



VIRTEL Screen Redesigner

HOWTO Guide

Version: Version 4.55-2

Date: 31/07/2015

NOTICE

Reproduction, transfer, distribution, or storage, in any form, of all or any part of the contents of this document, except by prior authorization of ***SysperTec Communication***, is prohibited.

Every possible effort has been made by ***SysperTec Communication*** to ensure that this document is complete and relevant. In no case can ***SysperTec Communication*** be held responsible for any damages, direct or indirect, caused by errors or omissions in this document.

As ***SysperTec Communication*** uses a continuous development methodology; the information contained in this document may be subject to change without notice. Nothing in this document should be construed in any manner as conferring a right to use, in whole or in part, the products or trademarks quoted herein.

SysperTec Communication and ***VIRTEL*** are registered trademarks. Names of other products and companies mentioned in this document may be trademarks or registered trademarks of their respective owners.

TABLE OF CONTENTS

TABLE OF CONTENTS	III
TABLE OF FIGURES	VII
SUMMARY OF AMENDMENTS	IX
1 INTRODUCTION.....	13
2 CONFIGURING THE MODERNIZED APPLICATION	14
2.1 BEFORE YOU START	14
2.1.1 Address for client line.....	14
2.1.2 Address for upload.....	14
2.2 UPDATING THE TRANSACTIONS	15
2.2.1 Choose a name for your application	15
2.2.2 Transaction “xxxxCAPT”	15
2.2.3 Transaction “xxxxAJAX”	17
2.2.4 Transactions “w2h”	19
2.3 CUSTOMIZING THE SCENARIO	20
2.3.1 Scenario in loadlib.....	21
2.3.2 Scenarios in a VIRTEL VSAM file.....	21
3 ACCESSING VIRTEL SR	22
3.1 APPLICATION MANAGEMENT	22
3.1.1 Install	22
3.1.2 User data folder	22
3.1.3 Uninstaller.....	22
3.2 STARTING VIRTEL SR	22
3.2.1 First access	22
3.2.2 Define a project.....	23
3.2.3 Components of the project.....	24
3.2.4 Migration	25
3.3 OPEN A PROJECT	26
3.3.1 Configuring Virtel SR	27
3.4 BASE URL FOR TESTS.....	27
3.5 URL FOR UPLOAD	27

3.6	TRANSACTION	27
3.7	BASE PAGE	27
4	UPDATING FILES TO THE MODERNIZATION DIRECTORY	28
4.1	HOW TO PROCEED	28
4.2	PRELIMIRAY FILES TO UPLOAD FOR A MODERNIZATION PROJECT	28
4.3	PRELIMINARY UPLOADS FOR THE DEVELOPMENT	29
4.3.1	<i>Custom.js</i>	29
5	SOURCE SECTION	30
5.1	CAPTURING A 3270 SCREEN	31
5.2	GENERATING THE MAP	31
5.3	LOADING A SELECTED MAP	32
5.4	MFS MAPS	32
5.4.1	<i>MFS Maps with Source</i>	32
5.4.2	<i>MFS Maps without Source</i>	32
6	WORKBENCH SECTION	33
6.1	GENERATION	34
6.2	CHOICE COLUMN	34
6.3	EXAMPLES	35
6.3.1	<i>Copy area example</i>	35
6.3.2	<i>Enterable "UNPROT" area example</i>	35
6.4	WORKBENCH MANAGEMENT	36
6.4.1	<i>Saving the workbench</i>	36
6.4.2	<i>Opening a saved workbench</i>	36
6.4.3	<i>Example workbench</i>	36
6.4.4	<i>Workbench import and export</i>	37
6.4.5	<i>Rescue</i>	37
7	TEMPLATE SECTION	38
7.1	CREATION	38
7.2	CONTROLS	39
7.3	SAVE IN TEST PAGE MODE	40

7.4	SAVE IN STANDARD PAGE MODE	41
7.5	BATCH INTERFACE	42
8	TESTING	43
8.1	UPLOAD TO VIRTEL DIRECTORY	43
8.2	MODES OF OPERATION	44
8.2.1	<i>Test mode</i>	44
8.2.2	<i>Standard mode</i>	45
9	TRY IT SECTION	46
9.1	TESTING IN “TEST” MODE	47
9.2	TESTING IN “STANDARD” MODE.....	48
9.2.1	<i>Base page</i>	48
9.2.2	<i>Navigation</i>	49
9.3	VIRTEL DEVELOPMENT REVIEW	50
9.3.1	<i>HTML code choice</i>	50
9.3.2	<i>Testing</i>	51
10	TROUBLESHOOTING	52
10.1	VIRTEL LOG	52
10.2	CLIWHOST.HTM.....	52
10.3	RESPONSE FORMAT ERRORS	53
11	MODERNIZED APPLICATION CUSTOMIZATION	54
11.1	STATEMENTS THAT MAY BE CUSTOMIZED	54
11.1.1	<i>The Interface side</i>	54
11.1.2	<i>The modernized application side</i>	54
11.2	STYLES CSS.....	54
11.2.1	<i>Change the background color</i>	54
11.2.2	<i>Replace the logo</i>	54
11.3	HTML MODIFICATIONS	55
11.3.1	<i>Changing the header (top of page)</i>	55
11.3.2	<i>Changing the trailer (bottom of the page)</i>	56
11.4	JAVASCRIPT STATEMENTS	56

11.5	NORMALIZING THE TAGS	57
11.5.1	<i>Title</i>	57
11.5.2	<i>Bottom of the page</i>	58
11.6	DEVELOPING A PROJECT WITH A JAVASCRIPT LIBRARY	59
12	CUSTOMIZING THE WORKBENCH	60
12.1	SUPPLIED ADDITIONAL CHOICES	60
12.1.1	<i>Snippet examples</i>	60
12.1.2	<i>Saving your changes</i>	62
12.1.3	<i>Your adaptation</i>	62
12.2	CREATING NEW CHOICE	63
12.2.1	<i>Display the choice</i>	63
12.2.2	<i>Processing this choice</i>	64
12.2.3	<i>Relating to rules and processes</i>	65
12.3	SAMPLES	65
13	TEMPLATES FOR VARIOUS LANGUAGES	66
13.1	YOUR LANGUAGE CHOICES	66
13.2	WORKBENCH RELATED CHOICES AND LANGUAGE SUPPORT	67
13.3	CHOICE OF THE TEMPLATE LANGUAGE	68
13.4	TEMPLATE CREATION	68
14	CONCLUSION	69
15	INDEX	71

TABLE OF FIGURES

Figure 1-1: VIRTEL– Virtel SR – A modernization project.....	13
Figure 2-1: VIRTEL message log indicating URL addresses assigned.....	14
Figure 2-2: TESTCAPT transaction definition (CICS)	16
Figure 2-3: TESTAJAX transaction definition (CICS)	18
Figure 2-4: Definition of a “w2h” transaction.....	19
Figure 2-5: Example scenario TESTAJAX	20
Figure 3-1: Starting when no project has been defined	22
Figure 3-2: Page to create a project	23
Figure 3-3: Interface after the creation of the project	23
Figure 3-4: Components of your project	24
Figure 3-5: Project sources for VirtelSR 4.54	25
Figure 3-6: Virtel SR interface for your project.....	26
Figure 3-7: Virtel SR configuration settings	27
Figure 5-1: VirtelSR source map	30
Figure 5-2: Virtel SR screen capture page.....	31
Figure 5-3: Dialog box for map capture	31
Figure 5-4: Virtel SR– MFS map source code	32
Figure 6-1: VirtelSR workbench	33
Figure 7-1: Virtel SR page template	38
Figure 7-2: Display of error messages after the template has been generated	39
Figure 7-3: Virtel SR – Constructing a test page and saving it locally.	40
Figure 8-1: Virtel SR facility for upload	43
Figure 8-2: Saving the HTML test page in CLI-DIR for display on browser	44
Figure 8-3: Saving the HTML page template in CLI-DIR for display on browser	45
Figure 9-4: Virtel SR Screenshot with the “view” icon (circled) and its dialog popup	47
Figure 9-5: Base page for standard mode of operation.....	48
Figure 9-6: Base page with sub-page “TRADER002.htm” loaded after modernization	49
Figure 9-7: Base page with classic sub-page loaded and before modernization.	50
Figure 9-8: Example of inserting HTML code	50

Figure 10-1: Page CLIWHOST.htm	52
Figure 11-1: Rules of the style sheet dealing with background color	54
Figure 11-2: CSS rule processing the logo display.....	55
Figure 11-3: HTML tags for the head band	55
Figure 11-4: Example of statements for the bottom of the page	56
Figure 11-5: Sample of JavaScript statements to be adapted	57
Figure 11-6: Sample of modified statement	57
Figure 11-7: Sample of preparing tags to be normalized.	58
Figure 12-1: Example of generated tags upon the wrapped named COPY-FROM choice.	60
Figure 12-2: Standard statements to process a wrapped named COPY-FROM choice while creating the template.	61
Figure 12-3: Adapted statements to process a “wrapped named COPY-FROM” choice while creating the template.	62
Figure 12-4: Statement to display the new choice “my title” in the choice select box of the workbench.....	63
Figure 12-5: Framework to guide the coding of the process of the new choice.	64
Figure 13-1: Setting icon.	66
Figure 13-2: Setting panel.....	66
Figure 13-3: Workbench choices.	67
Figure 13-4: Use sample.	67
Figure 13-5: Template language icons.	68
Figure 13-6: Language selection.	68

SUMMARY OF AMENDMENTS

Virtel 4.55 (VirtelSR 4.55-2 July 2015)

- The user data is located in a separate folder from the installation folder. This means that user data will be maintained across VirtelSR upgrades.
- The GUI interface is now based on NW.js which is an evolution of Node Webkit. The installation of NodeJS is therefore no longer necessary as was required in version VirtelSR455-1.

VirtelSR 4.55-1 (May 2015)

- The fileSystem HTML5 API has been removed from requirements of the W3C. Therefore we looked for another solution using new technologies, seeking again a pure HTML solution with the ability to access the files of the modernization projects, on the PC. This new version is based on Node Webkit which refers to NodeJS with its ability to access the file system. This implementation makes it easier for the user to deal with the files. The core of the features remains stable. A project developed by a previous version of Virtel SR can be continued with the new version without change.

VIRTEL 4.54 (February 2015)

- Correction of minor bugs.
- Additional elements in "project PC files".

VIRTEL 4.53 (July 2014)

- Correction of minor bugs.
- New verification to give error messages.

VIRTEL 4.52 (December 2013)

- New batch facility to generate templates and upload them to the VIRTEL directory (cf 7.5).
- New sort facility for the list of the sandbox files.

VIRTEL 4.50 (July 2013)

- Change of the set-up to use the API fileSystem of HTML5. The components required for the Virtel SR interface are installed in an instance of VIRTEL (which does not need to be the VIRTEL that contains the modernized project). The components for the modernization project are saved on the user work station, and, if required, uploaded to the VIRTEL that contains the modernized project.
- A project developed with VSR448 may be imported.
- New recovery capabilities (undo an action on the workbench, rescue a workbench which has not been saved before closing the interface).
- New capability to export/import workbench in spreadsheet format.

VIRTEL 4.48 (December 2012)

- Provision of a new way of processing a BMS map capture, and the view of a template rendering. The new version is based on the standard VIRTEL Web Access (VWA) interface. It requires version 4.48 of VIRTEL.

VIRTEL 4.47 (March 2012)

- Creation of a private client file for the workbench choice customization. Folder with choice samples.
- New logo for the CLIWHOST.htm page.
- Move of the file "toWorkbench.js" to the "Utils" folder.

VIRTEL 4.46 (21 Oct 2011)

- Multilingual template feature.
- Presentation of the upload procedure at the set up step since, with the new VIRTEL options, the directory CLI-DIR will be supplied empty.
- FMS map (IMS) processed as a standard way of creating a workbench file.
- Enhancement of the utility to enable the choice of a JavaScript library (Yahoo! User Interface Library, jQuery, jQuery mobile) to work with for modernizing a 3270 application.
- Better separation of the codes likely to be customized, with comments to explain how to handle them.
- The new default code for the modernization project separates the JavaScript statements and the HTML tags. The JavaScript statements specific to the project are contained in a JavaScript file which is common for both modes (test and standard)
- New workbench choices (e.g. "area definition") are intended to make it easier to customize workbench choices to generate HTML tags that may be related to JavaScript statements.
- Some workbench choices have been removed (e.g. "COPY bar") because they were specific cases of generic choices. (Example: "COPY bar" was a particular case of the new "wrapped named COPY-FROM" workbench choice. The way of wrapping the COPY-FROM could be customized to render a bar through specific HTML tags related to specific CSS rules.)
- Upgrade of YUI to 2.8.2r1.

VIRTEL 4.44 (21 Oct 2010)

- Update of document pictures.

VirtelSR 4.43M1 (11 Mar 2010)

- Miscellaneous bug fixes.
- Upgrade of YUI to version 2.8.0r4.
- New “template” and “test” modes to select which buttons are displayed.
- While the template is generated, the errors are traced. Clicking on an error message causes the related line of the workbench to be highlighted.

VIRTEL 4.41 (17 Mar 2008)

- Cut and paste workspace rows
- User-specified HTML code in workspace

VIRTEL 4.40 (20 Nov 2007)

- Initial release of VIRTEL Screen Redesigner

1 INTRODUCTION

This manual is a step-by-step guide for using **VIRTEL Screen Redesigner** (“Virtel SR”). It provides information on how to develop a modernization project where **VIRTEL** is a host/web protocol converter, running on the mainframe, and configured to serve 3270 applications through user-friendly web pages (GUI) to thin-client web browsers running different platforms and operating systems.

It would be advantageous if the reader is familiar with the Virtel product and has a general understanding of how Virtel works. In particular Virtel scenarios, Virtel HTML tags and Virtel Web Access (VWA) and Virtel Modernization (VWM). This information can be found in the Virtel User guide. Also, some familiarity with how Virtel connects to 3270 applications would be useful. This information can be found in the Virtel Connectivity guide.

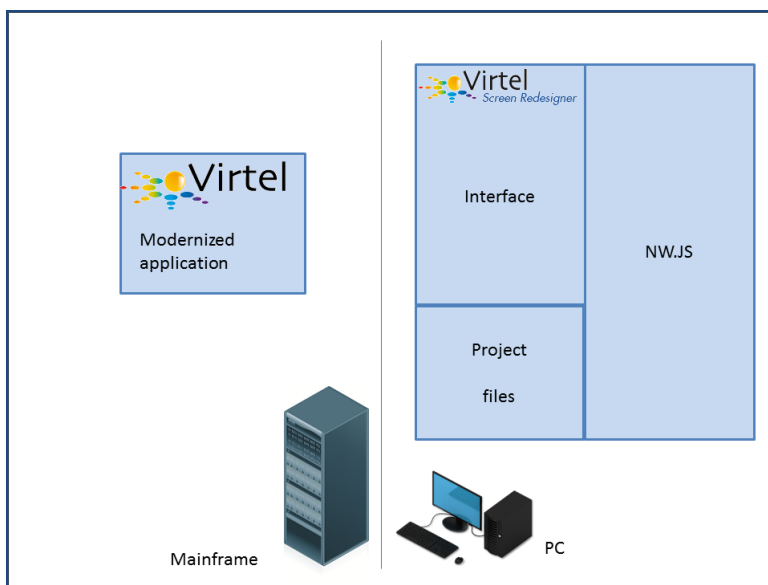


Figure 1-1: VIRTEL– Virtel SR – A modernization project

Virtel SR is a tool which helps you to build web pages and transform static mainframe 3270 application screens to a more GUI based presentation.

- **Virtel SR** is designed as a desktop application. The interface accesses user files which are stored locally on your workstation.
- **Virtel SR** helps you to design HTML pages based on a 3270 screen descriptions or **maps** (see chapter 30).
- **Virtel SR** uses a **workbench** (see chapter 6) which allows you to **decide** how each 3270 field should be presented on the HTML page.
- Based upon your presentation decisions, **Virtel SR** generates a **page template** (see chapter 7), which is a combination of HTML and **VIRTEL**-specific tags which takes into account context of the application.
- After generating the page template, **Virtel SR** helps you **test** the results in conjunction with the mainframe application (see chapter 8).
- Finally, **Virtel SR** itself can be **customized** by advanced users.

2 CONFIGURING THE MODERNIZED APPLICATION

The process of modernizing the application requires the developer to configure parameters and components used by **VIRTEL**. This chapter covers the configuration of **VIRTEL** on the mainframe.

2.1 BEFORE YOU START

Before you can use **Virtel SR**, the **VIRTEL** instance for the modernized project must be already installed and started. The **VIRTEL** installation procedure is described in the *VIRTEL Installation Guide* (if you have installed the full version of **VIRTEL**) or the file *INSTALL-FSE.txt* (if you have installed the Free Starter Edition of **VIRTEL**).

