NEWTEST DEVELOPER'S MANUAL

VERSION 7.6

VOL. 2:
SPECIFICS OF CLIENT ENVIRONMENTS
LANGUAGE FUNCTIONS

MAY 2015

CONTACT:

ip-label
Tel. + 33 1 77 15 37 50 | Fax + 33 1 49 64 03 80
Headquarters 90 boulevard National, 92250 La Garenne Colombes, France | **www.ip-label.com**
SAS with capital of 549,728 € | RCS Nanterre B 327 139 309 | info@ip-label.com

ip-label

Contents

1 INTRODUCTION

The purpose of this document, which supplements information on the general method of creating scenarios, is to provide you with the information you need to set up Newtest scenarios based on specific client environments.

Because the present document does not review general aspects of implementing scenarios, you should be familiar with the principles and operation of Newtest before continuing further. Background information is available in:

- *Newtest Overview*
- *Newtest Developer's Manual, Vol. 1*

Implementing Newtest Probe and Newtest Multi-Probe involves programming user scenarios.

2 THE BASICS

2.1 NOTIONS

- **RAS (Remote Access Service)**: this service enables Internet connections via PPP or SLIP protocols. It also contains information about the modem used to access the Internet.
- **TCP/IP (Transport Connected Protocol/Internet Protocol)**: this is the protocol used, in particular, by HTTP and FTP.
- **HTTP (HyperText Transfer Protocol)**: a protocol for transferring HyperText documents. These are written in HTML (HyperText Mark-up Language) and can be viewed using a suitable browser. Usually one page is viewed at a time. Other pages are viewed by clicking a link within the active page; each link is associated with an address called the URL (Universal Resource Locator). Each URL address points to a unique document, which is downloaded and displayed by the browser.
- **FTP (File Transfer Protocol)**: a protocol for file transfer. It allows files (binary or text) to be loaded to or from or a Web server.
- **SMTP (Simple Mail Transfer Protocol)**: protocol for sending electronic Internet messages (e-mail) to specific addresses via an SMTP server.
- **POP3 (Post Office Protocol)**: protocol for receiving electronic Internet messages (e-mail) sent to a given user on a POP3 server.
- **IMAP4 (Internet Message Access Protocol)**: protocol that offers more functionalities than POP3 for reception of electronic messages via the internet. It handles reception of messages to a given user on an IMAP4 server.
- **NNTP (Network News Transfer Protocol)**: makes it possible to access a newsgroup server.
- **DNS (Domain Name System)**: Internet address resolution server.
- **PING**: PING requests use the ICMP protocol (Internet Control Message Protocol) to verify that the user can reach a Host (remote machine).
- **LDAP (Lightweight Directory Access Protocol)**: a protocol for accessing information directories that supports TCP/IP. With directories, databases can be shared on an internal or external network. Such databases can contain any type of information, from addresses to system data.
- **WAP (Wireless Application Protocol)**: a communication protocol designed to offer access to Internet via mobile telephone networks (GSM, GPRS, UMTS). A WAP gateway ensures interconnection of mobile

telecoms and the internet. Communication between the mobile and the WAP gateway involves different layers: WAE (Application Environment), WSP (Session Protocol), WTP (Transaction Protocol), WTLS (Transport Layer Security) and WDP (Datagram Protocol).

2.2 ADVANCED NOTIONS

- There are two ways of accessing the web:
  - **Intranet**: an Internet network is made up of sites that are available to users only on their own IP network.
  - **Internet**: the worldwide network of communication systems, accessible via the public IP network via and an Internet service or access provider (ISP or IAP). Connection to a provider can be opened with a modem connected to the NEWTEST workstation, using Windows Dial-up Networking.
- **HTML pages**: the pages you see in your browser are in fact made up of different objects (graphics, sounds, other HTML pages, etc.). While you "surf", the browser is doing more than just loading the main page (the main page is the one that the URL address in the title zone of your browser refers to), but is also loading all the components included in the page. Note: an HTML page may contain other HTML pages, which themselves contain others, and so forth. Thus, if you wish to check for the presence of a particular string in a page, that string may in fact be located in one of the "sub-pages". The main page may also just be a "container", without any displayable objects.
- **HTML document**: the HTML document corresponds to an HTML page shown in the browser. There are two types of HTML document:
  - The main (or "parent") HTML document corresponds to the main HTML page shown in the browser (i.e. its URL appears in the browser's address bar).
  - HTML sub-documents (or "child" documents) correspond to secondary HTML pages displayed in a sub-window of the main HTML document. A sub-document may itself contain other sub-pages.
- **HTML elements**: These are the interactive objects that make up an HTML document (images, links, text boxes, applets, etc.). The elements can be grouped by type into collections (collection of links, collection of images, etc.).
- **Frames**: frames are HTML pages that are contained in a main page. They divide the display into separate and independent parts at the browser level.
- **Cookies**: http is a "stateless" protocol, meaning that each command is executed independently without any knowledge of the commands that came before it. For this reason, some applications (especially secure applications) must open a virtual session. The user types a user name and password, and the web application sends back an identifier for the temporary session. This session identifier can be stored in a « cookie », a local information container.
- **SSL** (Secure Sockets Layer): this is a protocol using a private key to encrypt and decrypt data. It is used for secure HTTP exchanges.
- **Passive mode**: this is a special FTP client mode, used when you try to reach a server through a firewall. This mode avoids incoming calls.
- **DN** (Distinguished Name - LDAP): each entry in an LDAP directory is uniquely identified by an LDAP distinguished name (DN). Comparable to the "Path" of a UNIX file, the DN represents the absolute access path to an entry (ex.: ou=Experimental, o=University of Michigan, c=us).
- **Attribute** (LDAP): objects referenced in an LDAP directory have attributes that describe the object's characteristics. Any object in the directory can be found by searching one or more of its attributes.

3 SCENARIOS BASED ON THE WINDOWS CLIENT ENVIRONMENT

3.1 INTRODUCTION

This section discusses the specifics of implementing Windows scenarios as a supplement to information given on the general method of creating scenarios, and addresses issues specific to the Windows client environment. You must therefore be familiar with the general method (see *Programming scenarios, Newtest Developer's Manual vol. 1*) before reading this section.

The Windows scenario is the default scenario type. It uses only those concepts strictly linked to the Windows interface.

3.2 NOTIONS RELEVANT TO THE WINDOWS CLIENT ENVIRONMENT

The Windows interface consists of a number of main or **parent windows**. A parent window may contain several zones sensitive to the mouse actions. Among these, the most relevant here are the **user menu**, the **title bar**, and the **system menu**. These zones are clearly distinguishable (when available).

A main window may contain sub-windows or **child-windows**, whose position may be fixed or variable. Sub-windows exist in many forms. Buttons and lists are examples, but sub-windows may also have their own system menu and title bar.

The user normally works in the Windows environment using the keyboard and the mouse to interact with programs. Actions are carried out in one window at a time. The sets of actions possible using the keyboard and the mouse are called **keyboard entry** (keystrokes) and **mouse entry** (mouse movements). These inputs are independent and unique. This means that each feed can be directed to different windows, but one feed cannot be directed to more than one window at a time.

Windows scenarios using these types of input cannot run simultaneously.

3.3 LEARNING MODE

The following dialog box appears at the start of the learning mode (see *Programming Scenarios in the Developer's Manual, vol. 1*):

The **Run...** button inserts an *AppExec* instruction in the script.

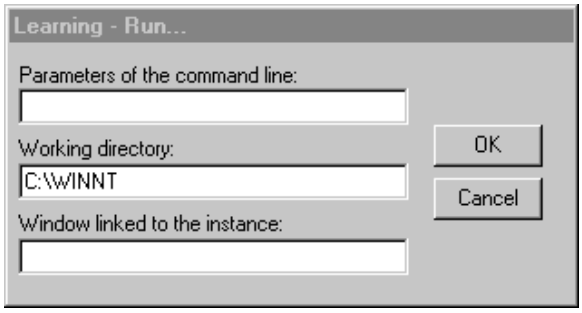
The **Terminate** button ends the recording phase.

Warning: Any keyboard or mouse action outside the learning mode window may be recorded.

This step entails inserting an application startup command into the script, then launching this application if it is not already open.

Click **Run**.

Type or select the name of the executable to launch, then click the **Open** button. The following dialog box appears:



The **Parameters of the command line** field contains a complement to the command line.

The **Working directory** field indicates the directory path for the application to run. This information is used when the *SetWorkDir* command is inserted into the script.

The **Window linked to the instance** field contains one of the words in the title of the active window of the application to be started. This information is optional.

Click **OK**.

Perform the tasks that you want to serve as the basis for the response time measurements that you want to obtain.

Suggestions:
Use the keyboard rather than the mouse.
Use menus rather than controls.
To activate a window, click its title bar.

Warning: Never run an application in a script in any way other than the one described in step 1.

Click the **Terminate** button.

3.4 WINDOWS MANAGER

The purpose of the windows manager is to enable you to deal with pop-ups that appear or disappear unexpectedly during a transaction. These may interfere with the scenario's execution, and can result in a false unavailability.

The windows manager can be used:

- either within the scenario script by explicitly activating or deactivating the rules with the instruction NEWTEST_SCRIPT:

  SetWindowManagerRule(ruleFileName AS STRING, active AS BOOL) AS BOOL

  ruleFileName: name of the rule file. The prefix "xxx: " introduces an explicit definition of rules **within** the character string. xxx is the name that must be used to terminate the function.

  active: to enable or disable rule file

- or within scenario properties by associating it with one or more rules files

Rules are written according to the following syntax (BNF)

```
RULE            ::= ON CRITERIA DO *ACTION
CRITERIA        ::= CRITERIA_AND *(OR CRITERIA_AND)
CRITERIA_AND    ::= CRITERIA_TERM * (CRITERIA_TERM)
CRITERIA_TERM   ::= (CRITERIA )
CRITERIA_TERM   ::= NOT CRITERIA_TERM
CRITERIA_TERM   ::= (TITLE | CLASS) [NOCASE] (MATCH | REGEXP | EQUAL | CONTAIN) PATTERN
CRITERIA_TERM   ::= (TOP | FOCUS | NEW | CLOSED)
PATTERN         ::= string
ACTION          ::= CLOSE | QUIT | CANCEL | OK | YES | NO | KILL
ACTION          ::= SENDKEY string
ACTION          ::= EXEC string
ACTION          ::= SCRIPT string
```

List of actions to run in order

The criterion is a Boolean expression

Typical action on dialog box type windows

KILL: terminates the application that generated the window

Sends a keystroke sequence to the keyboard

Runs a command line

Runs a script (scenario)

Type of criterion test:

- (TITLE | CLASS): tests the window's title or class
- (MATCH | REGEXP | EQUAL | CONTAIN): type of comparison

Window criterion:

- FOCUS: the focus must be on the window
- TOP: the window must be at the front of the desktop
- NEW: the window must be new
- CLOSED: the window has just disappeared

A file may contain more than one rule.

Example:

```
ON NOCASE TITLE CONTAIN "notepad" AND TOP AND NEW
DO CLOSE KILL
```

Closes any new window whose title contains the character string "notepad" when it appears at the front of the screen. Result: activation of the rule immediately closes any notepad that opens, but does not affect any that were already there.

Further examples:

SetWindowManagerRule("RULE4:ON TITLE NOCASE CONTAIN 'Windows Security' AND NEW DO SENDKEY 'ema{TAB}ip-label~{ENTER}' ",True)

SetWindowManagerRule("RULE1:ON TITLE NOCASE CONTAIN 'Manage AddOns' DO CLOSE ",True)

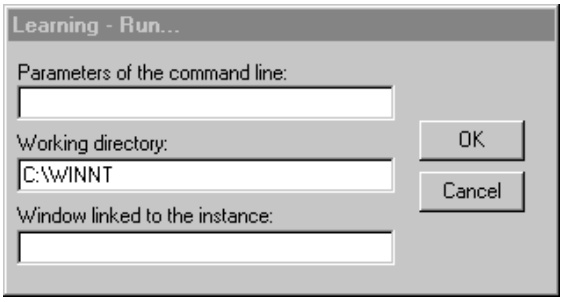
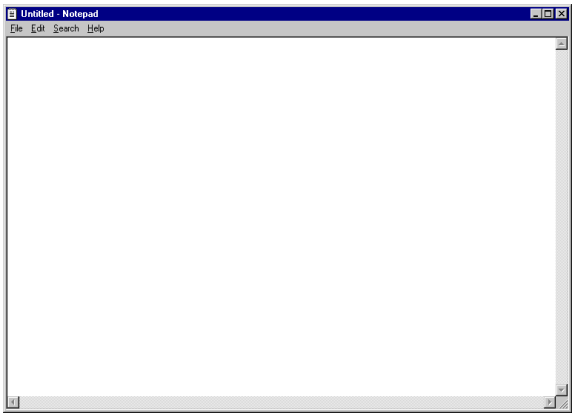
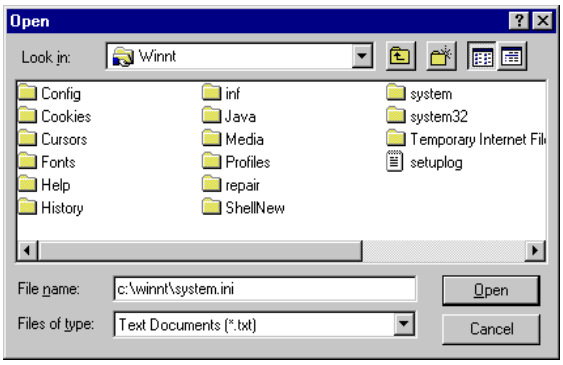
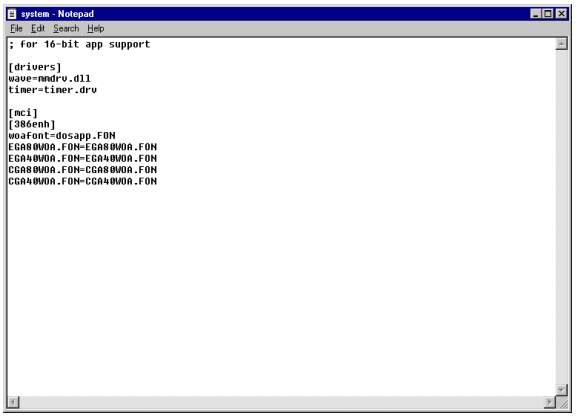
SetWindowManagerRule("RULE3:ON TITLE NOCASE MATCH 'Internet Explorer' DO SENDKEY '~{ENTER}' ",true)

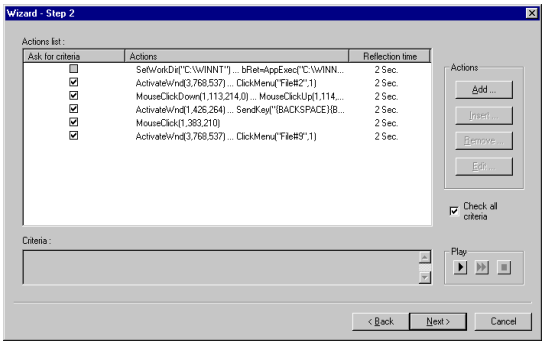
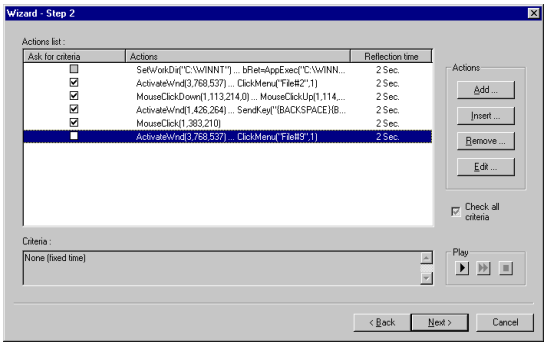
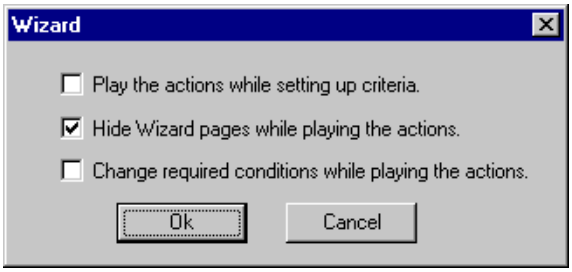
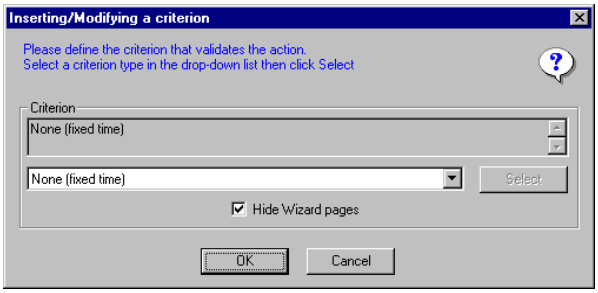
SetWindowManagerRule("RULE2:ON TITLE NOCASE CONTAIN 'Open Document' DO OK ",True)

3.5 EXAMPLE: CONFIGURING A WINDOWS SCENARIO

Below is an example that illustrates the actions required to create a Windows application scenario. The example scenario is designed to load Windows Notepad, open a file, and close Notepad.

Launch the wizard (Step 1): **Learning** menu / **Script generation wizard**, and then start the learning process by clicking **[Next]**.	
STEP 1... Click **Run....**	
Choose the Notepad accessory in WINDOWS.	

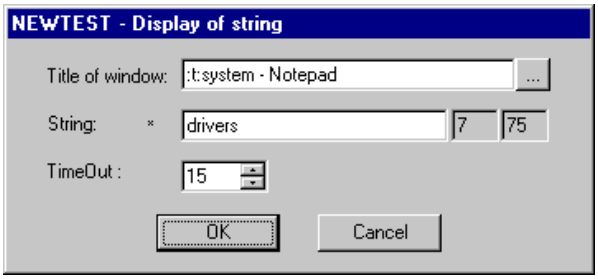
Make sure the working directory is the right one. Click **OK**.	
In the **File** menu, select **Open**.	
Type *c:\winnt\system.ini* in the **File Name** textbox (or *other equivalent file*). Click **Open**.	
In the **File** menu, select **Exit**.	

Click **Terminate**.	
STEP 2... The list of actions generated by the wizard appears. The first action group represents Notepad loading; you cannot change the predefined criterion here (checkbox in **Ask for criteria the column** is unchecked and unavailable). It is good practice to add a comment (after //) to each of the actions that characterize the action in order to facilitate maintenance later on.	
Clear the checkboxes of the action blocks for which you do not want to define criteria. In order to set a criterion for loading the SYSTEM.INI file, the checkbox of the *MouseClick* action group (representing the mouse "click" action on the Open button) must remain checked. Click **Next**.	
If you choose not to play back the actions, NEWTEST proceeds to the step in which you choose criteria (useful when you made a mistake concerning one or more criteria). You can hide the wizard page while defining criteria so that it does not overlap the application. The last option, when available, allows you to select the current object by one of its attributes, for example when the name of the object is variable.	
The scenario is compiled and played back. It stops on each action group whose checkbox is checked. The **Inserting/Modifying a criterion** dialog box prompts you to select the type of criterion to detect. Choose "Display string" in the drop-down list and then click **Select**.	

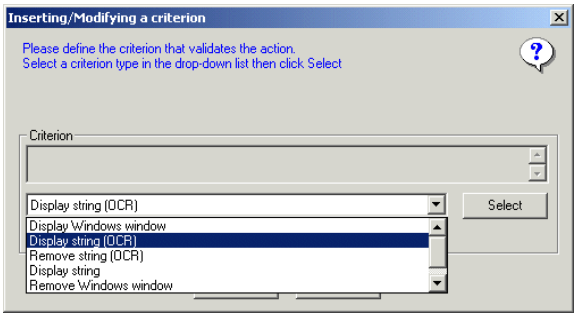
In the settings, click the **[...]** button to select the "system.ini – Notepad" window. Type "drivers" in the String textbox and click Test to make sure NEWTEST locates this string in the designated window.

The criterion for the presence of the "drivers" string in the Notepad window is now defined.

Click **OK** in this dialog box, then click **OK** again in the previous **Insertion / Modification of a criterion** window.

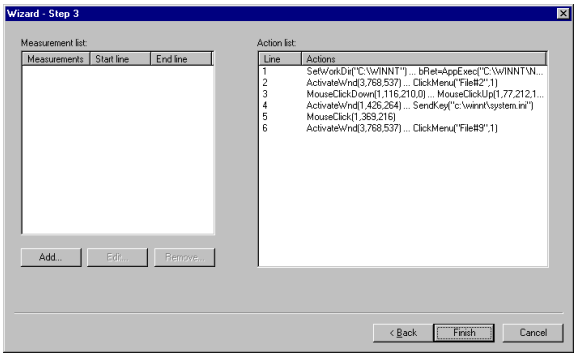


It is also possible to choose a **Display string (OCR)** criterion in order to identify a word or phrase in a display area for which the preceding method would not be adequate (text within an image, for example).

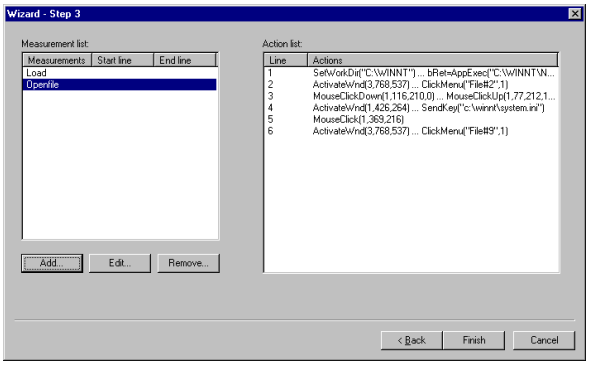


STEP 3...

NEWTEST displays the list of actions. For the time being, no measurement has been defined. Click **Add** and name the first measurement "Load". Click Add a second time and call this new measurement "Openfile".

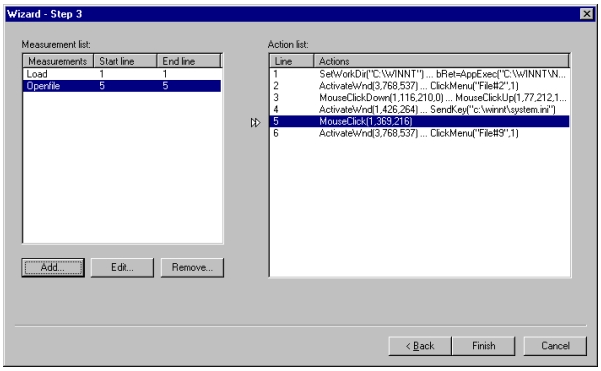


The two measurement names appear in the list.



Click the first block of actions (corresponding to the loading of the NOTEPAD application) and move the selection to the line corresponding to the “Load” measurement. Do the same for the action block corresponding to opening the SYSTEM.INI file (i.e. MouseClick...) for the “Openfile” measurement.

Once the measurement points are defined, click **Finish**.



The complete script of the scenario is generated and displayed.

3.5.1 WINDOWS SCENARIO GENERATED SCRIPT

```
// ————————————————————————————
//   Notepad.SCE - 23/04/2001 11:22:39
// ————————————————————————————
DIM Load AS ORDER(TRESP)
DIM Openfile AS ORDER(TRESP)
DIM TimeOut AS INT
DIM bRet AS BOOL
TimeOut = 15
TopChrono (Load)
SetWorkDir("C:\WINNT")
bRet=AppExec("C:\WINNT\Notepad.exe ", "")
IF bRet=FALSE THEN
        RET Error(Load)
        GOTO ERROR
ELSE
        RET LogTps(Load)
ENDIF
Wait(2)
IF WaitWnd(":t:Untitled - Notepad",TimeOut)=FALSE THEN
        RET ErrorLine("Err001")
        GOTO ERROR
ENDIF
ActivateWnd(3,768,537)
ClickMenu("File#2",1)
IF WaitWnd(":t:Open",TimeOut)=FALSE THEN
        RET ErrorLine("Err002")
        GOTO ERROR
ENDIF
Wait(2)
IF WaitWnd(":t:Open",TimeOut)=FALSE THEN
        RET ErrorLine("Err003")
        GOTO ERROR
ENDIF

ActivateWnd(1,426,264)
SendKey("system.ini~{ENTER}")
IF WaitWnd(":t:system - Notepad",TimeOut)=FALSE THEN
        RET ErrorLine("Err004")
        GOTO ERROR
```

```
ENDIF
IF WaitWndStr("drivers",TimeOut)=FALSE THEN
        RET ErrorLine("Err005")
        GOTO ERROR
ENDIF
Wait(2)
TopChrono (Openfile)
IF WaitWnd(":t:system - Notepad",TimeOut)=FALSE THEN
        RET Error(Openfile)
        GOTO ERROR
ENDIF

ActivateWnd(1,768,537)
ClickMenu("File#9",1)
IF WaitDspWnd(TimeOut)=FALSE THEN
        RET Error(Openfile)
        GOTO ERROR
ELSE
        RET LogTps(Openfile)
ENDIF
Wait(2)
Wait(5)
END
ERROR:
HardCopy(1)
Wait(5)
```

4 SCENARIOS BASED ON THE WEB TRANSACTION CLIENT

4.1 INTRODUCTION

This section discusses the specifics of implementing **Web Transaction Client (WTC)** scenarios as a supplement to information given on the general method of creating scenarios, and addresses issues specific to Web Transaction Client. You must therefore be familiar with the general method (see *Programming scenarios, Developer's Manual vol. 1*) before reading this section.

