Virtel Macros

Note: *This Newsletter applies to Virtel 4.56 (updt5484 and above) running with default parameters or in "Compatibility mode". References to the file w2hparm.js refer to the default file distributed in the W2H directory. This file can be found in the distributed sources.zip file which can be downloaded from the Administration portal. See *Appendix B* in this newsletter for an example of the distributed w2hparm.js file.*

Introduction

Virtel macros capture keystroke operations which can subsequently be used to automate 3270 functions. These user captured macros are stored within a file called macros.json. This file is a JavaScript array of JSON objects, with each object representing a user macro. Here is an example:-

```
{"macros":[
{"name":"mylogon","rev":2,"def":[{"txt":"sptholt"},"ENTER",{"txt":"bastet"},"ENTER"
,"ENTER","ENTER"],"mapping":{"key":"ctrl","keycode":76}},
{"name":"logoff","rev":1,"def":[{"txt":"=x"},"ENTER",{"txt":"logoff"},"ENTER"],"map
ping":{"key":"ctrl","keycode":79}},
{"name":"logon","rev":2,"def":["Tab","Down",{"txt":"sptholx"},"ENTER",{"txt":"MOONR
AY"},"ENTER","ENTER","ENTER"],"mapping":{"key":"alt","keycode":76}}
],"fmt":2}
```

Macro modes

The macros.json file can be maintained locally, known as local mode, or can be downloaded from a centralized repository on the host through the Virtel's Dynamic Directory Interface (DDI). This is known as remote or DDI mode. See: *Using macros with the DDI interface* for further information. The macros.json file is maintained through either the local macro interface, as launched by the user from the Virtel toolbar, or by an Administrator through the DDI interface. The macros.json file can be edited outside of these interfaces but this is not recommended unless you understand the macro structure and are familiar with JavaScript. There are various synchronization options which will determine how the central DDI repository is synchronized with the user's local macros. By default Virtel maintains macros in local mode. In both modes, local and DDI, the macros.json file is stored in the browsers local storage.

The User Macro Interface

Macro functions

You can capture and list macros by using the two macro ICON functions displayed in the Virtel tool bar. These are the red record/stop button and the green/blue triangular display button. Green indicates local mode and blue DDI mode. When the ICON is blue macros will be synchronized between the users local storage and the DDI repository. The type of synchronization will depend on the options set in the w2hparm.js member.
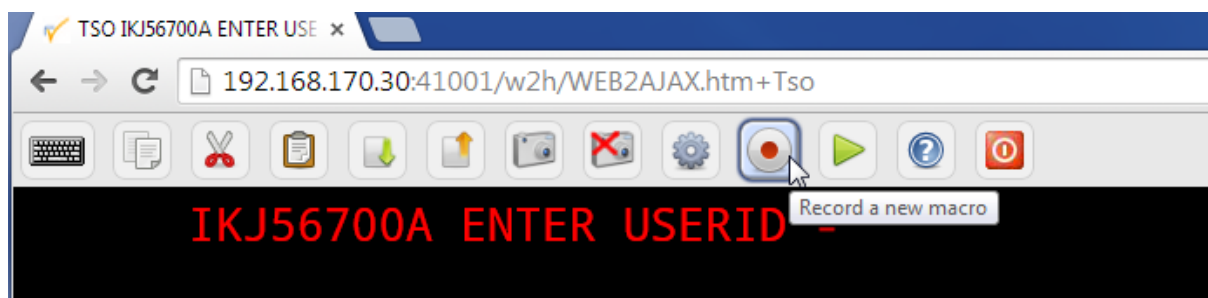
The Record function

Figure 1 - The Macro Record ICON

The record function is an on/off button that will record key strokes. When recording, the ICON will flash until it is clicked at which point it will stop recording and save the key strokes. A save panel will be displayed asking for the name of the macro entry. Note that an ENTER/PFK key must be pressed at least once, i.e. data must be sent by some key operation in order to create and save an entry within the macro.JSON file.
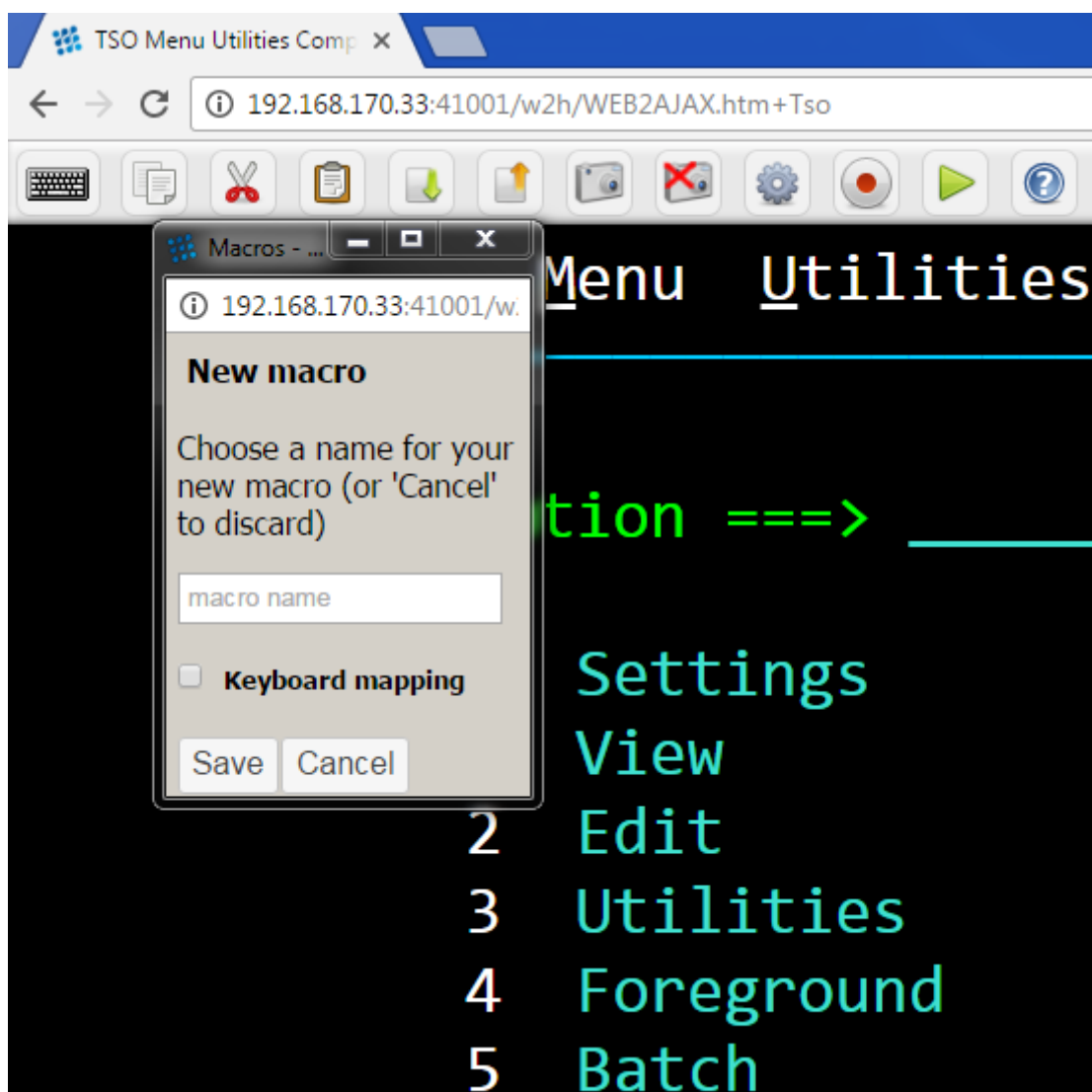