Your first task is to identify the URL addresses assigned to **VIRTEL**. Use SDSF, or a similar utility, to browse the message log (JESMSGLG) of the **VIRTEL** started task. You will find several VIRHT02I messages which indicate the URL assigned to each HTTP line defined in **VIRTEL** (see example below):

```
VIRHT02I LINE HTTP-CLI (C-HTTP) HAS URL http://192.168.235.30:41002
VIRHT02I LINE HTTP-W2H (W-HTTP) HAS URL http://192.168.235.30:41001
```

Figure 2-1: VIRTEL message log indicating URL addresses assigned

2.1.1 ADDRESS FOR CLIENT LINE

Line HTTP-CLI (C-HTTP) represents the port used for client applications. Note the URL shown in the message. This is the “Base URL for tests” which you will need later.

2.1.2 ADDRESS FOR UPLOAD

Line HTTP-W2H (W-HTTP) represents the port used for administration of the VIRTEL system. Note the URL shown in the message. This is the “URL for upload” which you will need later.

What is installed with VIRTEL

During **VIRTEL** installation the following elements are installed as standard:

- VIRTEL line W-HTTP (port 41001) is used for VIRTEL administration
- VIRTEL line C-HTTP (port 41002) is reserved for customer applications
- Entry point WEB2HOST is used for VIRTEL administration
- Entry point CLIWHOST is reserved for customer applications
- Directory W2H-DIR is used for VIRTEL administration pages and scripts
- Directory CLI-DIR is reserved for customer applications

Note: the port numbers are customizable and may be different at your installation

These definitions are sufficient for most installations. With more knowledge of **VIRTEL** you can add additional lines, entry points, and directories if required.

2.2 UPDATING THE TRANSACTIONS

VIRTEL refers to mainframe applications using “transactions” defined in the **VIRTEL** configuration file. This section describes how to define the transactions which access your application.

2.2.1 CHOOSE A NAME FOR YOUR APPLICATION

You must choose a four character name to identify the application which you are going to redesign. When using **Virtel SR**, each redesigned application requires two **VIRTEL** transactions named xxxxCAPT and xxxxAJAX, where xxxx is the four-character name you chose for your application:

- The xxxxCAPT transaction is used for capturing 3270 screens so that they can be processed by **Virtel SR**.
- The xxxxAJAX transaction is used for running the modernized application.

Two transactions TESTCAPT and TESTAJAX are already defined for you when **VIRTEL** is installed. They can be used as samples to define your own xxxxCAPT and xxxxAJAX transactions.

2.2.2 TRANSACTION “XXXXCAPT”

Transaction CLI-31C (external name TESTCAPT) is supplied by **VIRTEL**.

It can be copied to create a new transaction named xxxxCAPT. Your transaction must be customized so that it matches:

- Your 3270 application VTAM applid
- The transaction name used to start your application

TRANSACTION DETAIL DEFINITION

The procedure for defining your transaction is as follows:

- LOGON to VIRTEL from a 3270 screen and press F1 (Admin), or point your browser at the address of line HTTP-W2H (W-HTTP) which you previously noted in section 2.1.2, then select Admin from the VIRTEL Web Access menu.
- At the configuration menu, press F3 (Entry Points)
- Put the cursor on the entry point CLIWHOST
- Press F4 to display the list of transactions
- Put the cursor on the transaction whose external name is TESTCAPT
- Press F12 to display the transaction detail definition screen, shown below

TRANSACTION DETAIL DEFINITION	
Internal name	CLI-31C
External name	TESTCAPT
Description	Logon to TEST application (screen capture)
Application	DBDCCICS
PassTicket	
Name	
Application type	1 = VTAM
Pseudo-terminals	CLVTA
Logmode	
How started	1 = menu
Security	1 = basic
H4W commands ?	
Logon message	
TIOA at logon	Signon&/w&*7D4C7D&'114BE9'&U&'114CF9'&P&/Acomp1 ete&/w DEMO&/A
TIOA at logoff	
Initial Scenario	
Final Scenario	
Input Scenario	CAPTURE
Output Scenario	

Figure 2-2: TESTCAPT transaction definition (CICS)

Make the following changes to this screen:

INTERNAL NAME

If you decided to name your application something other than TEST, then you should also change the *Internal name* from CLI-31C to a new available one (e.g. CLI-50C). We suggest that you use the “C” suffix for all of the capture transactions so that you can quickly identify all the transactions that are capture transactions.

EXTERNAL NAME

If you decided to name your application something other than TEST, then you should also change the *External name* from TESTCAPT to xxxxCAPT.

DESCRIPTION

If you decided to name your application something other than TEST, then you should also change the *Description* to show that change.

APPLICATION NAME

In the *Application* field, replace DBDCCICS with the VTAM APPLID (ACB name) of your 3270 application name (if not previously changed).

SCRIPTS

The *TIOA at logon* and *TIOA at logoff* fields contain the necessary commands to automatically logon and logoff to/from your 3270 application.

TIOA AT LOGON

The script supplied is suitable for the standard signon procedure used in current versions of CICS (GMTRAN=CESN). If your CICS system uses a nonstandard signon screen (or if your application is not a CICS application) then you will need to adapt this script. Typically, you will only need to change the transaction name imbedded within the command. In the example above, you would change DEMO to your initial CICS transaction ID.

TIOA AT LOGOFF

The *TIOA at logoff* field contains the commands which your application requires when the user disconnects the session. For example, *&*6D CESF LOGOFF* issues Clear followed by a CICS logoff command.

Further information about Connection / Disconnection Scripts see the manual – *Virtel Connectivity*, available from Syspertec.

REGISTERING THE CHANGES

When you have completed the necessary changes:

- Press F1 to save the changes
- Press F3 to return to the list of transactions
- Press F3 again to return to the list of entry points
- **Press F1 again** to update the entry point

SCENARIO

This transaction refers to an input scenario CAPTURE which is supplied as an assembled module in the VIRTEL loadlib. Depending on your configuration scenarios can be located within a loadlib or a Virtel directory. See the Virtel Connectivity manual for more information on scenario location or Chapter 2.3.

If the entry point of your project (default is CLIWHOST) is configured in order to retrieve scenarios from a dedicated **VIRTEL** directory other than the default SCE-DIR directory, then the scenario file CAPTURE.390 must be copied to that **VIRTEL** directory. Scenario .390 files are delivered in the SCE-DIR directory within a VIRTEL VSAM file.

This scenario is supplied as a standard in the **VIRTEL** directory "SCE-DIR". If necessary, please use the VIRTEL administration interface to copy the scenario to the target **VIRTEL** directory.

2.2.3 TRANSACTION "XXXXAJAX"

Transaction CLI-31U (external name TESTAJAX) is supplied by **VIRTEL**.

It can be copied to create a new transaction named xxxxAJAX. Your transaction must also be customized so that it matches:

- Your 3270 application id.
- The transaction name used to start your application.

The procedure for defining your transaction is as follows:

TRANSACTION DETAIL DEFINITION

In the List of Transactions screen (see procedure in section 2.2.2 above):

- Put the cursor on the transaction whose external name is TESTAJAX.
- Press F12 to display the transaction detail definition screen, shown below.

The screenshot shows the 'TRANSACTION DETAIL DEFINITION' window. It contains the following fields and values:

Internal name	CLI-31U
External name	TESTAJAX
Description	Logon to CICS TEST application (AJAX mode)
Application	DBDCCICS
PassTicket	
Name	
Application type	1 = VTAM
Pseudo-terminals	CLVTA
Logmode	
How started	1 = menu
Security	1 = basic
H4W commands ?	
Logon message	
TIOA at logon	Signon&/w&*7D4C7D&'114BE9'&U&'114CF9'&P&/Acomp1 ete&/w DEMO&/A
TIOA at logoff	
Initial Scenario	
Final Scenario	
Input Scenario	
Output Scenario	TESTAJAX

Figure 2-3: TESTAJAX transaction definition (CICS)

Make the following changes to this screen:

INTERNAL NAME

If you decided to name your application something other than TEST, then you should also change the *Internal name* from CLI-31U to a new available one (e.g. CLI-51U). We suggest that you use the “U” suffix for all of the AJAX transactions so that you can quickly identify all the transactions that are AJAX transactions.

EXTERNAL NAME

If you decided to name your application something other than TEST, then you should also change the *External name* from TESTAJAX to xxxxAJAX.

DESCRIPTION

If you decided to name your application something other than TEST, then you should also change the *Description* to show this change.

APPLICATION NAME

In the *Application* field, replace DBDCCICS with the VTAM APPLID (ACB name) of your 3270 application (if not previously changed).

SCRIPTS

The *TIOA at logon* and *TIOA at logoff* scripts may need to be adapted, as described in the previous section, including changing the transaction ID at the end of the *TIOA at logon* field.

OUTPUT SCENARIO

If you decided to name your application something other than TEST, then you should also change the *Output Scenario* from TESTAJAX to xxxxAJAX.

REGISTERING THE CHANGES

When you have completed the necessary changes:

- Press F1 to save the changes
- Press F3 to return to the list of transactions
- Press F3 again to return to the list of entry points
- **Press F1 again** to register the changes to the entry point

2.2.4 TRANSACTIONS “W2H”

Under the entry point “CLIWHOST”, the following transaction must be defined to **VIRTEL** within the CLIWHOST entry point:

The screenshot shows the 'TRANSACTION DETAIL DEFINITION' window with the following values:

Internal name	CLI-03CJ
External name	w2h
Description	Directory for custom.js
Application	CLI-DIR
PassTicket	
Name	
Application type	4 = PAGES
Pseudo-terminals	CLLOC
Logmode	
How started	2 = sub-menu
Security	0 = none
H4W commands ?	
Check URL Prefix	/w2h/custom-js
TIOA at logon	
TIOA at logoff	
Initial Scenario	
Final Scenario	
Input Scenario	
Output Scenario	

Figure 2-4: Definition of a “w2h” transaction

If the transaction is not there, you will need to execute the batch mainframe job “CUSTCSS” from the VIRTEL SAMPLIB to create it.

In the definition of this transaction, set the field “Application” to “CLI-DIR” (instead of the default “W2H-DIR”).

When you have completed this change on the definition:

- Press F1 to save the changes,
- Press F3 to return to the list of transactions,
- Press F3 again to return to the list of entry points,
- **Press F1 again** to update the entry point.

2.3 CUSTOMIZING THE SCENARIO

VIRTEL allows a “scenario” to be associated with any transaction. A scenario is a user-written module which can intercept and modify the processing of the dialog between the mainframe application and the browser.

Each modernized application has an associated scenario for operation in standard mode (will be described in 8.2.2). The scenario analyzes the 3270 screen to determine which HTML page template to display.

```
* ----- *
*  SCENARIO FOR VIRTEL SCREEN REDESIGNER SAMPLE APPLICATION  *
* ----- *
TESTAJAX  SCREENS  APPL=TESTAJAX
          SCENARIO OUTPUT
*
* Analyze the screen identifier at row 1 column 2 length 6
*
PIVOT     IF$   (1,2,6),EQ='DEMOM1',THEN=MENU01
          IF$   (1,2,6),EQ='DEMOM2',THEN=MENU02
          GOTO$ CLASSIC
*
* Select page template for screen TEST01
*
MENU01    DS     0H
          SET$   PAGE,'PGTEST01.html'
          SCENARIO END
*
* Select page template for screen TEST02
*
MENU02    DS     0H
          SET$   PAGE,'PGTEST02.html'
          SCENARIO END
*
* Select default page template for all other screens
*
CLASSIC   DS     0H
          SET$   PAGE,'VSRajax.html'
          SCENARIO END
*
          SCRNEED
          END
```

Figure 2-5: Example scenario TESTAJAX

You will find other samples in the supplied folder “samples\scenario”.

VIRTEL offers two ways for dealing with a scenario:

2.3.1 SCENARIO IN LOADLIB

The member SELECTPG is supplied at installation time and can be found in the **VIRTEL** SAMPLIB library (a PDS on your Z/OS mainframe) or in the SUBLIB library (z/VSE) with the name "SELECTPG.A" to be retrieved through a PUNCH LIBR. It serves as a model scenario for the XXXXAJAX scenario, one of the two scenarios required for modernization. To create your scenario, copy SELECTPG into the **VIRTEL** CNTL library dataset, renaming it to xxxxAJAX, where xxxx is the name of the transaction to be tested. In the above example it would be TEST. The member name must match the value entered in the *Output Scenario* field of the TESTAJAX transaction in section 2.2.3.

It is assumed that each screen presented within an application can be uniquely identified. In the following examples screen names are used as a means of identification. After customizing the scenario to match the application's screen names, run the job ASMSCEN in the **VIRTEL** CNTL library to assemble and link the scenario into a **VIRTEL** LOADLIB. It is best to keep user assembled routines in a USER.LOADLIB concatenated before the installation **VIRTEL** LOADLIB.

If you later modify and reassemble the xxxxAJAX scenario, the following z/OS command is necessary to force **VIRTEL** to reload the updated module:

```
F VIRTEL,NEW=xxxxAJAX
```

Note: The message VIR0061W PROGRAM xxxxAJAX NOT IN MEMORY is normal if the program has not yet been loaded. Subsequently the command should complete with the message VIR0060W PROGRAM xxxxAJAX IS A NEW COPY

2.3.2 SCENARIOS IN A VIRTEL VSAM FILE

As of version 4.48, **VIRTEL** offers the ability of coding the scenario with "**VIRTEL Studio**". The sample above might be copied as a basis to build the source (file with .vsc extension) for your project. In this case, the compiled scenario (file with .390 extension) is only required to be uploaded to the **VIRTEL** directory provided that the entry point has been set with the appropriate parameters. The **VIRTEL** directory will be associated with a VSAM file. See "VIRTEL Installation" and "VIRTEL Connectivity" manuals for more information on user VSAM files and directories.

3 ACCESSING VIRTEL SR

Virtel SR is a desktop application which user the “NW.js”, an application runtime environment. **Virtel SR** has to be downloaded and installed before use. See the following instructions on how to install **Virtel SR**.

3.1 APPLICATION MANAGEMENT

3.1.1 INSTALL

Download the latest version of **Virtel SR** (vsr-nw455-2.zip) according to your installation platform. Unzip the file to create the **Virtel SR** installation folder. Within the installation folder locate the file “VirtelSR.exe”. This is the program that will launch **VirtelSR**. Create an ICON on your desktop or create a Shortcut to make **Virtel SR** easily available from the desktop or the start menu.

3.1.2 USER DATA FOLDER

Launching VirtelSR for the first time will create a folder dedicated to **Virtel SR** user data. The location of this folder is the default user data location depending on the operating system: e.g. “C:\Users\FRED\AppData\Local\VirtelSR\data”.

3.1.3 UNINSTALLER

To uninstall the application just delete the **Virtel SR** installation folder. There is no uninstall procedure.(see 3.1.1). The user data folder can be kept for future **Virtel SR** installations or upgrades. If the user data folder is not required that should also be deleted.

3.2 STARTING VIRTEL SR

To start **Virtel SR**, click the icon or Shortcut that relates to the “VirtelSR.exe” installation file (see 3.1.1).

3.2.1 FIRST ACCESS

At your first access without any user data, or with empty user data, no project is defined.