The **Web Transaction Client environment** is designed to handle web transaction navigation scenarios in a single-user environment in order to supply information about response times as perceived by the user. It is not intended to operate with Newtest Multi-Probe.

4.2 GENERAL

Scenarios based on the Web Transaction Client environment allow you to automate a series of actions that a user would perform on the graphic interface of a web browser.

In order to use the Web Transaction Client, **Microsoft Internet Explorer** or **Firefox** must be installed on the workstation which necessarily runs Windows 2000 or Windows XP. In addition, a RAS connection or IP network access is required to connect to the internet.

4.3 WTC RESTRICTIONS

Please keep in mind the following restrictions when building WTC scenarios:

WTC cannot accommodate URLs that are fully or partially built on the result of a Java script (in other words, when the elements that make up a URL are not all present in the parent page's HTML code)

- WTC does not support downloading of files
- WTC does not support protocols other than HTTP and HTTPS
- WTC cannot reproduce Windows user actions (mouse click or keyboard input) on Java applets or ActiveX controls unless these actions cause a navigation (for example, the response required by security dialog boxes)
- WTC does not provide for measurements of Flash elements or streaming in Internet Explorer
- WTC cannot measure the time it takes to load each individual element of an HTML page
- WTC cannot measure the time it takes to load HTML pages where navigation is considered to be in progress in Internet Explorer until an action is performed by a user

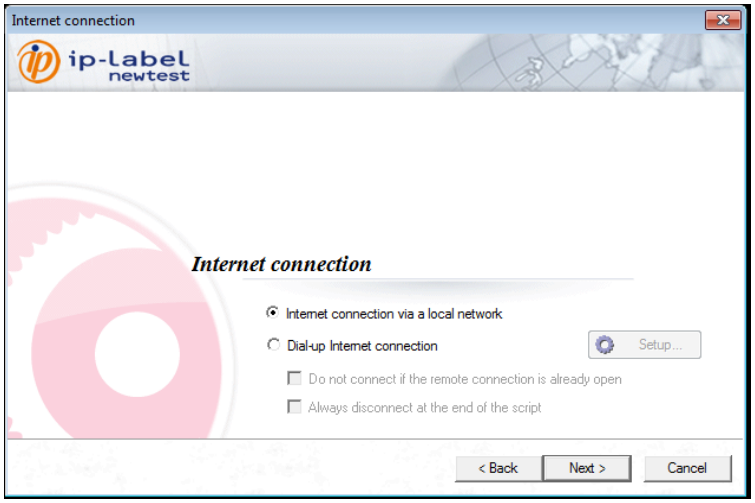
Other restrictions may be identified as website technologies evolve. This list will be updated to reflect such developments.

4.4 BUILDING A WEB TRANSACTION CLIENT SCENARIO

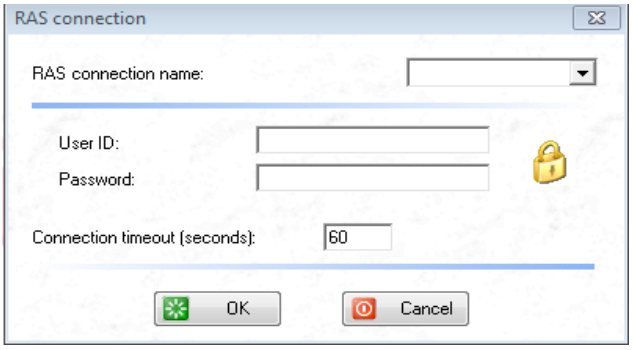
4.4.1 LEARNING MODE

The following page appears at the beginning of the learning phase:

The first step in building a Web Transaction Client scenario consists of choosing either an **Internet connection via a local network** or a **Dial-up Internet connection** (Windows RAS) to access the internet.



If you choose a remote access service connection (**Dial-up Internet connection**), using a modem to connect to the Internet, the **[Setup...]** button becomes available. Click it to display the following setup dialog box, which prompts you to specify a user name and password:



RAS connection name: in the drop-down list, select the name of the RAS connection to use for this scenario. The list shows all the dial-up connections that are configured for Windows.

User ID: enter the user ID (user/login) for this RAS connection.

Password: type the password that corresponds to this user ID.

Connection timeout (seconds): default setting is 60 seconds. This option allows you to set the maximum time in seconds for the opening of a RAS connection before attempt is abandoned (the default value is enough in most cases).

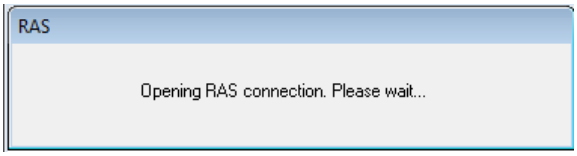
The following two options are also available in the Internet connection window:

Do not connect if the remote connection is already open: check the box to make Newtest use the selected RAS connection if it is already active at the beginning of the scenario. If you do not select this option, Newtest hangs up and then redials.

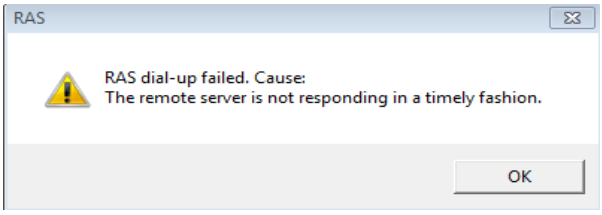
IMPORTANT: if you want to receive statistics showing the breakdown of RAS response times each time a scenario runs, do not select this option.

Always disconnect at the end of the script: select this option to make Newtest systematically hang up at the end of a scenario execution.

Click **[Next]**. If you opted for a RAS connection. Newtest now dials up.

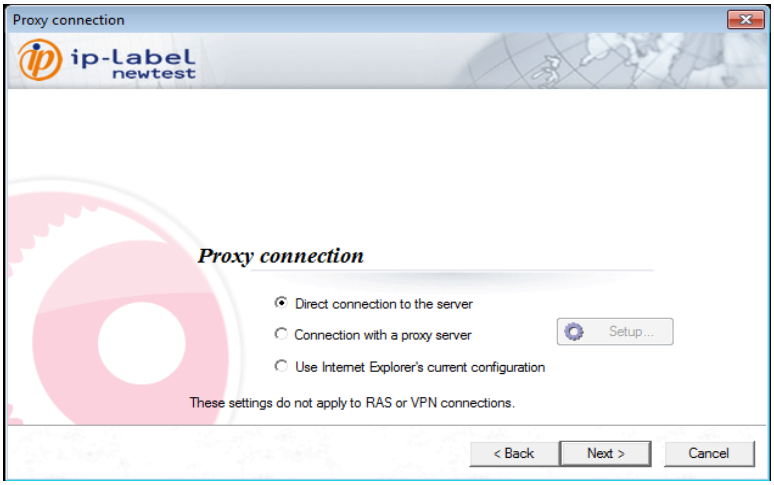


A message appears if the connection fails:



Click **OK** and then make the necessary changes to your RAS settings. Test the connection by clicking **[Next]** again.

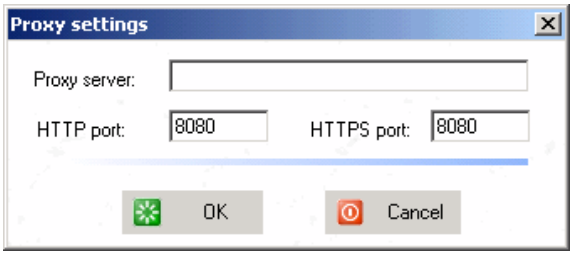
The next step in building the scenario consists of defining settings for the way Internet Explorer connects to the internet:



Direct connection to the server: select this option to make the Web Transaction Client connect directly to the monitored server without using a proxy. If the computer's Internet Explorer options call for a proxy server for connecting to the web, NEWTEST will ignore this configuration and will connect directly.

Connection with a proxy server: select this option to access the web via a proxy server. Click **[Setup...]**.

Next enter the name or IP address of the proxy server for the HTTP and HTTPS protocols. If the computer's Internet Explorer options call for a direct connection to the web (without proxy), NEWTEST will ignore the configuration and will use the proxy parameters that you specify.

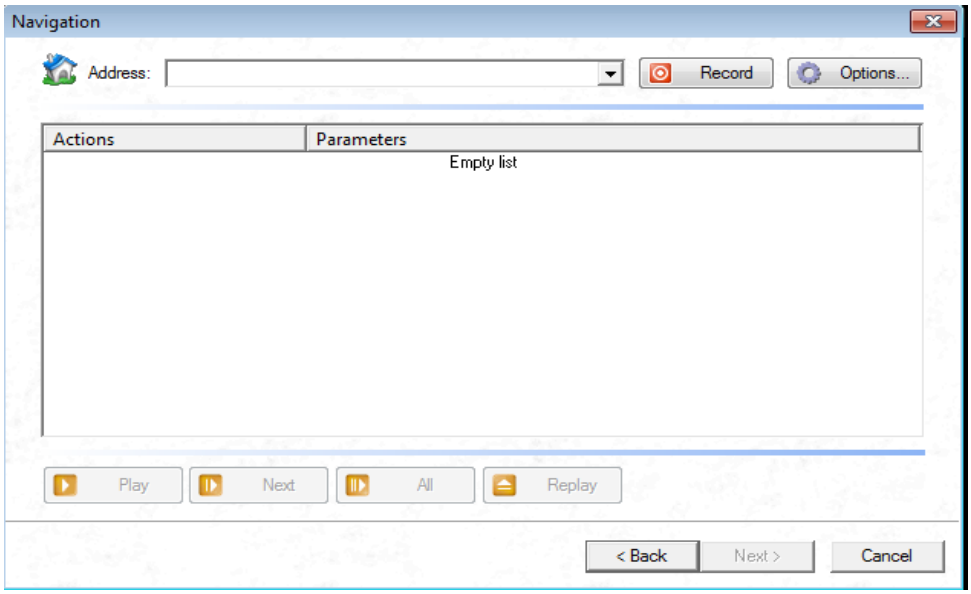


- **Proxy server**: enter the name or IP address of the proxy to use
- **HTTP port**: type the number of the port to use on the proxy for the HTTP protocol.
- **HTTPS port**: type the number of the port to use on the proxy for the HTTPS protocol.

Use Internet Explorer's current configuration: click this radio button if you want the Web Transaction Client to use the connection options defined on the computer's Internet Explorer (direct connection, proxy, use a proxy parameter configuration script, etc...).

For authenticated proxy connections, additional options allow you to set up this security in the next step.

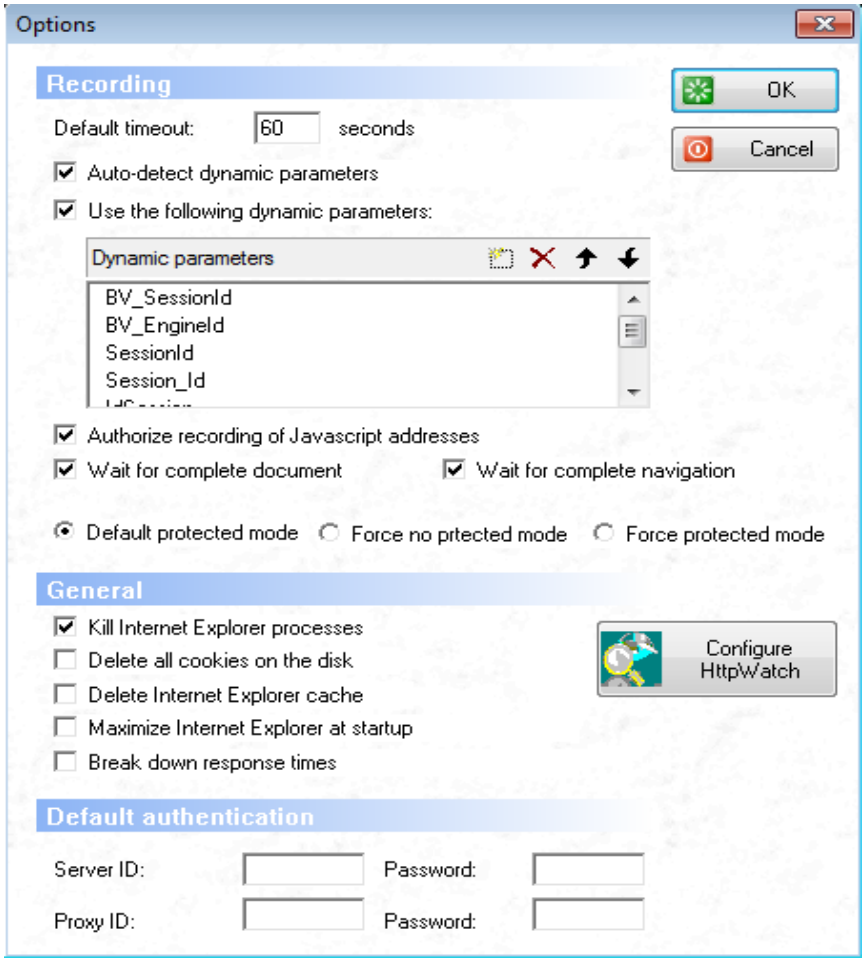
Click **[Next]** to proceed to the Navigation learning steps during which the scenario "learns" the actions it will perform:



In the upper part of the Navigation window, specify the address (URL) to connect to, launch and stop learning, and access settings for the learning phase.

The default settings are appropriate in most cases. If you need to modify them, it is preferable to do so prior to starting the learning phase, as some parameters are called on during recording of the transaction.

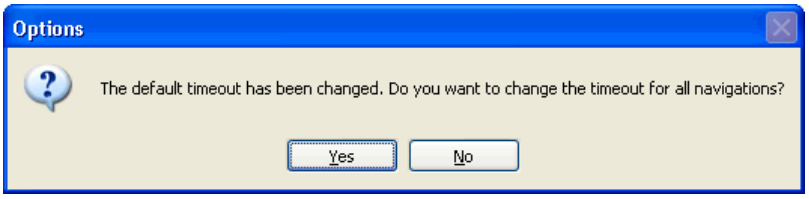
Click the **[Options...]** button:



Recording:

These options affect the modalities of navigation and the way Newtest records the addresses of the monitored site. They are important for the validity of the recorded scenario. For most of these options, modifications cannot take effect until learning is re-launched. This is why it is important to make adjustments prior to the start of recording.

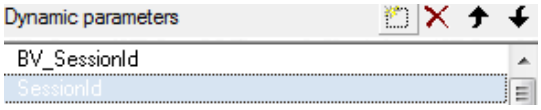
Default timeout (default is 60 seconds): set the maximum lapse of time to fully download pages before they are considered to be unavailable. This value is very important: it determines the tolerance for the full loading time of a page before it signals the unavailability of the monitored site. This value is applied to all the transaction steps to which a specific value has not been assigned. If you change this value after a recording session, a dialog box appears when you close serthe Options page. Indicate whether you want to apply the change without re-recording.



Click **[Yes]** to assign the new default timeout to all the steps of that scenario.

Auto-detect dynamic parameters (enabled by default): check this box if you want Newtest to use its own method of finding dynamic elements in URLs (session IDs, for example). ***It is advisable to leave this box checked*** unless you want to impose manually defined dynamic parameters in the script. For this option to take effect, it is necessary to relaunch the recording session. For more detailed information, refer to the section on *Editing navigation actions*.

Use the following dynamic parameters (enabled by default): Check this box to make Newtest use the elements in the list below as dynamic variables for the URLs (POST data fields and URL parameters):



Click  to add a dynamic parameter name, or  to delete the parameter selected in the list.

This option is useful when Newtest is unable to integrate the dynamic parameter with its **Auto-detect dynamic parameters** option, or simply when you know the list of parameters specific to the monitored site and you want to make sure they are used. For this option to take effect, it is necessary to relaunch the recording session.

Authorize recording of Javascript addresses (enabled by default): check this box to authorize Newtest to record addresses that initiate javascripts (eg: *javascript:document.MINIBE_FORM.BOUTON.value=.../...*). In this way, when the script is replayed, Newtest passes this address to Internet Explorer which executes the associated script in the browser. (Most of the time, the execution of this script moves the navigation to a new page). This function works well for most sites.

However, this approach may not allow the Newtest script to replay correctly afterwards. This happens particularly when the script that is launched is involved in retrieving information that the user must input to the fields in the current page. When the script replays and these fields remain empty (no user input), the script retrieves empty data.

Disabling the recording of javascript addresses makes it possible for Newtest to capture during learning, not the address, but the navigation request generated by the execution of the script (in general a POST request) with all the necessary parameters.

Before deciding to disable the recording of javascript addresses, carry out learning using this option. If you notice that the information entered in the interfaces of the monitored site are not integrated into the generated request, try a new learning session, this time disabling the option.

Wait for complete document: select this option if you want Newtest to consider that the navigation is over only when all of the elements (pages and contents) are loaded. The criteria set for this step will be verified only after the document is fully loaded and complete.

Wait for complete navigation: select this option if you want Newtest to consider that the navigation is over only when Internet Explorer has sent an end-of-navigation message. This option is cumulative with the preceding one to determine that Internet Explorer considers the navigation entirely complete and also that all object loading is finished. **Unselect both options if you want WTC to consider the navigation over as soon as the criteria set for the step in question have been fulfilled.**

General

Kill Internet Explorer processes (enabled by default): check this box if you want Newtest to kill the IEXPLORE process(es) when Internet Explorer does not close normally at the end of a scenario (when Internet Explorer remains open, a dialog box prevents closing, etc.). This is a rare occurrence, but in this way you can ensure that the system is restored if the browser freezes. However, ALL open browser windows will be closed.

Do not uncheck this box unless you are sure that the instances of Internet Explorer are always correctly closed at the end of your scenarios, and you want to avoid every browser instance being systematically closed (this is an extremely rare case).

Delete all cookies on the disk (enabled by default): check this box to make Newtest delete all the cookie files saved to the disk at the end of the scenario. This can be useful in cases where the connection information (user sessions) are stored in this type of cookie. Deleting them at the end of the scenario ensures that the next time the scenario runs, the situation will be exactly the same (and still require, for example, a validation on the site).

Delete Internet Explorer cache: select this option if you want the Internet Explorer cache to be emptied systematically at the beginning of the scenario. The purpose is to avoid using cached objects instead of downloading them directly from the web site, which would alter performance measurements in subsequent tests.

Maximize Internet Explorer at startup: select this option for a full-screen view of Internet Explorer when the script starts up. When enabled, this option ensures a screen shot with a maximum of information for better analysis in case of error. Furthermore, with full-screen, the scenario in question will run on all the robots under the same conditions and with the same behavior.

Break down response times: select this option to break down response times into client time, network time, server time and total time, for each measurement in the scenario (please refer to the explanations in the Annexes for further information).

Configure HttpWatch: this displays the configuration dialogue for the HTTPWatch diagnostic in the event of an unavailability of a scenario step.

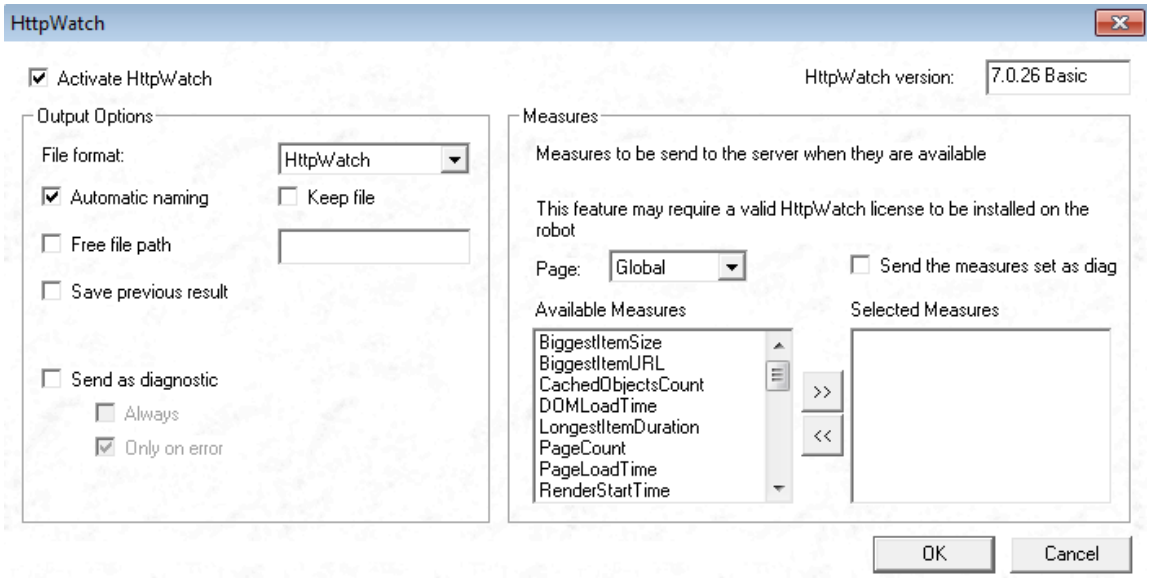
This functionality is available only if HttpWatch is installed on the robot. The version installed determines the extent of the information generated by the diagnostic.

You can propagate the HttpWatch diagnostic globally with the freely available basic edition.

To extract measurements (pageCount, DOMLoadTime) directly from the HttpWatch tool, an enterprise-type (paid) license is required. The 8.x version must be used.

To find out more about HttpWatch, please visit their website: http://www.httpwatch.com

When you click the button [Configure HttpWatch], the following window appears:



Activate HttpWatch: select this option to activate HttpWatch.

Output Options

File format: choose the format of the result file for HTTPWatch: HTTPWatch native, XML or CSV.

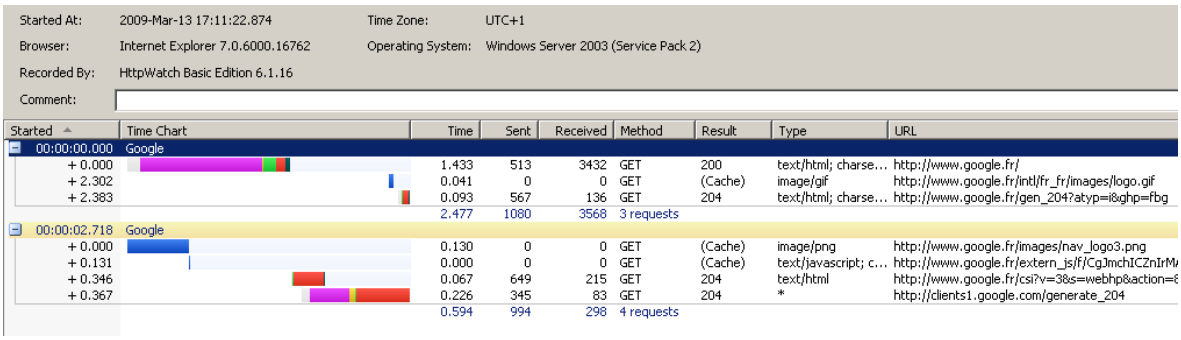
Automatic naming: select this option to make Newtest automatically generate the name of the file it creates. It will be located in the NTBR or Newtest Robot's TMP directory.

Free file path: allows you to name the file (name and path) in which the HTTPWatch result will be generated.

Save previous result: select this option in order to generate a backup (.BCK) of the previous result before generating the new result.

Send as diagnostic: select this option to generate a diagnostic that breaks down pages into objects.

- **Always**: with each execution
- **Only on error**: in the event of unavailability



Measures

Page: select the metrics to transmit according to script page.

Send the measures set as diag: sends derived measurements as a diagnostic.

Default authentication

Server ID / Password: login and password for connection to sites that require validation upon connection (this identification is sent in the URL by the syntax *login:password@http://server_url*)

Proxy ID / Password: login and password for connection to the proxy used to connect to the internet. This option is valid for proxies that require authentication, whether you selected ***Connection with a proxy server*** configured with the Newtest wizard or ***Use Internet Explorer's current configuration*** in the preceding page of the script generation wizard.

Click **[OK]** in the Options window to confirm your choices. To quit the Options settings window without applying the modifications, click **[Cancel]**.

In the **Address** text box, enter the URL where you want to begin the scenario's learning process. Alternatively, select an address in the drop-down list of addresses previously used in the Newtest script generation wizard.

HTTP is the default protocol. If you want to connect to a secure site (SSL), you must type "**HTTPS://**" in front of the URL.

Click the **[Record]** button to start learning.

If a scenario script already exists, a message asks whether you want to overwrite existing actions and measurements. Choose **[No]** in order not to save, or **[Yes]** to confirm that you want to delete the existing actions and measurements.

Internet Explorer opens and loads the URL that was entered in the text box. The **[Record]** button changes into a **[Stop]** button so that you can end learning at any time. The Address field disappears, and a message indicates that recording is underway. You cannot access the **[Options]** button during recording.

