Genie Macros

Overview

Genie includes macro capability which allows for keystroke automation. For example, users can be brought to specific menus or programs automatically, or certain screens can be bypassed by automatically entering data for users and pressing keys automatically.

Genie macros are created using XML files that are stored on the IFS.

XML File Creation

Genie looks for macros in folder PROFOUNDUI_SERVER_ROOT/macros.

In a default installation, this folder is:

/www/profoundui/macros

Macros are created by placing XML documents into this folder. Files must be created with the ".xml" extension.

For example, to create a macro named "mymacro", the file name should be:

"mymacro.xml"

XML File Encoding

Genie supports the following file encodings for macros:

- UTF-8 (CCSID 1208)
- UTF-16 (CCSID 1200)
- ISO-8859-1 (CCSID 819 or CCSID 1252)

For customers outside of the United States, it's recommended to use UTF-8 encoding.

The encoding type for Genie macros must be specified using an XML declaration, for example:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

The file encoding specified in the XML declaration must match to the actual file encoding as controlled by the file's CCSID.

The CPY command can be used to change the CCSID of an IFS file by copying over (replacing) an existing file and using the TOCCSID and DTAFMT(*TEXT) parameters.

XML Document Syntax

The specifics of XML document syntax are outside the scope of this document. For information on this, see the W3C XML reference here:

http://www.w3.org/TR/REC-xml/

The W3C also has a tutorial here:

http://www.w3schools.com/xml/default.asp

Genie Macro XML Elements

Genie macros are created using the following XML elements:

- <macro>

  This element is the XML document root element that contains all other elements.

- <detect>

  This element allows you to specify a location on the screen to locate certain text. If successful, the actions defined inside this element will be performed.

  This element serves as a container that holds other elements that define actions. The actions will be performed any time that the detection is successful in the session.

- <detectonce>

  This element is similar to <detect> except that the actions will happen only 1 time in a session – the first time the screen is detected.

- <id row=”1” col=”1”>Text</id>

  This element specifies the location and text to search for on the screen.

  Attriubtes “row” and “col” are required on this element. They specify the row and column locations to search for text on the screen. The row/col numbers given should be based on the element id assigned to the field in Genie.

  This element must also contain content which defines the text to search for at the given location. The text is case sensitive, but leading or trailing blanks will not affect a match.

- <input row=”1” col=”1”>MyText</input>

  This element allows you to automatically input text into a 5250 entry field.

Attriubtes “row” and “col” are required on this element. They specify the row and column locations for the entry field. The row/col numbers given should be based on the element id assigned to the field in Genie.

This element must also contain content which defines the text to place into the entry field. The text will be put into the field exactly as specified – including any leading blanks.

- <key>F1</key>

  This element allows you to automatically press a function key. The key is specified as text content in the element. For example:

  - F1
  - F16
  - Enter
  - RollUp
  - RollDown

- <cursor row=”1” col=”1” />

  This element allows you to set the cursor to a specific location. This is useful for those screens that have cursor-sensitive functionality – such as selecting a subfile record where the cursor is located.

  Attriubtes “row” and “col” are required on this element. They specify the row and column locations for a field where the cursor will be placed. The row/col numbers given should be based on the element id assigned to the field in Genie.

- <close />

  This element closes down the Genie session.

Genie Macro Processing

Genie macros are created by specifying <detect> or <detectonce> elements with one or more <id> elements and placing other elements inside to perform actions.

This is called a “detect block”.

There are 2 types of detect blocks:

- “Input” detect blocks contain at least one <key> element. They may also contain one or more <input> elements and a <cursor> element.

- “Close” detect blocks contain only a <close> element which is used to shut down the session.

Either type of detect block requires one or more <id> elements. An error will result if your detect blocks do not conform to one of the above types.

The ordering of detect blocks in a macro does not matter since only 1 will function at a time, based on screen detection.

Macro Example

The following macro makes use of both types of detect blocks. The macro will:

- Bypass the Display Program Messages screen the first time it is encountered in a session by pressing the enter key.
- Enter the command WRKSPLF into the command line the first time the i5/OS main menu screen is detected in a session.