The interface opens by requesting a project to be defined. In the examples that follow the project TEST will be used:

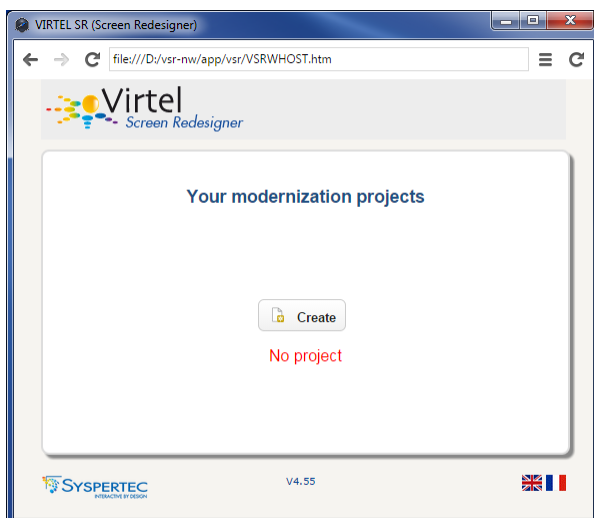
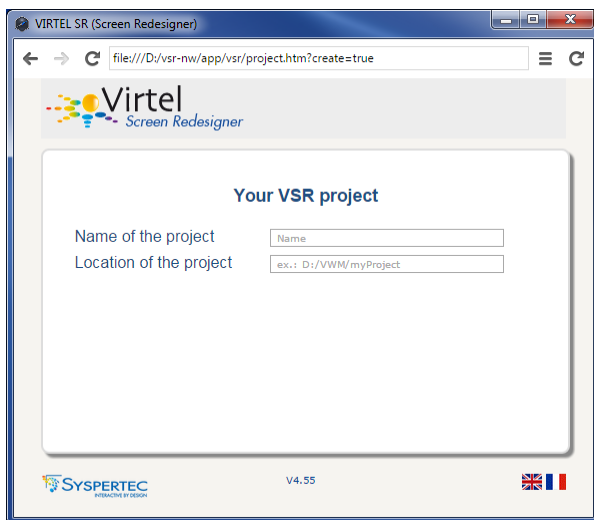


Figure 3-1: Starting when no project has been defined

3.2.2 DEFINE A PROJECT

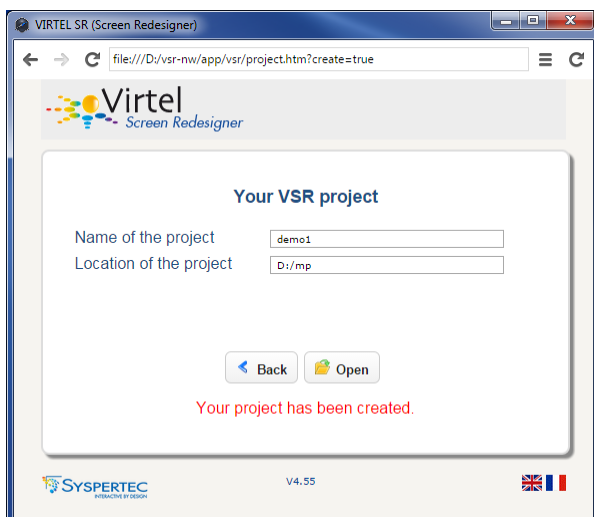


The screenshot shows a web browser window titled 'VIRTEL SR (Screen Redesigner)'. The address bar contains 'file:///D:/vsr-nw/app/vsr/project.htm?create=true'. The page header features the Virtel logo and 'Screen Redesigner'. The main content area is titled 'Your VSR project' and contains two input fields: 'Name of the project' with a placeholder 'Name' and 'Location of the project' with a placeholder 'ex.: D:/VWM/myProject'. The footer includes the SYSPERTEC logo, version 'V4.55', and flags for the UK and France.

Figure 3-2: Page to create a project

In creating a project you must

- Create a folder on your PC, in order to receive the project components.
- Provide details of the project to the project creation wizard.
- Enter the absolute path for the location of the folder (Must be created before creating the project).
- Validate your input by clicking the "Create" button.



The screenshot shows the same web browser window as Figure 3-2, but the 'Name of the project' field now contains 'demo1' and the 'Location of the project' field contains 'D:/mp'. Below the input fields are two buttons: 'Back' and 'Open'. A red message 'Your project has been created.' is displayed below the buttons. The footer remains the same with the SYSPERTEC logo, version 'V4.55', and flags for the UK and France.

Figure 3-3: Interface after the creation of the project

3.2.3 COMPONENTS OF THE PROJECT

The project validation creates sub-directories, some containing default or sample files.

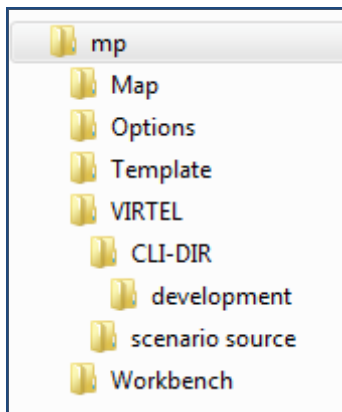


Figure 3-4: Components of your project

In the directory of your project, you will then find the following folders

Map

This folder is intended to receive the BMS/MFS maps for the definition of the 3270 screens.

If original 3270 BMS/MFS maps are available these can be placed into this folder.

Options

This folder contains default options files some of which can be customized.

Template

Templates generated by the **Virtel SR** interface will be placed into this folder.

Workbench

Workbenches generated by the **Virtel SR** interface will be placed into this folder.

VIRTEL

This folder contains the default components designed for the **VIRTEL** directory dedicated to the project. These are required for the modernization of a project. See Chapter 4.2.

The default customization sub-folder is called “CLI-DIR”. If you intend to dedicate another **VIRTEL** directory to your project other than CLI-DIR then rename this sub-folder accordingly.

A sub-folder “scenario source” is provided as an archive to save the modernization files/components.

3.2.4 MIGRATION

If you have already developed a project with a previous version of **Virtel SR** (from 4.50 to 4.54), you will be familiar with the component layout. In the previous versions, the modernization file structure was:-

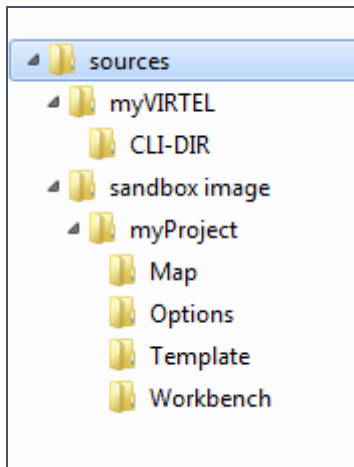


Figure 3-5: Project sources for VirtelSR 4.54

Both structure formats are compatible with **VirtelSR**. To use VirtelSR4.55 using the structure developed with VirtelSR 4.54, you only need to define the project by entering the name of the project (myProject), and the location of the folder “myProject” of your existing sources during the creation process. If you define your project with the new version of the interface, your existing files will be migrated accordingly.

Note: The VIRTEL “CLI-DIR” folder is only used manually by the user. It therefore can be left within the previous structure location.

3.3 OPEN A PROJECT

Once you have defined your project, you can access it with the **Virtel SR** interface:

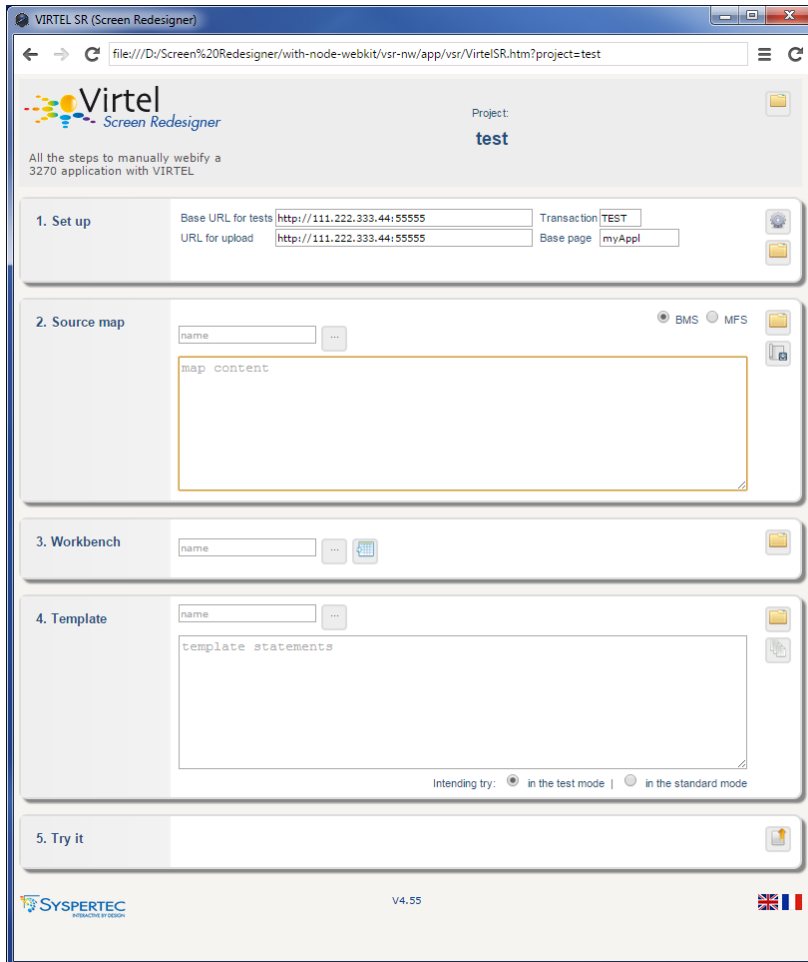


Figure 3-6: Virtel SR interface for your project

When launching **Virtel SR**, this will be the default page. The project icon (top right) will enable you to go back to the definition/selection page.

3.3.1 CONFIGURING VIRTEL SR

When using **Virtel SR** for a new project you must configure certain parameters to match your mainframe VIRTEL settings.

The first part of the **Virtel SR** user interface screen contains these configuration parameters:

Figure 3-7: Virtel SR configuration settings

3.4 BASE URL FOR TESTS

In this box, enter the URL address of the client line which you noted in Section 2.1.1.

3.5 URL FOR UPLOAD

In this box, enter the URL for upload which you noted in Section 2.1.2.

3.6 TRANSACTION

In this box, enter the four character name XXXX which identifies your host application. This name must match the prefix of the transactions XXXXCAPT and XXXXAJAX which you defined in Section 2.2.1. The name TEST is recommended for your first application as these transactions are already defined in VIRTEL.

3.7 BASE PAGE

The “base page” field contains the name of the HTML base page for the standard mode of operation. This field is initialized to **myAppl** at installation. See Section 8.2 for a description of the two modes of operation that are provided by **Virtel SR**. It is recommend to use default name myAppl to begin with.

Note: **Virtel SR** will append **.htm** to the name you specify here. It is the filename of an HTML page template in the source folder “\myProject\VIRTEL\CLI-DIR”.

When you have entered all of the parameters, you are ready to start using the **Virtel SR** interface to modernize your project. The parameters are automatically saved in JSON format in the **Mainframe.txt** file in the **Options** folder of “Your project” file structure. The parameter settings are always visible on the **Virtel SR** interface and may be changed at any time.

The modernization examples are based on the **VIRTEL** target directory, which is by default supplied empty. Before you can start a **VIRTEL** modernization project, a number of files need to be uploaded to that directory. This is the subject of the following chapter.

4 UPDATING FILES TO THE MODERNIZATION DIRECTORY

For the modernized project, the target **VIRTEL** modernization directory is by default CLI-DIR (A separate directory can be created for each modernization project). Files, or page templates, must be uploaded from the installed Virtel SR directory to the Virtel modernization directory in order to be able to modernize a project.

4.1 HOW TO PROCEED

Use the following procedure to upload a page template to VIRTEL:

- In the fifth section of the **Virtel SR** interface ("Try it"), click the icon on the right to open the upload drag/drop interface.
- Select the files from your **Virtel SR** installation directory on your PC. The files are listed in Section 4.2
- Drag them and drop them onto the area "CLI-DIR" (or another modernization directory) of the upload interface.
- The files will be uploaded and an upload report produced.
- Refresh your browser screen after uploading the files. CTRL-R can be used to do this.

4.2 PRELIMINARY FILES TO UPLOAD FOR A MODERNIZATION PROJECT

Your **Virtel SR** user source folder (see 3.2.3) "myProject\VIRTEL\CLI-DIR" contains the following which must be uploaded into the default (CLI-DIR) or other modernization directory files:

CLIWHOST.HTM

This is a file whose name depends upon the entry point name. The default is CLIWHOST. Do not change this name without first contacting **SysperTec** for support. This page will be displayed on your browser if only the URL for Virtel Client line is entered. (See chapter 2.1.1).

FAVICON.ICO

This is the icon which is displayed at the left of the URL or in your favorites. The **SysperTec** icon is displayed in your web page by default. This can be customized.

VSRAJAX.HTML

This is the default template for constructing screens using the XXXAJAX transaction (See chapter 2.3). It renders the 3270 screen presentation. It should not be changed until you are familiar with the VIRTEL template process.

ERROR.HTM

This template can be chosen to display error messages upon return from an application.

CUSTOMFUNCTIONS.JS

This file contains the JavaScript statements which are specific for this project. They can be customized.

MYAPPL.HTM

This file is the HTML home page for the project. Within this page modernized sub-pages will be inserted.

VISU.CSS

This file is the default style sheet for the modernization project. This file must be uploaded to the modernization **VIRTEL** directory.

4.3 PRELIMINARY UPLOADS FOR THE DEVELOPMENT

If you wish to further develop the presentation and look and feel of your application the CUSTOM.JS file can be uploaded to provide additional development capabilities. The project folder « \myProject\VIRTEL\CLI-DIR\development» contains the following file:

4.3.1 CUSTOM.JS

The file “custom.js” is only to be used for the development of the project. It extends Virtel Web Access to **Virtel SR** capabilities for capturing the BMS maps (See Chapter 5) or viewing the rendering of a template (See Chapter 9.1). This file must be uploaded to the **VIRTEL** modernization directory selected by the transaction described in Section 2.2.4. The custom.js does not belong to the components of the modernized application in the production mode. In production mode the Virtel production “custom.js” options will be utilized.

5 SOURCE SECTION

A “map” is a field-by-field description of a 3270 screen for a mainframe application. **VIRTEL** manages special tags which make the link between the fields of the 3270 screen and the data on the corresponding HTML page which the **VIRTEL** HTTP server sends to the browser. See Virtel Web Access User guide for further information.

The information in the “map” is used by **Virtel SR** to construct the HTML and **VIRTEL** tags which convert the 3270 screen to HTML format.

The second section of the **Virtel SR** user interface, shown below, is dedicated to this function:

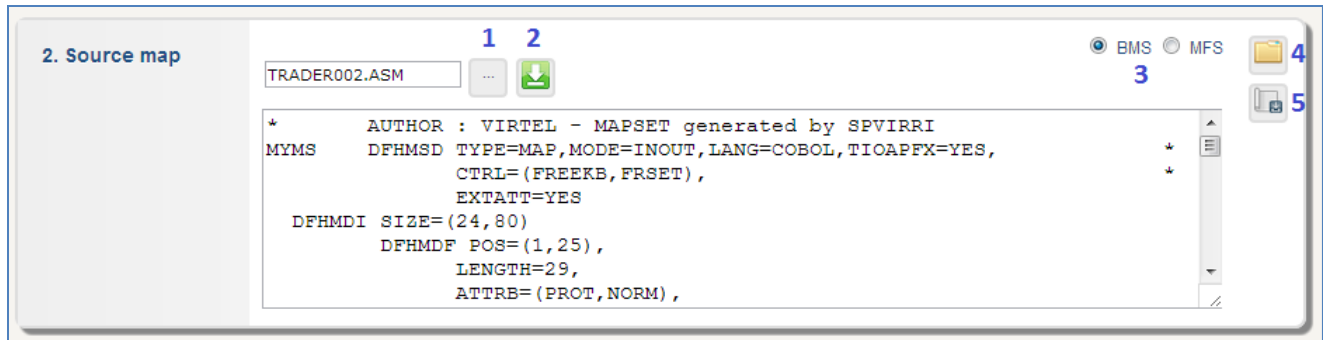


Figure 5-1: VirtelSR source map

Within this section, the following areas (denoted by their number in the screen shot above) provide the following actions:-

1. Switch between input and select field,
2. Load a map,
3. Choose between BMS and MFS formats.
4. Open the interface for the list of the files in the “Map” folder.
5. Open VIRTEL Web Access with **Virtel SR** functionalities to capture a map.

Sections 6.1 through **Erreur ! Source du renvoi introuvable.** describe how to capture a CICS BMS map from a screen. The specific case of the MFS (IMS) maps is the subject of Section 5.4.

5.1 CAPTURING A 3270 SCREEN

Press the “Capture the map” button (See number 5). This starts execution of the mainframe application in a browser window. The application will be displayed in the browser in the VIRTTEL Web Access mode. Navigate to the screen you wish to capture. Note that two additional icons will appear on the Virtel Toolbar, one of them being the Capture button/ICON. When you click on the icon circled in the screenshot below, a BMS map for the displayed 3270 screen will be created.

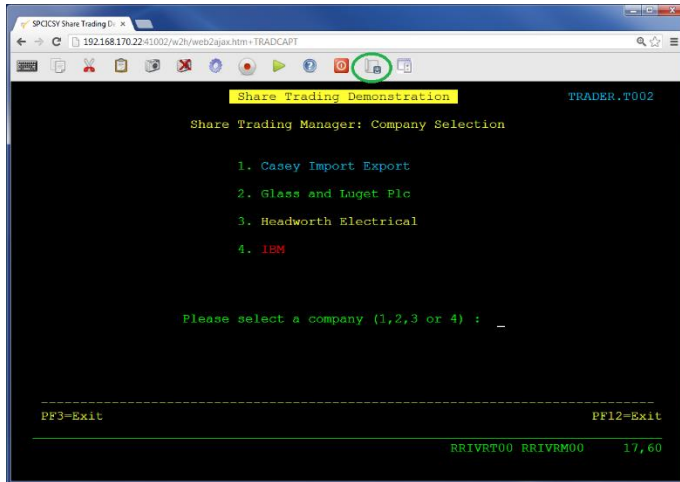


Figure 5-2: Virtel SR screen capture page

You can then navigate through the mainframe application until you reach the desired screen. Repeating the process to generate a BMS map for each screen you wish to modernize.

5.2 GENERATING THE MAP

When you click the capture ICON (encircled in green in the screenshot above) a dialog box will appear which allows you to save the generated BMS map file with a particular name.