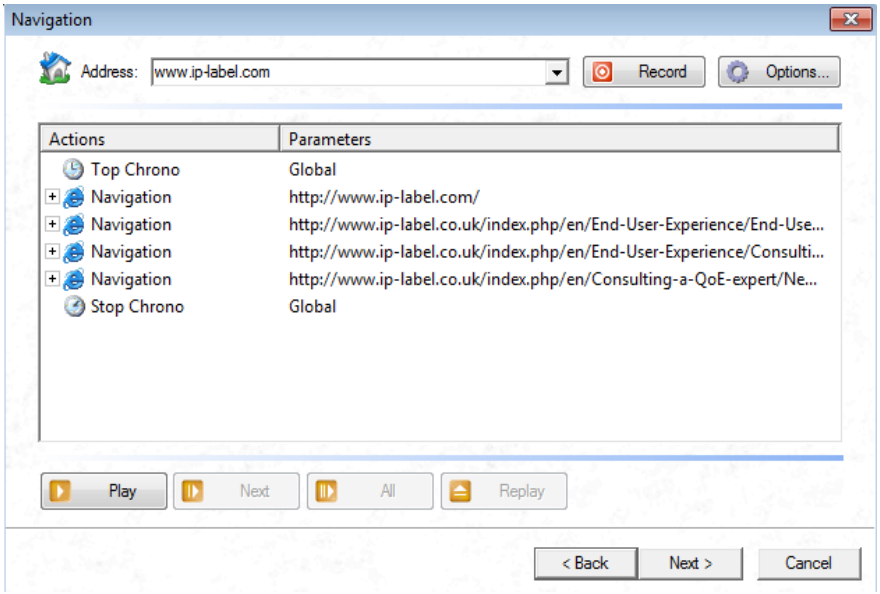
Navigate as you normally would with Internet Explorer. **Use the site in exactly the same way a user would after connecting to the site.** Newtest records the transactions and registers each parameter in order to reproduce the navigation (within the limits of the operating restrictions indicated above in *WTC Restrictions*).

Unlike what usually happens in a browser, Internet Explorer windows disappear at the start of each navigation. This is to prevent actions (mouse clicks or keystrokes) until the page is completely loaded, and guarantees correct recording of the navigation.

NOTE: it may happen that the browser window disappears while you are entering information or even not doing anything in the interface. This is due to the fact that navigation is induced (by a JavaScript, for example). This is a normal occurrence. Simply wait for window to reappear, and proceed to enter your data or click on a new link.

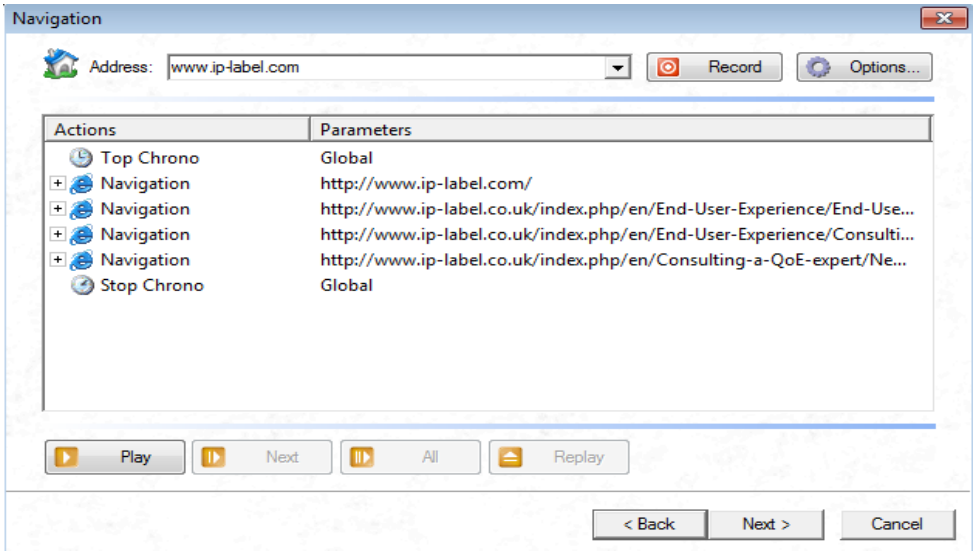
When you have finished the entire navigation, click the **[Stop]** button to bring recording to a close.

Any Internet Explorer windows that were open during the navigation are now closed, and the list of scenario actions appears in the interface of the script generation wizard.



4.4.2 SETTINGS

The list of actions (in the **Actions** column) is a chronological list of navigations resulting from the latest recording session, start and stop chronometers (measurement points) and criteria for the validation of the correct progression of each step (character string checked in the HTML code of downloaded pages):



Each **navigation** is symbolized by the icon and represents the loading of a page or frame by Internet Explorer. It may stem from the browser opening onto a URL, from a click on a link in a loaded page, or any other event (playing of a JavaScript, for example) that leads Internet Explorer to download new elements. The URL of that navigation appears in the **Parameters** column.

The **chronometers** or timers, symbolized by "start" and "stop" icons, encompass the measurement. At the end of recording, Newtest sets a default measurement (*Global*), which starts at the loading of the first URL of the navigation and stops after the last navigation. It is possible to add any number of measurements to the scenario. The **Parameters** column shows the name of the measurement.

Each **control string** is symbolized by an icon and represents the Newtest action that consists of checking for the presence of a character string within the last document loaded in order to verify the correct progression of the navigation (criterion). These actions are dependent on the navigations. Expand the tree (click the +) at the corresponding navigation action to view them. You can add as many criteria as you wish to each navigation. The default criterion is the title of the HTML pages loaded by each navigation (the document's *<TITLE> .../... </TITLE>* tags). The **Parameters** column shows the character string controlled in the HTML page.

4.4.3 PLAYING THE SCENARIO SCRIPT

The navigation bar above the list of actions offers options for playing the scenario, in a single run or step by step.

IMPORTANT: editing of HTML pages and verification of criteria (controlling of strings) can take place only after Newtest plays the corresponding actions. This is because elements must be downloaded from the site in order to be edited.

It is therefore necessary to play the scenario in order to define the settings described hereafter. Whenever necessary, it will be specified whether or not you need to be in "Play" mode to carry out a specific action.

Play: click this button to begin playing the scenario, step by step or in continuous play mode. The engine that plays the scenario script is initialized and the buttons Next and All become available:

Stop: click at any time to stop playing the scenario script.

Next: click to skip to the following action and play it. Each time you click the button, the following action in the scenario script is played (step-by-step mode).

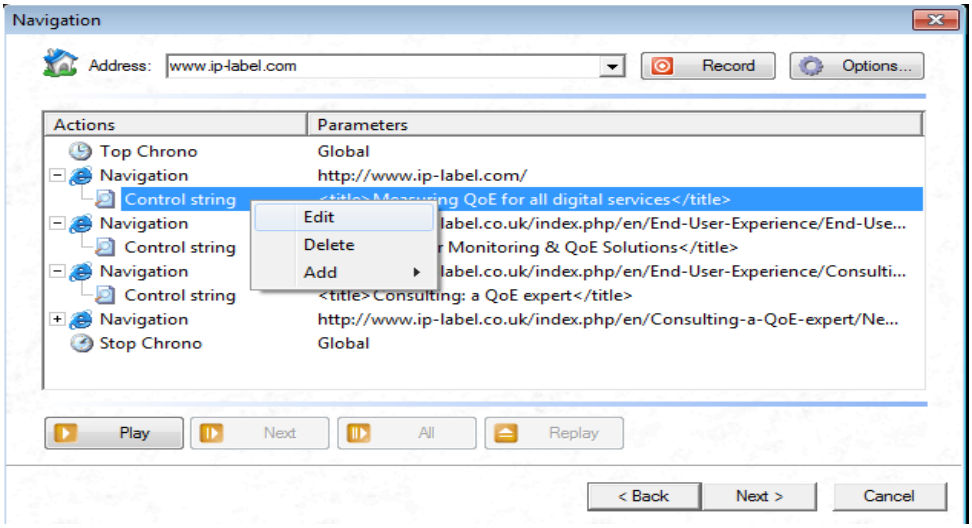
All: click this button to enter continuous play mode. All actions of the script are played one after another from start to finish. You can begin playing the scenario script step-by-step and enter continuous play mode at any time.

Replay (available when at least one action has been played): click to replay the most recently played action. This is useful for checking any changes made (criterion, URL etc.).

4.4.4 EDITING CONTROL STRINGS (CRITERIA)

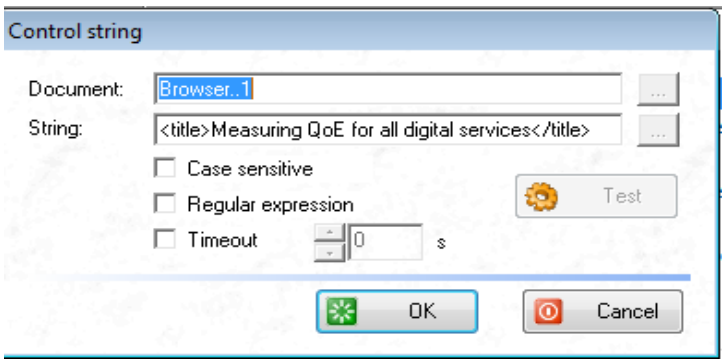
Whenever possible, Newtest automatically sets one criterion per navigation within a scenario. This default criterion is the title of the loaded HTML page (*<TITLE> Titre of the page </TITLE>*). If the page's HTML code does not have *<TITLE></TITLE>* tags, you will have to set your own criteria. In addition, you can add as many criteria as you wish to each navigation.

To edit a control string, double-click the corresponding element in the list of actions, or right-click and select *Edit* in the pop-up menu:



IMPORTANT: editing a control string check is possible ONLY IN PLAY MODE and after you carry out the corresponding navigation.

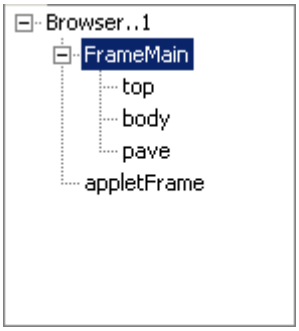
A window for editing the control string appears:



Document: indicates the document (by default, the most recent one downloaded) that is tested for the control string. However, you may need to change the name of the document that appears here.

To edit the entry, click: 

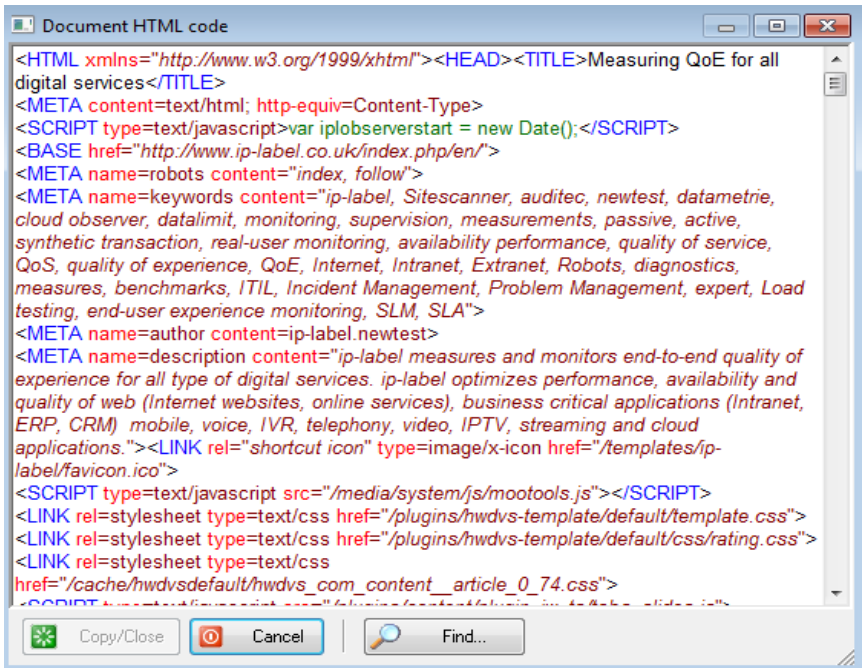
Select a browser in the tree that appears and, if necessary, select the frame in which to search for the control string:



In the example above, **Browser..1** designates the main document of browser 1 (the HTML page loaded in the first browser opened). **FrameMain** is the name of the *FrameSet* element loaded in the document. It, in turn, contains three frames named **top**, **body** and **pave**. Another frame is loaded by the main document in browser 1; it is called **appletFrame**.

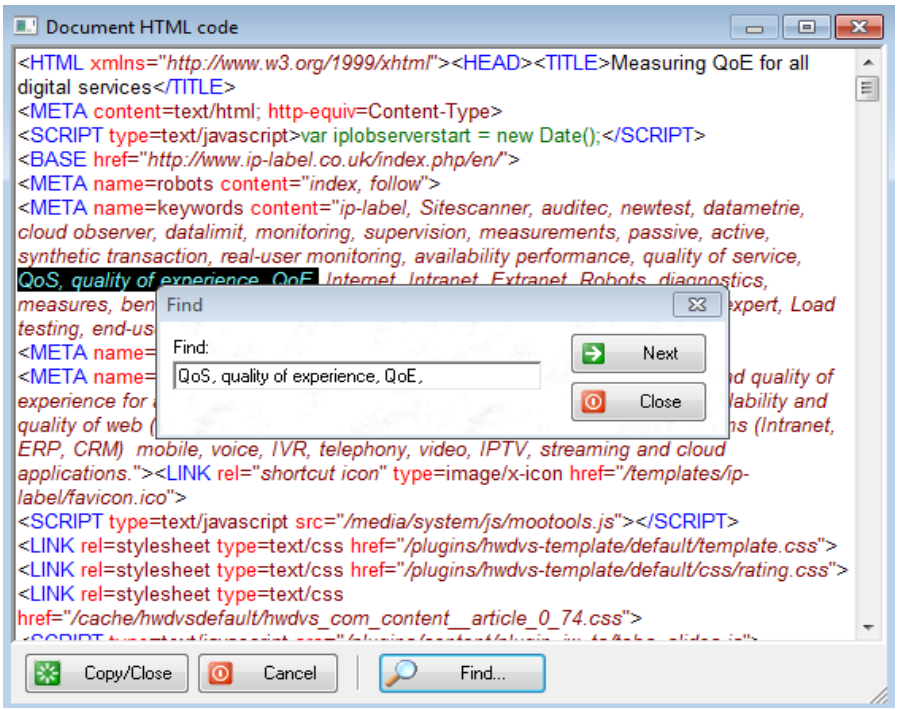
Click the name of the document in which you want search for the control string. A character string designating this document (i.e., its path) appears in the **Document** textbox:

String: in this textbox, enter the string to search for in the HTML code of the designated document, or click the  button to edit the document's source code:



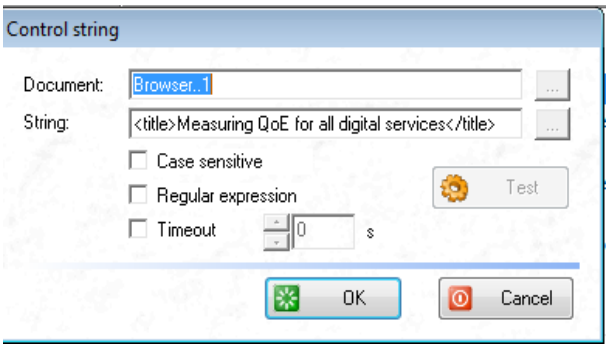
Select the character string in the source code and click **Copy/Close** to use it as your control string.

You can search for a string in the document's source code by clicking **Find**:



In the **Find** textbox, type in the string you want to find and then click **Next** as many times as necessary to locate the specific occurrence that you are seeking. Click **Close** to end the search.

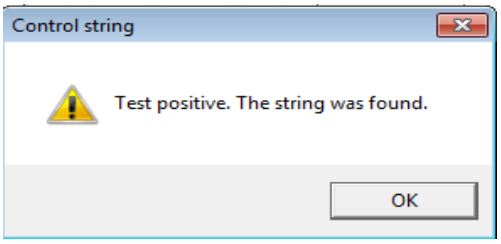
Cancel closes the HTML edit window without taking up the selection.



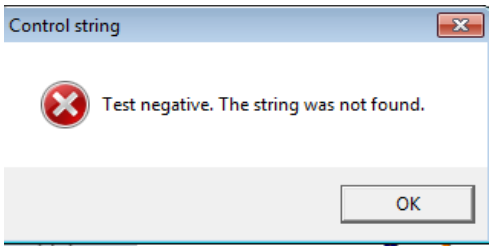
Case sensitive: makes the search sensitive to the case (upper/lower case) of the characters in the control string.

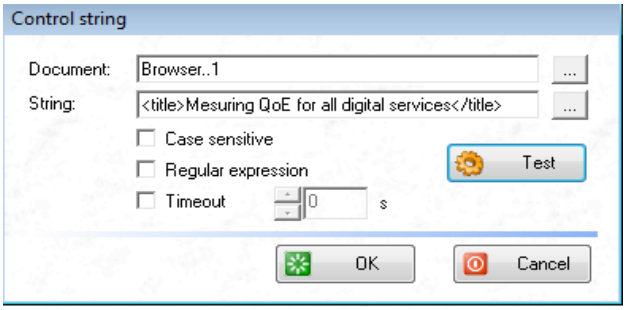
Regular expression: select this option to indicate that the string found in the **String** textbox is expressed in regular expression form (formula containing logical operators for building a variable expression). *Refer to the section on Regular Expressions*.

[Test]: click this button for an immediate test of the control string in the HTML document. A message informs you of the results of the test run:



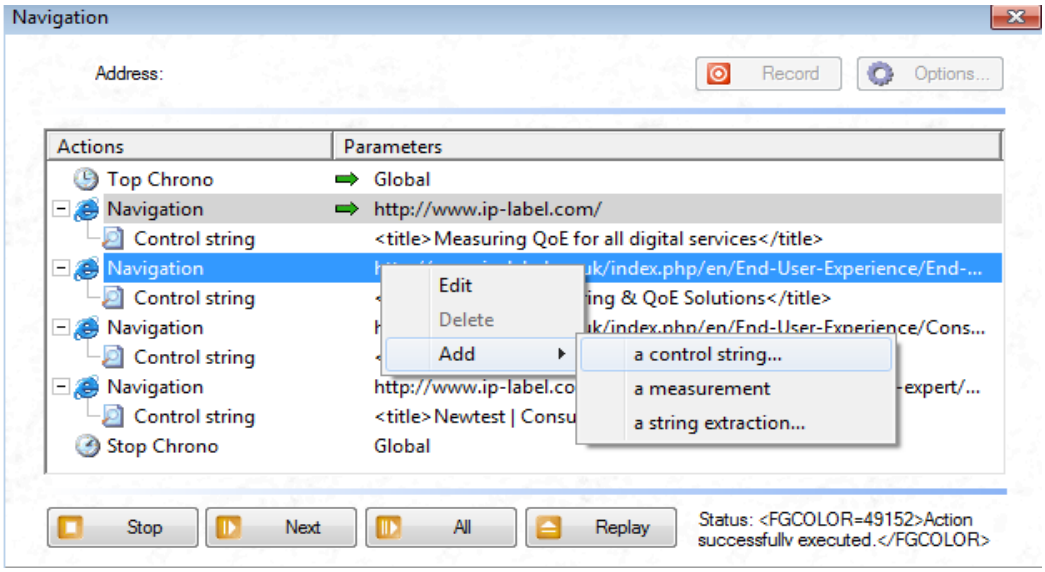
or





Click **[OK]** to validate your control string, or **[Cancel]** to quit without saving the changes.

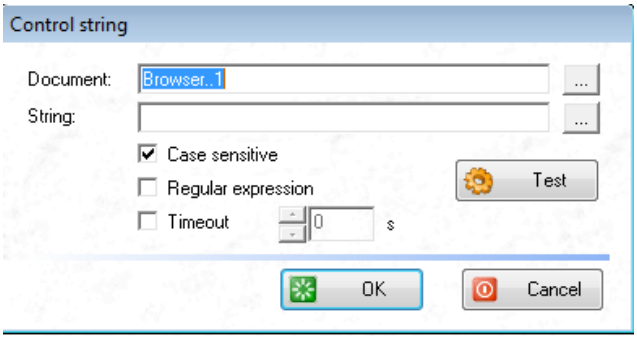
4.4.5 ADDING AND DELETING CONTROL STRINGS



To delete a control string that features in the list of actions, right-click the entry concerned and, in the pop-up menu, select **Delete**.

To add a new control string, right-click the navigation to which you wish to add this new criterion. In the pop-up menu, select **Add** and then select **a control string**.

The control string dialog box appears:



Refer to section on *Editing control strings checks (criteria)* for information about configuration.

You can add as many controls as you wish for a single navigation. The validating condition for this navigation will be all the programmed criteria.

IMPORTANT: Control strings can be added only in REPLAY mode, and after the corresponding navigation has been carried out.

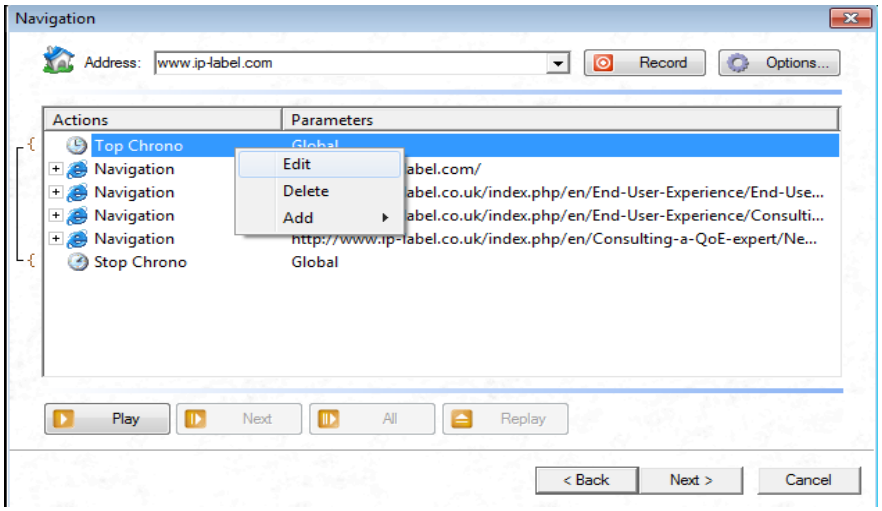
4.4.6 ADDING, EDITING AND DELETING MEASUREMENTS

A scenario's measurement points can be edited or deleted at any time either in replay or in stop mode.

All measurements have a starting point, **Top Chrono**, and an ending point, **Stop Chrono**. Each measurement in a scenario has a unique name limited to 15 characters. You can create aliases with a longer name in the NMC interface.

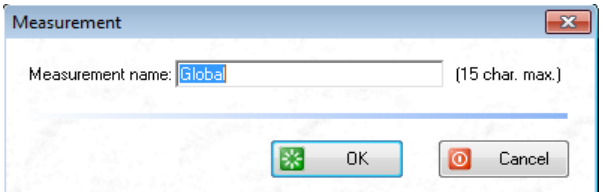
The measurement's starting point (Top Chrono) is the moment at which the chronometer/timer starts ticking. It always precedes the first navigation whose playing time you want to measure. The ending point (Stop Chrono) indicates the when the chronometer/timer stops ticking and returns the result. It always follows the last navigation.

To edit an existing measurement, right-click its starting point or its ending point in the list of actions:



A pop-up menu appears. In it, you can choose to:

Edit a measurement (shortcut – double-click the measurement in the list of actions): to change the name of the measurement in the following dialog box:

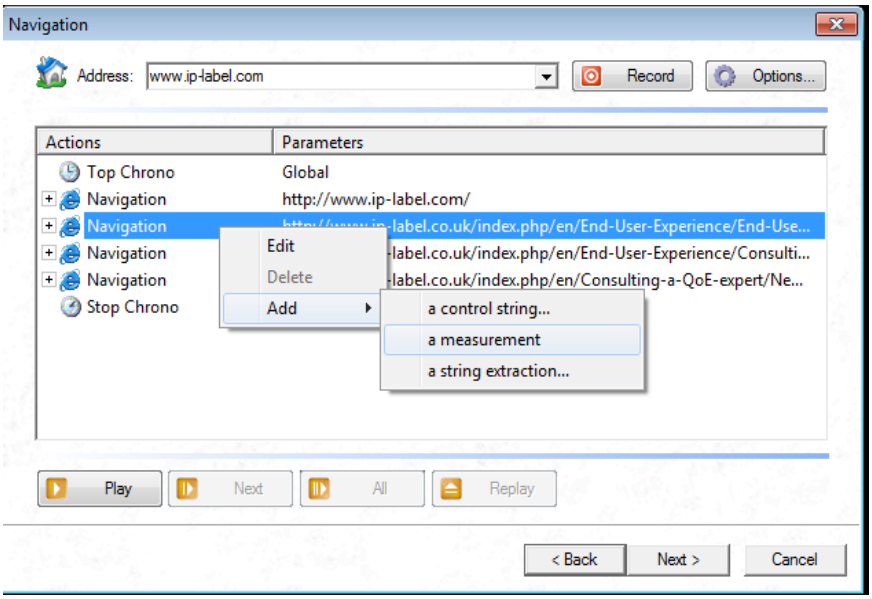


Click **[OK]** to confirm the change, or **[Cancel]** to quit editing without saving changes.