Figure 2 - Saving a macro

Keyboard Mapping

When saving the macro you have the option of assigning a "hot key" or shortcut to the macro through keyboard mapping. Keyboard mapping can be a combination of ALT or CTRL keys and another keyboard key (F1 thru F12, A thru to Z, 1 thru 9). Beware that some keyboard combinations may be reserved for the operating system or Virtel functions. For example, CTRL-R is a browser refresh option. Allocating this combination as a hotkey will only invoke the refresh option and not the Virtel macro. Keyboard mapping is a feature that is turned on through a parameter in the w2hparm.js file. By default, keyboard mapping is set to false. To turn on keyboard mapping specify the following in the w2hparm.js member:-::

        w2hparm.keymapping=true

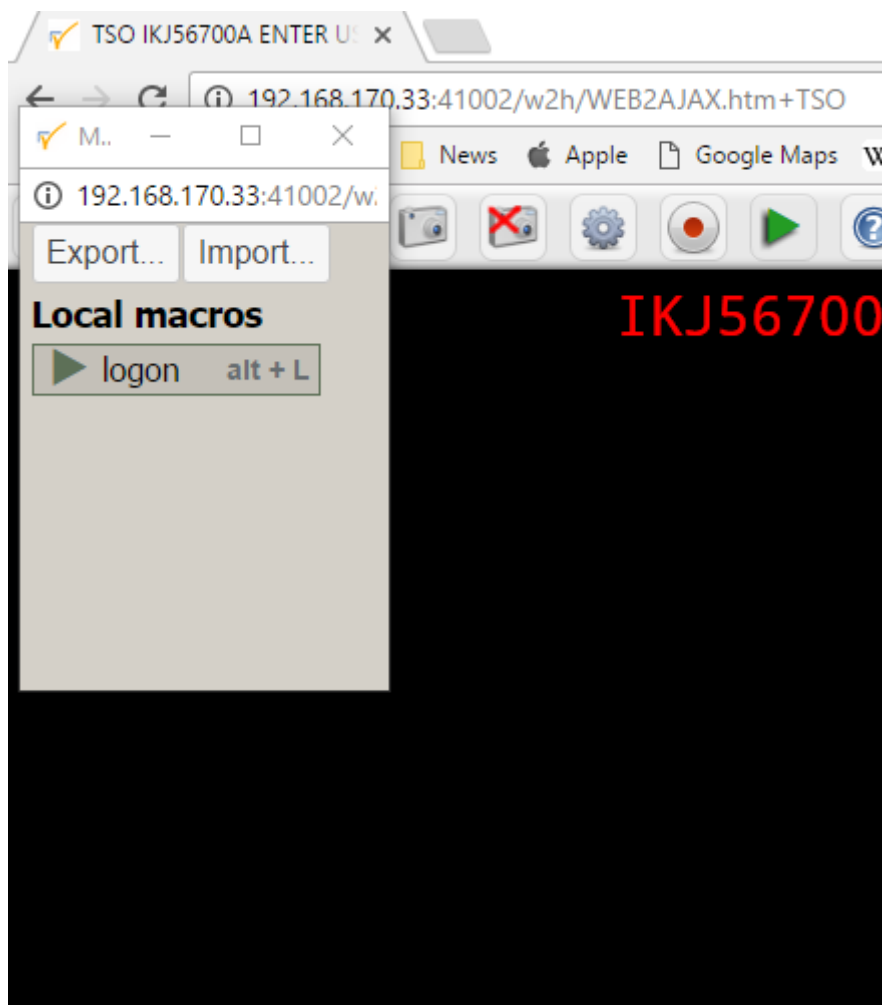With keyboard mapping enabled the macro interface will display the associated key mapping against the macro.