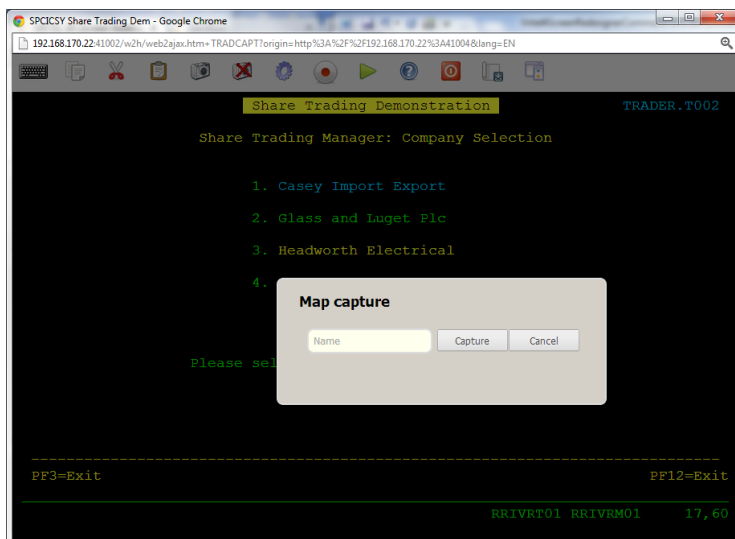


Figure 5-3: Dialog box for map capture

Enter the chosen name for this particular map or 3270 display. **Virtel SR** will save the map to the folder “Map” of your project and display a report showing the successful transfer of the file(s).

*Note: As an alternative to generating a map using **Virtel SR**, you can also download the source file of an existing BMS map from the mainframe and save it directly in the Map subdirectory for your project. If you have the current map source, this may be preferred as the name of the fields in the map will be more meaningful; however, care must be taken as use of a prior version of the map source may not match the application screen layout currently in use and can therefore cause unpredictable results.*

5.3 LOADING A SELECTED MAP

- Return to the **Virtel SR** interface and in Section 2:-
- Click on the three dots “...” to the right of the map name input field. The input field changes to a selection list
- Open the selection list and locate the name of the map file which you saved in the previous step. Click on this name.
- Press the LOAD ICON to load the map,
- The source map area will be populated with the MAP definition.

5.4 MFS MAPS

If the application you want to modernize has been developed on IMS with MFS maps, there are two possibilities:

5.4.1 MFS MAPS WITH SOURCE

If the MFS map source is available. Select MFS format in the top right corner and Load the map accordingly:

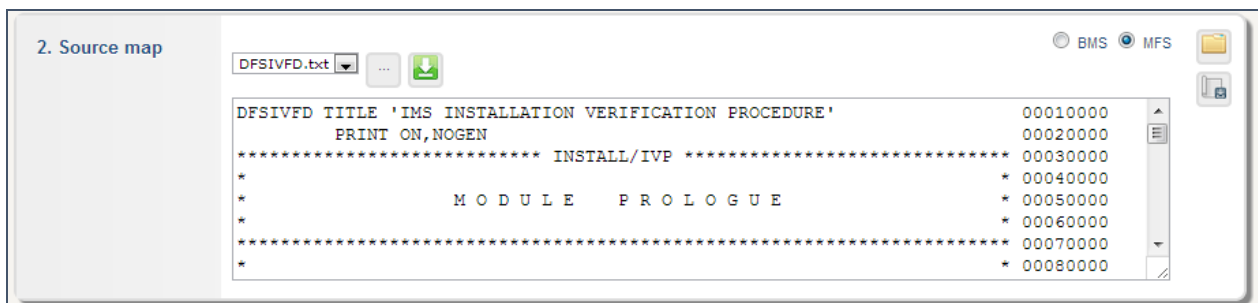


Figure 5-4: Virtel SR– MFS map source code

5.4.2 MFS MAPS WITHOUT SOURCE

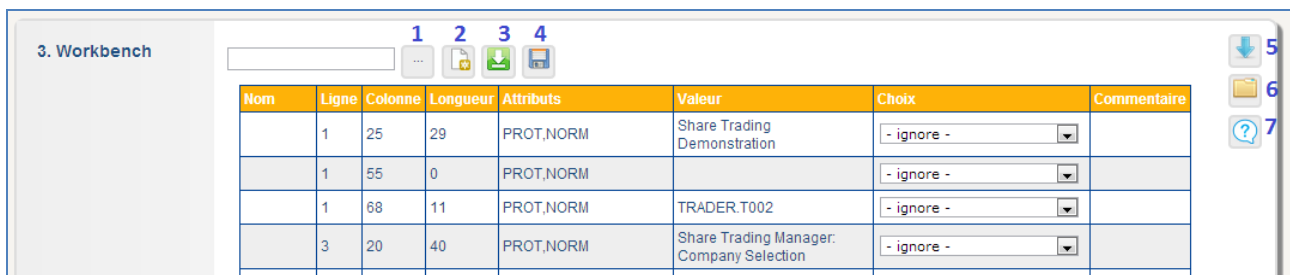
If the MFS map source is not available, it is suggested that you request **VIRTEL** to build the description with the BMS format (see sections 7.1 through 7.4 above). This will allow you to create a WORKBENCH (Section 3) file which is safe to use to create your new presentation templates. These will work within an IMS environment. You must set the radio button to the “BMS” option.

6 WORKBENCH SECTION

The source map captured / or provided in the SOURCE map section contains the base information for coding the Virtel HTML tags of a web page. Each field of the 3270 screen is defined within the WORKBENCH display by the following information:

Name	The name of the 3270 field.
Line	The 3270 screen row where the field starts.
Column	The 3270 screen column where the field starts (this is the column of the attribute byte which precedes the data)
Length	The length of the 3270 field (excluding the attribute byte)
Attributes	The field attributes. These determine, among other things, whether the field is an input field.
Value	The initial value of the field displayed on the 3270 screen.

Virtel SR extracts this information for each field in the map and presents it in the form of a table, together with a “Choice” column which allows you to specify how you want each field to be processed by **VIRTEL**. This table is called the “Workbench” and it is displayed in the third section of the **Virtel SR** user interface. An example is shown below:



Nom	Ligne	Colonne	Longueur	Attributs	Valeur	Choix	Commentaire
	1	25	29	PROT,NORM	Share Trading Demonstration	- ignore -	
	1	55	0	PROT,NORM		- ignore -	
	1	68	11	PROT,NORM	TRADER.T002	- ignore -	
	3	20	40	PROT,NORM	Share Trading Manager: Company Selection	- ignore -	

Figure 6-1: VirtelSR workbench

Within this section, the following areas (denoted by their number in the screen shot above) provide the following actions:-

1. Switch between input and select field,
2. Create the workbench based on the map,
3. Load the workbench,
4. Save the workbench to the folder “Workbench” of the project,
5. Navigate to the bottom of the workbench,
6. Open the interface for list of the files in the folder “Workbench”,
7. Display/hide the tooltips for the workbench.

6.1 GENERATION

With the source code for the map still displayed in the “Source map” window, press the “Create from map (2)” button to create the workbench table. When **Virtel SR** has created the workbench, you will see the columns as described above. One row is generated for each 3270 field described in the map. All cells in the workbench can be modified. To change the value of a cell, click on the cell and overtype the value, then click OK.

Rows in the workbench may be added, duplicated, copied and pasted, or deleted by right clicking on any cell in the row and selecting the desired operation from the context menu. The last column of each row allows you to enter notes or comments.

6.2 CHOICE COLUMN

Each row of the workbench has a “Choice” column. This column governs how Virtel will process this particular 3270 field represented by the “Workbench” table row. The choice column is a drop-down menu with the following options:

- ignore -	Do not process this field
literal	A constant field whose value is the contents of the “Value” column. If the literal contains the characters ‘<’, ‘>’, ‘”’, and ‘&’, they will be replaced by the corresponding HTML character-entity sequence when the page template is generated. This ensures that these characters are displayed as-is and not interpreted as HTML.
area definition*	Create a VIRTEL tag which relates a name to a 3270 screen area defined by the 3270 screen position.
COPY-FROM	Copy the 3270 field contents from the specified position on the screen.
named COPY-FROM	Copy the 3270 field contents using field name. An additional tag will be generated which associates the name to the screen position. This name can have been defined manually, either by the area-definition or by the column-definition choice (in this case, the nth choice (named COPY-FROM) will refer to the nth item of the column). If the name has not been previously defined, the definition will be generated automatically for the area.
HTML code	The contents of the Value column is inserted into the generated page as HTML code. This choice is similar to literal, except that the characters ‘<’, ‘>’, ‘”’, ‘&’ are not displayed but are interpreted according to HTML syntax by the browser.
GENERATE-HTML	Automatically recognizes the field attributes (input, output, color, and highlight) and generates the appropriate HTML code
named GENERATE-HTML	Similar to GENERATE-HTML but specifies 3270 field by name. An additional tag will be generated which associates the name to the screen position.

PFK button generates a program function key button.
This option modifies the meaning of certain cells in the workbench:
 The **Name** column contains the name of the key (PF1-PF24, PA1-PA3, CLEAR, or ENTER) which will be sent to the mainframe application when the button is pressed. The **Value** column contains the text for the button.

Notes:

1. In the test mode, the names of the input fields have already been allocated by VIRTEL Web Access, and will not be the same as those expected in your test template. In standard mode names will be adhered to.
2. Do not use function keys which cause application disconnection. The disconnect button on the base page, myappl.htm, is meant for this purpose.

At the right of the WORKBENCH section, the “Help (On/Off)” button allows you to toggle between display and non-display of tool tips. The meaning of the data in the various columns of the workbench varies depending on the value selected in the **Choice** column. When you hover the mouse over a “Choice” cell the tool tip explains how the cell is interpreted by the selected choice.

The choices listed are the default choices. Advanced users can also create and add the own choice options.

6.3 EXAMPLES

Here are some suggestions for your first tests:

6.3.1 COPY AREA EXAMPLE

- Within the WORKBENCH identify the rows you want to be displayed.
- For each row open the selection list in the **Choice** column and select **COPY-FROM**.

6.3.2 ENTERABLE “UNPROT” AREA EXAMPLE

- Within the WORKBENCH table identify a row which is “enterable” i.e. does not have “PROT” in the attribute column.
- Open the selection list in the **Choice** column and select **GENERATE-HTML**.

These selected areas can now be used to generate the template, which is the subject of Section 7, template generation. Before looking at how to generate the HTML templates from the selected WORKBENCH areas we will review how to save and restore a WORKBENCH section.

6.4 WORKBENCH MANAGEMENT

The workbench can be saved in the **Workbench** subdirectory of the **Virtel SR** directory. Two navigation arrows, “↓” situated at the top of the workbench and “↑” at the bottom of the workbench, allow you to quickly reposition the **Virtel SR** window.

6.4.1 SAVING THE WORKBENCH

- In the workbench filename input field, type the name under which you wish to save the workbench (for example, “demo.json”).
- Press the “Save” button.
- A confirmation message appears.

*Note: the workbench data is saved in JSON format. You may use any extension for the filename, however if you do not code an extension then **Virtel SR** adds the default extension **.json**.*

6.4.2 OPENING A SAVED WORKBENCH

- Click on the three dots (“...”) to the right of the workbench filename input field.
- The input field changes to a selection list.
- Open the selection list and find the name of the workbench file which you previously saved.
- Press the “Reload” icon.

6.4.3 EXAMPLE WORKBENCH

A sample workbench is supplied in “\samples\workbench samples\basic\basic.json” in the **Virtel SR** components. It uses the standard choices. This file is a separate zip download found within the Virtel SR FTP directory.

6.4.4 WORKBENCH IMPORT AND EXPORT

Some users find it easier to modernize their project using a spreadsheet program. If there are a lot of fields to modernize this may be the solution to consider. To modify a workbench with a spreadsheet, you can export the WORKBENCH by clicking the related **Virtel SR** export icon. The file will be exported and can then be opened by a relevant spreadsheet program. The EXPORT button is identified by the GREEN Square in the snapshot below.

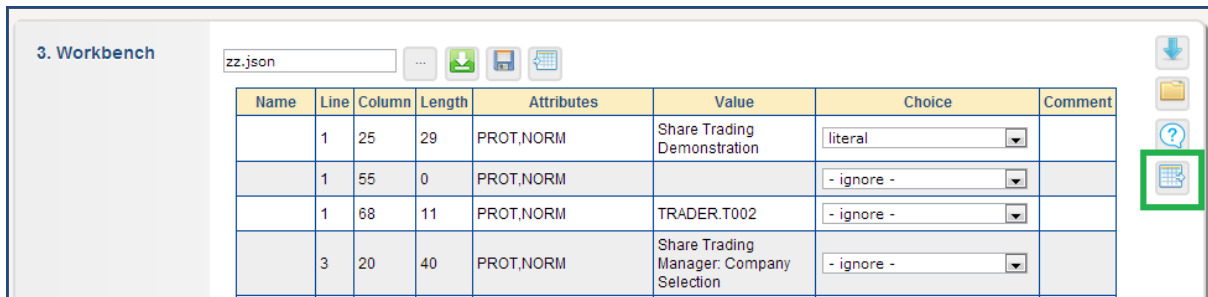


Figure 6-2: Export button

The default export format is "xml". You can modify it by accessing the settings panel (click on the setting icon at the right of the first **Virtel SR** section). An alternative choice is "txt".

You may choose the format depending on your spreadsheet version (later versions can handle the "xml" format) or on the specific characters you are using (with "txt" format, you might need to open the spreadsheet, then import data after selecting the file origin in Unicode UTF-8). With the spreadsheet you can freely work on the workbench cells.

Once the file has been modified, you can import it back to **Virtel SR** by dragging the spreadsheet file and dropping it to the **Virtel SR** import icon.

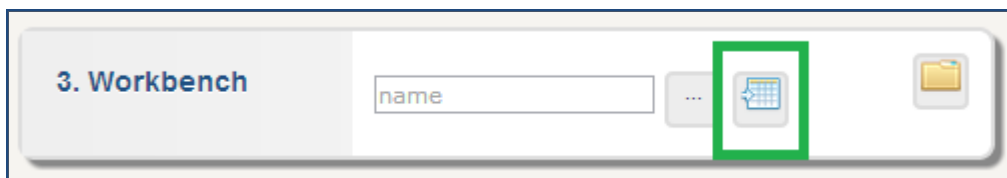


Figure 6-3: Import button for workbench import

6.4.5 RESCUE

If you close the **Virtel SR** interface without having saved a modified workbench, a popup will propose you to remain on the HTML page. If you leave without saving, in the next **Virtel SR** session, a rescue icon will appear.

Clicking on it leads to load the latest version of your workbench.

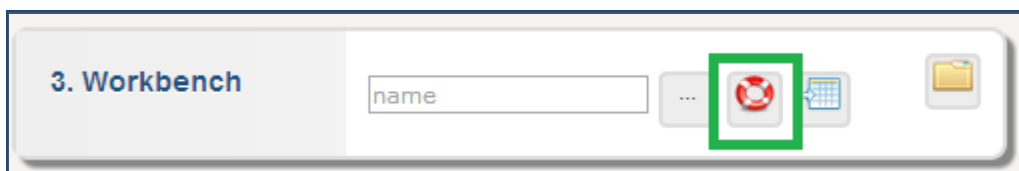


Figure 6-4: Rescue button

7 TEMPLATE SECTION

The purpose of the **Virtel SR** utility is to generate a “page template” from 3270 fields. A page template is an HTML page containing special **VIRTEL** tags which dynamically generates HTML code and content using the 3270 screen image of the mainframe application.

Virtel SR uses the information in the WORKBENCH section to generate a page template in the TEMPLATE section. The figure below shows the “Template” section of the **Virtel SR** user interface. If necessary, you can reposition to the Template section by pressing the “↓” arrow situated at the top of the workbench.