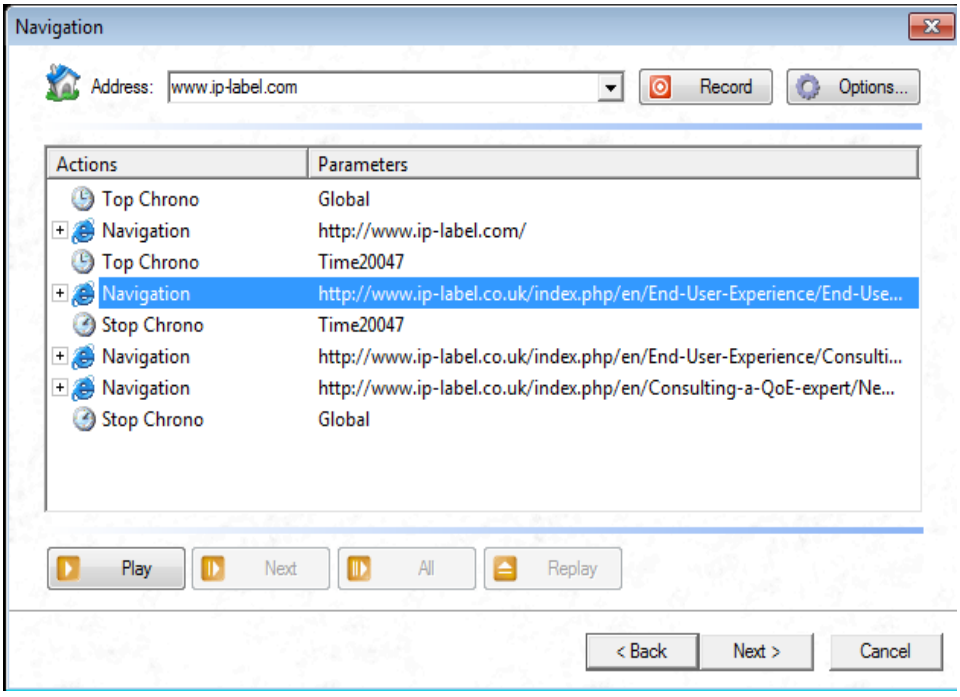
Delete: for deleting the selected measurement. When you choose this option, the two actions (**Top Chrono** and **Stop Chrono**) that correspond to the selected measurement are deleted.

A measurement can be added to a single navigation either in replay or in stop mode. However, to add a measurement covering more than one navigation, you *must* be in STOP mode.

Add: to add a measurement, right-click the navigation whose preceding chronometer/timer must be started in the list of actions. In the pop-up menu, select **Add**, and then **a measurement**:



Two new actions are added to the list: a Top Chrono and a Stop Chrono. The new measurement is automatically assigned a temporary name, such as *Time20047* in the example below:

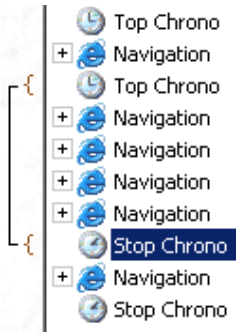


Click once on Top Chrono or Stop Chrono. Holding down the left mouse key, drag the action to the position of your choice in the list. You can thus (outside replay mode) change the start and stop point of the chronometer.

Obviously, you cannot move Stop Chrono to a place in the list above Top Chrono. Likewise, Top Chrono cannot be placed after Stop Chrono.

No two measurements can share the same name.

When you click a measurement in the list (i.e. Top Chrono or Stop Chrono), brackets appear in the left margin and are joined together to show which navigations the measurement encompasses:

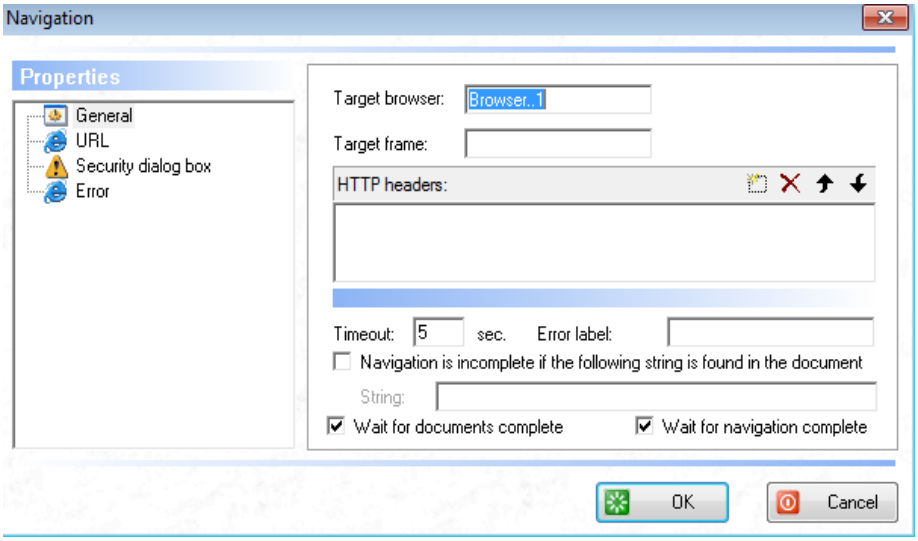


4.4.7 EDITING NAVIGATION ACTIONS

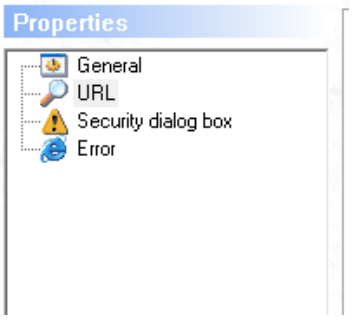
In most cases, the parameters of navigations recorded during the scenario learning phase are correct and operational. Nevertheless it may happen that advanced settings and some tuning are required to adapt the scenario to the security mechanisms and architecture of the monitored site.

Certain topics in this section presuppose a solid knowledge of HTML.

To edit a navigation, right-click its line in the list of actions and then select **Edit** from the pop-up menu (shortcut: double-click the navigation). An edit window appears:



The pane on the left displays a list of the navigation's properties. On the righthand side of the window are shown the settings of the selected property.

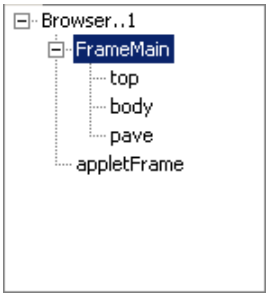


Use the address recorded (default option)**:** select this option if you want Newtest to use the address recorded during learning. You cannot edit the entry.

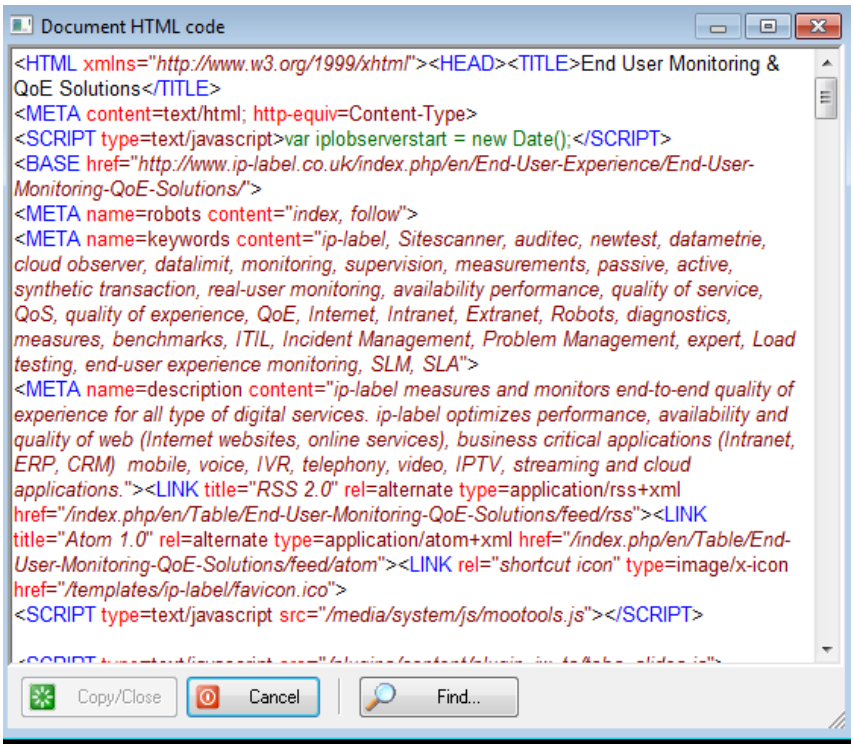
Find the address in the parent document: select this option if the URL's base address for this navigation is variable and has to be found in the HTML code of an already downloaded document.

Parent document: designates the HTML document in which to locate the base URL to be used for the navigation.

Click the  button to display a tree in which you can select the document.



Start delimiter: enter the string that immediately precedes the URL to retrieve. Locate it in the source code of the document designated in **Parent document** by clicking the  button to display the HTML editor. Here you can find and copy the start delimiter:



Occurrence: occurrence of the **start delimiter** in the HTML code. This element is supplied automatically if you use the HTML code editor to define the **start delimiter**.

End delimiter: enter the string that immediately follows the URL to retrieve. Locate it in the source code of the document designated in **Parent document** by clicking the button to display the HTML editor. Here you can find and copy the end delimiter.

Address found: this textbox is not editable. It displays the result of the URL retrieved using the settings defined.

Encoding (none, by default): enables/disables UTF-8 encoding of the retrieved string (URL).

EXAMPLE: You know that the URLs in a document may vary in certain conditions. One of these URLs must systematically be accessed by the Web Transaction Client environment. You also know where the URL appears in the HTML code. Therefore, the only variable part is the value of the URL in the HTML code. The source code containing the Url is:

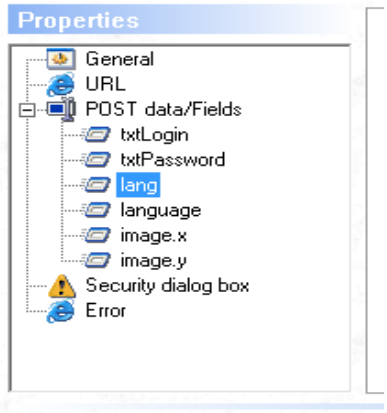
<FORM action=http://fr.search.yahoo.com/search/fr><MAP name=m><AREA alt=Finance coords=0,0,52,52 href="bf" shape=RECT><AREA alt=Tchatche coords=53,0,121,52 href="tc" shape=RECT><AREA alt=Courrier coords=122,0,191,52 href="be" shape=RECT><AREA alt=Nouveautés coords=441,0,510,52 href="jb" shape=RECT><AREA alt="Mon Yahoo!" coords=511,0,579,52 href="bi" shape=RECT><AREA alt=Aide coords=580,0,637,52 href="hf" shape=RECT></MAP><IMG alt="Yahoo! France" border=0 height=60 src="http://eur.i1.yimg.com/eur.yimg.com/i/fr/g/m5.gif" useMap=#m width=638>

The URL to retrieve is always found right after the string "*<FORM action=*" and is closed by the character ">":

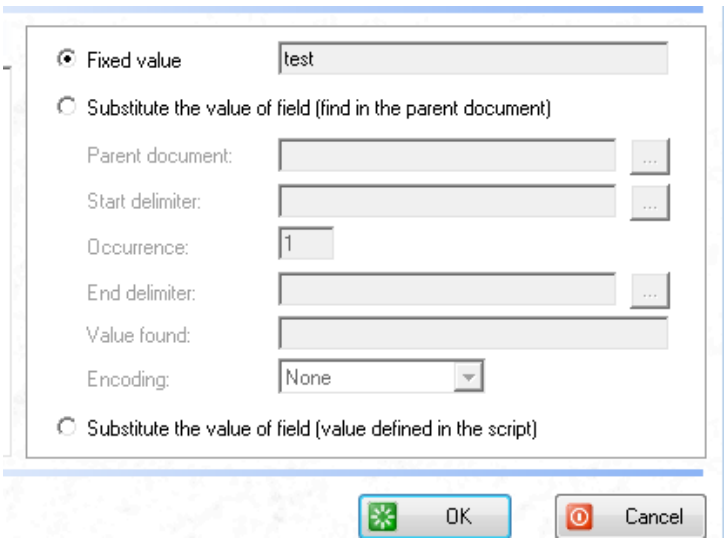
<FORM action=http://fr.search.yahoo.com/search/fr>

In this example, the Start delimiter is "<FORM action=" and the End delimiter is ">".

If the URL has dynamic parameters, each parameter appears in the tree under **Url/Parameters** (_DLG, _CMD, _LANG, etc. in the image below).



When you select one of the URL's dynamic parameters, a panel for setting up its identification and use appears:



In the same way as for defining use of a dynamic URL, you can find the value of the dynamic parameter in an already loaded document.

Fixed value: select this option to keep the value recorded for this parameter during learning (constant, no dynamic parameter).

Substitute the value of field (find in the parent document): to find the parameter's value in the HTML code of an already loaded document.

Parameters of this configuration are identical to those used for finding a dynamic URL (see above).

Substitute the value of field (value defined in the script): externalizes the parameter's value to the scenario script by using a string variable. It is thus possible to control, within the scenario, the initialization of this value.

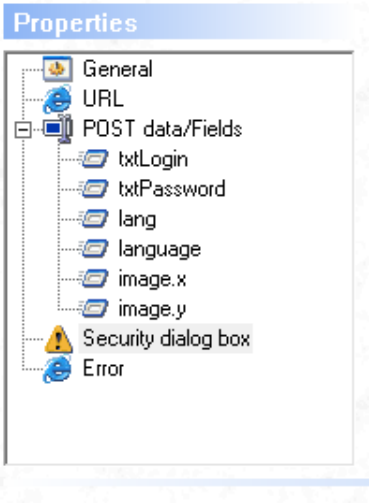
> EXAMPLE: you connect to a bank site and your transaction records the different pages that you consult about an account (statement, transfers, etc.). Each of these pages appears with only a "*SessionID=*" field in the URL, in order to ensure that the user passed the identification phase. Two hours after creating the scenario, these pages no longer display correctly. When you launch a new learning session, search the parent documents of the pages concerned for the session identifier that is present in the recorded URL. You will see the places in the HTML code where it will have to be retrieved systematically.
> This is the sort of code you would see:

FRAME frameBorder=NO marginHeight=0 marginWidth=0 name=FrameWork noResize src="/cgi-bin/emcgi?sessionid=02CA307F161FE84D0568A2A492E5DD29F558ABE6A2AF826F715F09FB10649344"><

The dynamic parameter to retrieve is always located after the string "*sessionid*=" and ends with the character ":

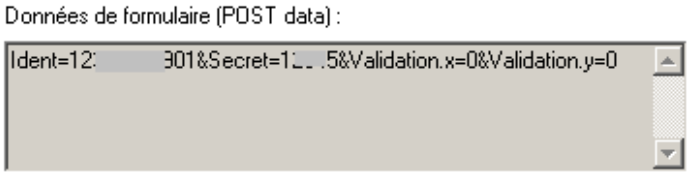
sessionid=02CA307F161FE84D0568A2A492E5DD29F558ABE6A2AF826F715F09FB10649344".

In this example, the Start delimiter is "sessionid=", and the End delimiter is ".



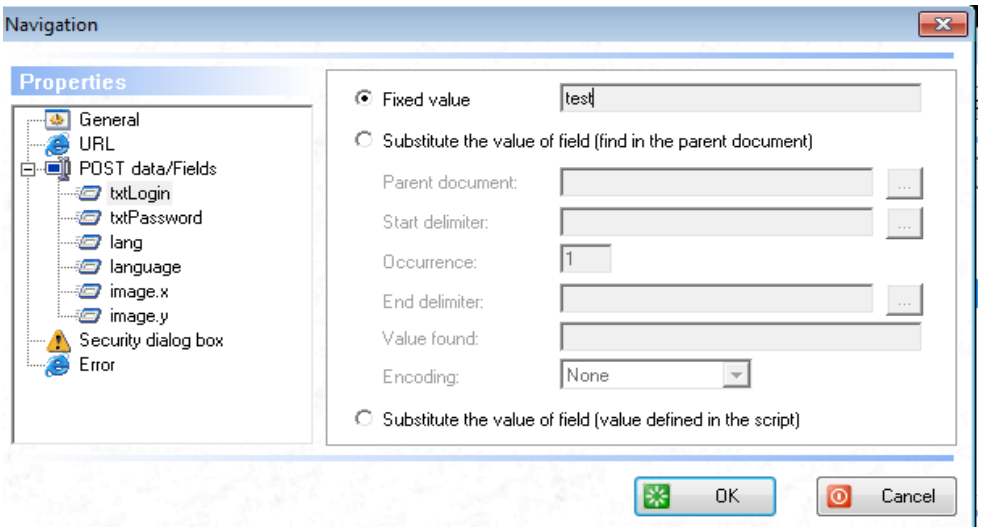
POST data/Fields: as for URL parameters, data in the POST field can have fixed or dynamic parameters.

If you select **POST data/Fields**, the following information appears:



You can view (but not edit) the request's entire POST field.

In the tree under **POST data/Fields** are listed the different fields of the POST. When you click on the name of a field, the following options appear:



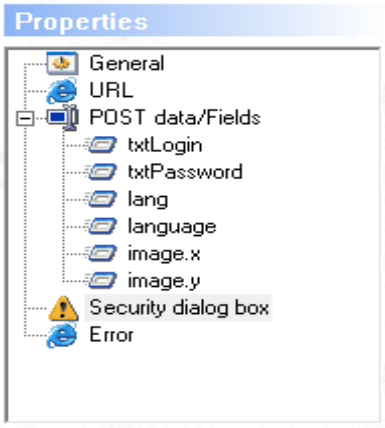
Fixed value: select this option to keep the value recorded for this POST field during learning (constant, no dynamic parameter).

Substitute the value of field (find in the parent document): to find the value of the POST field in the HTML code of an already loaded document.

Parameters of this configuration are identical to those used for finding a dynamic URL (see above).

Substitute the value of field (value defined in the script): externalizes the POST field's value to the scenario script by using a string variable. It is thus possible to control, within the scenario, the initialization of this value.

Dynamic mode for **POST data** operates in the same way as dynamic URL parameters (explained above).



Security dialog box: this parameter allows you to deal with any security dialog boxes (Windows dialog box in Internet Explorer or not) that require keyboard or mouse input from the user. The conditions are: a) that the user action is necessary for the navigation's continuity, and b) that a single action (keystroke or click on a button) is required in order to resume the navigation.

Example: an Internet Explorer security dialog box warning that the user is entering unsecured mode, or a validation window (login/password), etc.

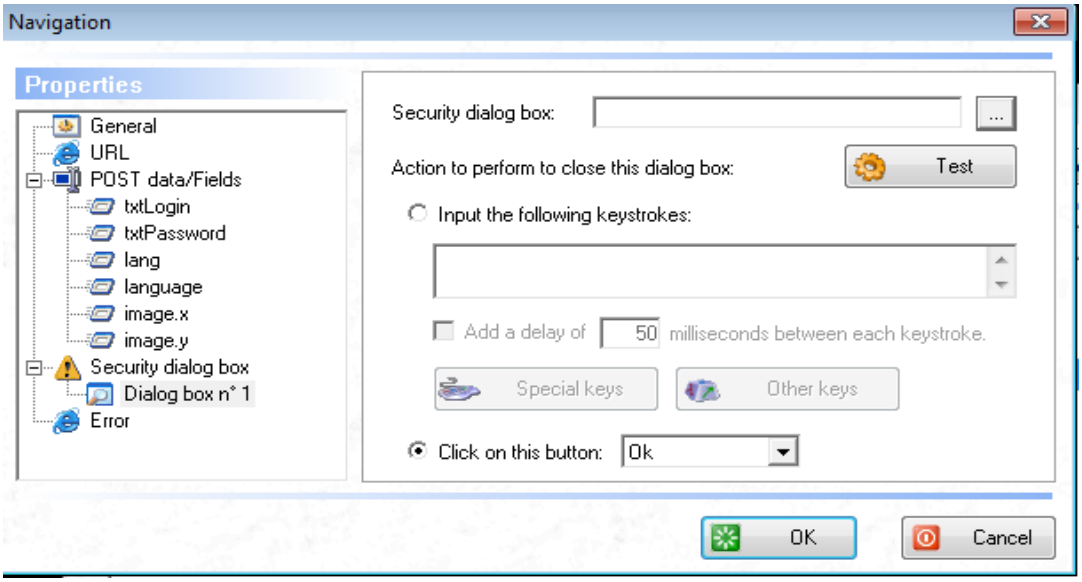
It is always preferable to avoid building into the scenario any Windows dialog boxes that block the navigation. Only security/authentication boxes that the scenario cannot do without should be included in the scenario.

During learning, Newtest automatically detects these elements, whenever feasible, and provides for their integration into the scenario. When this is not possible, or if dialog boxes must be dealt in each later

connection, right-click **Security dialog box** and then select either **New security dialog box** or **New authentication dialog box**.

This action should always be carried out in step-by-step REPLAY mode when the Windows dialog box to navigate is present.

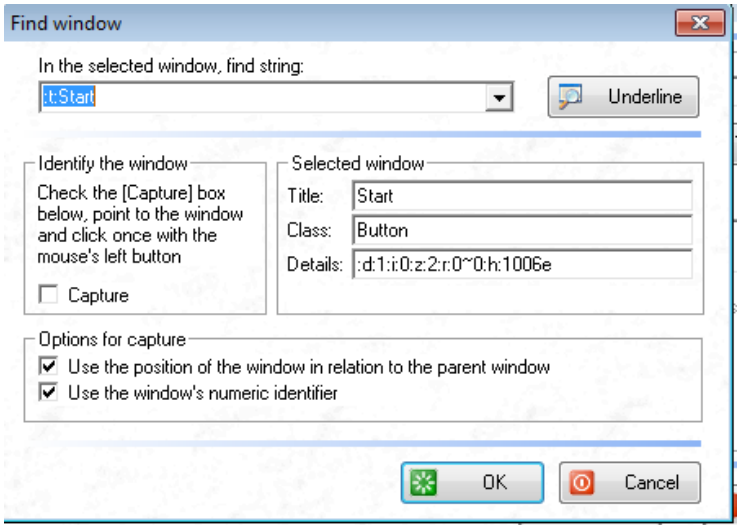
An element appears in the list. Click once to display its **Properties**:



The icon symbolizes a security dialog box

Settings

Security dialog box: allows you to identify the window to navigate. Click the button to launch the **Find window** configuration panel:



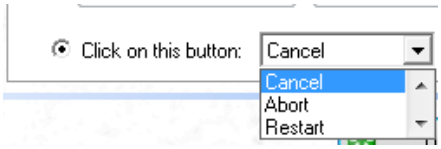
This panel is the equivalent of the configuration interface for the “display window” criterion in the Windows client environment. Refer to *Scenarios based on the Windows Client Environment* for details.

Action to perform to close this dialog box: the two choices are a sequence of keystrokes or a mouse action.

Input the following keystrokes: select this option to send a sequence of keystrokes to the dialog box. In the textbox below, type the sequence of keystrokes to input to the security dialog box. To include in the sequence special keys or key combinations, use the following buttons:

Add a delay of [n] milliseconds between each keystroke: check this box and set a value in milliseconds to slow the keystrokes of the sequence sent to the dialog box.

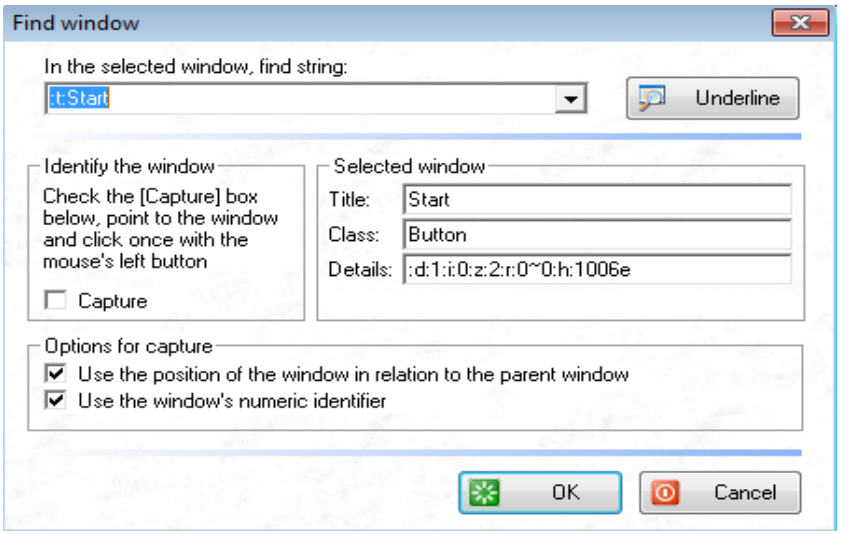
Click on this button: select this option to perform a mouse action (click) on a button in the dialog box. Choose the button to click on in the drop-down list:



The icon symbolizes an authentication dialog box

Settings

Authentication dialog box: allows you to identify the window to navigate. Click the button to launch the **Find window** configuration panel:



This panel is the equivalent of the configuration interface for the “display window” criterion in the Windows client environment. Refer to *Scenarios based on the Windows Client Environment* for details.

Login: enter the identifier to use (*login*).

Password: enter the corresponding password.

The **[Test]** button allows you to make sure the dialog box is correctly dealt with during operation.

4.4.8 EXTRACTING A STRING TO A VARIABLE

In some cases it is necessary to retrieve part of the HTML code of a downloaded document.

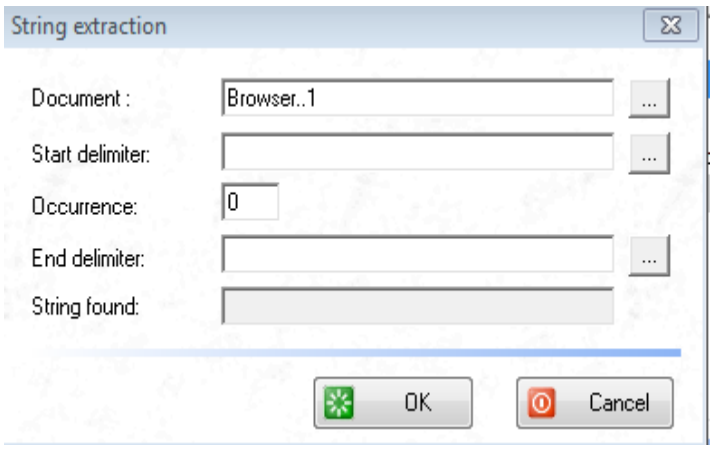
This can make it possible to memorize, in a Newtest variable (STRING), information present on the monitored site and to save this information in Newtest data (ORDER(APPLI)). It is also possible to convert such information into INTEGER data and to perform mathematical operations with it. In addition, this function can also allow you to retrieve several parts of the HTML code in order to build a dynamic URL, for example.