Figure 2a - Saving a macro with key mapping

The Display button

The green triangular button will display the local storage macros. From here a context menu can be opened against each macro using the mouse right click. This will provide delete, edit, save as and run functions.
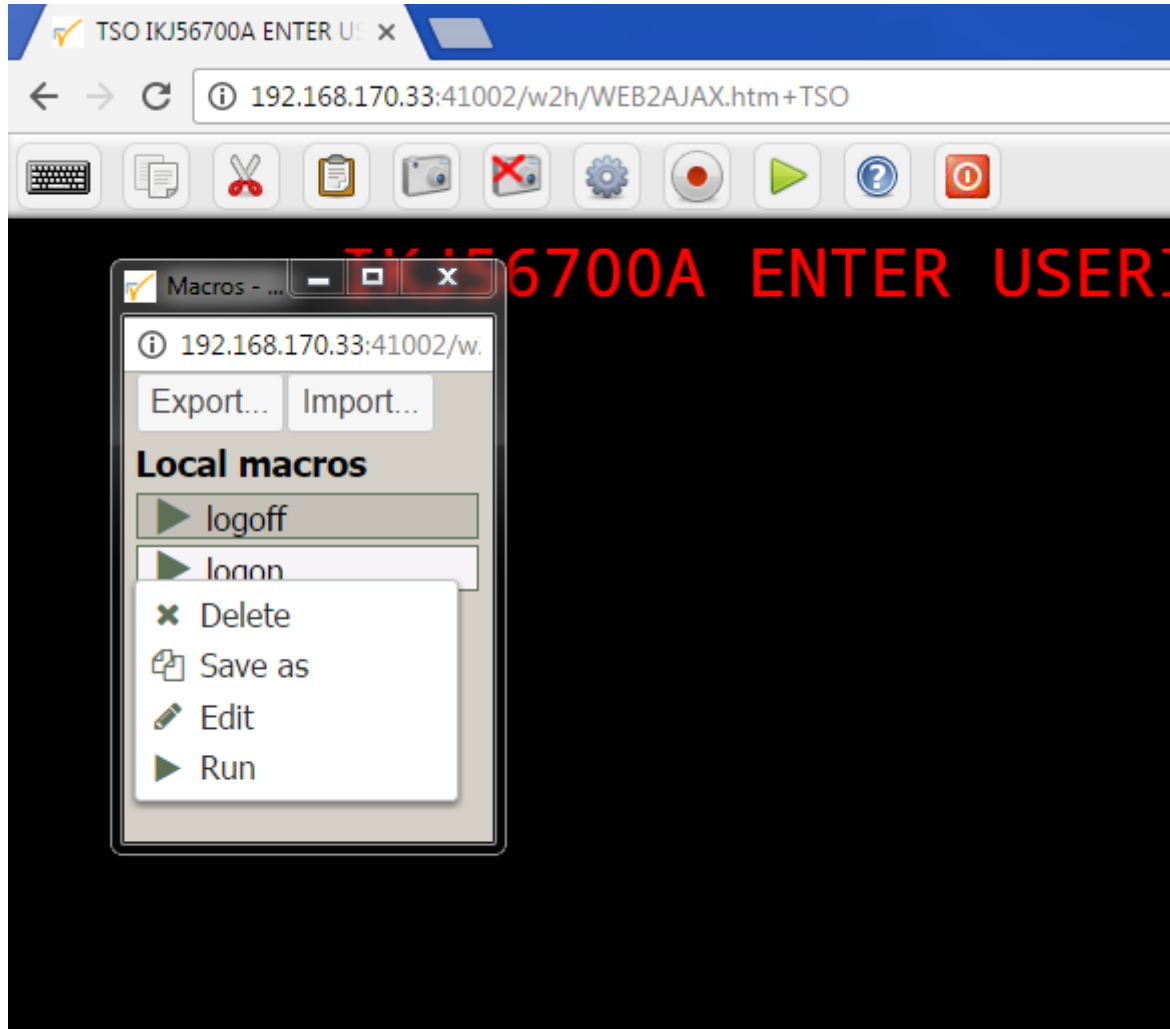

Figure 3 Context menu associated with a macro

Export and Import options

Macros can be exported or imported using the Export and Import buttons. On export, the macros.json file will be created. If you plan to migrate to using the DDI option you will need to export the macros and then upload the relevant macros.json file through the DDI interface. See: *Using macros with the DDI interface* for more information.

Using macros with the DDI interface

An alternative to maintaining macros in a user's local browser storage is to maintain and manage macros in a central repository on the mainframe. Within the central repository all of the macros can be managed by a Virtel administrator. The advantage of maintaining macros in a central repository is that the administrator has control over the macros and how they are maintained and distributed. A user can now be provided with site controlled macros as part of their Virtel interface – they do not have to write the own unsupported macros. The repository is part of the DDI interface which can be used to administer and manage files – Virtel uses the DDI feature to manage the macro file macros.json.

For any user the repository can hold three distinct levels of a macro. The user, group and global level. A user's user and group level are assigned based upon their corresponding security subsystem userid and group names. The global macro level contains macros applicable to all users. A user may maintain macros at their user level only whereas an Administrator can maintain macros at all three levels.

DDI Installation

In this example the DDI macro interface is turned on for transactions defined in the CLI custom directory, normally associated with port 41002. By default, the transaction CLI-03P identifies the w2hparm.js as belonging to the W2H directory. This transaction needs to be modified so that the directory for w2hparm.js (Application field) will now be the CLI directory. The transaction should be updated so the path w2h/w2hparm-js points to the CLI-DIR directory. A modified w2hparm.js file, with the macros enabled option set, should then be uploaded to the CLI directory. We use the CLI directory for customer modifications to avoid any changes being lost during a Virtel upgrade.

Figure 4 Modified CLI-03P transaction

To use the DDI interface the options must be set in the w2hparm.js member. The various options control the synchronisation between the user's local macro storage and the DDI central repository. Specify one of the following options to turn on DDI or remote macro mode:-

```
"w2hparm.useVirtelMacros":[false | true | never | once | daily | session ],
```

Macro options:

Option	Description
Never	Do not synchronise with DDI unless the user presses the refresh button.
Once	Synchronise with DDI only when local storage hasn't been initialised
Daily	Once a day
Session	Every time a new browser session is opened
True	Same as never
False	DDI macros disabled. This is the default

The w2hparm.js member can be found in the sources.zip file. This can be downloaded using the following url: myVirtel:41001/w2h/sources.zip or from the Administration portal. Extract the zip file and locate the w2hparm.js file. Modify the w2hparm definition to include the above option then use the Virtel file upload facility to upload the modified w2hparm.js to the CLI directory.

The next step is to run the ARBOLOAD JCL with the VMACROS=YES option set. The ARBOLOAD job can be found in the CNTL or SAMPLIB library and can only be run when Virtel is down. This will add the relevant Virtel transactions and elements to support DDI.

Managing macros with DDI

Obtain a "default" copy of a macros.json file. This could be one that has been exported from local storage. This file needs to be copied, using DDI, to Global directory within the DDI interface using the DDI upload interface.

