**New MP Command “Load HTML Form”**

Table of Contents

[1.1 Assignment 2](#_Toc451860028)

[1.2 Implementation 2](#_Toc451860029)

[1.2.1 New MP Command “Load HTML Form” 2](#_Toc451860030)

[1.2.2 “User Input For DataShare File” Dialog 2](#_Toc451860031)

[1.2.3 HTML Form Rules 3](#_Toc451860032)

[1.2.3.1 Output Argument Name 3](#_Toc451860033)

[1.2.3.2 Output Argument Type 3](#_Toc451860034)

[1.2.3.3 HTML <input> Tag 4](#_Toc451860035)

[1.2.3.3.1 “text” Type (Text Field) 4](#_Toc451860036)

[1.2.3.3.2 “number” Type (Text Field) 4](#_Toc451860037)

[1.2.3.3.3 “radio” Type (Radio Buttons) 4](#_Toc451860038)

[1.2.3.3.4 “file” Type (File Browser) 4](#_Toc451860039)

[1.2.3.3.5 “checkbox” Type (Checkboxes) 4](#_Toc451860040)

[1.2.3.4 HTML <select> Tag (Drop-Down List) 4](#_Toc451860041)

[1.2.4 MP Script Example 5](#_Toc451860042)

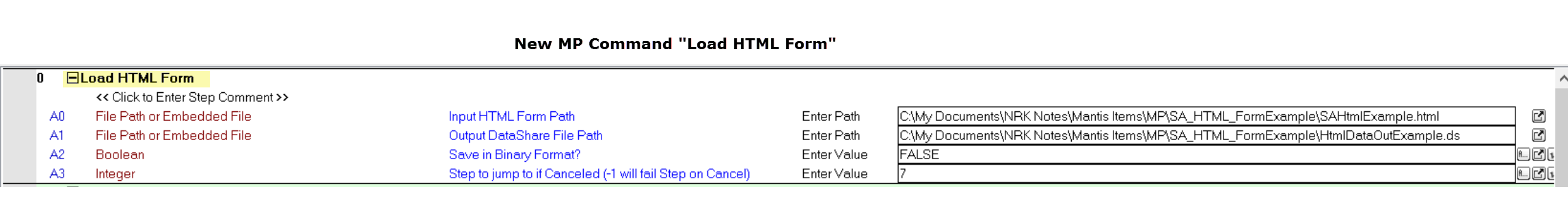
## Assignment

It is required to add MP command that would allow to retrieve user input from a prompted custom HTML file and save it to DataShare file.

## Implementation

### New MP Command “Load HTML Form”

A new MP command “Load HTML Form” has been introduced to the application under “File Operations/DataShare operations” category:



|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Argument Type** | **Argument Name** | **Description** |
| A0 | File Path or Embedded File | Input HTML Form Path | The HTML custom form file location.  If file doesn’t exists, the command fails. |
| A1 | File Path or Embedded File | Output DataShare File Path | The output DataShare file location where the extracted from HTML form user data will be saved.  If DataShare file doesn’t exist, new file is automatically created.  If file exists and has data, then the application tries to restore data from DataShare file to HTML form presenting values from the file. New arguments are automatically appended to the existing DataShare file on Save operation. |
| A2 | Boolean | Save in Binary Format? | The argument defines file format for DataShare output file. |
| A3 | Integer | Step to jump to if Canceled (-1 will fail Step on Cancel) | The argument defines a MP step in case of operation has been cancelled. |