The basic idea is to associate a navigation with one or more strings extracted from the existing HTML document(s).

Right-click the line corresponding to the navigation for which you wish to extract a string and then select **String extraction.**

IMPORTANT: Strings can be extracted only in REPLAY mode, and after the corresponding navigation has been carried out.

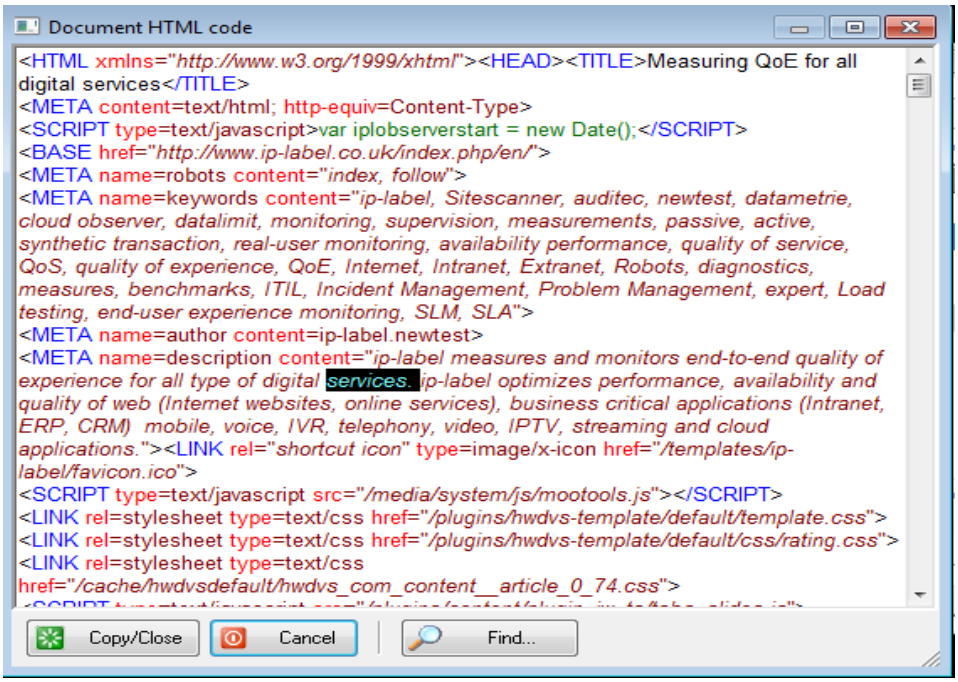
The following dialog box appears:



Document: designates the document from which you want to extract the string. The default document is the most recently downloaded one. However, you may wish to change this setting.

To edit, click the  button to display the browser and, if applicable, the frame from which to extract the string.

Start delimiter: enter the string that immediately precedes the string to capture. Locate it in the source code of the document designated in **Document** by clicking the  button to display the HTML editor. Here you can find and copy the start delimiter:



Occurrence: occurrence of the **start delimiter** in the HTML code. This element is supplied automatically if you use the HTML code editor to define the **start delimiter**.

End delimiter: enter the string that immediately follows the string to extract. Locate it in the source code of the document designated in **document** by clicking the button to display the HTML editor. Here you can find and copy the end delimiter.

String found: this textbox is not editable. It displays the result of the string retrieved using the settings defined, i.e. the string in the selected document that is located between the start delimiter and the end delimiter.

Extracting a string functions in the same was as locating a dynamic URL (see *Editing navigation actions* above).

The specified string will be extracted by the Newtest Basic function **ExtractDocumentString** and will be stored in a *FoundStr* variable.

Whatever the number of extractions programmed for the scenario, the same variable will be automatically used by the script generation wizard to store them.

It is therefore necessary to make the adequate modifications to preserve the information in several different variables if the need arises.

Here is the part of Newtest Basic code added by the string extraction function:

```
FoundStr = ExtractDocumentString(NavId,"Browser..1","Welcome ",1,"<")
```

This command extracts the character string from the main document downloaded in the browser indexed 1. The string is located between the first occurrence of the string *Welcome* and the < , and the result is stored in the STRING variable *FoundStr.*

4.4.9 SPECIAL CASE: DESIGNATING WINDOWS BY NAME

The fields **Target browser** (*Navigation* properties window), **Parent document** (*URL/parameters*) and **Document** (*control string*) designate the browser window concerned by the step or the operation performed. Browser windows are numbered automatically, in the order in which they appear (Browser..1, Browser..2 etc.).

It is advisable to allow Newtest to set these parameters during learning and to refrain from editing them.

Nonetheless, this naming method may cause problems when the scenario is replayed, for instance when popups that were not present during learning (generally advertisements) appear between two steps or, on the other hand, no longer appear after allowance was made for them during learning.

In such cases, the fields can be manually edited in the relevant steps to replace automatic naming by the real name of the document displayed in the window (*Title* field of the HTML document as displayed on screen in the browser's title bar, minus the mention "Microsoft Internet Explorer"). There is now wizard dedicated to this operation. You must enter the exact name of the document in the correct case (case sensitive). The following wildcard and truncation symbols may be used:

¤? : replaces a character within a name

¤* : replaces one or more successive characters

Examples:

"Auditec Home" (full title of the document)

or

"Auditec¤" (document beginning with "Auditec")*

or

"¤ Home" (document ending with "Home")*

Use this solution only if absolutely necessary. It is not infrequent to make mistakes when applying these changes and thereby to cause malfunctions whose cause is particularly difficult to detect.

4.5 SCRIPT GENERATED BY LEARNING

```
//
************************************************************************
// * This script was generated by the NEWTEST - WEB Script Generation Wizard
// * 25/03/2003 10:59:04
//
************************************************************************

DIM User AS STRING
DIM Password AS STRING
DIM bRet AS BOOL
DIM NavId AS INT
DIM Global AS ORDER(TRESP)


// Initialize WEB Engine - Url: www.auditec-newtest.com
// ————————————————
bRet = StartNavigation(1,0,1,2,"",8080,8080)     // Launches Internet Explorer
```

```
IF bRet=FALSE THEN                              // Tests correct launching of IE
        RET ERRORLINE("Erreur interne")
        GoTo LABEL_ERROR
ENDIF
Wait(2)

TopChrono(Global)                               // Starts a chronometer

// Navigation 1 properties
// ———————
NavId = 1
SetNavigationUrl(NavId,"http://www.auditec-newtest.com/")       // Sets the target URL
SetNavigationBrowserName(NavId,"Browser..1")  // Designates the browser
SetNavigationFrameName(NavId,"")                // Designates the frame
SetNavigationUrlSearchMethod(NavId,0,0,"","","",1)        // Defines the dynamic URL


// Navigation 1
// ———————
bRet = PlayNavigation(NavId,60)          // Launches the navigation request
IF bRet=FALSE THEN                       // Tests the result of the navigation
        Ret Error(Global)
        Ret ErrorLine("Echec navigation 1")
        GoTo LABEL_ERROR
ENDIF

// Criteria evaluation for navigation 1
// ———————————————
bRet = CheckDocumentString(NavId,"Browser..1",0,0,"Newtest is Service Level Management (SLM) ")
                                         // Tests the criterion (control string)
IF bRet=FALSE THEN                       //
        Ret Error(Global)                //
        Ret ErrorLine("Echec navigation 1")      //
        GoTo LABEL_ERROR                 //
ENDIF

// Navigation 2 properties                          // Next navigation
// ———————
NavId = 2
SetNavigationUrl(NavId,"http://www.auditec-
newtest.com/uk/pages/principal.asp?f=Support%2FInfos%2Easp")
SetNavigationBrowserName(NavId,"Browser..1")
SetNavigationFrameName(NavId,"")
SetNavigationUrlSearchMethod(NavId,0,0,"","","",1)

// Navigation 2
// ———————
bRet = PlayNavigation(NavId,60)
IF bRet=FALSE THEN
        Ret Error(Global)
        Ret ErrorLine("Echec navigation 2")
        GoTo LABEL_ERROR
ENDIF

// Criteria evaluation for navigation 2
// ———————————————
bRet = CheckDocumentString(NavId,"Browser..1",0,0,"NEWTEST range: Measurement and monitoring of
response times")
```

```
IF bRet=FALSE THEN
        Ret Error(Global)
        Ret ErrorLine("Echec navigation 2")
        GoTo LABEL_ERROR
ENDIF


// Navigation 3 properties
// ———————
NavId = 3
SetNavigationUrl(NavId,"http://www.auditec-newtest.com/uk/pages/Support/infos.asp")
SetNavigationBrowserName(NavId,"Browser..1")
SetNavigationFrameName(NavId,"main")
SetNavigationUrlSearchMethod(NavId,0,0,"","","",1)

// Navigation 3
// ———————
bRet = PlayNavigation(NavId,60)
IF bRet=FALSE THEN
        Ret Error(Global)
        Ret ErrorLine("Echec navigation 3")
        GoTo LABEL_ERROR
ENDIF

// Criteria evaluation for navigation 3
// ———————————————
bRet = CheckDocumentString(NavId,"Browser..1",0,0,"information systems as seen by user")
IF bRet=FALSE THEN
        Ret Error(Global)
        Ret ErrorLine("Echec navigation 3")
        GoTo LABEL_ERROR
ENDIF

Ret LogTps(Global)                      // Stops chronometer; returns value

GoTo LABEL_END

// Error - Hardcopy
// ———————
LABEL_ERROR:
Wait(2)
HardCopy(1)                             // Screenshot for diagnostic

// Close WEB Engine
// ———————
LABEL_END:
Wait(2)
bRet = StopNavigation()                 // Closes browser
IF bRet=FALSE THEN                      // Tests correct closing of browser
        RET ERRORLINE("Erreur interne")
ENDIF


Wait(2)
END
```

4.6 REGULAR EXPRESSIONS

This section presents the regular expressions that can be used for control strings.

These regular expressions are correctly interpreted in control strings only when the option **Regular expression** is selected in the interface (see above, *Adding and deleting control strings*).

The following table contains the complete list of metacharacters and their behavior in the context of regular expressions:

CHARACTER	DESCRIPTION
\	Marks the next character as either a special character, a literal, a backreference, or an octal escape. For example, 'n' matches the character "n". '\n' matches a newline character. The sequence '\\' matches "\" and "\(" matches "(".
^	Matches the position at the beginning of the input string.
$	Matches the position at the end of the input string.
*	Matches the preceding subexpression zero or more times. For example, zo* matches "z" and "zoo". * is equivalent to {0,}.
+	Matches the preceding subexpression one or more times. For example, 'zo+' matches "zo" and "zoo", but not "z". + is equivalent to {1,}.
?	Matches the preceding subexpression zero or one time. For example, "do(es)?" matches the "do" in "do" or "does". ? is equivalent to {0,1}
{*n*}	*n* is a nonnegative integer. Matches exactly *n* times. For example, 'o{2}' does not match the 'o' in "Bob," but matches the two o's in "food".
{*n*,}	*n* is a nonnegative integer. Matches at least *n* times. For example, 'o{2,}' does not match the "o" in "Bob" and matches all the o's in "foooood". 'o{1,}' is equivalent to 'o+'. 'o{0,}' is equivalent to 'o*'.
{*n*,*m*}	*m* and *n* are nonnegative integers, where *n* <= *m*. Matches at least *n* and at most *m* times. For example, "o{1,3}" matches the first three o's in "fooooood". 'o{0,1}' is equivalent to 'o?'. Note that you cannot put a space between the comma and the numbers.
?	When this character immediately follows any of the other quantifiers (*, +, ?, {*n*}, {*n*,}, {*n*,*m*}), the matching pattern is non-greedy. A non-greedy pattern matches as little of the searched string as possible, whereas the default greedy pattern matches as much of the searched string as possible. For example, in the string "oooo", 'o+?' matches a single "o", while 'o+' matches all 'o's.
.	Matches any single character except "\n". To match any character including the '\n', use a pattern such as '[.\n]'.
(*pattern*)	Matches *pattern* and captures the match. The captured match can be retrieved from the resulting Matches collection, using the SubMatches collection in VBScript or the $0...$9 properties in JScript. To match parentheses characters ( ), use '\(' or '\)'.
(?:*pattern*)	Matches *pattern* but does not capture the match, that is, it is a non-capturing match that is not stored for possible later use. This is useful for combining parts of a pattern with the "or" character (|). For example, 'industr(?:y|ies) is a more economical expression than 'industry|industries'.
(?=*pattern*)	Positive lookahead matches the search string at any point where a string matching *pattern* begins. This is a non-capturing match, that is, the match is not captured for possible later use. For example 'Windows (?=95|98|NT|2000)' matches "Windows" in "Windows 2000" but not "Windows" in "Windows 3.1". Lookaheads do not consume characters, that is, after a match occurs, the search for the next match begins immediately following the last match, not after the characters that comprised the lookahead.
(?!*pattern*)	Negative lookahead matches the search string at any point where a string not matching *pattern* begins. This is a non-capturing match, that is, the match is not captured for possible later use. For example 'Windows (?!95|98|NT|2000)' matches "Windows" in "Windows 3.1" but does not

	match "Windows" in "Windows 2000". Lookaheads do not consume characters, that is, after a match occurs, the search for the next match begins immediately following the last match, not after the characters that comprised the lookahead.
x|*y*	Matches either *x* or *y*. For example, 'z|food' matches "z" or "food". '(z|f)ood' matches "zood" or "food".
[*xyz*]	A character set. Matches any one of the enclosed characters. For example, '[abc]' matches the 'a' in "plain".
[^*xyz*]	A negative character set. Matches any character not enclosed. For example, '[^abc]' matches the 'p' in "plain".
[*a-z*]	A range of characters. Matches any character in the specified range. For example, '[a-z]' matches any lowercase alphabetic character in the range 'a' through 'z'.
[^*a-z*]	A negative range characters. Matches any character not in the specified range. For example, '[^a-z]' matches any character not in the range 'a' through 'z'.
\b	Matches a word boundary, that is, the position between a word and a space. For example, 'er\b' matches the 'er' in "never" but not the 'er' in "verb".
\B	Matches a nonword boundary. 'er\B' matches the 'er' in "verb" but not the 'er' in "never".
\c*x*	Matches the control character indicated by *x*. For example, \cM matches a Control-M or carriage return character. The value of *x* must be in the range of A-Z or a-z. If not, c is assumed to be a literal 'c' character.
\d	Matches a digit character. Equivalent to [0-9].
\D	Matches a nondigit character. Equivalent to [^0-9].
\f	Matches a form-feed character. Equivalent to \x0c and \cL.
\n	Matches a newline character. Equivalent to \x0a and \cJ.
\r	Matches a carriage return character. Equivalent to \x0d and \cM.
\s	Matches any whitespace character including space, tab, form-feed, etc. Equivalent to [\f\n\r\t\v].
\S	Matches any non-white space character. Equivalent to [^ \f\n\r\t\v].
\t	Matches a tab character. Equivalent to \x09 and \cI.
\v	Matches a vertical tab character. Equivalent to \x0b and \cK.
\w	Matches any word character including underscore. Equivalent to '[A-Za-z0-9_]'.
\W	Matches any nonword character. Equivalent to '[^A-Za-z0-9_]'.
\x*n*	Matches *n*, where *n* is a hexadecimal escape value. Hexadecimal escape values must be exactly two digits long. For example, '\x41' matches "A". '\x041' is equivalent to '\x04' & "1". Allows ASCII codes to be used in regular expressions.
num	Matches *num*, where *num* is a positive integer. A reference back to captured matches. For example, '(.)\1' matches two consecutive identical characters.
n	Identifies either an octal escape value or a backreference. If *n* is preceded by at least *n* captured subexpressions, *n* is a backreference. Otherwise, *n* is an octal escape value if *n* is an octal digit (0-7).
nm	Identifies either an octal escape value or a backreference. If *nm* is preceded by at least *nm* captured subexpressions, *nm* is a backreference. If *nm* is preceded by at least *n* captures, *n* is a backreference followed by literal *m*. If neither of the preceding conditions exists, *nm* matches octal escape value *nm* when *n* and *m* are octal digits (0-7).
nml	Matches octal escape value *nml* when *n* is an octal digit (0-3) and *m* and *l* are octal digits (0-7).
\u*n*	Matches *n*, where *n* is a Unicode character expressed as four hexadecimal digits. For example, \u00A9 matches the copyright symbol (©).

Note: In a Windows environment regular expressions can also be used in the OCR wizard (see Volume 1, §2.5.7), by checking the box for this option in the wizard:

5 SCENARIOS BASED ON THE IP ANALYSER CLIENT ENVIRONMENT

5.1 INTRODUCTION TO IP ANALYSER SCENARIOS

This section discusses the specifics of implementing **IP Analyser** scenarios as a supplement to information given on the general method of creating scenarios, and addresses issues specific to IP Analyser. You must therefore be familiar with the general method (see *Programming scenarios, Developer's Manual vol. 1*) before reading this section.

When IP Analyser operates with **Newtest Multi-Probe**, the scenarios run in parallel queues using their specific means of communication (modem or other).

It is therefore necessary, when creating RAS connection entries, to **plan to assign resources to each queue** in order to avoid competition for access to communications elements during scenario execution.

5.2 PREPARATION

Microsoft Internet Explorer 6 or higher must be installed. If you are using an Internet connection, the **RAS** (Remote Access Service) must also be installed.

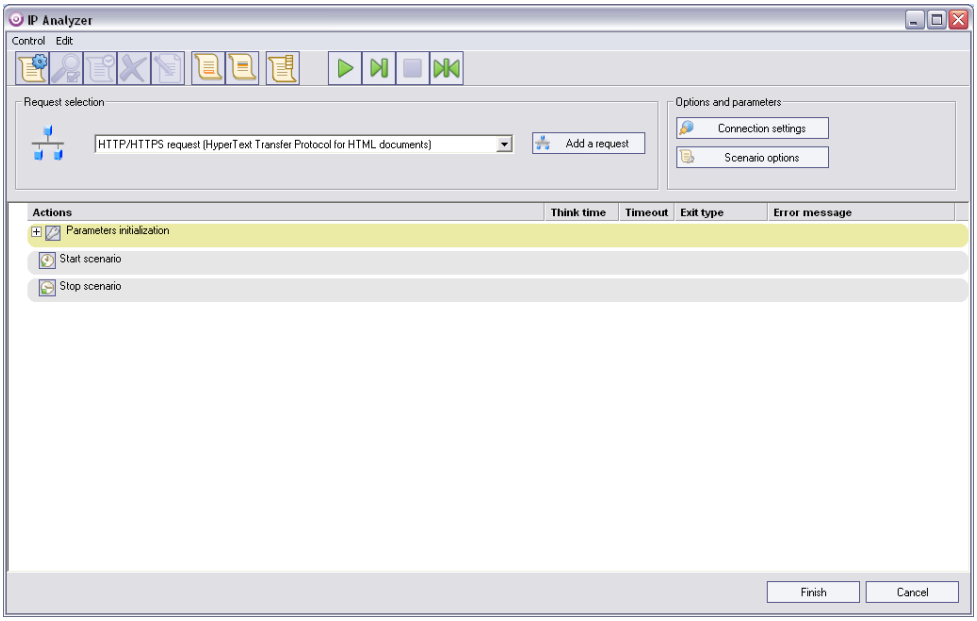
Make sure each of your Internet/Intranet applications runs with Internet Explorer (configure the application for this browser if necessary).

5.3 CONFIGURING IP ANALYSER SCENARIOS

Define scenario settings as described in *Programming scenarios (Newtest Developer's Manual vol. 1)*, selecting **IP Analyser** as the client environment.

5.3.1 BUILDING AN IP ANALYSER SCENARIO WITH THE SCRIPT GENERATION WIZARD

The main window of the script generation wizard for the IP Analyser client environment allows you to set the list of requests to run in addition to the scenario's parameters and options.



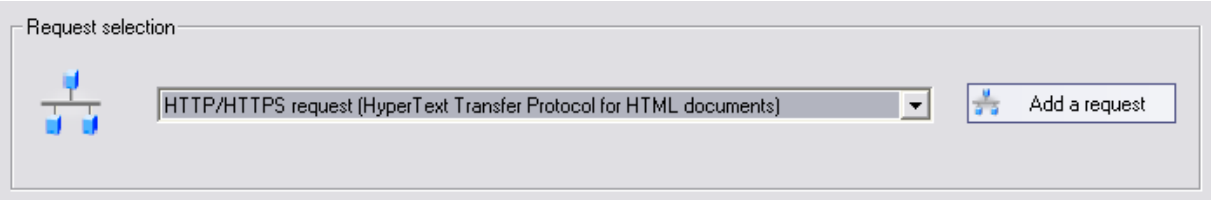
The upper part of the interface is for selecting the protocol on which to create a test. The button Add a request adds this test to the list of existing requests.

5.3.2 CREATING AN IP ANALYSER REQUEST

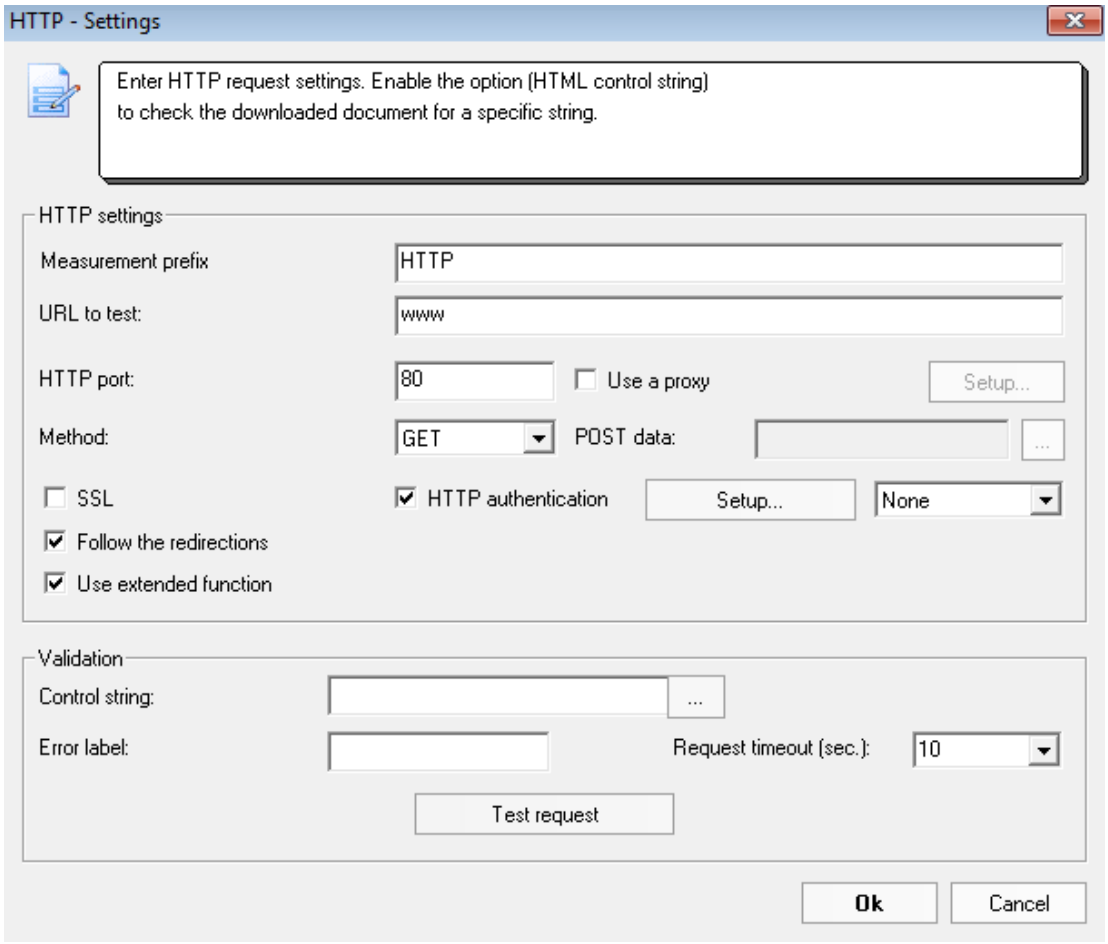
In the **Request selection** drop-down list, select the type of request to add:

- HTTP/HTTPS request (GET/PUT test of a web or intranet site)
- FTP request (GET or PUT test of an FTP server)
- Port test (connection to a TCP port)
- DNS request (resolution of an address on a given DNS server)
- PING request
- SMTP request (sending of a message)
- POP3 request (check of the reception of a message on a POP3 server)
- IMAP4 request (check of the reception of a message on an IMAP4 server)
- SMTP/POP3 request (full test of the sending and receiving of a message with a POP3 server)
- SMTP/IMAP4 request (full test of the sending and receiving of a message with a IMAP4 server)
- LDAP request (check of the presence of an entry in an LDAP database)
- NNTP request (check of the presence of a newsgroup)

5.3.2.1 CONFIGURING AN HTTP REQUEST



The wizard's **HTTP - Settings** page offers options for setting up a GET or POST HTTP request:



Measurement prefix: allows you to set a prefix for the measurements made using this request in the scenario. A prefix is indispensable for identifying the measurements of a request uniquely with respect to those of other requests and transmitting the information in an identifiable way to NMC.