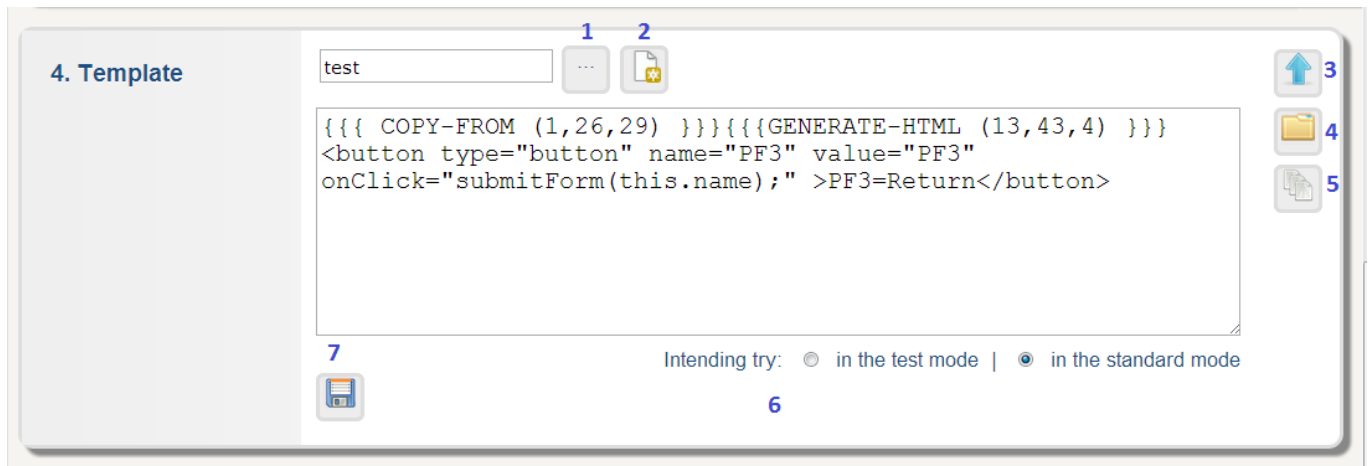


Figure 7-1: Virtel SR page template

Within this section, the following areas (denoted by their number in the screen shot above) provide the following actions:-

1. Switch between input and select field,
2. Create the template based on the workbench,
3. Navigate to the top of the workbench,
4. Open the interface for the list of the files in the folder “Template”,
5. Open the batch interface
6. Choose the mode (test/standard),
7. Save the template to the folder “Template” of the project.

7.1 CREATION

To create a page template using the contents of the current WORKBENCH, press the “Create” button (denoted by the number 2 in the screenshot above). The generated HTML code will appear in the template edit box (see figure above).

7.2 CONTROLS

While generating a template, **Virtel SR** checks if the information is consistent. If not it displays a corresponding error message.

3. Workbench

Nom	Ligne	Colonne	Longueur	Attributs	Valeur	Choix	Commentaire
...	- ignore -	...
23	1	79	PROT,NORM			- ignore -	
24	1	8	PROT,NORM	PF3=Exit		PFK button	
24	71	9	PROT,NORM	PF12=Exit		- ignore -	

4. Template

Choice "PFK button", line 24, column 1. The fieldname is missing.

```
<button type="button" name="" value="" onClick="submitForm(this.name); "
>PF3=Exit</button>
```

Intending try: ☒ in the test mode | ☐ in the standard mode

Figure 7-2: Display of error messages after the template has been generated

If the user clicks on an error message, the related workbench line is highlighted.

7.3 SAVE IN TEST PAGE MODE

Virtel SR contains a development tool (“Test Page Mode”) to help you validate your workbench choices. During your initial testing, we recommend that you use this “test page” mode, which requires you to first save your page in a special format known as a test page.

To save as a test page:

- Enter the name of your page in the template name field.
- Check that the radio button “intending try in the test mode” is checked.
- Press the “Save” button (denoted by the number 7 in the screen shot in section 8.2).

The HTML code in the edit box will be saved as an HTML test page in the **Template** folder of your project.

*Note: You may use any extension for the filename, however if you do not code an extension then **Virtel SR** adds the default extension **.html***

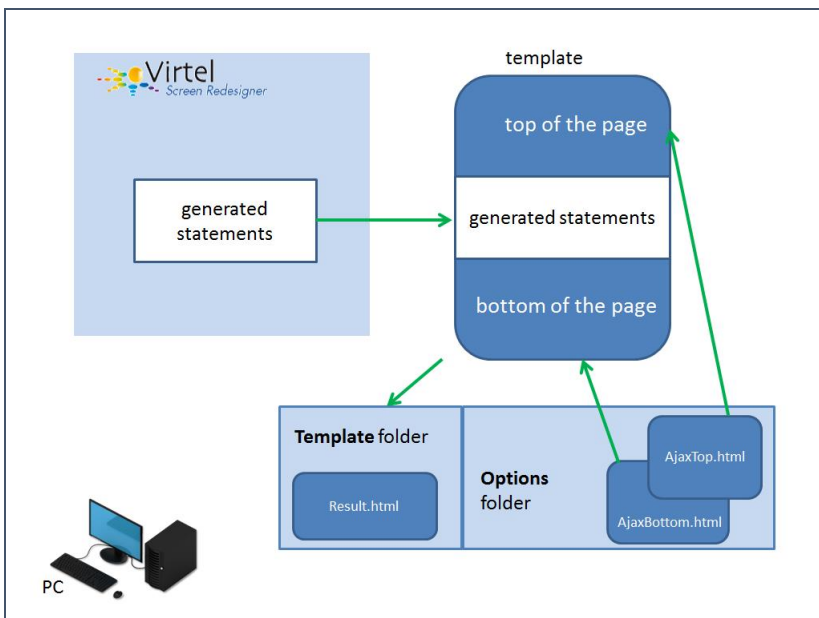


Figure 7-3: Virtel SR – Constructing a test page and saving it locally.

7.4 SAVE IN STANDARD PAGE MODE

When you have completed all the steps in “test page” mode (sections 9.1, 9.2, 9.3), you can move on to the standard mode of operation by saving the standard page template. In STANDARD mode, you page is saved as a sub-page within the default base page.

To save a standard page template:

- Go back to the step where you have created a page template (see section 7.1 above)
- Enter the name of your page in the template name field.

It is strongly recommended that you use different names for the test page and for the standard page. For example, you can use `TestDEMOM1.html` for the test page, and `PGDEMOM1.html` for the standard page.

- Press the “Save for standard mode” button.
- The HTML code in the edit box will be saved as an HTML standard page template in the **Template** folder of your project.

Note: You may use any extension for the filename, however if you do not code an extension then **Virtel SR** adds the default extension **.html**

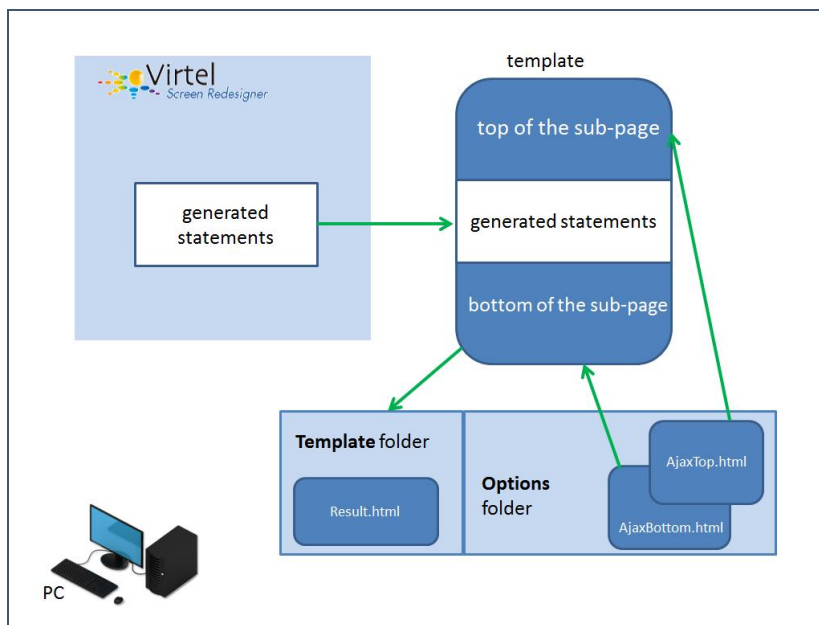


Figure 7-4: Virtel SR - Constructing a standard page template and saving it locally

7.5 BATCH INTERFACE

While developing a project it can help to generate the templates and upload them in a batch mode.

Suppose you decide to modify a way of particular WORKBENCH choice works and create a new choice snippet called “mySnippets.js”. Maybe your modification doesn’t need any modification in the WORKBENCH field selections but needs to be included when generating and uploading the templates to the **VIRTEL** directory. The “Batch” interface processes enables a batch generation/upload. It opens with the list of the project’s WORKBENCH’s. At this point you can select all the workbenches that you want to include you modified choice. At the bottom of the table, two action buttons are available; one to generate the templates and another to uploaded.

8 TESTING

Your page template has now been created locally on your workstation. The next step is to test how **VIRTEL**, functioning as an HTTP server, merges the page template with the mainframe application data and displays the results in your browser.

8.1 UPLOAD TO VIRTEL DIRECTORY

Before you can test the newly-created page template you must first upload it to **VIRTEL** where it will be stored in the client directory (CLI-DIR) or your designated modernization directory.

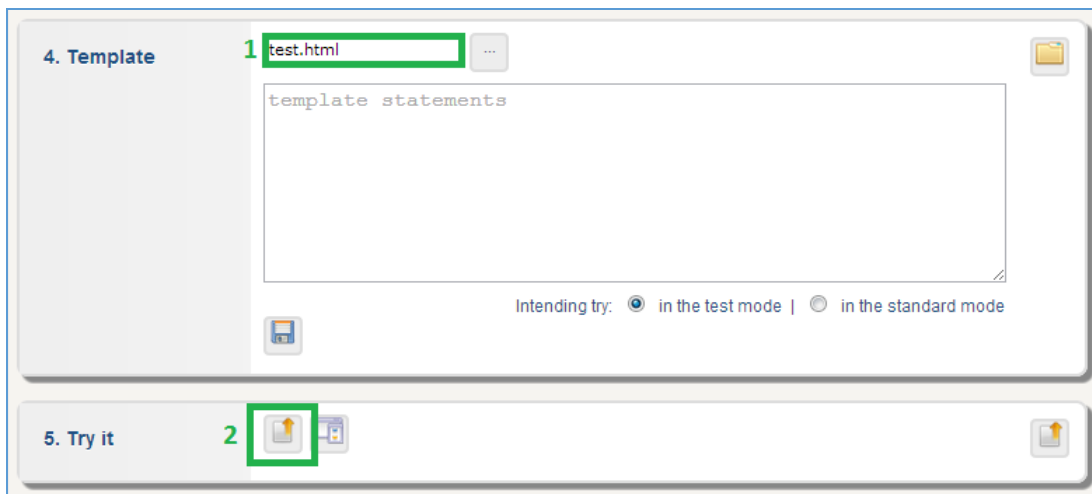


Figure 8-1: Virtel SR facility for upload

Once a name has been filled in (box 1 above), a button is displayed in the "Try it" section of the interface (box 2 above) to launch the upload of this template from the Template folder to the **VIRTEL** directory dedicated to the modernization project. A window opens with an upload report. The message:-

```
OK : <CLI-DIR > RETURN CODE IS: 00
```

Indicates a successful upload.

8.2 MODES OF OPERATION

8.2.1 TEST MODE

In “test page” mode, **VIRTEL** sends a complete HTML page to the browser and this is rendered accordingly.

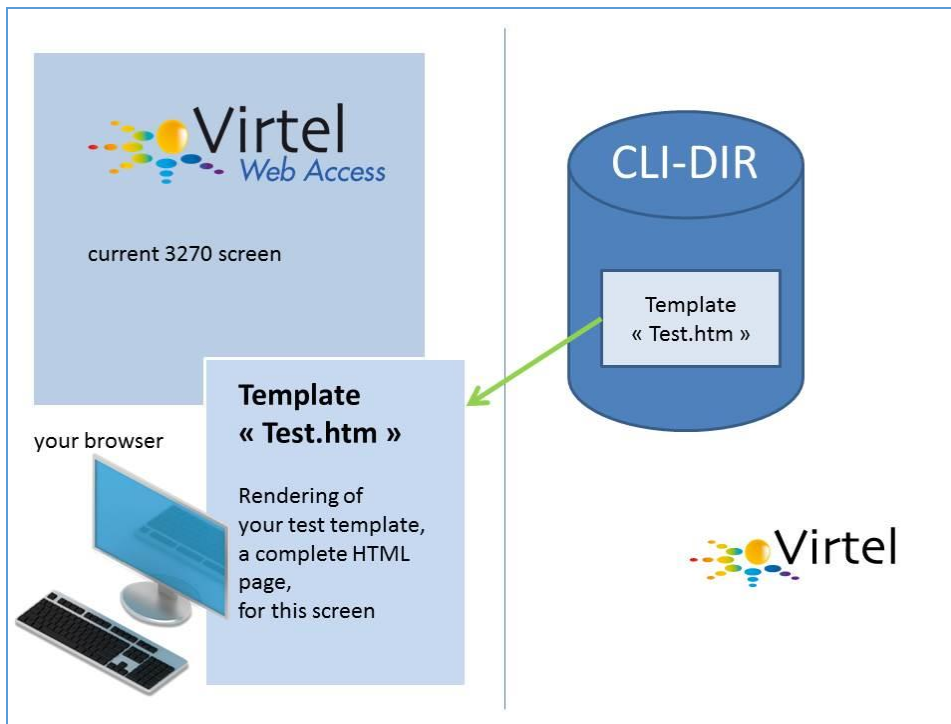


Figure 8-2: Saving the HTML test page in CLI-DIR for display on browser

8.2.2 STANDARD MODE

In the standard mode of operation, **VIRTEL** first sends the base page (default supplied is **myAppl.htm**) to the browser (1). At this stage, no connection has yet been made to the mainframe application. Subsequent exchanges with the mainframe application are made in the form of “Ajax requests” (2). In response to these requests, **VIRTEL** sends the page template combined with the application data in the form of HTML code. This HTML code is dynamically inserted into the base page as a sub-page. (3). In STANDARD MODE, your generated template is treated as a sub-page of the default base page. In production this is how Virtel works.

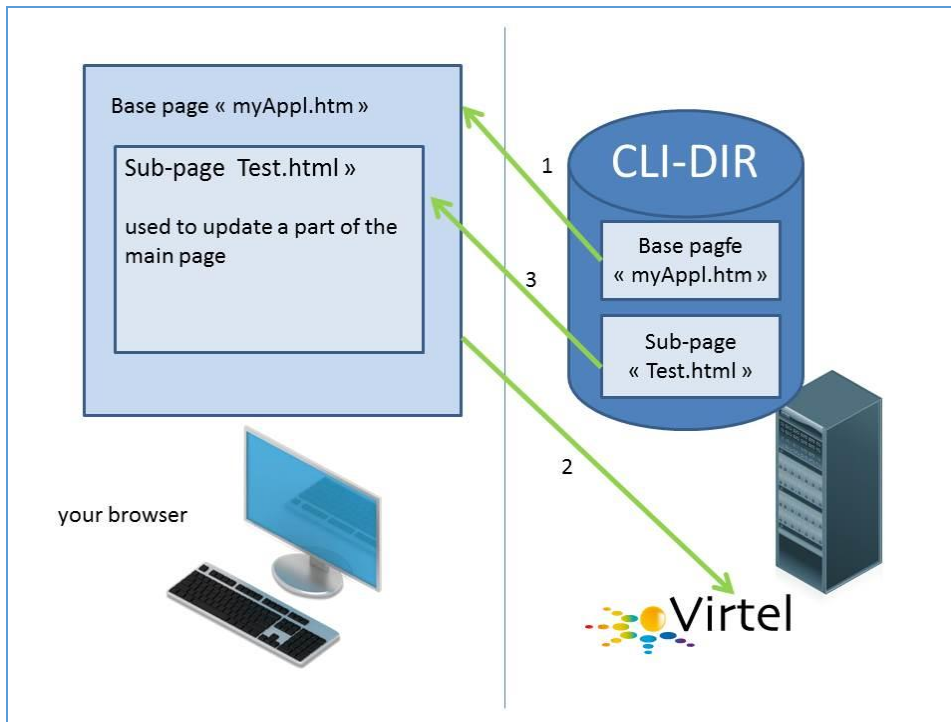


Figure 8-3: Saving the HTML page template in CLI-DIR for display on browser

9 TRY IT SECTION

To perform both “TEST” mode and “STANDARD” mode tests you must:

- Upload the page template to a **VIRTEL** directory.
- Send the necessary HTTP request from your browser.