The normal practice for an administrator in managing the macro repository would be to capture and create a macro in their user directory, upload it to the mainframe repository using the DDI interface, modify it for general use, and then copy it to any group, user or global directory for general availability. The upload process is through a drag and drop operation within DDI interface. To copy a macros.json to another user or group requires downloading the macro first then dragging the file into the relevant DDI panel area designated by the dotted lines.

Uploading a macro.json file to the Global directory

Open the DDI interface, select Global. The Global directory will open. Drag the macros.json file into the area designated by a series of dotted lines. This will initiate a file upload into the global directory of the macro repository:-
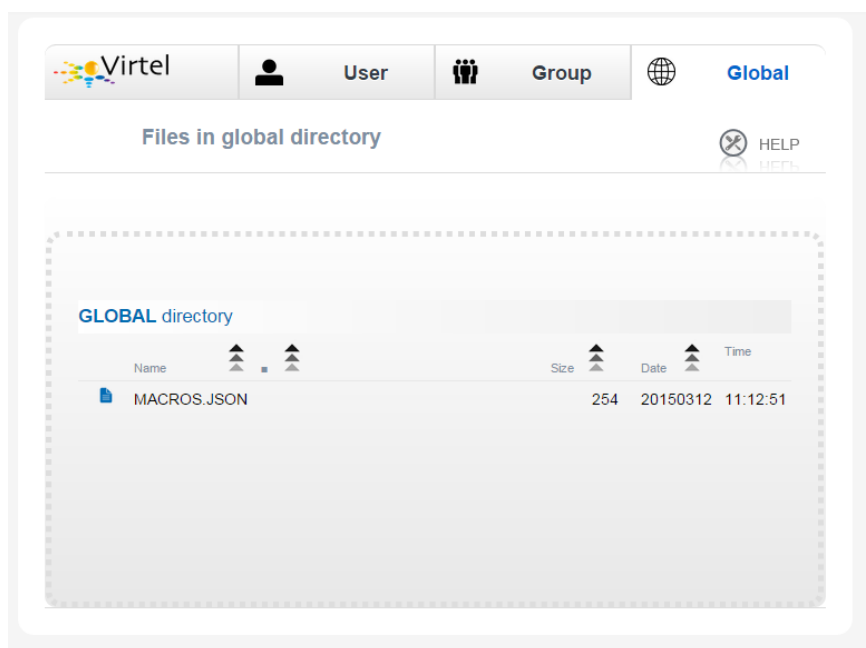

Figure 5 The DDI Global Upload

Once you have a copy in the global area, right click on the macros.json file to open a context menu. This menu provides some additional functions such as Copy, Download, Delete and Edit. For macros.json the copy function is not applicable.
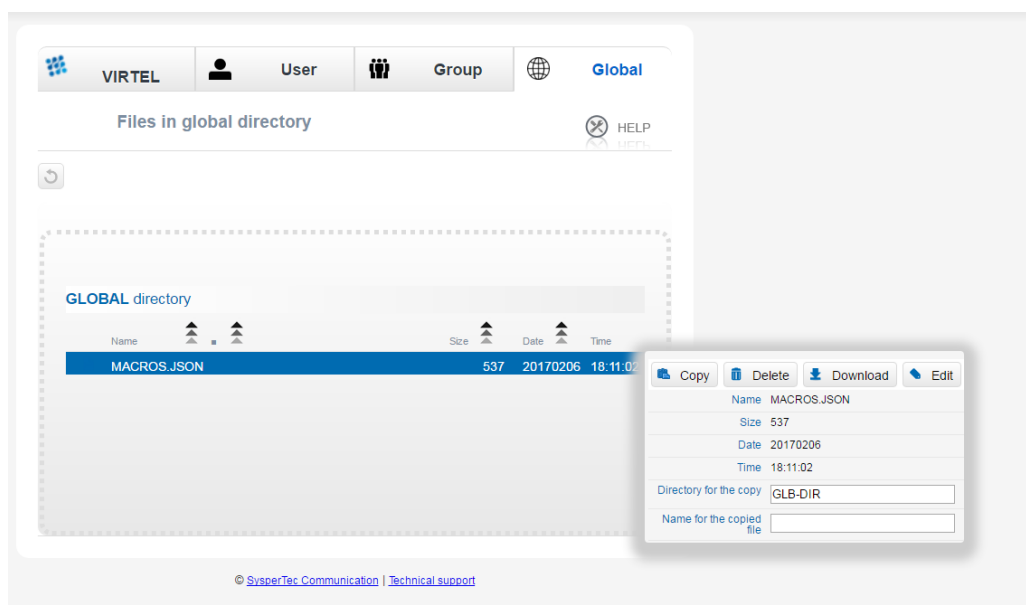
Figure 6 Additional file functions in DDI

Selecting "Edit" will open the same editor interface that was used when editing macros in local storage. Here the administrator can modify the macros held within the macros.json file. Macros can be saved with different names and associated with different key mappings. Options like Save As, Edit and Delete are available within a context menu when you right click on a macro.
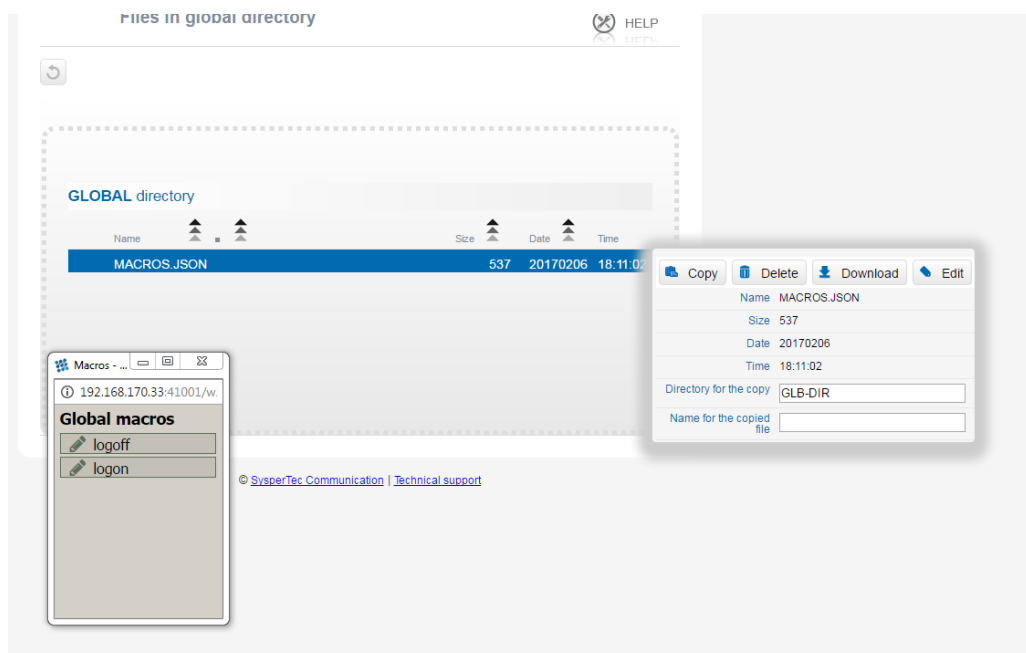
Figure 7 Updating macros with the Global Directory

Making macros available to users.