The prefix you indicate will be added, followed by an underscore (_), to the name of each measurement. The maximum length of the prefix is 6 characters.

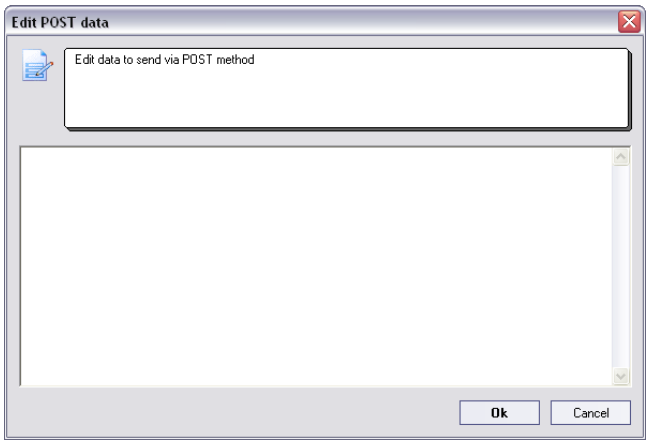
URL to test: enter the URL on which the request will be performed (***without*** the **http://** prefix).

HTTP port: type the port of the HTTP server. (Default port is 80 for HTTP; if you have selected SSL, the default port is 443).

Connection via proxy server: select this option if the HTTP server is accessed via a proxy server. Click **Setup...** to define proxy server access settings (see *Proxy server properties* below).

Method: select the HTTP method to use (GET or POST). GET is the default method.

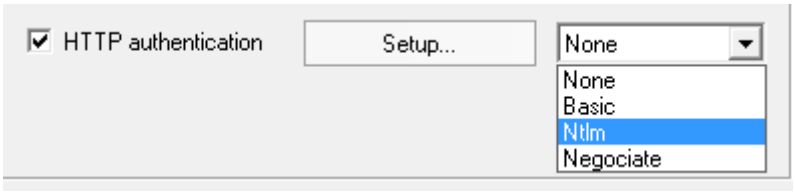
POST data: enter POST data (textbox is available only if you selected POST as your Method). For a more readable data entry field when dealing with lengthy POST parameters, click  :



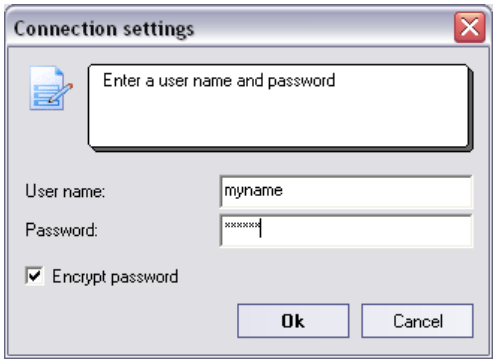
Use SSL: check the box if you are using SSL.

HTTP authentication: opt for this feature if connection to the site requires a login and password.

Choose first the type of authentication (Basic HTTP digest, NTLM, etc.):



Click [**Setup**] to enter authentication information:

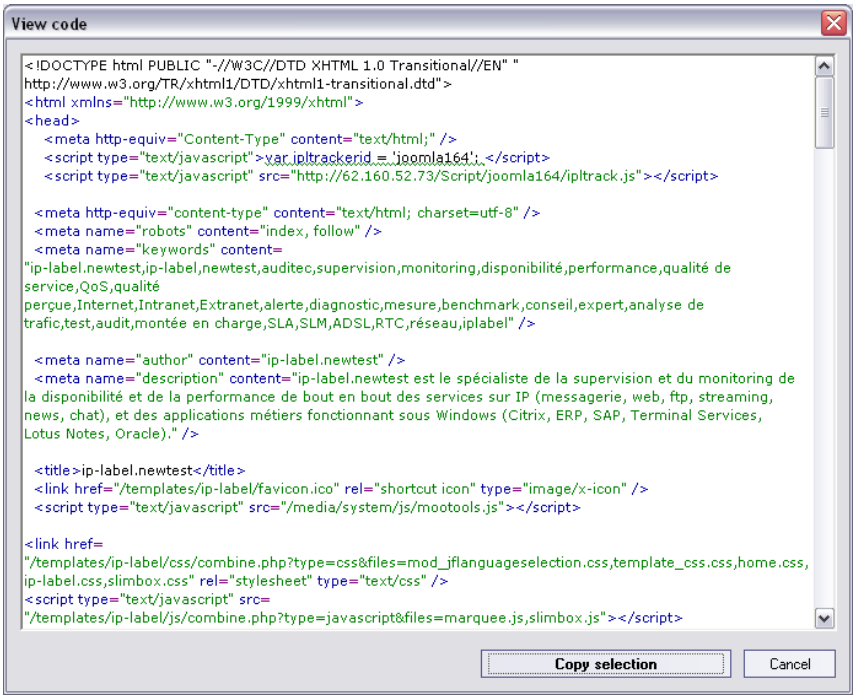


- **User Name**: login to use
- **Password**: password to use
- **Encrypt password**: select this option if you want the password to be encrypted in the scenario (function GetPassword)

Follow the redirection: select this option if you want Newtest to follow the redirection of the page it is testing. If you choose to do so, the robot will launch a second request to the URL corresponding to the redirect and will measure it. If you do not select this option, the request will test the first page containing the redirect.

Control string: when you launch the request, an HTML page is downloaded. To make sure it is the right page, choose a string of significant characters in that page. The request will check for this string each time you download the page.

NOTE: you may be able to see a string in a web page without it really being there! (This is the case with images, accents, included HTML pages, etc.). Solve this problem by searching for the string in the page's source code; click [...].



The HTML code resulting from the request appears in the **View code** window. Select a string in the code and click **Copy selection** to automatically copy the selected text to the "control string" field, and then close the active window.

If the specified string is not found in the page while the request is running, the request becomes unavailable.

Error label: text recorded by the ERRORLINE function in the script in the event the request fails.

Request timeout (sec.): enter the timeout of the HTTP request in seconds.

Test request... button: click for an immediate test of the HTTP request using the current parameters. A dialog box appears at the end of the test.

Click **OK** to validate your parameters.

Description of response time measurement

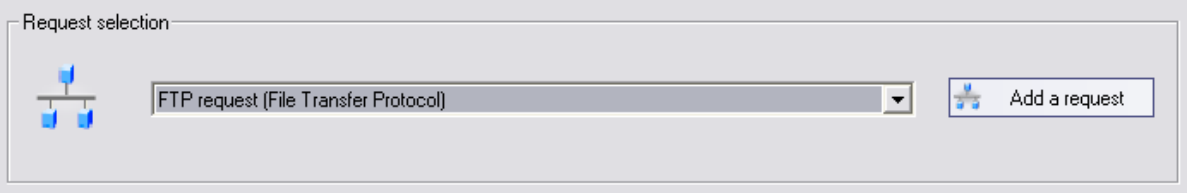
Measurement	Description
<Prefix>_Test	Total HTTP request time (to 1/100th of a second)

Description of numeric measurements

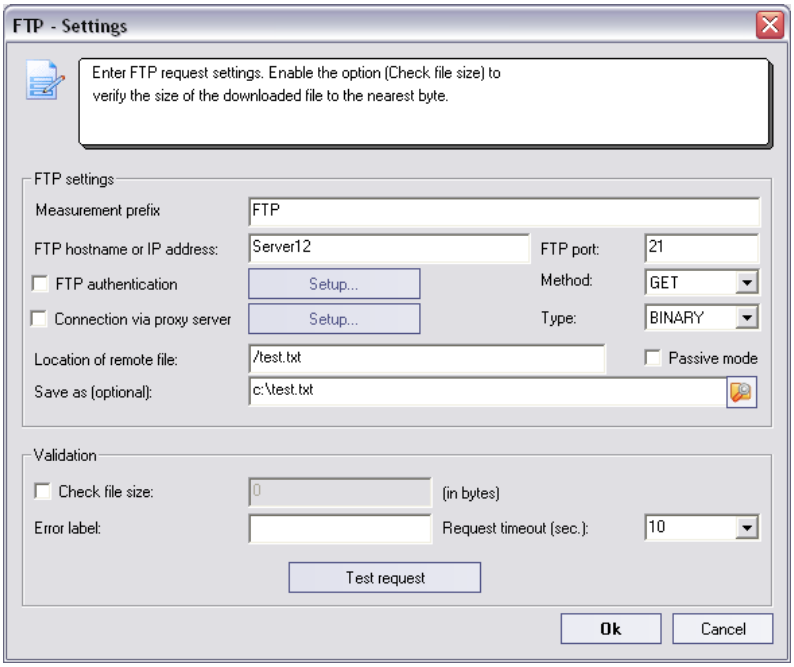
Measurement	Description
<Prefix>1_Dns	Time to DNS resolution (in ms = milliseconds)
<Prefix>2_Connect	Time to connect to HTTP server (ms)

<Prefix>3_Firstbyte	Time to receive the first byte of data (ms)
<Prefix>4_GetTime	Total loading time (ms)
<Prefix>5_GetRate	Transfer rate (in bytes/second)

5.3.2.2 CONFIGURING AN FTP REQUEST



The wizard's **FTP - Settings** page offers options for setting up a GET or PUT FTP request. It is also possible to rename or delete a file on the FTP server.



Measurement prefix: allows you to set a prefix for the measurements made using this request in the scenario. A prefix is indispensable for identifying the measurements of a request uniquely with respect to those of other requests and transmitting the information in an identifiable way to NMC.

The prefix you indicate will be added, followed by an underscore (_), to the name of each measurement. The maximum length of the prefix is 6 characters.

FTP server name or IP address: enter the name or IP address of the FTP server.

FTP port: type the port of the FTP server. (Default port is 21.)

FTP authentication: select this option if a user name and password are required to access the FTP server. Then click [Setup...] to supply the user name and password:

- **User Name**: login to use
- **Password**: password to use
- **Encrypt password**: select this option if you want the password to be encrypted in the scenario (function GetPassword)

Method: select the FTP method to use (GET, PUT, DELETE, RENAME). GET is the default method.

Connection via proxy server: select this option if the FTP server is accessed via a proxy server. Click Setup... to define proxy server access settings (see *Proxy server properties* below).

Type: choose the type of file to download or send (ASCII or BINARY).

The following fields depend on the method selected for the FTP request:

Method GET

Location of remote file: type path to the remote file from the working directory available to you upon connection to the FTP server (ex: /test.txt).

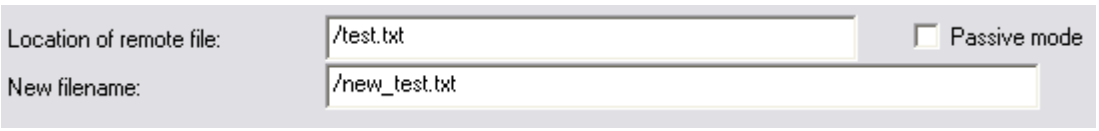
Save as (optional): indicate the path to the location (local) to which to save the file retrieved on the FTP server. The button  enables you to browse to locate the directory and the file to save.

Method PUT

Location of remote file: type path to the remote file from the working directory available to you upon connection to the FTP server (ex: /test.txt).

Save as (optional): indicate the path to the local file to send to the FTP server. The button  enables you to browse to locate the directory and the file to send.

Method RENAME



Location of remote file: type path to the remote file from the working directory available to you upon connection to the FTP server (ex: /test.txt).

New filename: provide the new name and path to the file on the FTP server. If the path is different from the one shown in **Location of remote file**, the file will be renamed and moved.

Method DELETE

Location of remote file: type path to the remote file from the working directory available to you upon connection to the FTP server (ex: /test.txt).

Passive mode: check the box if you are trying to reach the FTP server through a firewall.

Check file size (for GET method only): select this option if you wish to check the size of the file retrieved by GET FTP. If you opt to check file size, you must enter the exact size (in bytes) of the file to retrieve. If the size is incorrect, the request will become unavailable. You can fill in the **Check file size** text box automatically when you test the request in the Configurator (see "Test request" below).

Error label: text recorded by the ERRORLINE function in the script in the event the request fails.

Request timeout (sec.): enter the timeout of the FTP request in seconds.

Test request button: click for an immediate test of the FTP request using the current parameters. A dialog box displays the result of the test.

If the option **Check file size** was enabled during the text, the size of the loaded object is also tested against the parameter value. If these to not match, the following message appears:

If the option **Check file size** was not selected, the value in the parameter is automatically filled in with the size of the object loaded.

Click **Ok** to confirm your settings.

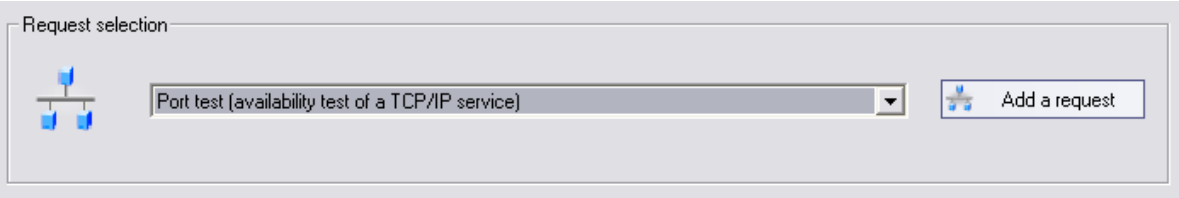
Description of response time measurement

Measurement	Description
<Prefix>_Test	Total FTP request time (to 1/100th of a second)

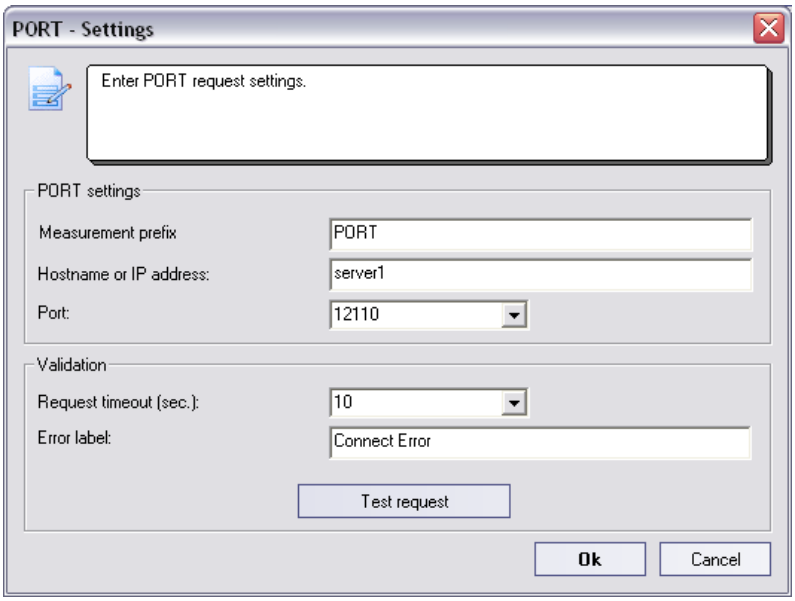
Description of numeric measurements

Measurement	Description
<Prefix>1_Dns	DNS resolution time (in ms = milliseconds)
<Prefix>2_Connect	Time to connect to FTP server (ms)
<Prefix>3_Auth	Time to authentication on the FTP server (ms)
<Prefix>4_OpenData	Time to open a data channel (ms)
<Prefix>5_GetTime	Total loading time (ms)
<Prefix>6_GetRate	Transfer rate (in bytes/second)

5.3.2.3 CONFIGURING A PORT REQUEST



The wizard's **PORT - Settings** page offers options for setting up a request that consists of testing the availability of the service associated with the specified port (connect).



Measurement prefix: allows you to set a prefix for the measurements made using this request in the scenario. A prefix is indispensable for identifying the measurements of a request uniquely with respect to those of other requests and transmitting the information in an identifiable way to NMC.

The prefix you indicate will be added, followed by an underscore (_), to the name of each measurement. The maximum length of the prefix is 6 characters.

Hostname or IP address: enter the host name or IP address.

Port: enter the port you wish to test. Type a port number or select one from the drop-down list.

Request timeout (sec.): enter the timeout of the port request in seconds.

Error label: text recorded by the ERRORLINE function in the script in the event the request fails.

Test request button: click for an immediate test of the port request using the current parameters.

Click **Ok** to confirm your settings

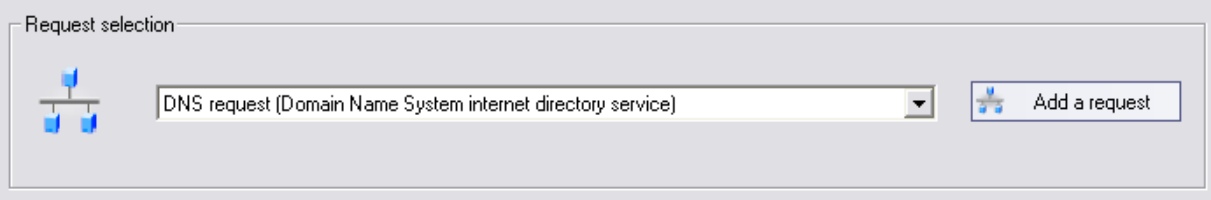
Description of response time measurement

Measurement	Description
<Prefix>_Test	Total port request time (to 1/100th of a second)

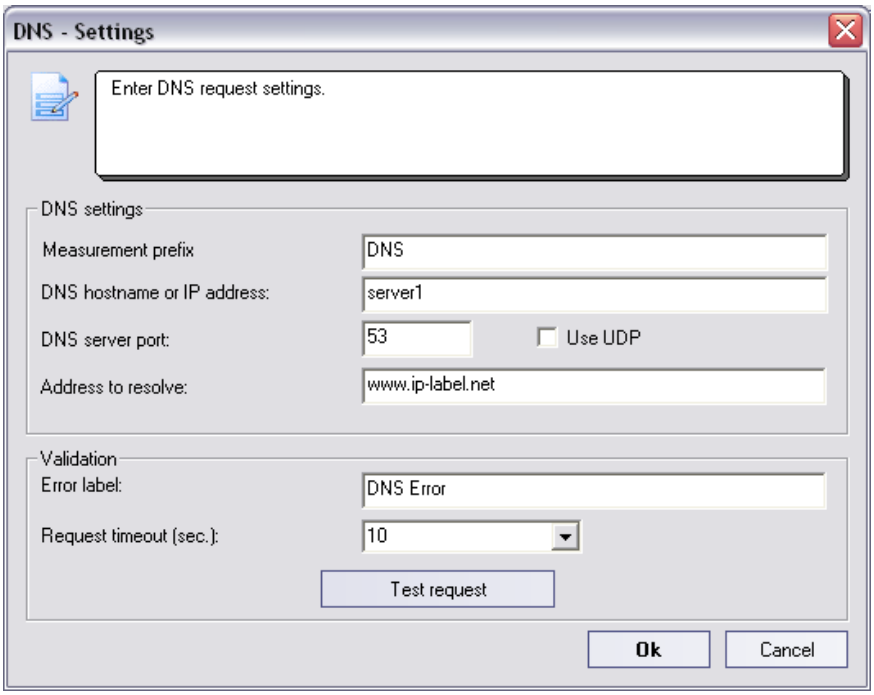
Description of numeric measurements

Measurement	Description
<Prefix>1_Dns	DNS resolution time (in ms = milliseconds)
<Prefix>2_Connect	Time to connect to server (ms)

5.3.2.4 CONFIGURING A DNS REQUEST



The wizard's **DNS - Settings** page offers options for setting up a DNS (Domain Name System) request. A DNS request asks a given DNS server to resolve a domain name (i.e. obtain the corresponding IP address).



Measurement prefix: allows you to set a prefix for the measurements made using this request in the scenario. A prefix is indispensable for identifying the measurements of a request uniquely with respect to those of other requests and transmitting the information in an identifiable way to NMC.

The prefix you indicate will be added, followed by an underscore (_), to the name of each measurement. The maximum length of the prefix is 6 characters.

DNS hostname or IP address: enter the name or IP address of the DNS server.

DNS server port: type the port of the DNS server. (Default port is 53.)

Use UDP: select this option if you want the DNS request to be generated in UDP mode. (The default setting for requests is TCP.)

Address to resolve: enter the address to resolve (e.g. www.auditec-newtest.com). The server will attempt to obtain the corresponding IP address.

Error label: text recorded by the ERRORLINE function in the script in the event the request fails.

Request timeout (sec.): enter the timeout of the DNS request in seconds.

Test request button: click for an immediate test of the DNS request using the current parameters.

Click **Ok** to confirm your settings.

Description of response time measurement

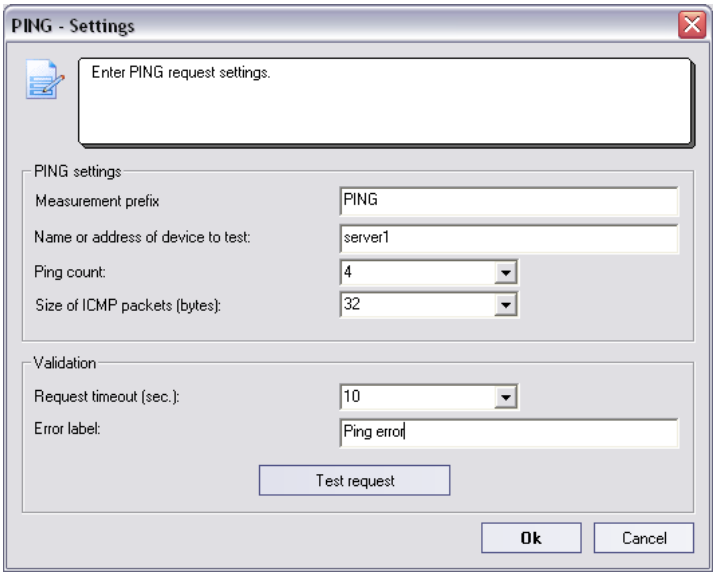
Measurement	Description
<Prefix>_Test	Total DNS request time (to 1/100th of a second)

Description of numeric measurements

Measurement	Description
<Prefix>1_Dns	DNS resolution time (in ms = milliseconds)

5.3.2.5 CONFIGURING A PING REQUEST

The wizard's **PING - Settings** page offers options for setting up a ping request.



Measurement prefix: allows you to set a prefix for the measurements made using this request in the scenario. A prefix is indispensable for identifying the measurements of a request uniquely with respect to those of other requests and transmitting the information in an identifiable way to NMC.

The prefix you indicate will be added, followed by an underscore (_), to the name of each measurement. The maximum length of the prefix is 6 characters.

Name or address of the device to test: enter the name or IP address of the device you wish to test.

Number of pings to send: type the number of basic requests to send for this test.

Size of ICMP packets (bytes): enter the size of the buffer memory zone to send in the ping.

Timeout of each ping (ms): enter the timeout of the ping request.

Error label: text recorded by the ERRORLINE function in the script in the event the request fails.

Test request button: click for an immediate test of the ping request using the current parameters.

Click **Ok** to confirm your settings.

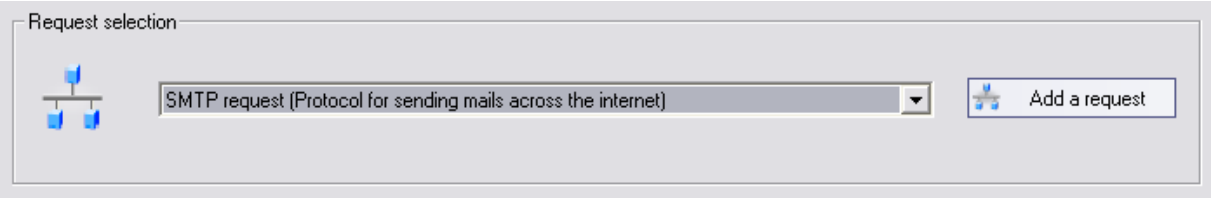
Description of response time measurement

Measurement	Description
<Prefix>_Test	Total time of the series of ping requests (to the 1/100th of a second)

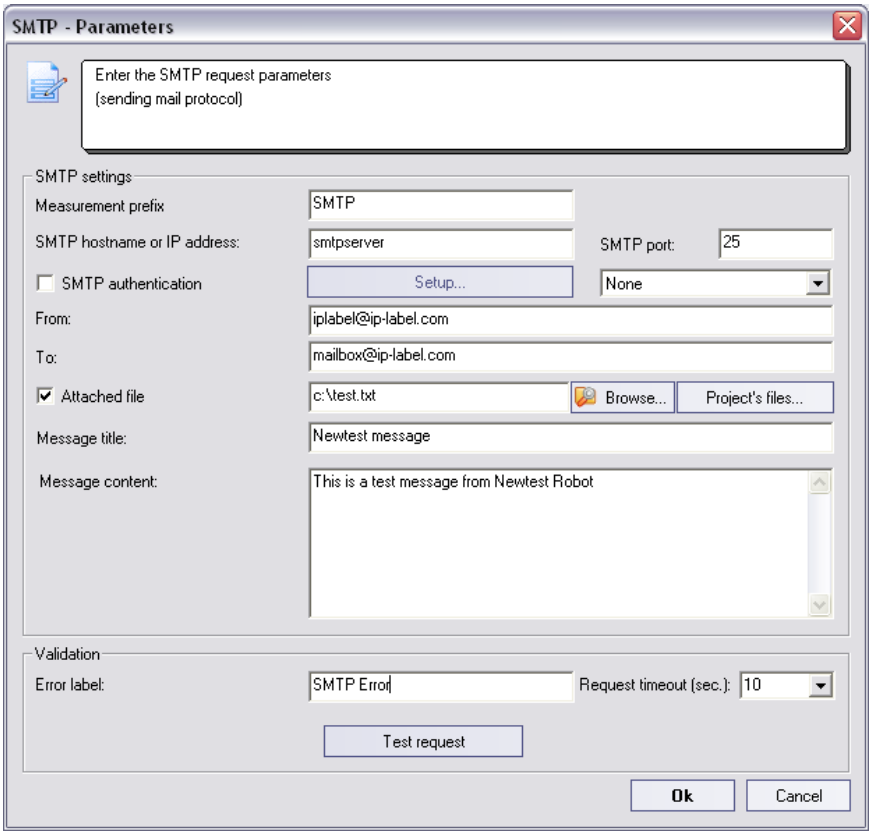
Description of numeric measurements

Measurement	Description
<Prefix>1_DNSTime	DNS initial resolution time (in ms = milliseconds)
<Prefix>2_MaxTime	Maximum response time obtained from the series of pings (ms)
<Prefix>3_MinTime	Minimum response time obtained from the series of pings (ms)
<Prefix>4_AvgTime	Average response time obtained from the series of pings (ms)
<Prefix>5_NbSent	Number of pings sent
<Prefix>6_NbRcvd	Number of successful pings
<Prefix>7_NbLost	Number of pings lost
<Prefix>8_LostPct	Percentage of pings lost. The result is a whole number between 0 and 100 inclusive.