The final section of the **Virtel SR** user interface, called the “TRY IT” section, contains controls to help you to carry out these actions:

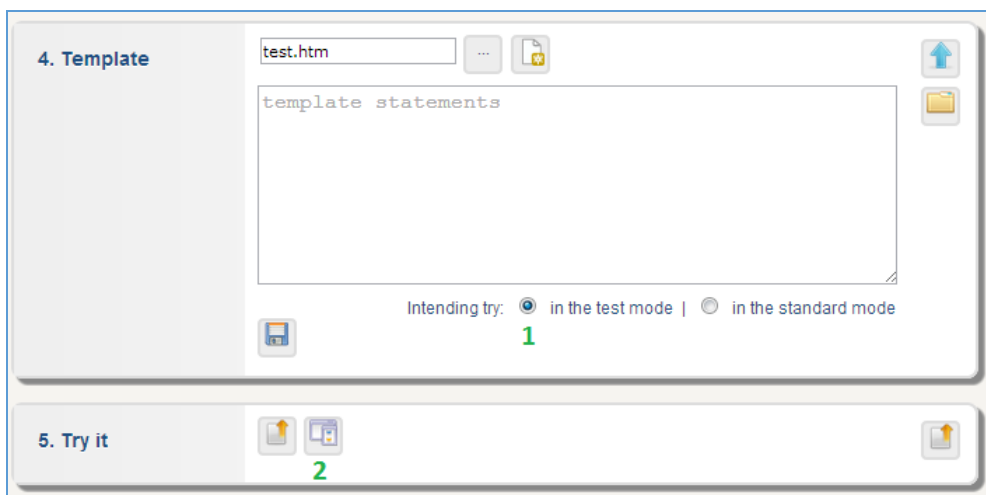


Figure 9-1: Virtel SR – save area for test mode

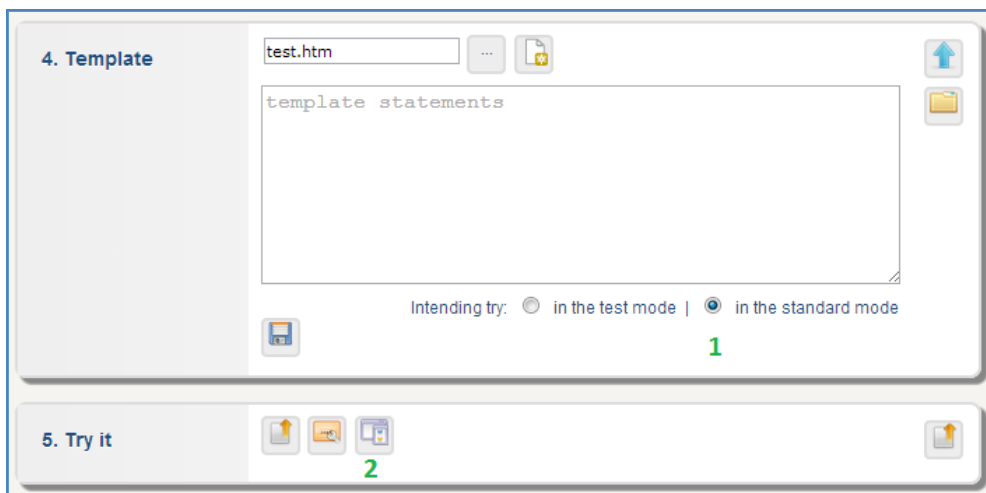


Figure 9-2: Virtel SR – Save area for standard mode

In both screenshots above, you choose using the radio button at the bottom of section 4 to select a mode of operation. Depending on the selected mode, a different process will be performed after clicking the “Try” button (Number 2).

9.1 TESTING IN “TEST” MODE

To test a page in “test” mode, you should open a classic 3270 window in the browser, navigate to the chosen screen, then open a second browser window and display the same application screen using your new page template.

You can use the capture window which you opened in section 5.1 above. If necessary, you can open the window again by pressing the button marked “Try the test page on Capture window”.

- Click the “view” icon circled in the screenshot below: a dialog popup appears.
- Type the name of the HTML test page which you just uploaded to VIRTEL.
- Press the “OK” button.
- A new window will open, showing the same 3270 screen presented using the specified page template.

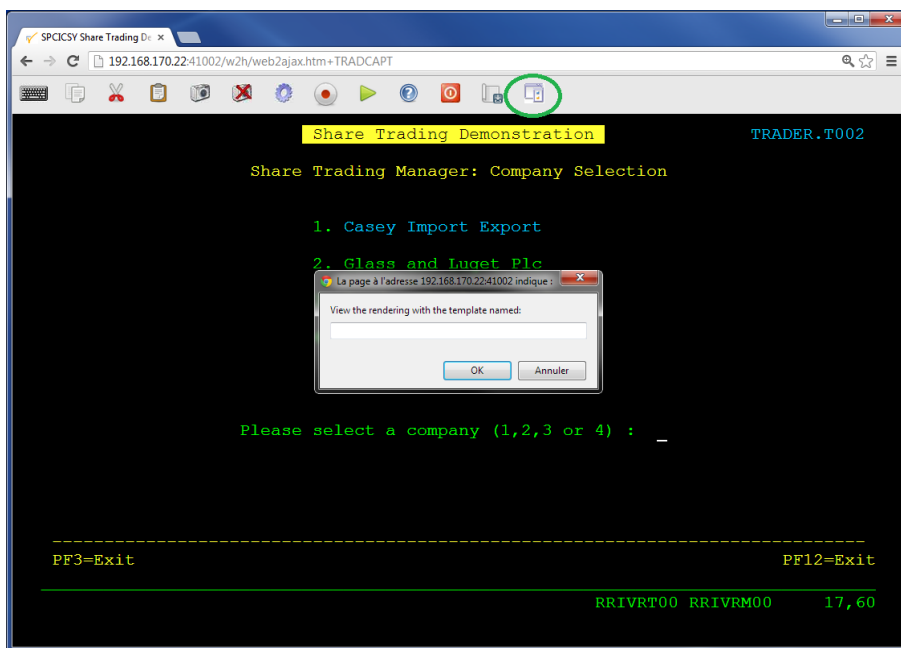


Figure 9-4: Virtel SR Screenshot with the “view” icon (circled) and its dialog popup

Note: This mode is intended only for rapid viewing of a page using the test template. The ability to navigate within the application is disabled in the test template (navigation can be done using the capture window).

9.2 TESTING IN “STANDARD” MODE

Before making a test using the standard mode of operation, you must first create a scenario and then assemble and link-edit it into the **VIRTEL** load library, as described in section 2.3 above.

9.2.1 BASE PAGE

Press the button marked “Try the standard mode” in the **Virtel SR** user interface. **Virtel SR** opens a new browser window using a URL derived from the **Virtel SR** configuration. The default base page (myAppl.htm) is displayed:

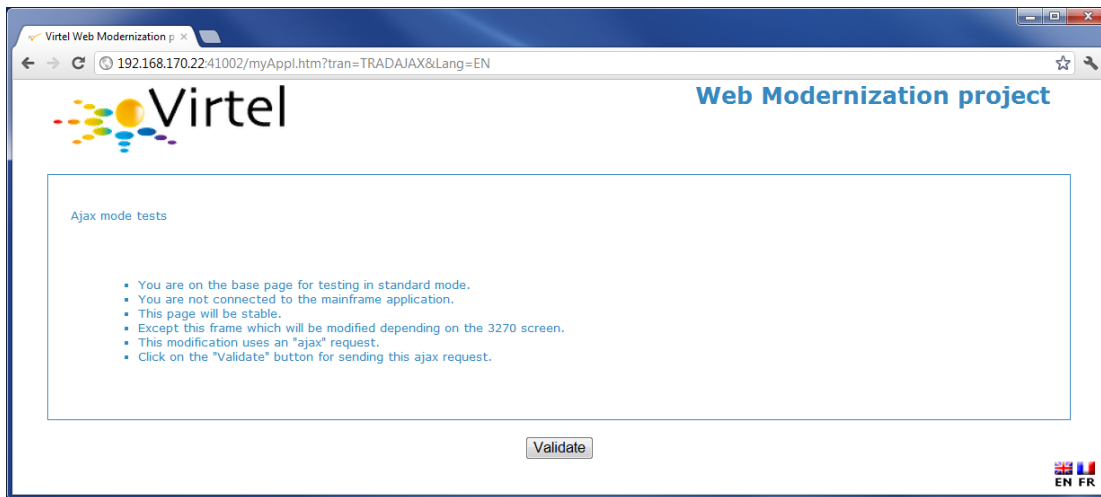


Figure 9-5: Base page for standard mode of operation

9.2.2 NAVIGATION

Press the “Validate” button on the base page to connect to the mainframe application.

Each screen presented by the mainframe application is first analyzed by the XXXXAJAX scenario (see section 2.3) which selects the appropriate HTML sub-page template. This template is then merged with the mainframe application data to form what is referred to as a “sub-page”. The sub-page is not a complete HTML page, but contains HTML code which is dynamically inserted into the base page in the browser window. Here is an example of the application screen TRADER 002 displayed using sub-page TRADER002.html. After modernization the page looks like this:-

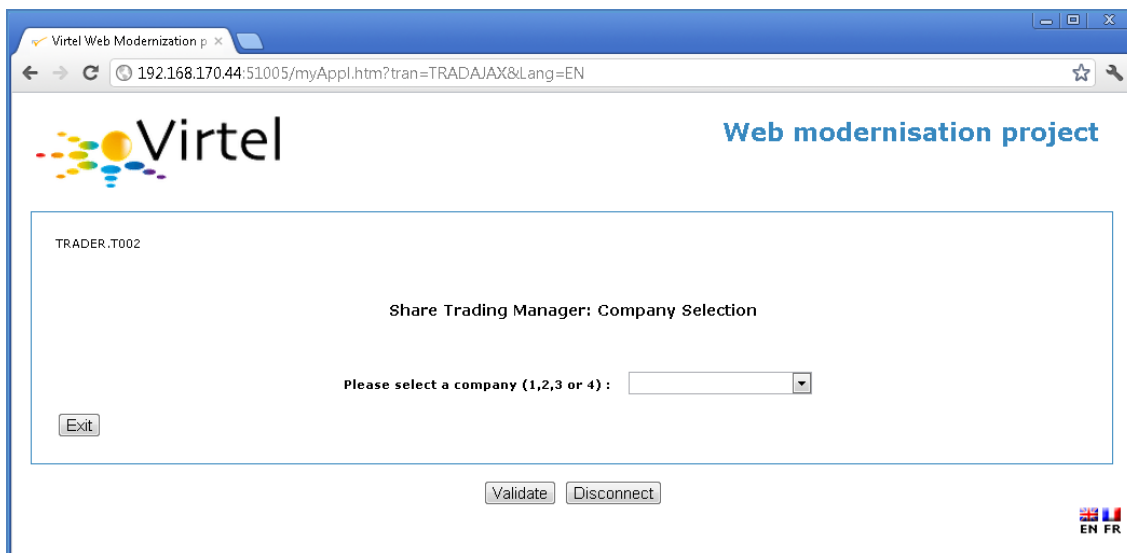


Figure 9-6: Base page with sub-page “TRADER002.htm” loaded after modernization

In this example we have modernized several fields of the 3270 display. The presentation now includes a select box. The modernization presentation has also reduced the area required to display 3270 information. One of the advantages of modernization is the ability to combine several classic 3270 displays into one GUI presentation. This reduces the amount of navigation a user has to go through. Notice also that the page name, TRADER.T002 has been relocated to the left hand side. In modernizing a project sometimes it is best to “draw” or map out the before and after presentation so that the relevant fields can be identified from the original source. These can then be mapped, positioned or transformed accordingly to achieve the expected GUI presentation.

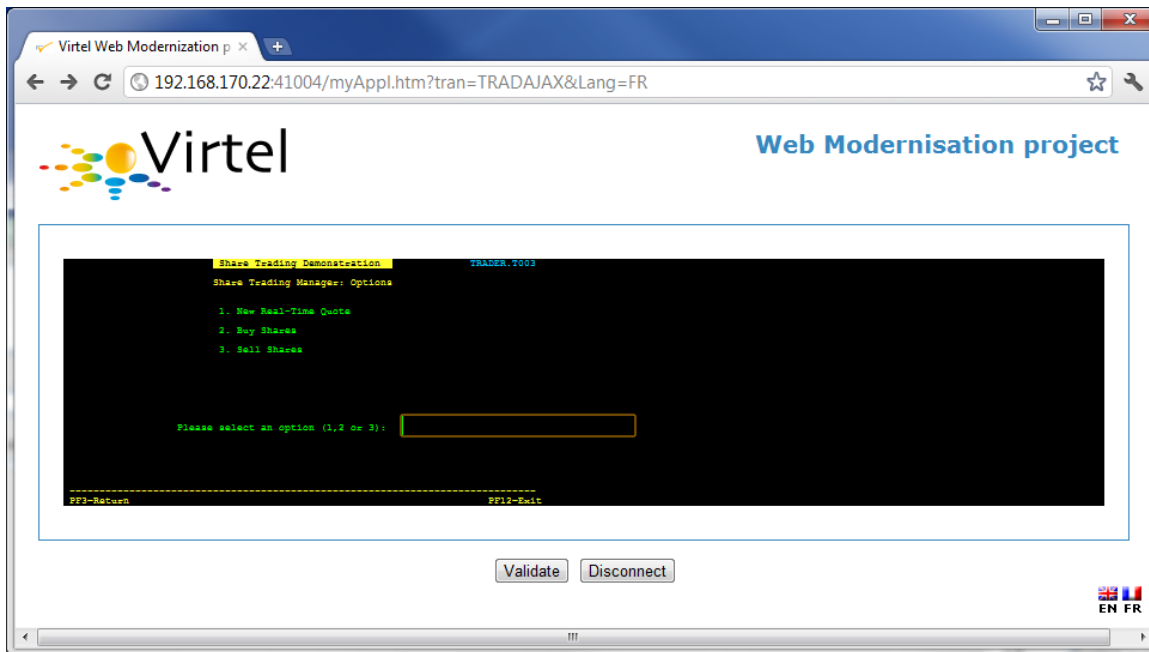


Figure 9-7: Base page with classic sub-page loaded and before modernization.

Note: . If the XXXXJAX scenario cannot identify with the application screen presented, then the scenario loads a default sub-page **VSRAjax.html** which displays the data in 3270 classic format. For this option to work be sure that you have uploaded the template “sources\myVIRTEL\CLI-DIR\VSRAjax.html” to your target **VIRTEL** directory. Failure to do so will have unpredictable results

9.3 VIRTEL DEVELOPMENT REVIEW

Virtel SR aims first and foremost to automatically generate the tags needed to allow **VIRTEL** to make the link between the mainframe application and its web presentation. The interface enables also to structure and enhance the template through HTML tags:

9.3.1 HTML CODE CHOICE

The HTML code can be redefined within the workbench itself. By inserting additional rows into the workbench and selecting “HTML code” in the **Choice** column, you can include your own HTML code in the **Value** column, and this code will be included in the generated page template.

Name	Line	Column	Length	Attributes	Value	Choice	Comment
	1	25	29	PROT,NORM	Share Trading Demonstration	- ignore -	
	1	55	0	PROT,NORM		- ignore -	
	1	68	11	PROT,NORM	TRADER.T002	- ignore -	
					<h1>	HTML code	
	3	20	40	PROT,NORM	Share Trading Manager: Company Selection	COPY-FROM	
					</h1>	HTML code	
	6	26	2	PROT,NORM	1.	- ignore -	

Figure 9-8: Example of inserting HTML code

This method allows you to refine the generated HTML code while still using all the functionality of the workbench. You may use any valid HTML command.

9.3.2 TESTING

While you are working on a page, you can keep the same testing context so that you can view the modernized page while you make modifications.

TEST MODE

By keeping the capture screen open (see section 9.1 above), you can request “View with test template” each time you upload the test page to the target **VIRTEL** directory.

STANDARD MODE

From your currently displayed test screen (see section 9.2.2), choose an operation such as “Previous”, if provided by your application, which allows you to return and redisplay the previous screen once you have uploaded the new version of your page template into VIRTEL’s CLI-DIR directory.

If your application does not allow you to return to the previous screen, you must press the “Disconnect” button, then restart the test procedure. Be sure to always press “Disconnect” before closing the browser window, in order to free up the current session with the mainframe application.

*Note: It may be necessary to clear your browser’s cache to ensure that you are using the latest version of your page template. With the version of **Virtel SR** using “NW.js”, you might open the « Developer Tools » panel while developing, and, and check the setting « disable cache (while DevTools is open) ».*

10 TROUBLESHOOTING

10.1 VIRTEL LOG

If you do not succeed in connecting to the mainframe application via the browser, the **VIRTEL** log will contain an error message indicating the cause of the problem. To display the **VIRTEL** log, use SDSF (z/OS) or a similar utility and select the JESMSGLOG dataset of the VIRTEL started task. A list of messages and their meanings may be found in the VIRTEL Messages and Codes manual.

10.2 CLIWHOST.HTM

One HTML page has the same name as the **VIRTEL** entry point: **CLIWHOST**. In accordance with standard **VIRTEL** principles of operation, whenever the user disconnects from the mainframe application, **VIRTEL** will display this page CLIWHOST.htm as the default page.