Any macros an administrator creates can be made available to users either at the Global, Group or Userid level. To make macros available at a userid level, select the User TAB from the DDI main panel. Enter the user's USERID in the user name field and press the "Green Tick" icon. This will create an area within the DDI macro repository. Upload the macros.json file you wish to make available to this user by dragging the file into the area designated by the dotted lines. This will initiate an upload of the macro into the users userid area.

Figure 8 Creating a macros.json file for a user

As with any file within the DDI repository a context menu can be brought up by right clicking on the file name.
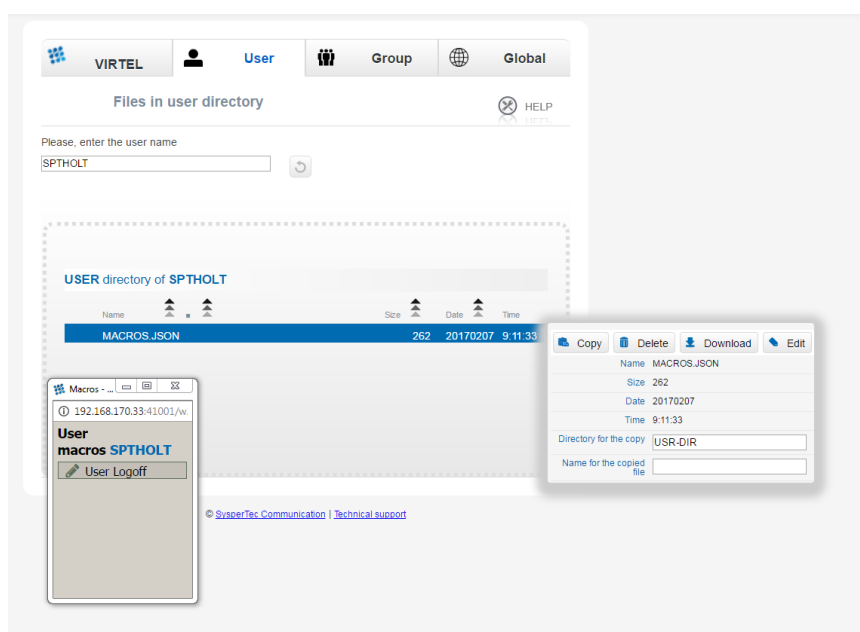
Figure 9 - Editing a global macro