5.3.2.6 CONFIGURING AN SMTP REQUEST



The wizard's **SMTP - Parameters** page offers options for setting up a mail (SMTP) request sending a mail message to a given address.



Measurement prefix: allows you to set a prefix for the measurements made using this request in the scenario. A prefix is indispensable for identifying the measurements of a request uniquely with respect to those of other requests and transmitting the information in an identifiable way to NMC.

The prefix you indicate will be added, followed by an underscore (_), to the name of each measurement. The maximum length of the prefix is 6 characters.

SMTP hostname or IP address: enter the name or IP address of the SMTP server.

SMTP port: type the port of the SMTP server. (Default setting is 25.)

SMTP authentication: allows Extended SMTP. Click **Setup...** to enter the connection settings.

- **User Name**: login to use
- **Password**: password to use
- **Encrypt password**: select this option if you want the password to be encrypted in the scenario (function GetPassword)

Encryption: select from the drop-down list the type of encryption you wish to use (none, SSL or TLS) for the SMTP request.

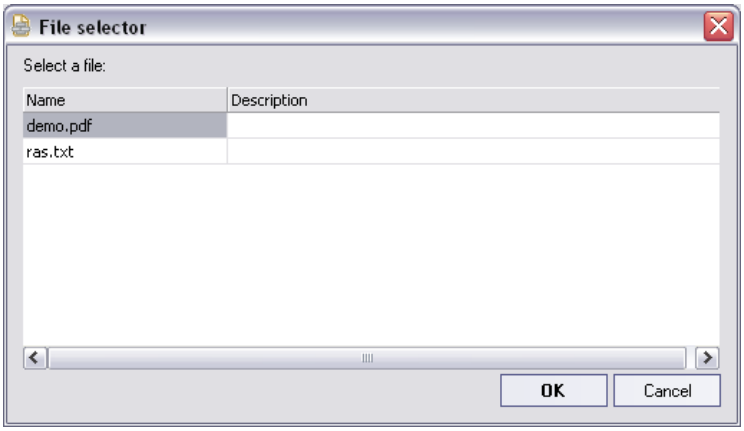
From: enter the e-mail address of the message sender.

To: enter the e-mail address of the message recipient.

Attached file: select this option to attach a file to the message. The buttons 

are for selecting the file to send.

- **Browse...**: to browse for the file to send.
- **Project's files...**: opens a dialog box to find the file to send from among the list of files saved in the current NTBR project (File tab page in the main interface of the NTBR project).



Message Title: enter the title of the message to send.

Message content: type the text of the message to send.

Error label: text recorded by the ERRORLINE function in the script in the event the request fails.

Request timeout (sec.): enter the timeout of the entire request (sending/receiving) in seconds, after which the request becomes unavailable.

Test request... button: click for an immediate test of the mail request using the current parameters.

Click **Ok** to confirm your settings.

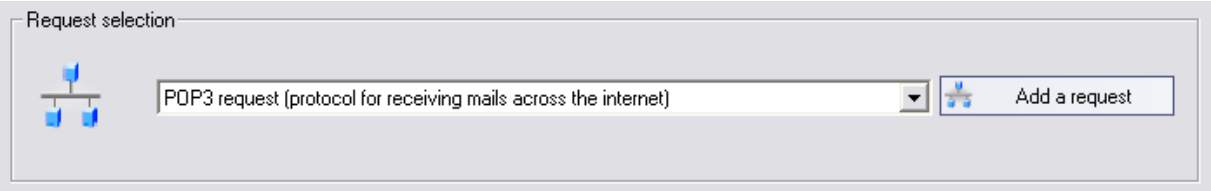
Description of response time measurements

Measurement	Description
<Prefix>_Test	Total outbound request time (to 1/100th of a second)

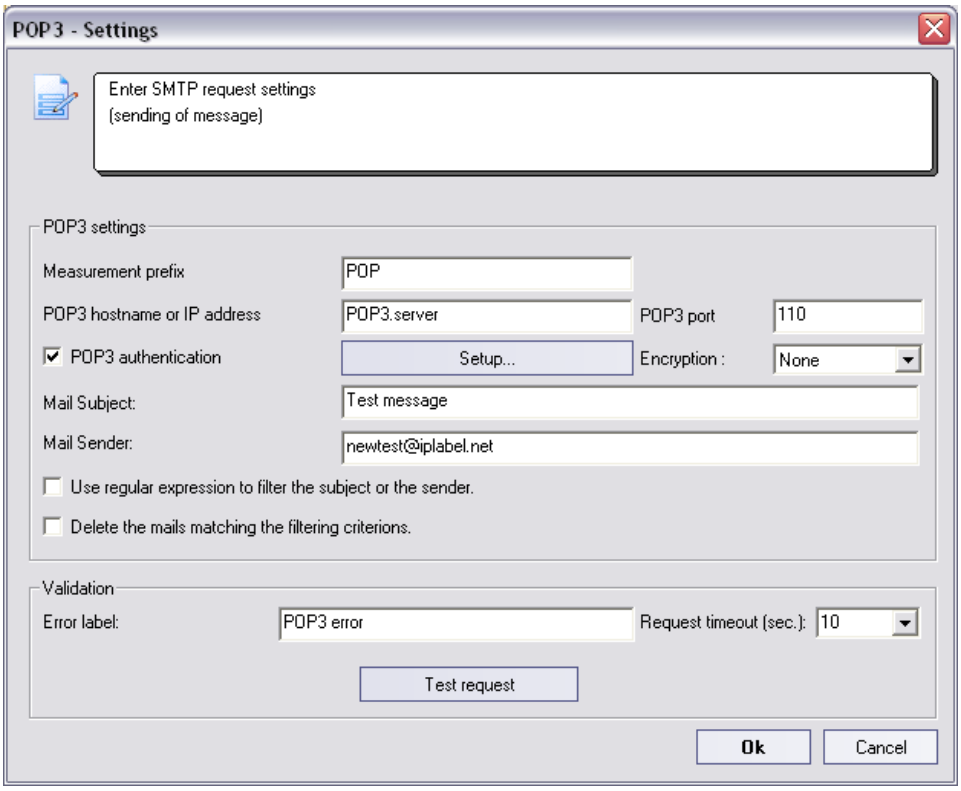
Description of numeric measurements

Measurement	Description
<Prefix>1_Dns	SMTP name DNS resolution time (in ms = milliseconds)
<Prefix>2_Connect	Time to connect to SMTP server (ms)
<Prefix>3_Auth	Time to authenticate on the SMTP server (ms)
<Prefix>4_SendTime	Time to send a message on the SMTP server (ms)
<Prefix>5_SendRate	Transfer rate (bytes/second)

5.3.2.7 CONFIGURING A POP3 MAIL REQUEST



The wizard's **POP3 - Settings** page offers options for setting up a request to check mail in a POP3 mailbox. A POP3 request consists of connecting to the POP3 server and checking/waiting for a message with a certain subject and/or sender. Purging the messages found on the POP3 server is also a possibility.



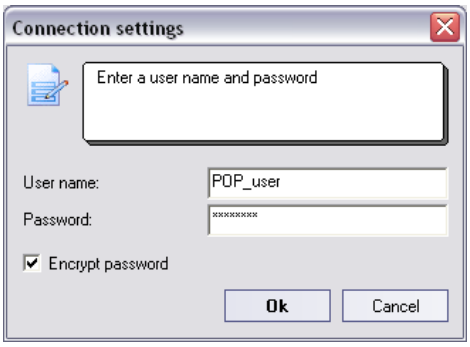
Measurement prefix: allows you to set a prefix for the measurements made using this request in the scenario. A prefix is indispensable for identifying the measurements of a request uniquely with respect to those of other requests and transmitting the information in an identifiable way to NMC.

The prefix you indicate will be added, followed by an underscore (_), to the name of each measurement. The maximum length of the prefix is 6 characters.

POP3 hostname or IP address: enter the name or IP address of the POP3 server.

POP3 port: type the port of the POP3 server. (Default port is 110.)

POP3 authentication: click [Setup...] to access settings for connecting to the POP3 mailbox:



- **User name**: login to use
- **Password**: password to use
- **Encrypt password**: select this option if you want the password to be encrypted in the scenario (function GetPassword)

Encryption: select from the drop-down list the type of encryption you wish to use (none or SSL) for the SMTP request.

Mail Subject: subject of the message to find in the POP3 mailbox.

Mail Sender: sender of the message to find in the POP3 mailbox.

Use regular expression to filter subjects or senders: select this option to enable use of regular expressions in the name of the *Mail Sender* and in the *Mail Subject* in the message to find. The request will be considered successful if at least one message matches these criteria.

Delete the mails matching the filtering criteria: select this option to delete all messages matching *Mail Subject* and *Mail Sender* selection criteria from the POP3 mailbox at the end of the request.

Error label: text recorded by the ERRORLINE function in the script in the event the request fails.

Request timeout (sec.): enter the timeout of the entire request (sending/receiving) in seconds, after which the request becomes unavailable.

Test request... button: click for an immediate test of the mail request using the current parameters.

Click **Ok** to confirm your settings.

Description of response time measurements

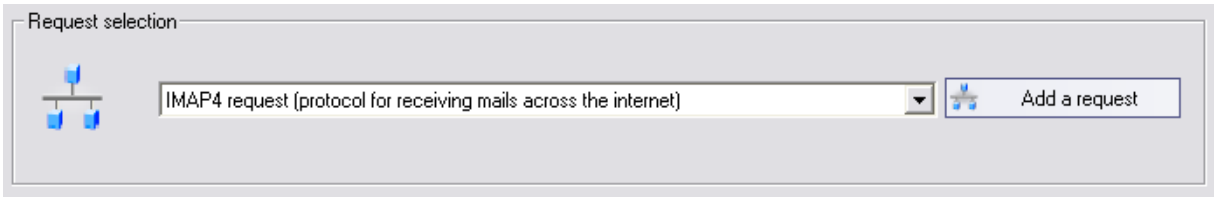
Measurement	Description
<Prefix>_Test	Total incoming request time (to 1/100th of a second)

Description of numeric measurements

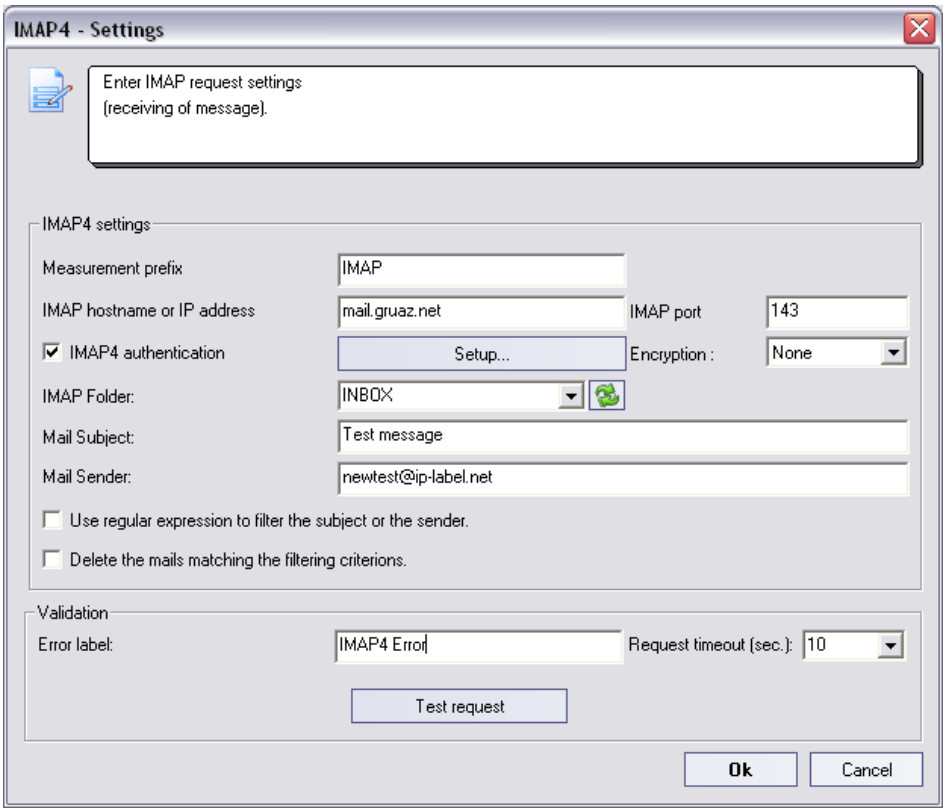
Measurement	Description
<Prefix>1_Dns	POP3 name DNS resolution time (ms)
<Prefix>2_Connect	Time to connect to the POP3 server (ms)
<Prefix>3_Auth	Authentication time on the POP3 server (ms)
<Prefix>4_WaitMail	Time until detection of new mail (ms)

<Prefix>5_ReceiveTime	Time to receive a message on the POP3 server (ms)
<Prefix>6_ReceiveRate	Transfer rate (bytes/second)

5.3.2.8 CONFIGURING AN IMAP4 MAIL REQUEST



The wizard's **IMAP4 - Settings** page offers options for setting up a request to check mail in a POP3 mailbox. A POP3 request consists of connecting to the POP3 server and checking/waiting for a message with a certain subject and/or sender. Purging the messages found on the POP3 server is also a possibility.



Measurement prefix: allows you to set a prefix for the measurements made using this request in the scenario. A prefix is indispensable for identifying the measurements of a request uniquely with respect to those of other requests and transmitting the information in an identifiable way to NMC.

The prefix you indicate will be added, followed by an underscore (_), to the name of each measurement. The maximum length of the prefix is 6 characters.

IMAP hostname or IP address: enter the name or IP address of the POP3 server.

IMAP port: type the port of the IMAP4 server. (Default port is 143.)

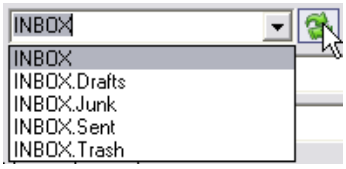
IMAP authentication: click [Setup...] to access settings for connecting to the IMAP mailbox.

- **User name**: login to use

- **Password**: password to use
- **Encrypt password**: select this option if you want the password to be encrypted in the scenario (function GetPassword)

Encryption: select from the drop-down list the type of encryption you wish to use (none, SSL or TLS) for the SMTP request.

IMAP Folder: allows you to designate the folder in the recipient's IMAP mailbox in which to find the message. *All folders* is the default setting. To view the list of available folders, click  . If the server and login parameters are correct, the drop-down list displays the names of the folders:



You can also type the folder name in this field if you know the exact syntax.

Mail Subject: subject of the message to find in the IMAP mailbox.

Mail Sender: sender of the message to find in the IMAP mailbox.

Use regular expression to filter subjects or senders: select this option to enable use of regular expressions in the name of the *Mail Sender* and in the *Mail Subject* in the message to find. The request will be considered successful if at least one message matches these criteria.

Delete the mails matching the filtering criteria: select this option to delete all messages matching *Mail Subject* and *Mail Sender* selection criteria from the IMAP mailbox at the end of the request.

Error label: text recorded by the ERRORLINE function in the script in the event the request fails.

Request timeout (sec.): enter the timeout of the entire request (sending/receiving) in seconds, after which the request becomes unavailable.

Test request... button: click for an immediate test of the mail request using the current parameters.

Click **Ok** to confirm your settings.

Description of response time measurements

Measurement	Description
<Prefix>_Test	Total incoming request time (to 1/100th of a second)

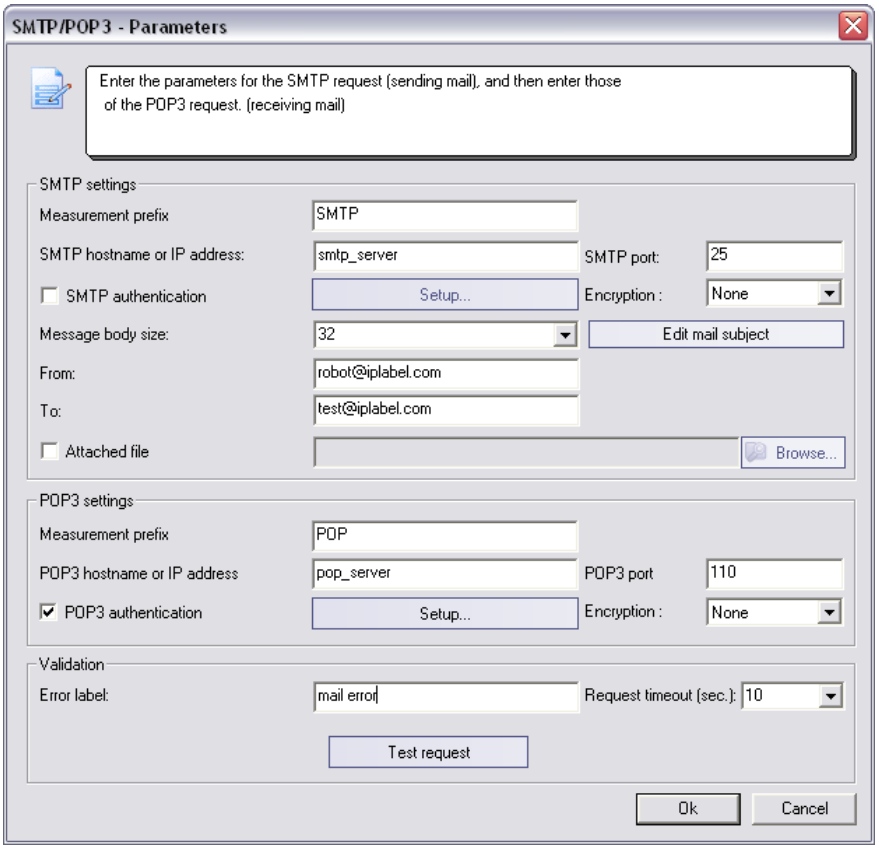
Description of numeric measurements

Measurement	Description
<Prefix>1_Dns	IMAP4 name DNS resolution time (ms)
<Prefix>2_Connect	Time to connect to the IMAP4 server (ms)
<Prefix>3_Auth	Authentication time on the IMAP4 server (ms)
<Prefix>4_WaitMail	Time until detection of new mail (ms)
<Prefix>5_ReceiveTime	Time to receive a message on the IMAP4 server (ms)

<Prefix>6_ReceiveRate	Transfer rate (bytes/second)

5.3.2.9 CONFIGURING A SMTP/POP3 REQUEST

The wizard's **SMTP/POP3 - Parameters** page offers options for setting up a mail (SMTP/POP3) request. A mail request (following connection to a given SMTP server) consists of sending a message on the SMTP server, and then receiving it on a given POP3 server. Purging the messages (selected by object) on the POP3 server is carried out beforehand.



.

SMTP settings

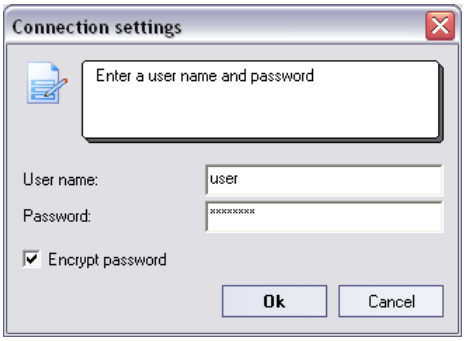
Measurement prefix: allows you to set a prefix for the measurements made using this request in the scenario. A prefix is indispensable for identifying the measurements of a request uniquely with respect to those of other requests and transmitting the information in an identifiable way to NMC.

The prefix you indicate will be added, followed by an underscore (_), to the name of each measurement. The maximum length of the prefix is 6 characters.

SMTP hostname or IP address: enter the name or IP address of the SMTP server.

SMTP port: type the port of the SMTP server. (Default setting is 25.)

SMTP authentication: allows Extended SMTP. Click **Setup...** to enter the connection settings.

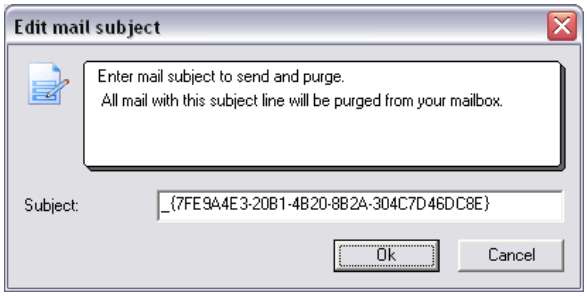


- **User name**: login to use
- **Password**: password to use
- **Encrypt password**: select this option if you want the password to be encrypted in the scenario (function GetPassword)

Encryption: select from the drop-down list the type of encryption you wish to use (none, SSL or TLS) for the SMTP request.

Message body size: type the size of the message to send (in bytes), or select a size from the drop-down list.

Edit mail subject button: the object of the message in an SMTP/POP3 test is generated automatically. It is then used to detect the reception of a mail in the Inbox. You may, however, change the object by clicking this button:

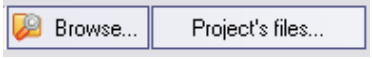


- **Subject**: type the object of the mail to send.

From: enter the e-mail address of the message sender.

To: enter the e-mail address of the message recipient.

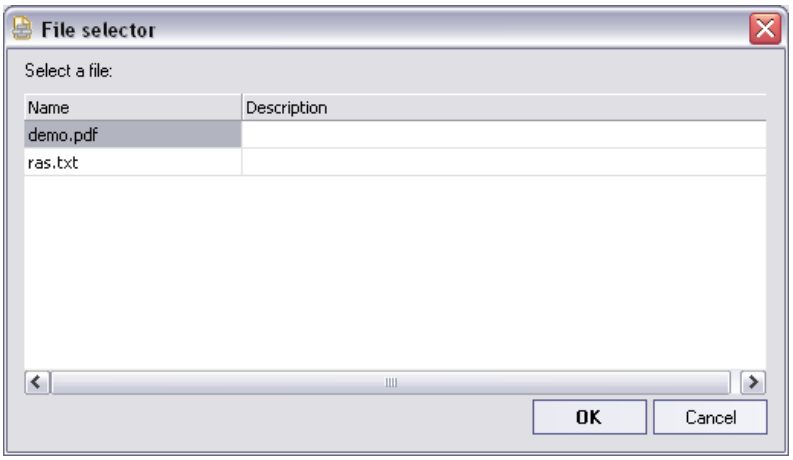
Attached file: select this option to attach a file to the message. The buttons



are for selecting the file to send.

Browse...: to browse for the file to send.

Project's files...: opens a dialog box to find the file to send from among the list of files saved in the current NTBR project (File tab page in the main interface of the NTBR project).



POP3 settings

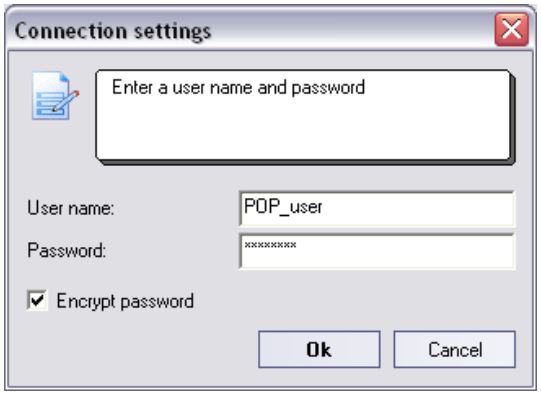
Measurement prefix: allows you to set a prefix for the measurements made using this request in the scenario. A prefix is indispensable for identifying the measurements of a request uniquely with respect to those of other requests and transmitting the information in an identifiable way to NMC.

The prefix you indicate will be added, followed by an underscore (_), to the name of each measurement. The maximum length of the prefix is 6 characters.

POP3 hostname or IP address: enter the name or IP address of the POP3 server.

POP3 port: type the port of the POP3 server. (Default port is 110.)

POP3 authentication: click [Setup...] to access settings for connecting to the POP3 mailbox:



- **User name**: login to use
- **Password**: password to use
- **Encrypt password**: select this option if you want the password to be encrypted in the scenario (function GetPassword)

Encryption: select from the drop-down list the type of encryption you wish to use (none or SSL) for the SMTP request.

