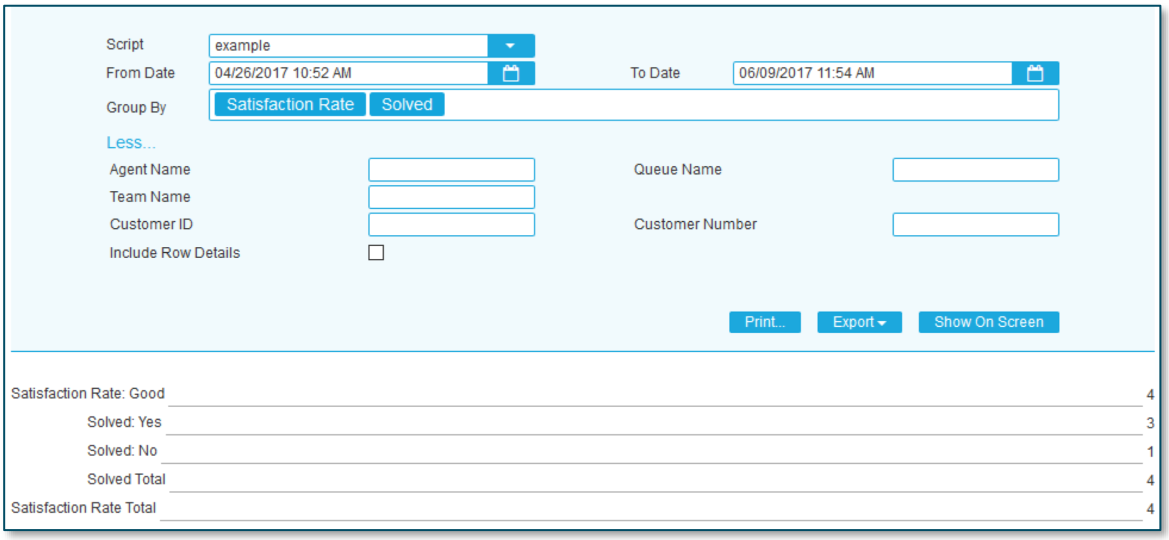
5.2. Usage

This section describes the reporting application from a supervisor's perspective. It is the primary tool that supervisors use to run reports.

5.2.1. Filters

A supervisor is able to create reports of users' activities using the following interface. Filters can be applied to the script selected.

Figure 6: Reporting Application



Filter options:

- **Script** – script to run report on.
- **Dates** – select date range for report. The calendar can be used to the select the date.
  - **From Date –** starting date for report.
  - **To Date –** ending date for report.
- **Group By** – Group data by local conclusions. Max three levels are supported.
- **Global conclusions** – filter data based on global conclusions in script. The global conclusions will differ based on the script selected.
- **Include Row Details** – display additional details per form. Do not use when script has more than 1000 forms created.

5.2.2. Actions

The following actions are available when running a report:

- **Show On Screen** – renders the report in the browser.
- **Print** – sends created report to the printer.
- **Export** – export the report to various formats.