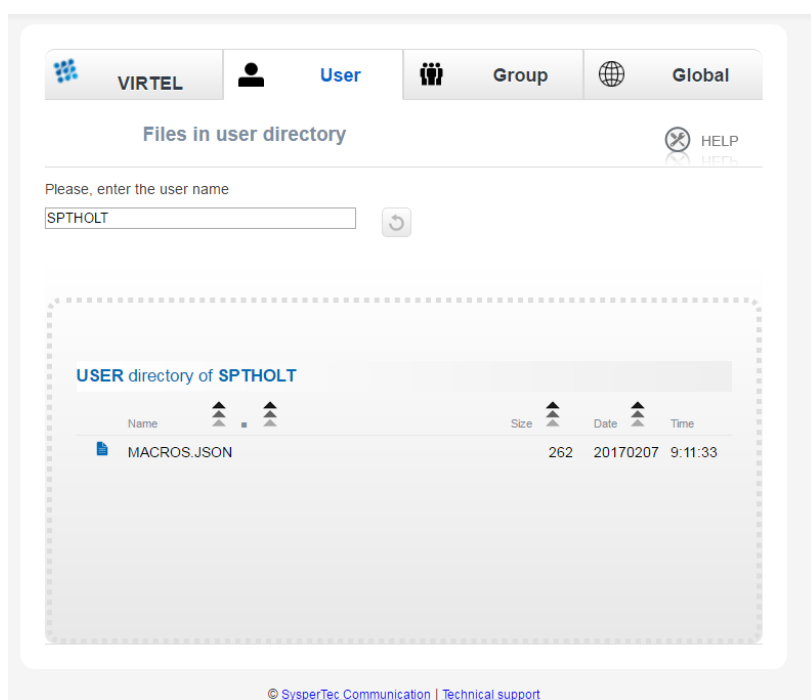An administrator can update any macros held within the DDI repository. Changes will be reflected in the user's local storage once the user open's their macro display with the "Green" ICON arrow. The following display shows the DDI repository macros as displayed in the user's browser. Remember that User macros are related to the users' Userid and the Group. This ability to associate macros to a userid and group means that all macros, related to business assets, can be managed and maintained in a central DDI repository. Maintenance can be applied centrally and distributed out to the user's browser. Users only have the option of modifying their User macros. Any changes will be uploaded to the mainframe and stored in the users' macro area within the DDI repository. This keeps the user macro modifications in sync with the repository.
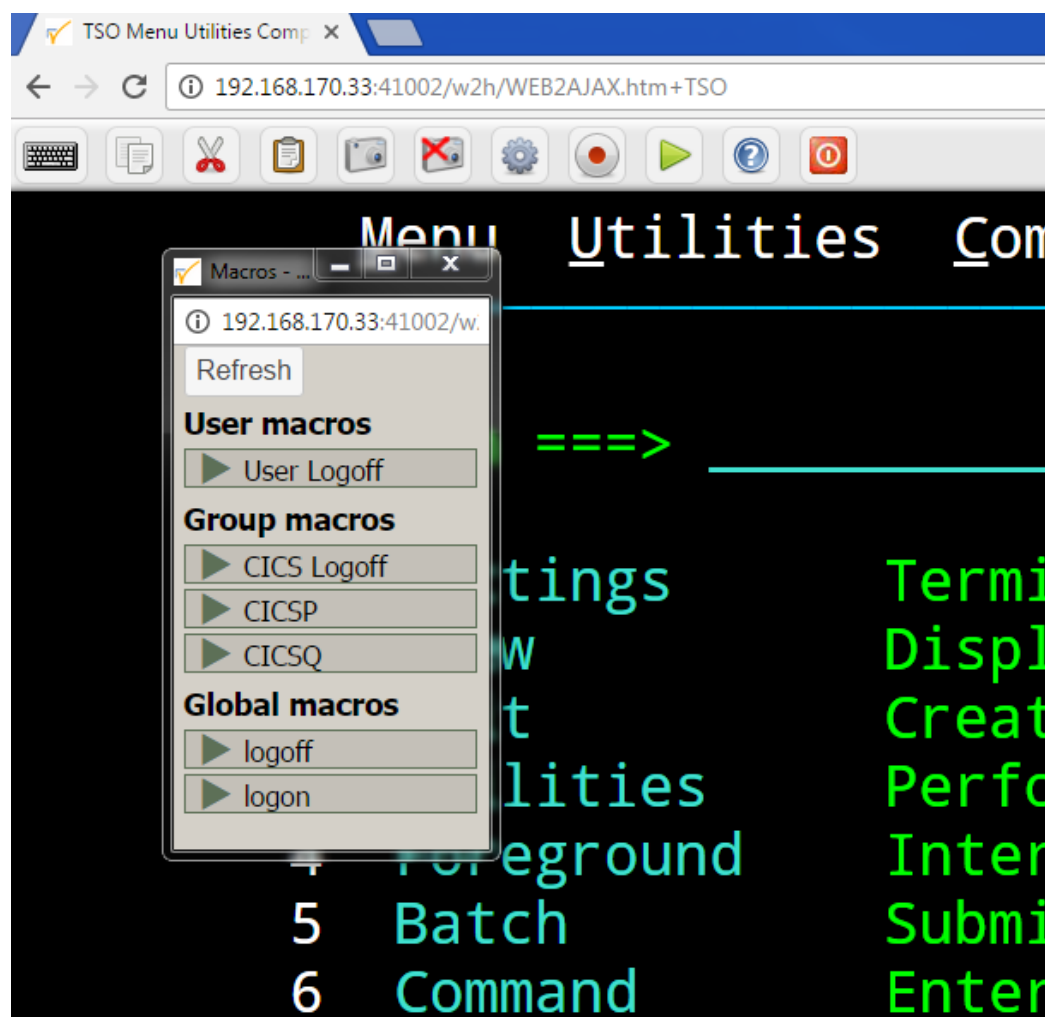
Figure 10 - User display of the DDI macro interface

Updating DDI macros at the Group level

To make macros available at a Group level, select the Group TAB from the DDI main panel. Enter the Group name in the group name field and press the "Green Tick" icon. This will create an area within the DDI macro repository were macros for this group can be held. Upload the Group macros.json file you wish to make available to this group by dragging the file into the area designated by the dotted lines. This will initiate an upload of the Group macro.
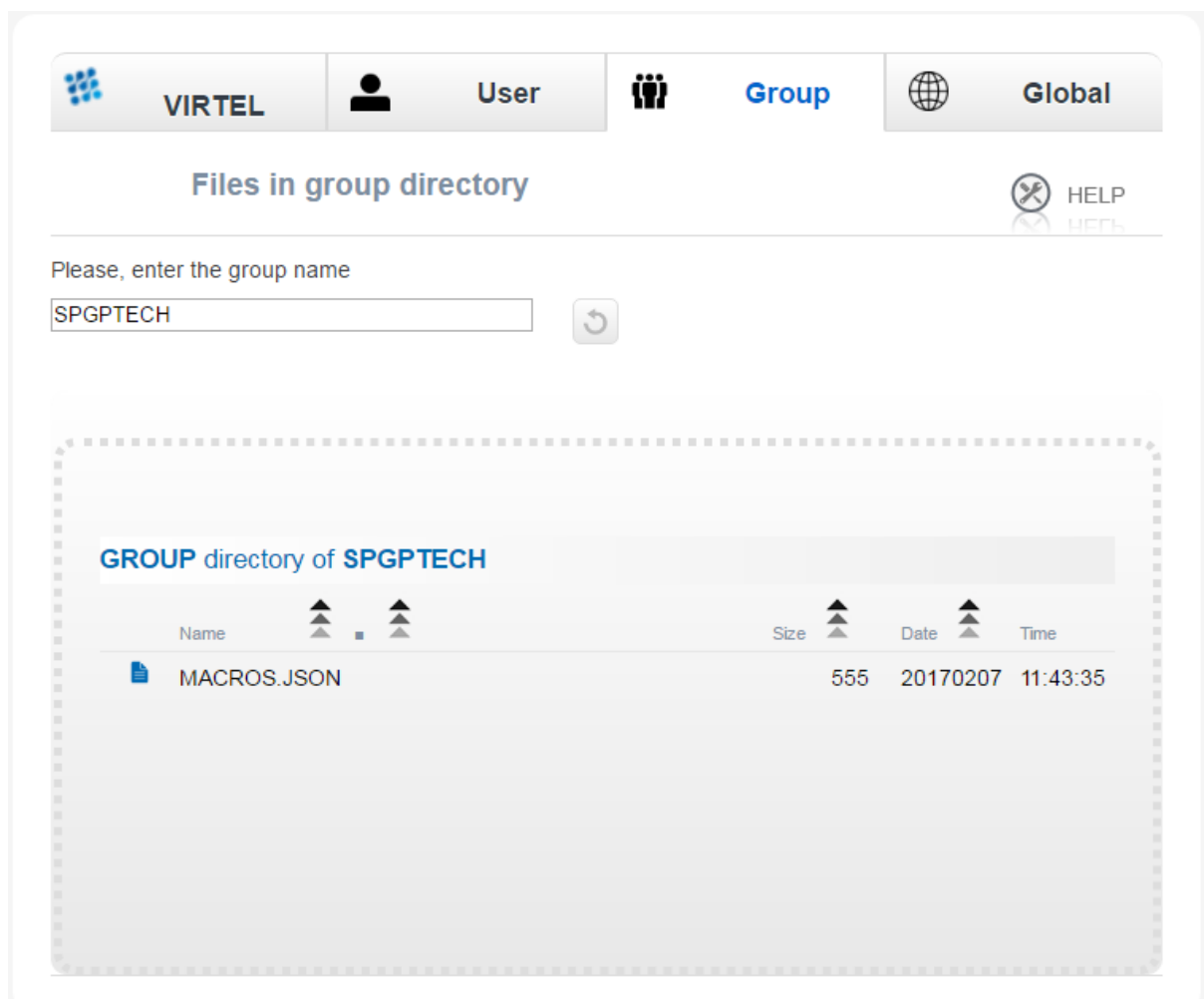
Figure 10 Macro Group Directory