### “User Input For DataShare File” Dialog

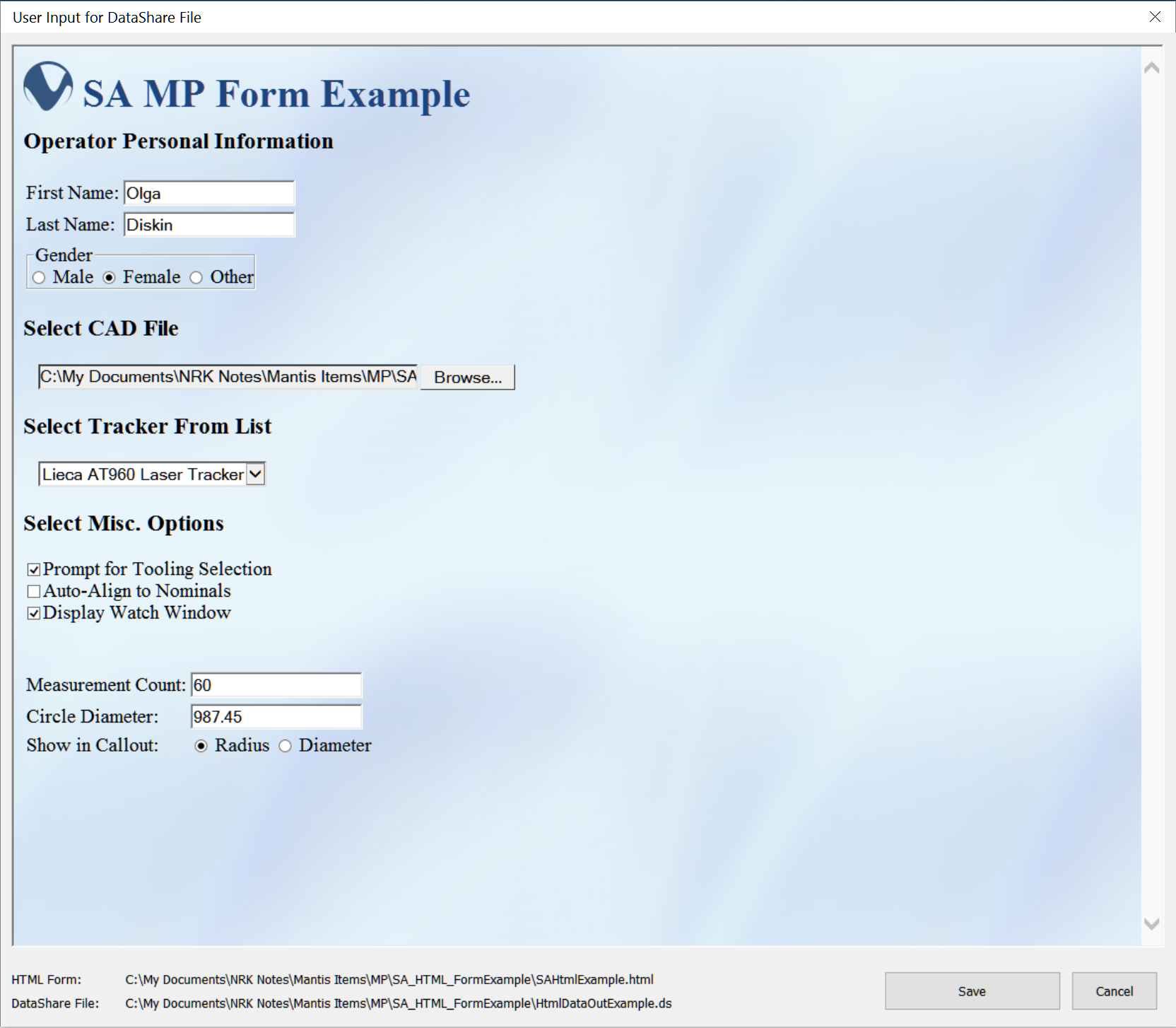
In runtime, the application presents the specified (see A0) custom HTML form in the SA dialog “User Input for DataShare File”. User can fill in the form.

The dialog provides two buttons that should be used – “Save” and “Cancel”.

On “Cancel” the MP step fails or jumps to another step (see Arguments Table above).

On “Save” the data are automatically saves in the DataShare file of specified type (binary or ASCII). Once DataShare file has been saved, it can be loaded by “Load DataShare File” MP command and used with the rest of the DataShare MP commands to handle data.

The following diagram shows an example of custom HTML form:



### HTML Form Rules

Although HTML form can be created per customer custom as it can be, the following rules should be followed when writing HTML code to ensure proper parsing and restoring operations to/from DataShare.

#### Output Argument Name

The argument name should be specified by HTML ***name*** attribute.

#### Output Argument Type

Currently, the application supports the following data types extracted from HTML form:

* String (S)
* Integer (I)
* Double (D)
* Boolean (B)

In order to identify each value the HTML ***id*** attribute must be specified. The id attribute can be unique as needed. In order to identify type of entered value, the application uses first character of HTML ***id*** attribute.

For example, id=”S123” or id=”S87654” will be identified as Strings. The examples below illustrate HTML <input> and <select> tags use.

#### HTML <input> Tag

##### “text” Type (Text Field)

The “text” type value can represent String, Integer, or Double value types.

<input type="text" id="S1" name="First Name" value="">

##### “number” Type (Text Field)

<input type="number" id = "D1" name="Circle Diameter" value=0>

<input type="number" id = "I" name="Desired Measurement Count" value=0>

##### “radio” Type (Radio Buttons)

It is important for radio buttons of the same group to have identical name attributes and data type. As in example below, id indicates String and name “Gender” for all <input> tags:

<fieldset> <legend>Gender</legend>

<input align="left" type="radio" id="S3" name="Gender" value="male" checked> Male

<input align="left" type="radio" id="S3" name="Gender" value="female"> Female

<input align="left" type="radio" id="S3" name="Gender" value="other"> Other

</fieldset>

##### “file” Type (File Browser)

The selected filename path should have String identification as in example below:

<input type="file" size = "50" id="S4" name="CAD File Path" value="">

**Note:** Before presenting HTML form to user, the application restores previously saved values from DataShare file.

An input field of “file” type has read only “value” attribute for security purpose. The application can’t populate (write) pre-stored filename to reopened HTML form. The re-opened form filename path initial value is always empty. User doesn’t have to re-enter filename again on the re-opened form; unless he wants to change the path.

##### “checkbox” Type (Checkboxes)

The HTML checkboxes represent Boolean values. See example below for proper type/name identification:

<input type="checkbox" id="B1" name="Prompt for Tooling Selection" value="">Prompt for Tooling Selection<br>

#### HTML <select> Tag (Drop-Down List)

The type/name identification should be added to <select> tag. See below:

<select id="S5" name="Tracker">

<option value="Faro">Faro Xi Laser Tracker</option>

<option value="Leica">Lieca AT960 Laser Tracker</option>

<option value="API">API Radian Laser Tracker</option>

</select>

### MP Script Example