At the beginning of your project, enter the IP address and port for the client line into a browser window:

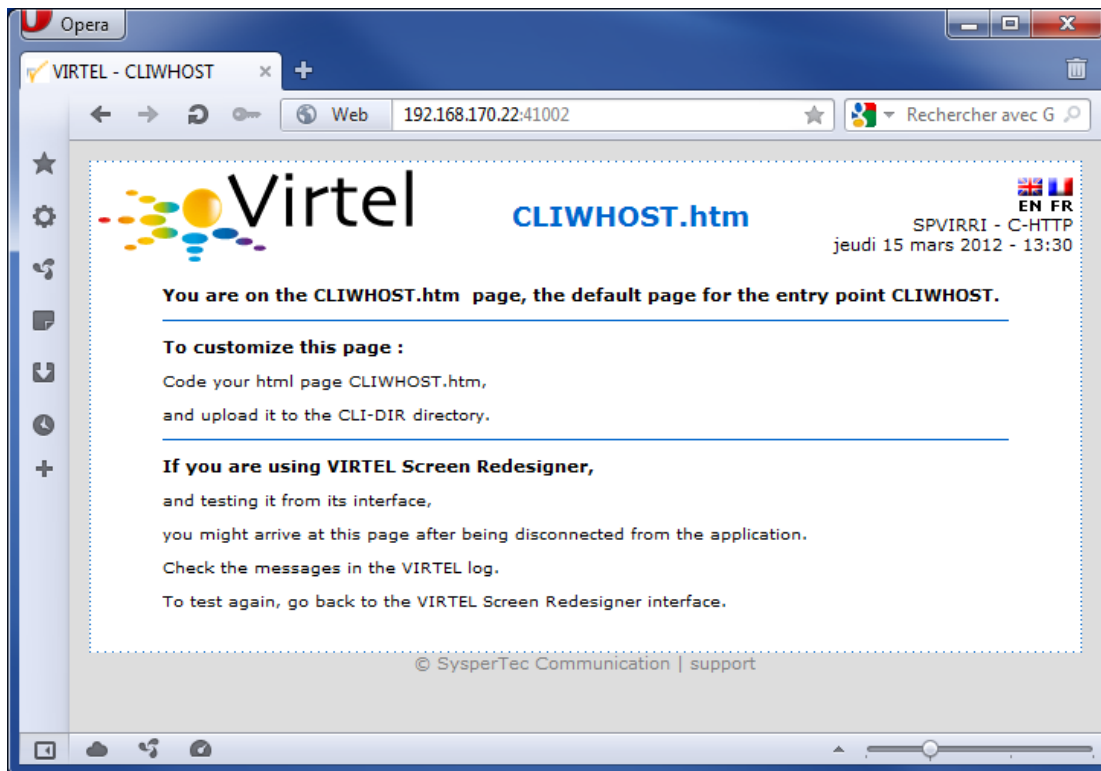


Figure 10-1: Page CLIWHOST.htm

When this default page is displayed, you should close the browser window and start again from the “Try it” section of the **Virtel SR** interface screen if you wish to perform further tests, or save the URL generated by **Virtel SR** as a favorite.

10.3 RESPONSE FORMAT ERRORS

The STANDARD mode of operation uses Ajax requests which expect a response to arrive in the form of an HTML sub-page. In the event of a problem in the mainframe application, the CICS connection, or the configuration of **VIRTEL**, the response might not be returned in the expected format. In this case, one of the following messages may be displayed in the browser:

- Form is missing in request response
- Request response is not an Ajax subpage

If either of these messages appears, refer to the **VIRTEL** log (section 10.1) to determine the cause of the problem.

11 MODERNIZED APPLICATION CUSTOMIZATION

The basic functions of **Virtel SR** can be customized to facilitate the process of screen modernization using **VIRTEL**.

11.1 STATEMENTS THAT MAY BE CUSTOMIZED

There are two kinds of statements which may be customized:

11.1.1 THE INTERFACE SIDE

The folder “«Options» of your project contains files which control the behavior of the interface which can be customized.

11.1.2 THE MODERNIZED APPLICATION SIDE

Your sources folder «\VIRTEL\CLI-DIR » of your project contains the standard files (HTML, js, css, images) used by your modernized application. They are intended to be customized. They are also required to be uploaded to the target **VIRTEL** modernization directory.

Here are some samples to illustrate how to customize these files according to your project’s requirements.

11.2 STYLES CSS

Your source file «\ VIRTEL\CLI-DIR\Visu.css » contains default definitions for the CSS rules for the project. It can be customized as much as desired. Here are some customization examples:

11.2.1 CHANGE THE BACKGROUND COLOR

The background color can easily be modified:

Open the file «\ VIRTEL\CLI-DIR\Visu.css» with a text editor.

You will find the following CSS rule :

```
html, body {
  background-color: #ffffff;
  color: #000000;
}
```

Figure 11-1: Rules of the style sheet dealing with background color

Replace the color **#ffffff** by the one you choose, using a color code from the standard HTML color table, and save your changes. Then upload this file to the target **VIRTEL** directory.

Note: It may be necessary to clear your browser’s cache to ensure that you are using the latest version of your page template.

11.2.2 REPLACE THE LOGO

The picture at the top of the page template can be replaced by a different image, such as a company logo. The logo display is processed by a CSS rule.

Open the file «\ VIRTEL\CLI-DIR\Visu.css » with a text editor. You will find a rule for the selector **#headBand** .

```
#headBand {
  width:90%;
  padding-left:5%;
  padding-right:5%;
  height:60px;
  background-image:url(../w2h/logovirt.gif); /* Image is customizable */
  background-repeat:no-repeat;
  background-position:left;
  margin:1em;
  font-family:Verdana, Arial, Helvetica, Sans-Serif;
  font-size: 2em;
  color: #3788BF;
  font-weight:bold;
  text-align:right;
}
```

Figure 11-2: CSS rule processing the logo display

You will notice that the image is referred to by the URL `../w2h/logovirt.gif`. You can replace this URL with the name of your image file (myImage.extension), and save the modified style sheet.

Then upload Visu.css and your image file to the target **VIRTEL** directory.

Note: It may be necessary to clear your browser's cache to ensure that you are using the latest version of your page template.

11.3 HTML MODIFICATIONS

When using both test and standard mode, the tags generated upon the workbench are wrapped by HTML tags which structure the page. These HTML tags can be customized. Here are some examples:

11.3.1 CHANGING THE HEADER (TOP OF PAGE)

The default web page presents a head band at the top of the page. The default title is **Web modernization project**. For modifying it, you must do the following:

IN TEST MODE

Open the file « \Options\PageTop.html » of your project with a text editor.

You will find the following tag with the mentioned text for the title:

```
<div id="headBand">Web modernization project</div>
```

Figure 11-3: HTML tags for the head band

You may replace this text with your wording, save this file. In order for the changes to take effect, you must recreate the template generated from the workbench file. This must be done for any template that uses the PageTop.html code. Then, the newly generated templates must be uploaded to the target **VIRTEL** directory as explained above.

IN STANDARD MODE

Open the file “\VIRTEL\CLI-DIR\myAppl.htm” with a text editor.

You will find the same tag that you see above. You must replace this text with your wording, and save this file. This saved file must be uploaded to the target **VIRTEL** directory, and the tests in the standard mode can then be continued.

Note: It may be necessary to clear your browser's cache to ensure that you are using the latest version of your page template.

11.3.2 CHANGING THE TRAILER (BOTTOM OF THE PAGE)

You can change the bottom of the page.

FOR THE TEST MODE

Open the file « \Options\PageBottom.html” with a text editor.

You will find the closing tag `</body>`.

You can insert above the `</body>` tag code similar to:

```
<div>Bottom of the page</div>
```

Figure 11-4: Example of statements for the bottom of the page

And save the file. To complete the change, follow the same procedure described in section 11.3.1 above for the test mode.

FOR THE STANDARD MODE

Open the file “\VIRTEL\CLI-DIR\myAppl.htm” with a text editor.

You will find the closing tag `</body>`. Insert ending code similar to what was described in the test mode above, and then follow the same procedure described in section 12.3.1 above for the standard mode.

11.4 JAVASCRIPT STATEMENTS

The source file “\VIRTEL\CLI-DIR\CustomFunctions.js” contains default definitions for the project JavaScript statements. *We adopted a coding methodology which separates the HTML tags and the JavaScript statements.*

The following is an example of building a template which includes a select box. The select box is built with HTML statements. But in a 3270 screen modernization project, we must separately check for the option of the box related to the value of the 3270 screen area. This is done by a JavaScript process. The following is an example of the JavaScript snippet required to perform this function.


```
function InitSelect(selectField,argVal){
  var initValue = argVal.replace(/\s/g,"");

  var index = -1;
  var val;
  var max = selectField.length;
  if ( max >= 0 ) {
    do {
      index++;
      val = selectField.options[index].value;
      if ( val == initValue ) selectField.options[index].selected = true ;
    }
    while(index < max-1 && val != initValue );

    if ( index === max-1 && val !== initValue ) {
      index++;
      selectField.options[index] = new Option(initValue,' ');
      selectField.options[index].selected = true ;
    }
  }
}
```

Figure 11-5: Sample of JavaScript statements to be adapted

In this sample, the highlighted statement processes the case where no select option value matched the 3270 screen initial value. This is the case of an empty value, inviting the user to select a choice. In the current example a blank will be displayed in the select box for this empty value. We can decide that it would be better to display text and replace the blank with the literal - select -.

In the JavaScript file, you will modify the statement as follows:

```
selectField.options[index] = new Option(initValue,'- select -');
```

Figure 11-6: Sample of modified statement

Save this file and upload it to the target **VIRTEL** modernization directory. You may then test your modernization templates in both “Test” and “Standard” mode of operation.

Note: It may be necessary to clear your browser’s cache to ensure that you are using the latest version of your page template.

11.5 NORMALIZING THE TAGS

In some 3270 application some information is consistently on every 3270 screen (e.g. a command field area, time, a title, textes related to PF keys, etc). We can design some elements of the project in order to set the related tags once only and then include them in other WORKBENCH templates.

Here are some samples:-

11.5.1 TITLE

Using the workbench, you can build the tags to render the title (copied from the 3270 screen and wrapped by <h1> tags).

3. Workbench

Name	Line	Column	Length	Attributes	Value	Choice	Comment
					<h1>	HTML code	
	1	25	29	PROT,NORM	Share Trading Demonstration	- ignore -	original
title	1	25	29	PROT,NORM	Share Trading Demonstration	named COPY-FROM	with name
					</h1>	HTML code	
...	- ignore -	...

4. Template

name

```

{{{DEFINE-DFHMDf-NAME (1,25,29) "title"}}}
<h1>{{{ COPY-FROM "title" }}}</h1>

```

Intending try:
☐ in the test mode
☒ in the standard mode

Figure 11-7: Sample of preparing tags to be normalized.

Then you can incorporate these resulting tags into your project by copying them from the area in the fourth section of the **Virtel SR** interface, and paste them to your sources files « \ Options\PageTop.html » and « \ Options\AjaxTop.html ».

Therefore the tags will always be added to the template while saving it (cf 7.3 and 7.4).

11.5.2 BOTTOM OF THE PAGE

In a similar way, you can incorporate tags for the bottom of the page by updating “\myProject\Options\PageBottom.html” and “ \myProject\Options\AjaxBottomp.html” and proceeding in a similar way.

11.6 DEVELOPING A PROJECT WITH A JAVASCRIPT LIBRARY

We mentioned the ability of customizing the supplied files (“myAppl.htm”, “visu.css”, “CustomFunctions.js”). This customization can be done by developing the project with user JavaScript library. This enables the modernization project to use code with short HTML tags to refer to complex widgets, designed for cross browser compatible. For example calendar related to a date fields, button with a graphical character etc).

The Virtel SR product provides some examples:

In the separately supplied folder “\samples” you will find components to start developing with several JavaScript libraries. “ReadMe” files give explanation on how to use the components. Files are presented with a set-up similar to that proposed for your sources.

Note: To make the set up flexible, in the declaration tags of the HTML pages the URL to the JavaScript library files are simply those of the library server repository. These URL can easily be modified to check a new version of the library. In a further step of the project, the useful files of the library server repository can be downloaded and then uploaded to the VIRTEL directory, and referred to by URL.

12 CUSTOMIZING THE WORKBENCH

The purpose of the **Virtel SR** is to replicate the template generation as much as possible. If a tag combination is expected to be repeatedly used, it can be built automatically through a related choice. **Virtel SR** seeks to make this easy to do as you can build your own choices. The number of the choices supplied as a standard is reduced in order to display only those you intend to use. Components are supplied prepared so that they can be added as choices.

12.1 SUPPLIED ADDITIONAL CHOICES

A useful choice is to build a combination of COPY-FROM and other HTML tags, maybe related to CSS rules or JavaScript processing. We supply a component “wrapped named COPY-FROM” which works as follows:

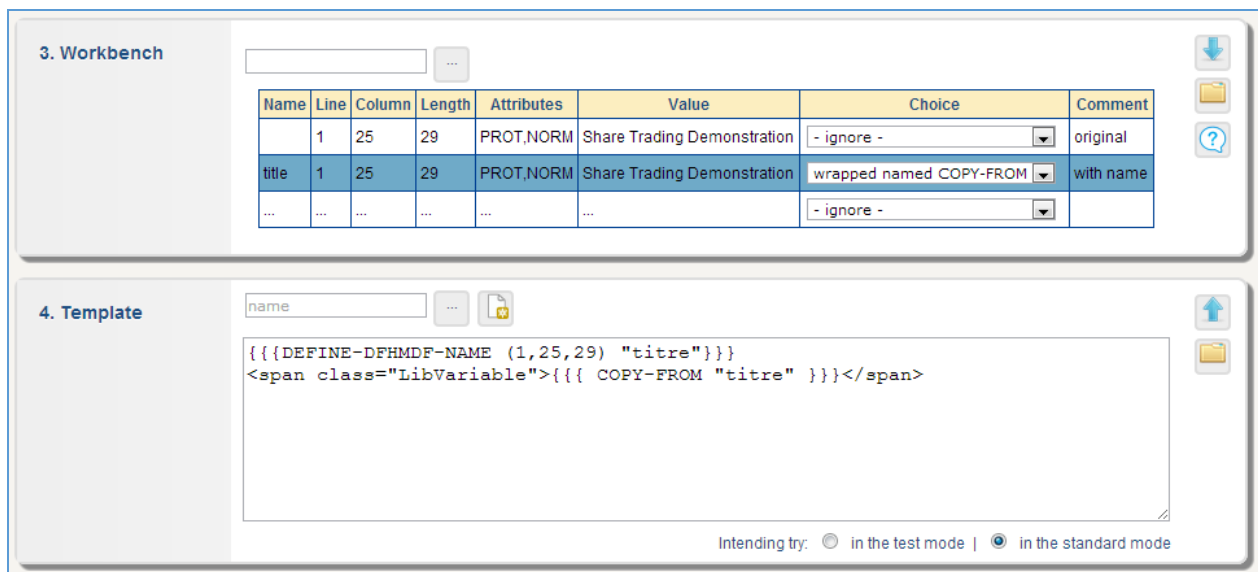


Figure 12-1: Example of generated tags upon the wrapped named COPY-FROM choice.

For this “wrapped named COPY-FROM” workbench choice, **Virtel SR** generates a **VIRTEL** tag for a named COPY-FROM wrapped by span tags within a CSS class.

12.1.1 SNIPPET EXAMPLES

In the separate supplied folder « \samples\snippets » you will find various components for different situations.

Let us take the supplied file “« \samples\snippets\wrapped named COPY-FROM\ mySnippets.js ».

The statements contained in this file process the implementation of the choice “wrapped named COPY-FROM”.

Open this file with a text editor. You will find a JavaScript array named “oTemplateBuilder.custom” indexed by the workbench choice. To each index is related a function which generates the tags for this choice.

The related function is coded as follows:

```

1. oTemplateBuilder.custom["wrapped named COPY-FROM"] = function(oRecord,str, index, that){
2.     var newString = str;
3.
4.     // get the useful workbench data
5.     var NAME      = elRecord.Nm; // Name
6.     var POS1      = elRecord.Li; // 3270 screen line number
7.     var POS2      = elRecord.Col; // 3270 screen column number
8.     var LENGTH    = elRecord.Lg; // 3270 screen length area
9.
10.    // add a HTML opening tag (could be customized)
11.    newString += "<span class=\"LibVariable\">";
12.
13.    // add the "named COPY-FROM" tag built through the standard Virtel SR choice
14.    // which leads to add the area definition tag for this name
15.    // with its controls on the data consistency
16.    newString = oTemplateBuilder.custom["named COPY-FROM"](oRecord, newString, index, that);
17.
18.    // add a HTML closing tag (could be customized)
19.    newString += "</span>";
20.
21.    return newString
22. }

```

Figure 12-2: Standard statements to process a wrapped named COPY-FROM choice while creating the template.

This function accepts the following parameters:

- oRecord : an array containing the workbench row data
- str : the text that has already been built for the resulting template
- index : the row number in the workbench
- that : the oTemplateBuilder builder

The template text is enriched through the following statements:

- The statements from line 5 to 8 retrieve the useful information of the workbench row.
- Line 11 adds the HTML span opening tag with the default CSS class.
- Line 16 adds the statement referring to the **Virtel SR** standard process for the choice "named COPY-FROM": This adds the **VIRTEL** tag for the definition of the 3270 screen area at the beginning of the text for the template after a consistency control; and this adds the **VIRTEL** tag intended to copy the value of the 3270 screen for the named area, at the end of the string being built.
- Line 19 adds the HTML span closing tag.

12.1.2 SAVING YOUR CHANGES

Open your source file «\ Options\mySnippets.js» to an editor.

At the bottom of this file, paste the copied selection.

Save this file.