To obtain the associated context menu right click on the file name – macros .json. This will provide the additional functionality to Edit, Download or Delete the macro.

Notes:

File macros.json

Macros are maintained in the macros.json file. Each macro mode, local or DDI (User, Group and Global) will maintain a separate macros.json file. Virtel's DDI can also be used to store other files other than the macros.json file. Things like corporate images, bespoke Javascript deployments etc. When copying files to DDI you must use the DDI drag and drop interface. Do not use the Administration drag and drop interface.

Save As feature

The "SAVE AS" function does not have the ability to re-assign key mapping. This can only be done through the macro editor. To clone a macro with a new key mapping perform the following operations:-

Open macro "A"
Save as "B"
Open macro "B" in Editor.
Save with key board mapping.

Autoupdate of macros in local storage.

Updates to DDI macros held in local storage can only be updated once the user refreshes or redisplays the macros. This triggers the download from the mainframe. Macros are not "automatically" updated in the background due to the performance. If an Administrator updates a macro on the mainframe that update will only be available to the user when the user redisplays or refreshes his local storage by using the "Green" display ICON.

Synchronisation of macros in multiple Virtel environments.

If you are running multiple images of Virtel, say in a SYSPLEX arrangement, using separate HTMLTRSF files then the "synchronizeVirtelMacros" option should be set to true. This ensures that macro changes are reflected in all DDI repositories and consequently associated local storage. Update the w2hparm.js file to include:-

```
"useVirtelMacros":true,
"synchronizeVirtelMacros":true,
```

Appendix A - Macro format and commands

The format of the macro.json is embedded JSON structures. Each name structure represents a keystroke macro identified by the "name" keyword.

Name   : The name of the macro entry.
Rev   : The "rev" is a user revision keyword.
Def   : The "def" keyword identifies the commands and entry values.

The macro editor supports the following commands:-

- "any string of characters to input into 3270 screen"
- move(pos)
- copy(startRow,startCol,endRow,endCol)
- paste(pos)
- paste(pos,nbRows,nbCols)
- key(keyIdentifier)

Macro example

```
{"macros":[
  {
     "name":"SDSFLOG",
     "rev":2,
     "def":["move(435)","ENTER",{"txt":"=M"},"ErEof","ENTER",{"txt":"6"},"ENTER"]
  },
  {
     "name":"SDSFDA",
     "rev":1,
     "def":[77,"ENTER",53,"ENTER",100,97,"ENTER"]
  }
],"fmt":1}
```

Appendix – B

Distributed w2hparm.js

```
var w2hparm = {
  "settingsGUI":{"version":"v2"},
  "font":"Droid Sans Mono",
  "fontsize":"window",
  "ctrl":"ENTER",
  "enter":"Newline",
  "home":"Home",
  "end":"ErEof",
  "shiftins":"Dup",
  "shifthome":"FieldMark",
  "shiftend":"End",
  "ctrlins":"PA1",
  "ctrldel":"PA2",
  "ctrlend":"End",
  "pgup":"PF7",
  "pgdn":"PF8",
  "pause":"CLEAR",
  "style":"3270"};
```

Appendix – C Virtel DDI Support for Macros

Virtel V4.56   Macros enhanced to include key mapping. Run ARBOLOAD with MACROS=YES to install feature.

Virtel V4.54   Macros are delivered as a part of the base product. Run ARBOLOAD with MACROS=YES to install feature.