  Note that the <cursor> element is not necessary here as the menu panel is not cursor sensitive. This is included simply to show how the element is specified.

  The example also shows using multiple <id> elements to detect this panel, although it's not really necessary.
- Automatically close the session any other time the i5/OS main menu screen is detected.

```
<?xml version="1.0" encoding="UTF-8" ?>
<macro>
    <detectonce>
        <id row="1" col="15">Display Program Messages</id>
        <key>Enter</key>
    </detectonce>
    <detectonce>
        <id row="1" col="1">MAIN</id>
        <id row="1" col="31">System i Main Menu</id>
        <cursor row="20" col="6" />
        <input row="20" col="6">WRKSPLF</input>
        <key>Enter</key>
    </detectonce>
    <detect>
        <id row="1" col="1">MAIN</id>
        <id row="1" col="31">System i Main Menu</id>
        <close />
    </detect>
</macro>
```

Since this example declares encoding type UTF-8, the IFS file should be created with CCSID 1208.

Substitution Variables in Macros

Genie allows substitution variables in macros to allow for dynamic processing. Substitution variables can be used in any element attribute or text content.

Substitution variables are defined by placing the variable name in brackets at the desired location. For example, the above macro could be adjusted to create a multipurpose macro for launching Genie to any OS command like this:

```
<?xml version="1.0" encoding="UTF-8" ?>
<macro>
    <detectonce>
        <id row="1" col="15">Display Program Messages</id>
        <key>Enter</key>
    </detectonce>
    <detectonce>
        <id row="1" col="1">MAIN</id>
        <id row="1" col="31">System i Main Menu</id>
        <cursor row="20" col="6" />
        <input row="20" col="6">[Command]</input>
        <key>Enter</key>
    </detectonce>
    <detect>
        <id row="1" col="1">MAIN</id>
        <id row="1" col="31">System i Main Menu</id>
        <close />
    </detect>
</macro>
```

In this case, a substitution variable named ‘Command’ is defined which is used to populate the value that is entered into the command line on the i5/OS main menu panel.

When Genie is launched, the substitution variable values can be passed in. Genie will replace the values before running the macro.

An error will result if Genie is unable to replace any substation variables defined in the document – i.e. they are not passed correctly.

Calling Genie to Run a Macro

To run a macro in Genie, the macro name must be specified as a query string parameter named “macro”. The macro name is the name of the XML file. The “.xml” extension should be omitted as Genie will append this automatically. The macro name is not case sensitive.

Assuming that the first example above was called “test.xml”, it could be launched as follows:

http://Systemi:8080/profoundui/genie?macro=test

Calling Genie to Run a Macro with Substitution Variables

When running a macro that includes substitution variables, all variable values must be passed as query string parameters.

This is done using query string parameters “varX” and “valueX”. For example, if you have 1 variable, you will use parameters “var1” and “val1”. If you have 2 variables, you will use “var1”, “value1”, “var2”, and “value2”.

“varX” gives the variable name, and “valueX” gives the value to associate with it.

For example, assuming the 2nd example macro above was called “test2.xml”, it could be launched like this:

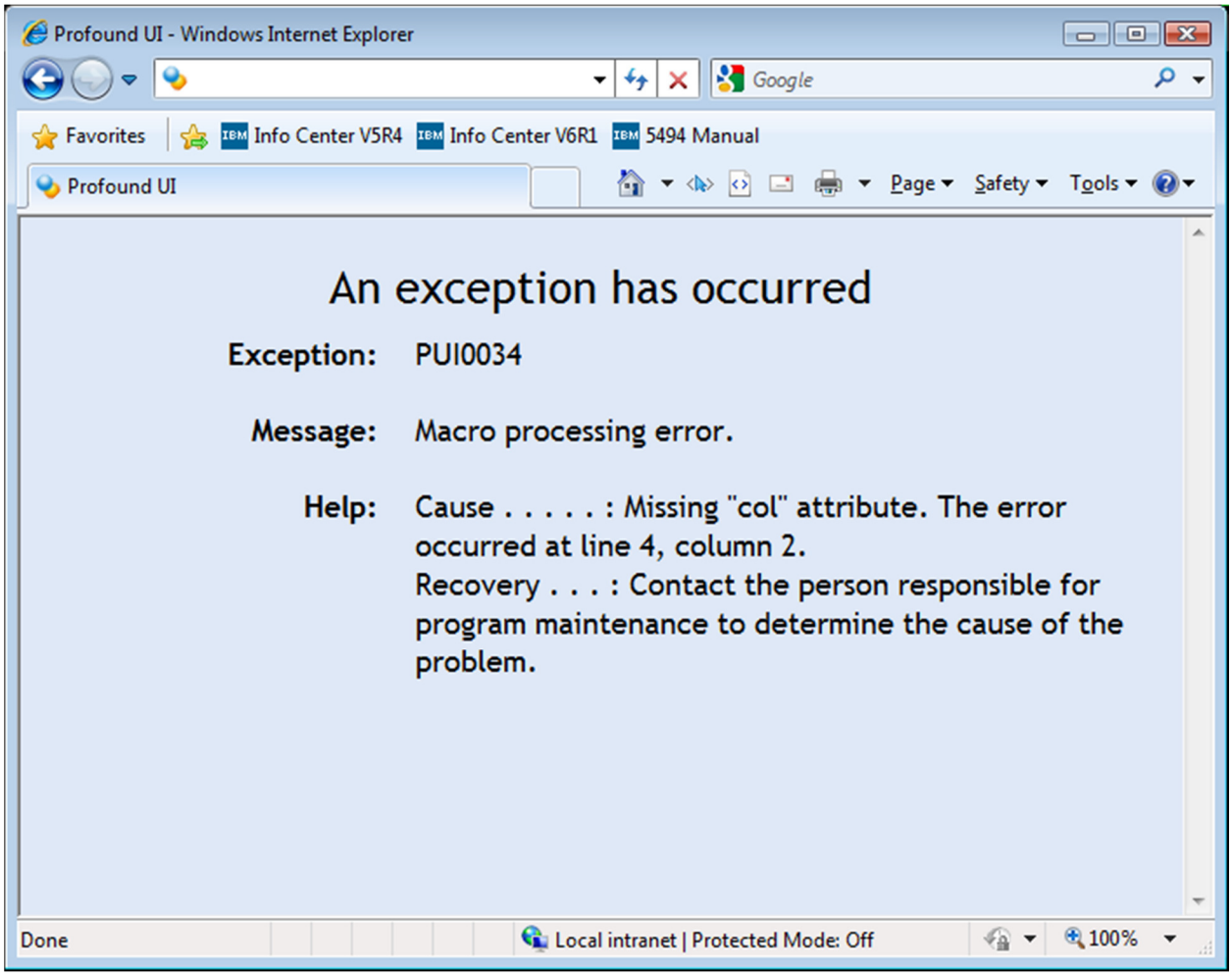
http://Systemi:8080/profoundui/genie?macro=test2&var1=command&value1=WRKMSG

The variable names are not case sensitive. The variable value will be placed into the macro exactly as specified.

Macro Errors

If Genie is unable to process the macro for any reason (i.e. file not found, no authority, XML syntax error, etc.), the session will not be started and an error screen will display.

If the error is inside the XML document itself, line and column numbers will display. For example:


Profound UI - Windows Internet Explorer
Google
Favorites
Info Center V5R4
Info Center V6R1
5494 Manual
Profound UI
Page
Safety
Tools
An exception has occurred
Exception: PUI0034
Message: Macro processing error.
Help: Cause . . . . . : Missing "col" attribute. The error occurred at line 4, column 2.
Recovery . . . : Contact the person responsible for program maintenance to determine the cause of the problem.
Done
Local intranet | Protected Mode: Off
100%