Refresh the **Virtel SR** interface.

The choice “wrapped named COPY-FROM” is available in the list of the workbench choices.

It can be used by customizing the css rules for the span tag with class « LibVariable » (cf 11.2).

It can also be adapted:

12.1.3 YOUR ADAPTATION

In one project we could process the «wrapped named COPY-FROM » by wrapping the COPY-FROM in a division with an identifier and a new CSS class.

The JavaScript statements described above will be adapted to wrap the COPY-FROM by a division with an identifier and a css class.

In your source file « \ myProject\Options\mySnippets.js » updated according the explanation above, you can modify the JavaScript statements for the function “oTemplateBuilder.custom[“wrapped named COPY-FROM”]”as follows:

```
oTemplateBuilder.custom["wrapped named COPY-FROM"] = function(oRecord, str, index, that){
    var newString = str;

    // get the useful workbench data
    var NAME      = elRecord.Nm; // Name
    var POS1      = elRecord.Li; // 3270 screen line number
    var POS2      = elRecord.Col; // 3270 screen column number
    var LENGTH    = elRecord.Lg; // 3270 screen length area

    newString += "<div id=\"\" + NAME + \"\" \"
                + \"class=\"newClass\">";
    newString = oTemplateBuilder.custom["named COPY-FROM"](oRecord, newString, index, that);

    newString += "</div>";

    return newString    };

```

Figure 12-3: Adapted statements to process a “wrapped named COPY-FROM” choice while creating the template.

The content of the name cell in the related workbench row is inserted as an identifier of division tags which wrap the COPY-FROM. A class “newClass” is set for the division.

To take into account this change, proceed as explained above (cf 12.1.2) starting at the step of saving « \ myProject\Options\mySnippets.js ».

12.2 CREATING NEW CHOICE

Virtel SR also allows you to create a new modernization choice to automate the building of complex combined statements which are suited to your project. Base code and comments are inserted in the supplied version of « \samples\snippets » to guide you.

Let us work on a sample for creating the choice “my title” which is designed to generate the same result as a sequence of the following various choices in the workbench:

- choice ‘HTML code’ with the value cell set to « <H1> »,
- choice ‘named COPY-FROM’,
- choice ‘HTML code’ with the value cell set to « </H1> ».

Since this combination is intended to be encountered frequently in your project, creating the choice “my title” will enable you to get the result by a single choice.

12.2.1 DISPLAY THE CHOICE

You could implement it as an adaptation of "wrapped named COPY-FROM" by proceeding as shown above and replacing the name of the choice. Here is another way of doing it.

Open the supplied file “\samples\snippets\outline for a workbench choice\mySnippets.js” with an editor. Copy all its content. It contains outline for a workbench choice with guiding comments.

Open your source file « \ myProject\Options\mySnippets.js » with an editor.

At the bottom of this file, paste the copied selection.

Through the editor, replace all “xxx” of the pasted part by “my title”.

The “Guide Comments” I invite you to remove comment marks to make the following definition instruction work.

```
myOptions[myOptions.length] = "my title";
```

Figure 12-4: Statement to display the new choice “my title” in the choice select box of the workbench.

Populate the texts for the tooltip. With the current design, the 3270 positions are needed, and the name is compulsory.

12.2.2 PROCESSING THIS CHOICE

To process this choice, you must define the related `oTemplateBuilder.custom` function you find in the pasted sequence of your file. It looks like this:

```
oTemplateBuilder.custom["my title "] = function(oRecord,str, index, that){
    var newString = str;

    var NAME      = elRecord.Nm;
    var POS1      = elRecord.Li;
    var POS2      = elRecord.Col;
    var LENGTH    = elRecord.Lg;

    var inconsistent = false;
    inconsistent = (NAME=== "");
    if (inconsistent) {
        var msg = "The name is missing";
        traceToTemplate.add(msg,index);
    }
    else {

        newString += "<H1>";
        newString = oTemplateBuilder.custom["named COPY-FROM"](oRecord, newString, index, that);
        newString += "</H1>";
    }

    return newString
};
```

Figure 12-5: Framework to guide the coding of the process of the new choice.

The function could be coded in the following way:

```
oTemplateBuilder.custom["my title "] = function(oRecord,str, index, that){
    var newString = str;

    var NAME      = elRecord.Nm;
    var POS1      = elRecord.Li;
    var POS2      = elRecord.Col;
    var LENGTH    = elRecord.Lg;

    var inconsistent = false;
    inconsistent = (NAME=== "");
    if (inconsistent) {
        var msg = "Name is missing";
        traceToTemplate.add(msg,index);
    }
    // standard Virtel SR control on the screen positions
    ctrlPos( POS1, POS2, LENGTH, null, index );

    newString += "<H1>";
    newString += CodeCopyDFHMDf( NAME );
    newString += "</H1>";

    return newString
};
```

To take into account this change, you must save the modified JavaScript file and proceed as explained above in 12.1.3.

12.2.3 RELATING TO RULES AND PROCESSES

COMBINING CHOICES WITH CSS RULES

The tags generated by new choices could be related to CSS rules, (for example, a CSS rule on the selector “H1” of your last choice sample). The CSS customization could be done as explained above (in section 11.2), or by adding a new style sheet.

JAVASCRIPT PROCESSING OF CHOICES

The tags generated by such new choices could be related to JavaScript processes. JavaScript customization could be done as explained above.

12.3 SAMPLES

The separately supplied samples folder « \samples\snippets\ » contains samples of other possible choices. See the Samples folder for further details.

13 TEMPLATES FOR VARIOUS LANGUAGES

Various languages require specific character encoding. **VIRTEL** supports this feature.

Please see the **VIRTEL** documentation for defining a language choice to **VIRTEL** on the mainframe. Here is the language configuration for **Virtel SR** side. This should be done only makes sense when your project requires different versions by language.

13.1 YOUR LANGUAGE CHOICES

At the right top of the interface, clicking the setting icon surrounded by a green circle in the screen shot below opens the language panel:

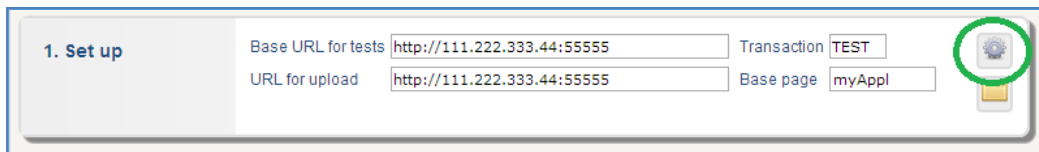


Figure 13-1: Setting icon.

The advanced settings panel contains a part for the template languages:

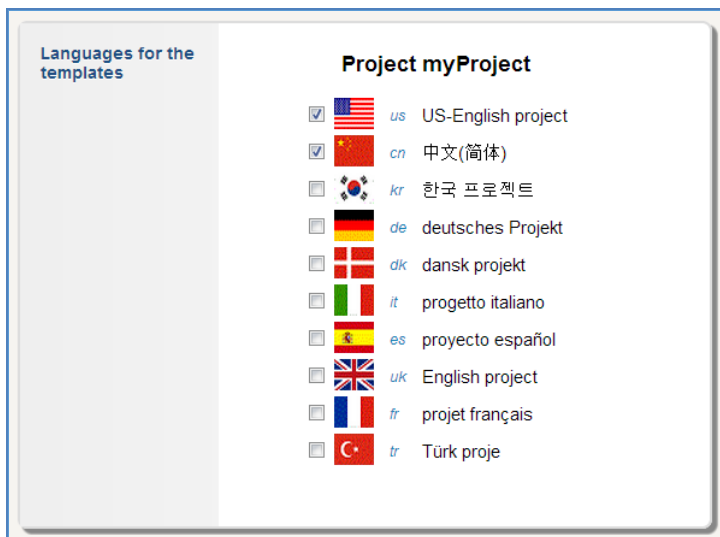


Figure 13-2: Setting panel.

Please check the boxes for the languages you want to support. Then, click the icon to go back to the **Virtel SR** interface (on the right top).

13.2 WORKBENCH RELATED CHOICES AND LANGUAGE SUPPORT

New choices will be inserted into the workbench. Beside the standard “literal”, you will see a “literal xx” for each “xx” selected language.

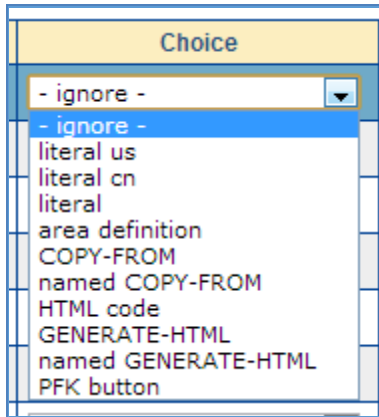


Figure 13-3: Workbench choices.

The choice “literal xx” is intended for the text which will only be displayed in the version for the language “xx”. This enables to work with a common workbench source for various language versions.

Value	Choice
test	literal us
测试	literal cn

Figure 13-4: Use sample.

If you wish to customize another workbench choice (e.g. “myChoice”) according to the checked templates languages,

- Open your source file « \myProject\Options\mySnippets.js » with an editor.
- Add the following statement at the bottom:


```
oTemplateLang.setBasicItem("myChoice");
```
- Save this file.
- Refresh the **Virtel SR** interface. The workbench choice "myChoice xx" (if the template language “xx” is selected) will be available.

13.3 CHOICE OF THE TEMPLATE LANGUAGE

The icons of the selected languages are displayed in section 4 of the **Virtel SR**, above the template creation button.

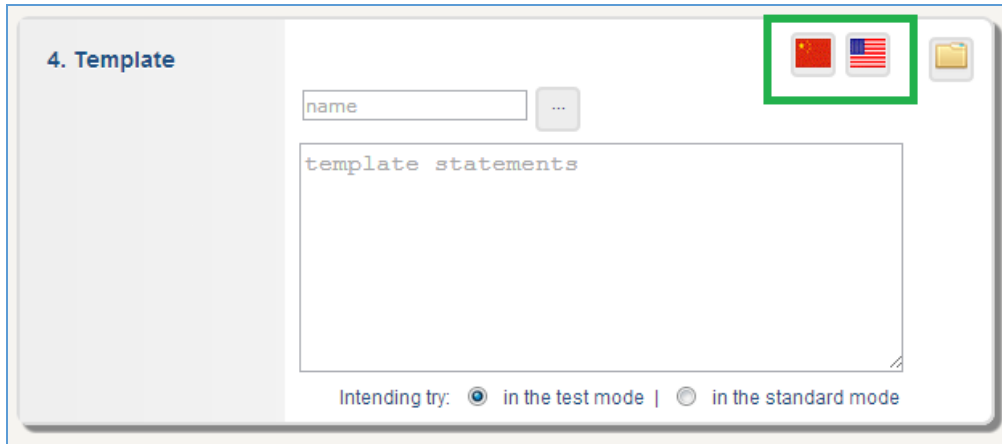


Figure 13-5: Template language icons.

Click the icon to choose the language.

The selected icon will be surrounded by a blue border.

The default location to save the template will be "Template/xx" (for the "xx" language).

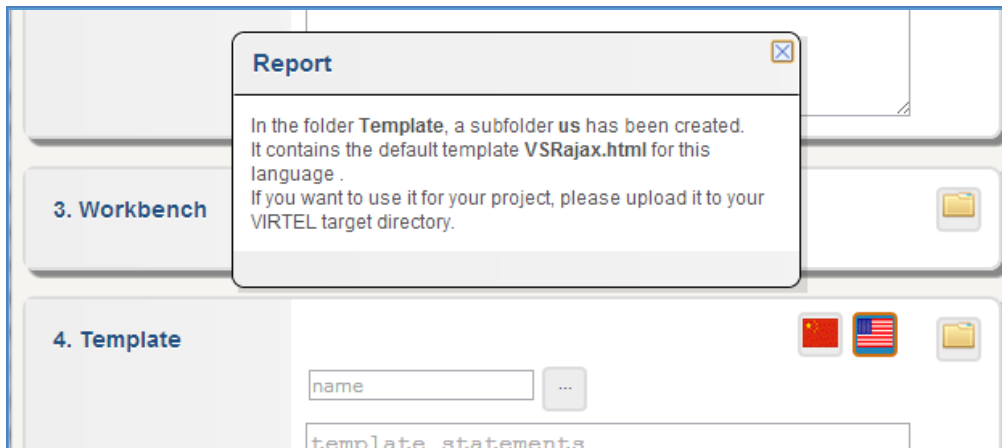


Figure 13-6: Language selection.

After the creation of the sub-folder "xx", **Virtel SR** creates a "VSRajax.html" template (see chapter 4.2) containing the **VIRTEL** tag for the character encoding of the "xx" language, and saves this template to the sub-folder "xx".

13.4 TEMPLATE CREATION

Click the button to create the template.

If the "xx" language is selected, the data of the "literal xx" choices will appear.

Save the template. The **VIRTEL** tag for the charset encoding of the "xx" language will be taken into account. By default the template will be saved into the "Html/xx" folder.

It then needs to be uploaded to the **VIRTEL** directory for testing.

14 CONCLUSION

This companion guide to the basic elements of our VIRTEL Web Modernization software, a product which helps you to modernize the user interface of mainframe applications.

If you would like further information or help with your project, we can offer:

- Training sessions.
- A consultancy service to help you with the development and implementation of your project.

To benefit from these services, simply email us at cominfo@sypertec.com.

Following your initial experience of VIRTEL Web Modernization, you may be interested in other possibilities offered by the VIRTEL product which contains many more features.

VIRTEL offers a range of functionalities, of which the following are some examples:

- PDF generation
- Application processing scenarios
- Access to external applications
- Data transfer to Excel spreadsheets
- XML data exchange
- Send and receive emails using SMTP
- Rendering presentation depending on the terminal (PC, smartphone, tablet...),
- ...

To learn more about these possibilities, please consult our web site www.virtelweb.com or send us your request by email to cominfo@sypertec.com.

In summary, here are some of the principal advantages of the **VIRTEL/WEB** software suite:

- 100% mainframe HTTP/SMTP server
- Dynamic support of VTAM <> TCP/IP
- Transparency for SNA-based applications
- Optimization and serialization of tasks
- HTTP/HTML/XML standards-based
- Communication by simple URL or by XML messages
- Provide and communicate with web services
- Browser-based navigation (no Applets)
- Numerous levels of modernization
- Uniform access to applications from distributed platforms (WAS, WebLogic, CRM, ERP, etc.)
- No need for additional connectors
- No requirement for Java on the mainframe
- No requirement for TN3270 on the mainframe
- Integrated with security tools: RACF, ACF2, TSS, SSO
- Support for secured dialog using https protocols (SSL/AT-TLS)

In daily use at hundreds of large IBM accounts,



Virtel enables your legacy applications

to interact with the other components of your information system,

simply, transparently, and securely..

15 INDEX

A

Ajax request, 45

Application

 Name, 15

Applid

 CICS, 16, 18

Attributes

 3270, 33

B

BMS, 31

 radio button, 32

C

CICS

 Applid, 16, 18

Conclusion, 69

Configuration

 Mainframe, 14

 Virtel SR, 27

Consultancy, 69

custom.js, 29

Customization, 54

 create a new workbench choice, 63

 display a new workbench choice, 63

 process a new workbench choice, 64

 supplied choice, 60

 tooltips for a workbench choice, 63

 workbench, 60

E

- Error messages, 52
- Error.htm, 28
- Excel, 69

H

- Home page for VWM, 29
- HTML
 - Generation, 38
 - Modification, 50

I

- Image, 54
- Import
 - Workbench, 37
- Installation, 22
- Introduction, 13

L

- Log
 - VIRTEL, 52
- Logo, 54

M

- Mainframe
 - Configuration, 14
- Map, 30
 - BMS, 31
 - Generate, 31
- Messages
 - Error, 52
- MFS

- map source, 32
- maps, 32
- radio button, 32
- without map source, 32

P

Page

- Standard mode, 45

PDF, 69

Project

- Create, 23

R

Rescue

- Workbench, 37

S

Scenario, 20

Scripts

- Logon and logoff, 16, 19

set up

- base page, 27

- basis upload, 28

- transaction name, 27

- url for test, 27

- url for upload, 27

SMTP, 69

snippets

- workbench choice, 65

Spreadsheet, 37

Standard mode

Sub-page, 45

Sub-page, 45

T

template

control, 39

creation, 38

standard mode, 41

Template

Batch, 42

Test, 44

Testing, 43

tooltips

for a workbench choice, 63

Training, 69

Transaction

xxxxAJAX, 17

xxxxCAPT, 15

Troubleshooting, 52

U

Upload, 43

procedure, 28

URL

Base, 14

Upload, 14

V

VIRTEL directory

Scenario, 21

update, 28

VirtelSR

Installation, 22

VSAjax.html, 28

W

Workbench, 33

Import, 37

Rescue, 37

TXT format, 37

XML format, 37

workbench choice

snippets, 65

X

XML, 69