Virtel V4.53   To upgrade your 4.53 system to include the new macro feature the following ARBO updates are required. Run this JCL to update your 4.53 ARBO to include the directories and transactions required for macros. You will also need to update transaction W2H-20 in WEB2HOST and changing the output scenario to ADMINVWM.

```
//VIRARBLD JOB 1,ARBOLOAD,CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*-----------------------------------------------------------*
//         SET  LANG=EN
//         SET  LOAD=SP000.VIRT453.LOADLIB
//         SET  ARBO=SP000.SPVIREH.ARBO
//         SET  VMACROS=YES
//*-----------------------------------------------------------*
//VIRLOAD PROC PROCESS='YES-OR-NO'
//VIRLOAD EXEC PGM=VIRCONF,PARM='LOAD,NOREPL,LANG=&LANG',REGION=2M
//STEPLIB  DD  DISP=SHR,DSN=&LOAD
//VIRARBO  DD  DISP=SHR,DSN=&ARBO
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  DDNAME=SYSIN&PROCESS
//SYSINNO  DD  DUMMY,DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//         PEND
//VMACROS EXEC VIRLOAD,PROCESS=&VMACROS
//SYSINYES DD  *                  <--START VMACROS SECTION-->
*-----------------------------------------------------------*
*
(EN)* Definitions for VIRTEL dynamic directories
(FR)* Dbfinitions des rbpertoires VIRTEL dynamiques
*
     SUBDIR  ID=GLB-DIR,
         (EN) DESC='Global files',
         (FR) DESC='Fichiers globaux',
             FSTYPE=V,
             DDNAME=HTMLTRSF,
             KEY=GLOBAL,
             NAMELEN=0064,
             AUTHUP=X,
             AUTHDOWN=X,
             AUTHDEL=X
     SUBDIR  ID=GRP-DIR,
         (EN) DESC='Group files',
         (FR) DESC='Fichiers de groupe',
             FSTYPE=V,
             DDNAME=HTMLTRSF,
             KEY=%GROUP%,
             NAMELEN=0064,
             AUTHUP=X,
             AUTHDOWN=X,
             AUTHDEL=X
     SUBDIR  ID=USR-DIR,
         (EN) DESC='User files',
         (FR) DESC='Fichiers utilisateur',
             FSTYPE=V,
             DDNAME=HTMLTRSF,
             KEY=%USER%,
             NAMELEN=0064,
             AUTHUP=X,
             AUTHDOWN=X,
             AUTHDEL=X
*
(EN)* Transactions to access VIRTEL dynamic directories via WEB2HOST
```

```
(FR)* Transactions pour accbder aux rbpertoires VIRTEL dynamiques via WEB2HOST
*
      TRANSACT ID=W2H-03A,
             NAME='w2h',
         (EN) DESC='Global files directory',
         (FR) DESC='Rbpertoire des fichiers globaux',
             APPL=GLB-DIR,
             TYPE=4,
             TERMINAL=DELOC,
             STARTUP=2,
             SECURITY=0,
             LOGMSG='/w2h/global'
      TRANSACT ID=W2H-03G,
             NAME='w2h',
         (EN) DESC='Group files directory',
         (FR) DESC='Rbpertoire des fichiers de groupe',
             APPL=GRP-DIR,
             TYPE=4,
             TERMINAL=DELOC,
             STARTUP=2,
             SECURITY=0,
             LOGMSG='/w2h/group'
      TRANSACT ID=W2H-03U,
             NAME='w2h',
         (EN) DESC='User files directory',
         (FR) DESC='Rbpertoire des fichiers utilisateur',
             APPL=USR-DIR,
             TYPE=4,
             TERMINAL=DELOC,
             STARTUP=2,
             SECURITY=0,
             LOGMSG='/w2h/user'
*
(EN)* Transactions to access VIRTEL dynamic directories via CLIWHOST
(FR)* Transactions pour accbder aux rbpertoires VIRTEL dynamiques via CLIWHOST
*
      TRANSACT ID=CLI-03A,
             NAME='w2h',
         (EN) DESC='Global files directory',
         (FR) DESC='Rbpertoire des fichiers globaux',
             APPL=GLB-DIR,
             TYPE=4,
             TERMINAL=CLLOC,
             STARTUP=2,
             SECURITY=0,
             LOGMSG='/w2h/global'
      TRANSACT ID=CLI-03G,
             NAME='w2h',
         (EN) DESC='Group files directory',
         (FR) DESC='Rbpertoire des fichiers de groupe',
             APPL=GRP-DIR,
             TYPE=4,
             TERMINAL=CLLOC,
             STARTUP=2,
             SECURITY=0,
             LOGMSG='/w2h/group'
      TRANSACT ID=CLI-03U,
             NAME='w2h',
         (EN) DESC='User files directory',
         (FR) DESC='Rbpertoire des fichiers utilisateur',
             APPL=USR-DIR,
             TYPE=4,
             TERMINAL=CLLOC,
             STARTUP=2,
             SECURITY=0,
             LOGMSG='/w2h/user'
*
(EN)* Transaction for the Dynamic Directory Interface application
(FR)* Transaction pour l'application Dynamic Directory Interface
*
      TRANSACT ID=W2H-07,
             NAME='dynadmin',
         (EN) DESC='Dynamic Directory Interface',
         (FR) DESC='Dynamic Directory Interface',
             APPL=VIR0022,
             TYPE=2,
```

```
                 TERMINAL=DELOC,
                 STARTUP=2,
                 SECURITY=1,
                 TRANSL=1,
                 EXITMSGI=SCDYNADM,
                 EXITMSGO=SCDYNADM
*
(EN)* Transaction to allow users to upload macros via CLIWHOST
(FR)* Transaction pour chargement des macros utilisateur via CLIWHOST
*
      TRANSACT ID=CLI-80U,
                 NAME='uplusr',
            (EN) DESC='Upload user macros',
            (FR) DESC="Chargement des macros de l'utilisateur",
                 APPL=VIR0041C,
                 TYPE=2,
                 TERMINAL=CLLOC,
                 STARTUP=2,
                 SECURITY=0,
                 LOGMSG=USR-DIR
*
(EN)* Transactions to allow administrator to upload macros
(FR)* Transactions pour chargement des macros par l'administrateur
*
      TRANSACT ID=W2H-66,
                 NAME='usrcap',
            (EN) DESC='Generate administrator upload capability token',
            (FR) DESC='Generation de jeton de capacitb administrateur',
                 APPL=$NONE$,
                 TYPE=2,
                 TERMINAL=DELOC,
                 STARTUP=2,
                 SECURITY=1,
                 TRANSL=0,
                 TIOASTA='&/S OK &/T',
                 EXITSTA=SCENUCAP
      TRANSACT ID=W2H-80A,
                 NAME='uplglb',
            (EN) DESC='Upload macros (GLB-DIR directory)',
            (FR) DESC='Chargement des macros (rbpertoire GLB-DIR)',
                 APPL=VIR0041C,
                 TYPE=2,
                 TERMINAL=DELOC,
                 STARTUP=2,
                 SECURITY=1,
                 LOGMSG=GLB-DIR
      TRANSACT ID=W2H-80G,
                 NAME='uplgrp',
            (EN) DESC='Upload macros (GRP-DIR directory)',
            (FR) DESC='Chargement des macros (rbpertoire GRP-DIR)',
                 APPL=VIR0041C,
                 TYPE=2,
                 TERMINAL=DELOC,
                 STARTUP=2,
                 SECURITY=1,
                 LOGMSG=GRP-DIR
      TRANSACT ID=W2H-80U,
                 NAME='uplusr',
            (EN) DESC='Upload macros (USR-DIR directory)',
            (FR) DESC='Chargement des macros (rbpertoire USR-DIR)',
                 APPL=VIR0041C,
                 TYPE=2,
                 TERMINAL=DELOC,
                 STARTUP=2,
                 SECURITY=1,
                 LOGMSG=USR-DIR
*-------------------------------------------------------------*
//*                           <--END VMACROS SECTION-->
```