Error label: text recorded by the ERRORLINE function in the script in the event the request fails.

Request timeout (sec.): enter the timeout of the entire request (sending/receiving) in seconds, after which the request becomes unavailable.

Description of response time measurements

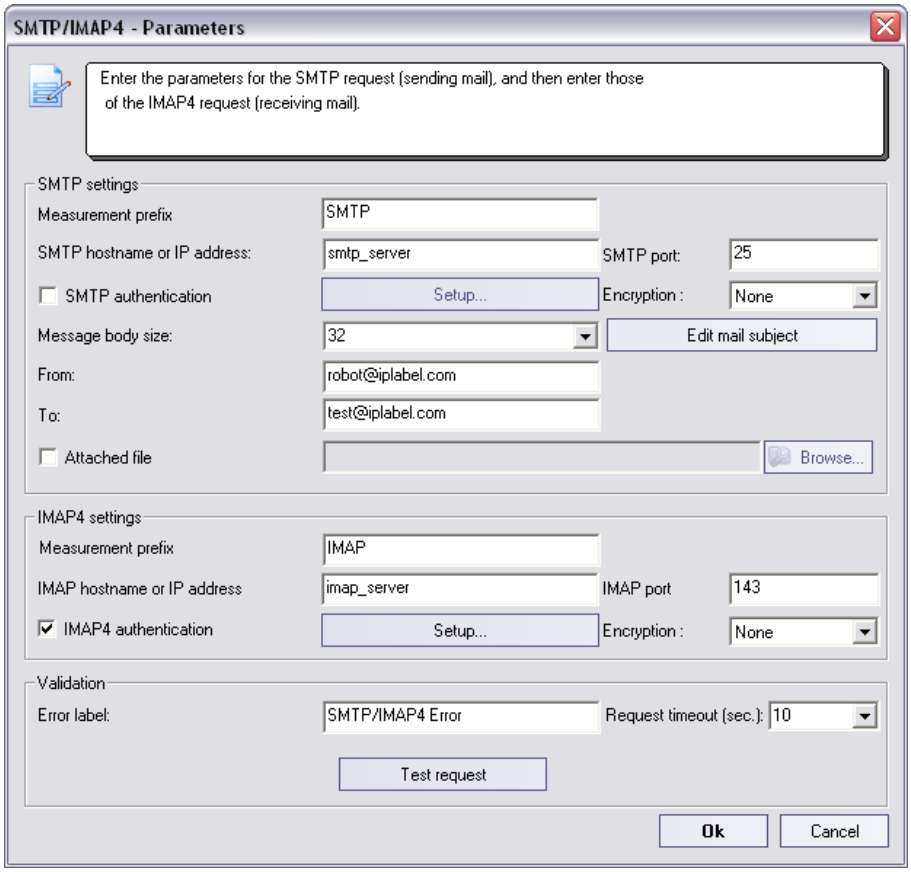
Measurement	Description
<Prefix>_Test	Total outbound request time (to 1/100th of a second)
<Prefix>_Test	Total incoming request time (to 1/100th of a second)

Description of numeric measurements

Measurement	Description
<Prefix>1_Dns	SMTP name DNS resolution time (in ms = milliseconds)
<Prefix>2_Connect	Time to connect to SMTP server (ms)
<Prefix>3_Auth	Time to authenticate on the SMTP server (ms)
<Prefix>4_SendTime	Time to send a message on the SMTP server (ms)
<Prefix>5_SendRate	Transfer rate (bytes/second)
<Prefix>1_Dns	POP3 name DNS resolution time (ms)
<Prefix>2_Connect	Time to connect to the POP3 server (ms)
<Prefix>3_Auth	Authentication time on the POP3 server (ms)
<Prefix>4_WaitMail	Time until detection of new mail (ms)
<Prefix>5_ReceiveTime	Time to receive a message on the POP3 server (ms)
<Prefix>6_ReceiveRate	Transfer rate (bytes/second)

5.3.2.10 CONFIGURING AN SMTP/IMAP4 REQUEST

The wizard's **SMTP/IMAP4 - Parameters** page offers options for setting up a mail (SMTP/IMAP4) request. A mail request (following connection to a given SMTP server) consists of sending a message on the SMTP server, and then receiving it on a given IMAP server. Purging the messages (selected by object) on the IMAP server is carried out beforehand.



SMTP settings

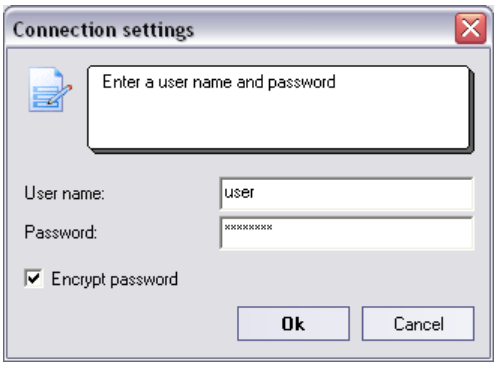
Measurement prefix: allows you to set a prefix for the measurements made using this request in the scenario. A prefix is indispensable for identifying the measurements of a request uniquely with respect to those of other requests and transmitting the information in an identifiable way to NMC.

The prefix you indicate will be added, followed by an underscore (_), to the name of each measurement. The maximum length of the prefix is 6 characters.

SMTP hostname or IP address: enter the name or IP address of the SMTP server.

SMTP port: type the port of the SMTP server. (Default setting is 25.)

SMTP authentication: allows Extended SMTP. Click **Setup...** to enter the connection settings.

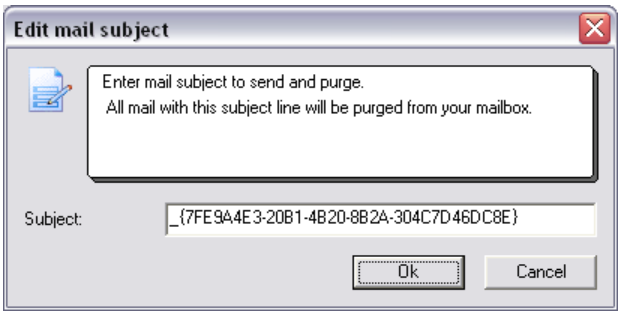


- **User name**: login to use
- **Password**: password to use
- **Encrypt password**: select this option if you want the password to be encrypted in the scenario (function GetPassword)

Encryption: select from the drop-down list the type of encryption you wish to use (none, SSL or TLS) for the SMTP request.

Message body size: type the size of the message to send (in bytes), or select a size from the drop-down list.

Edit mail subject button: the object of the message in an SMTP/POP3 test is generated automatically. It is then used to detect the reception of a mail in the Inbox. You may, however, change the object by clicking this button:



- **Subject**: type the object of the mail to send.

From: enter the e-mail address of the message sender.

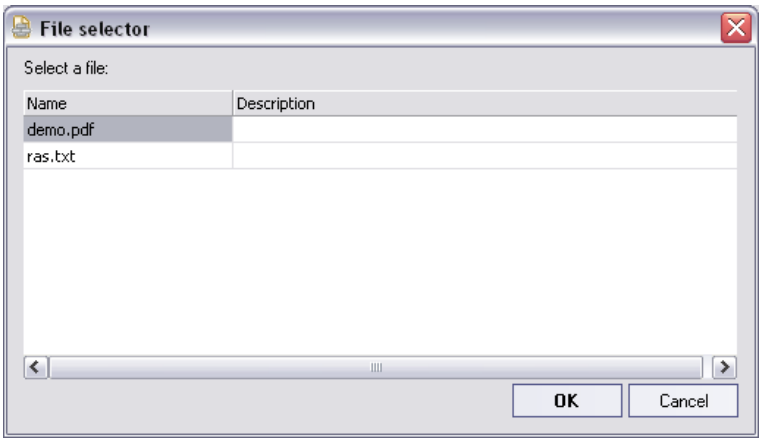
To: enter the e-mail address of the message recipient.

Attached file: select this option to attach a file to the message. The buttons 

are for selecting the file to send:

Browse...: to browse for the file to send.

Project's files...: opens a dialog box to find the file to send from among the list of files saved in the current NTBR project (File tab page in the main interface of the NTBR project).



IMAP4 settings

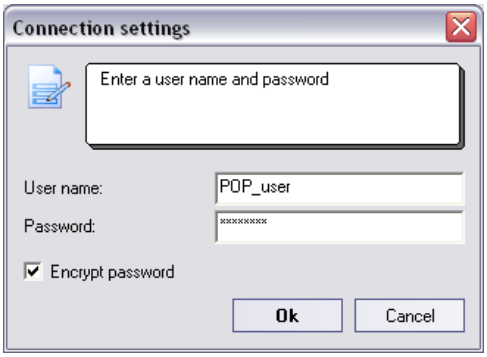
Measurement prefix: allows you to set a prefix for the measurements made using this request in the scenario. A prefix is indispensable for identifying the measurements of a request uniquely with respect to those of other requests and transmitting the information in an identifiable way to NMC.

The prefix you indicate will be added, followed by an underscore (_), to the name of each measurement. The maximum length of the prefix is 6 characters.

IMAP hostname or IP address: enter the name or IP address of the POP3 server.

IMAP port: type the port of the IMAP4 server. (Default port is 143.)

IMAP authentication: click [Setup...] to access settings for connecting to the IMAP mailbox:



- **User name**: login to use
- **Password**: password to use
- **Encrypt password**: select this option if you want the password to be encrypted in the scenario (function GetPassword)

Encryption: select from the drop-down list the type of encryption you wish to use (none or SSL) for the IMAP4 request.

Error label: text recorded by the ERRORLINE function in the script in the event the request fails.

Request timeout (sec.): enter the timeout of the entire request (sending/receiving) in seconds, after which the request becomes unavailable.

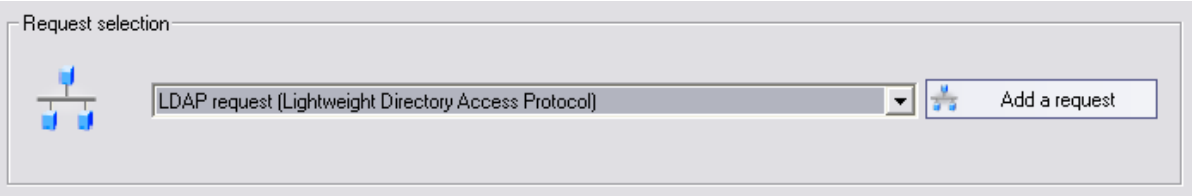
Description of response time measurements

Measurement	Description
<Prefix>_Test	Total outbound request time (to 1/100th of a second)
<Prefix>_Test	Total incoming request time (to 1/100th of a second)

Description of numeric measurements

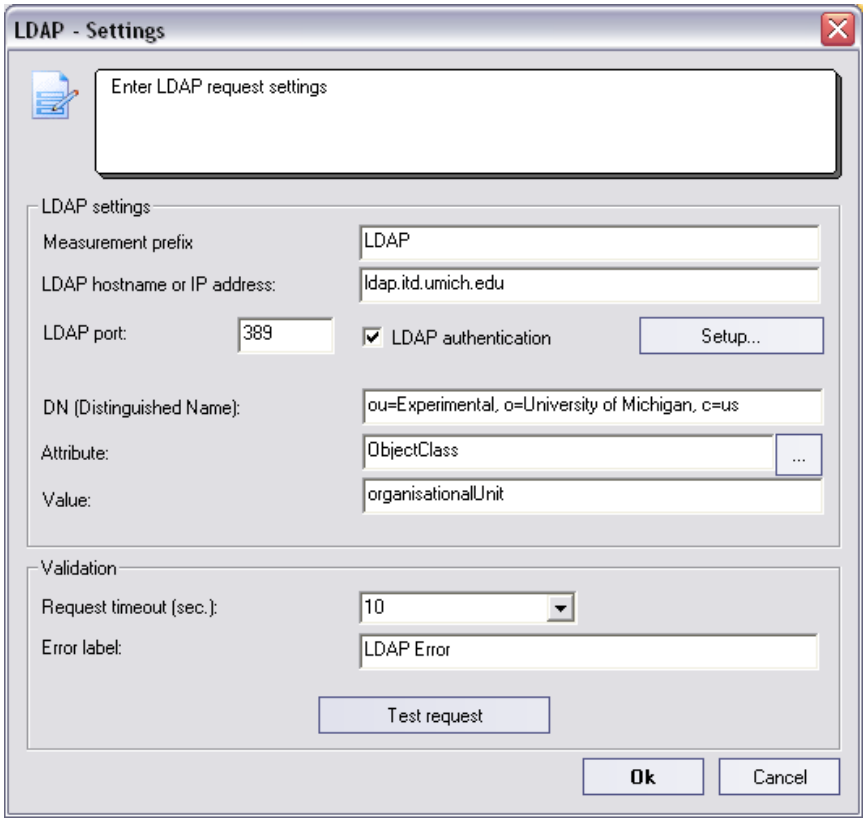
Measurement	Description
<Prefix>1_Dns	SMTP name DNS resolution time (in ms = milliseconds)
<Prefix>2_Connect	Time to connect to SMTP server (ms)
<Prefix>3_Auth	Time to authenticate on the SMTP server (ms)
<Prefix>4_SendTime	Time to send a message on the SMTP server (ms)
<Prefix>5_SendRate	Transfer rate (bytes/second)
<Prefix>1_Dns	IMAP4 name DNS resolution time (ms)
<Prefix>2_Connect	Time to connect to the IMAP4 server (ms)
<Prefix>3_Auth	Authentication time on the IMAP4 server (ms)
<Prefix>4_WaitMail	Time until detection of new mail (ms)
<Prefix>5_ReceiveTime	Time to receive a message on the IMAP4 server (ms)
<Prefix>6_ReceiveRate	Transfer rate (bytes/second)

5.3.2.11 CONFIGURING AN LDAP REQUEST



The wizard's LDAP settings page offers options for setting up a LDAP request.

LDAP requests entail checking the value of an attribute for a given entry (Distinguished Name) on an LDAP server.



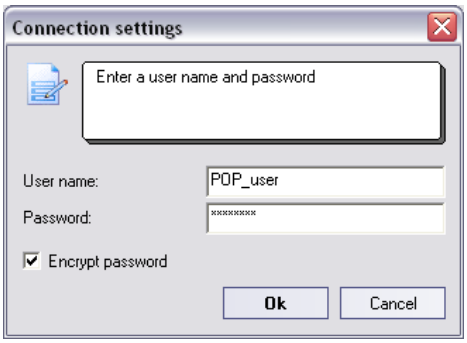
Measurement prefix: allows you to set a prefix for the measurements made using this request in the scenario. A prefix is indispensable for identifying the measurements of a request uniquely with respect to those of other requests and transmitting the information in an identifiable way to NMC.

The prefix you indicate will be added, followed by an underscore (_), to the name of each measurement. The maximum length of the prefix is 6 characters.

LDAP hostname or IP address: enter the name or IP address of the LDAP server to check.

LDAP port: type the port of the LDAP server. (Default port is 389.)

LDAP authentication: select this option if a user name and password are required to access the LDAP server. Then click the [Setup...] button to supply the user name and password:

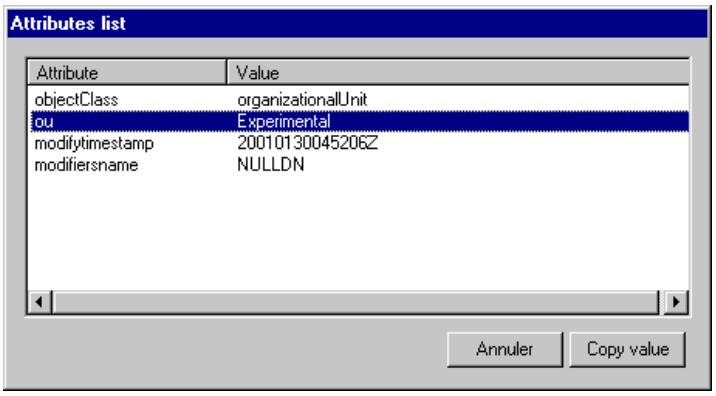


- **User name**: login to use
- **Password**: password to use
- **Encrypt password**: select this option if you want the password to be encrypted in the scenario (function GetPassword)

DN (Distinguished Name): type the name of the directory entry to check, in its LDAP notation. For example: **ou=Experimental, o=University of Michigan, c=us**.

Attribute: enter the name of the object you wish to test, e.g. **objectClass**.

If you do not know the attributes, click ... to select an attribute and the associated value from the Attributes list (see below).



This panel displays a list of attributes and their values for the specified Distinguished Name.

Select a line and click **Copy value** to automatically copy the values to the **Attribute** and **Value** text boxes.

Value: enter the value associated with the attribute, e.g. **organizationalUnit**. If the value returned when the request runs does not match the value you entered in the text box, the request will become unavailable.

If you do not know the value, click **Find attribute...** to select an attribute and the associated value from the **Attributes** list (see above).

Error label: text recorded by the ERRORLINE function in the script in the event the request fails.

Request timeout (sec.): enter the timeout of the request in seconds. If the anticipated response is not received from the LDAP server during this lapse of time, the request becomes unavailable.

Test request button: click for an immediate test of the LDAP request using the current parameters.

Click **Ok** to confirm.

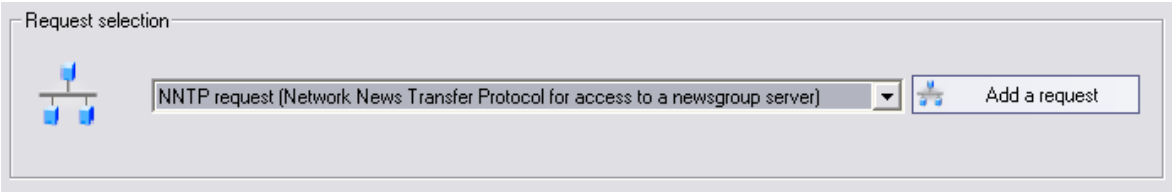
Description of response time measurement

Measurement	Description
<Prefix>_Test	Total LDAP request time (to 1/100th of a second)

Description of numeric measurements

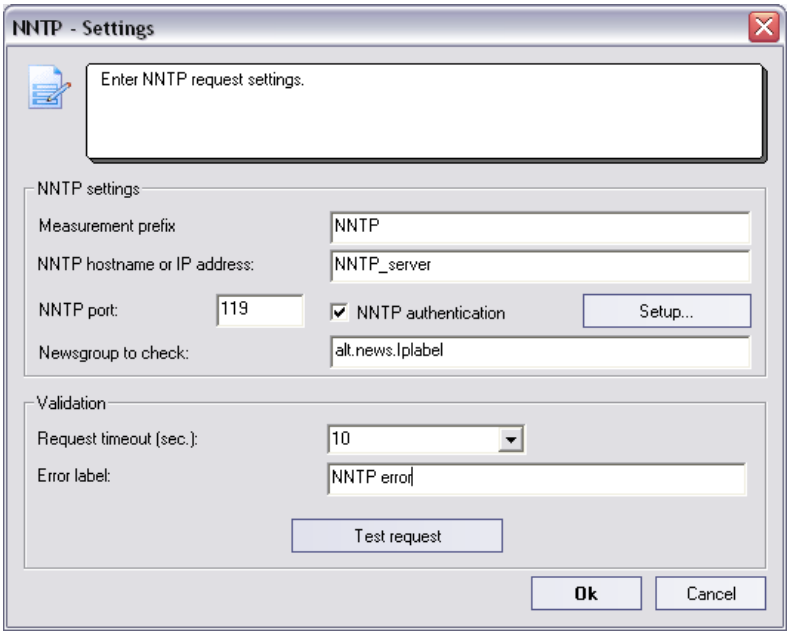
Measurement	Description
<Prefix>1_Dns	DNS resolution time (in ms = milliseconds)
<Prefix>2_Bind	Time to connect (BIND) to the LDAP server (ms)
<Prefix>3_Search	Time to find the value associated with the specified attribute (ms)

5.3.2.12 CONFIGURING AN NNTP REQUEST



The wizard's **NNTP - Settings** page offers options for setting up an NNTP (newsgroup) request.

NNTP requests test for the presence of a specific newsgroup among all the newsgroups available on the server.



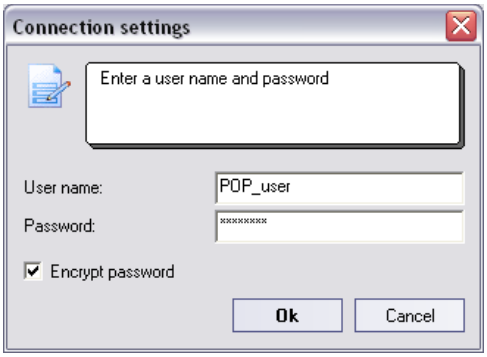
Measurement prefix: allows you to set a prefix for the measurements made using this request in the scenario. A prefix is indispensable for identifying the measurements of a request uniquely with respect to those of other requests and transmitting the information in an identifiable way to NMC.

The prefix you indicate will be added, followed by an underscore (_), to the name of each measurement. The maximum length of the prefix is 6 characters.

NNTP hostname or IP address: enter the name or IP address of the newsgroup server.

NNTP port: type the port of the NNTP server. (Default setting is 119.)

NNTP authentication: select this option if a user name and password are required to access the NNTP server. Then click **Setup...** to enter the user name and password:



- **User name**: login to use
- **Password**: password to use
- **Encrypt password**: select this option if you want the password to be encrypted in the scenario (function GetPassword)

Newsgroup to check: enter the name of a newsgroup to check among all the newsgroups on the server.

Error label: text recorded by the ERRORLINE function in the script in the event the request fails.

Request timeout (sec.): enter the timeout of the NNTP request in seconds.

Test request... button: click for an immediate test of the NNTP request using the current parameters.

Click **Ok** to confirm your settings.

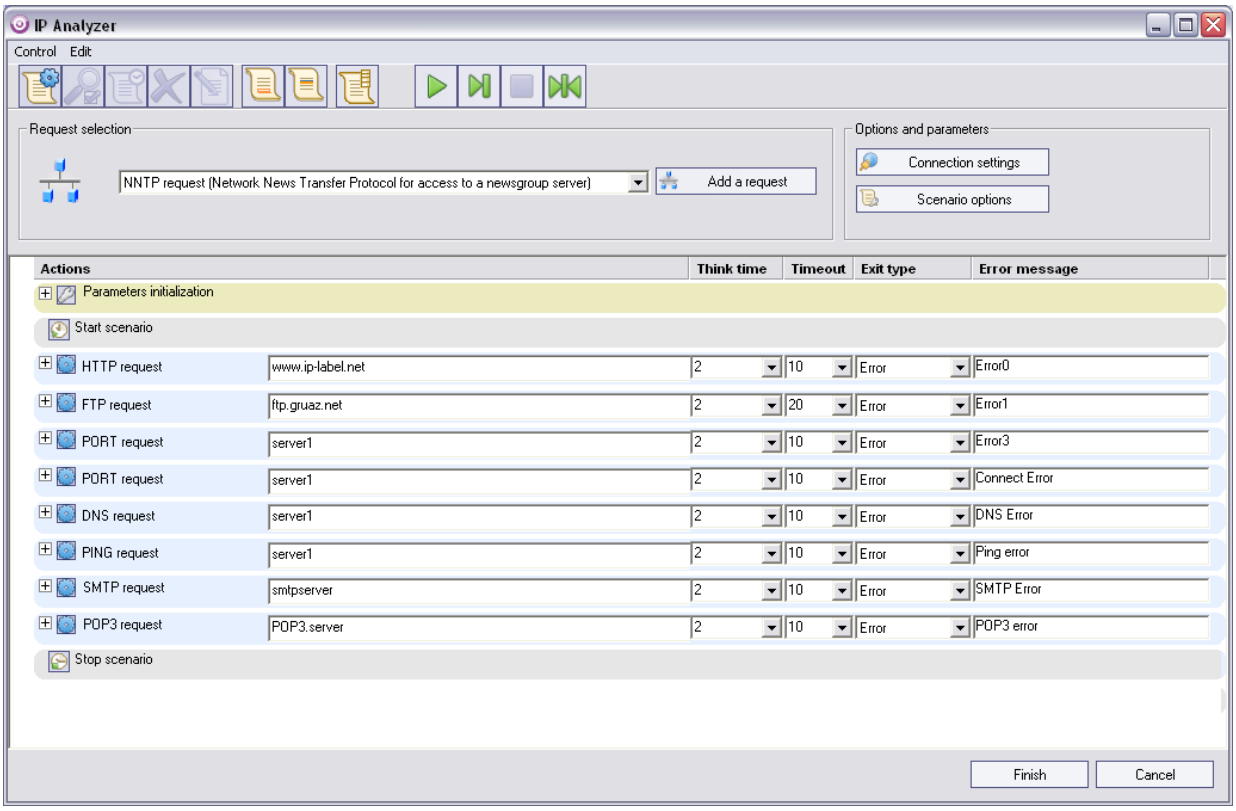
Description of response time measurement

Measurement	Description
<Prefix>_Test	Total NNTP request time (to 1/100th of a second)

Description of numeric measurements

Measurement	Description
<Prefix>1_Dns	DNS resolution time (in ms = milliseconds)
<Prefix>2_Connect	Time to connect to NNTP server (ms)
<Prefix>3_Auth	Time to authentication on the NNTP server (ms)
<Prefix>4_Getgroups	Time to read the list of newsgroups (ms)

5.3.3 USING THE IP ANALYSER WIZARD INTERFACE



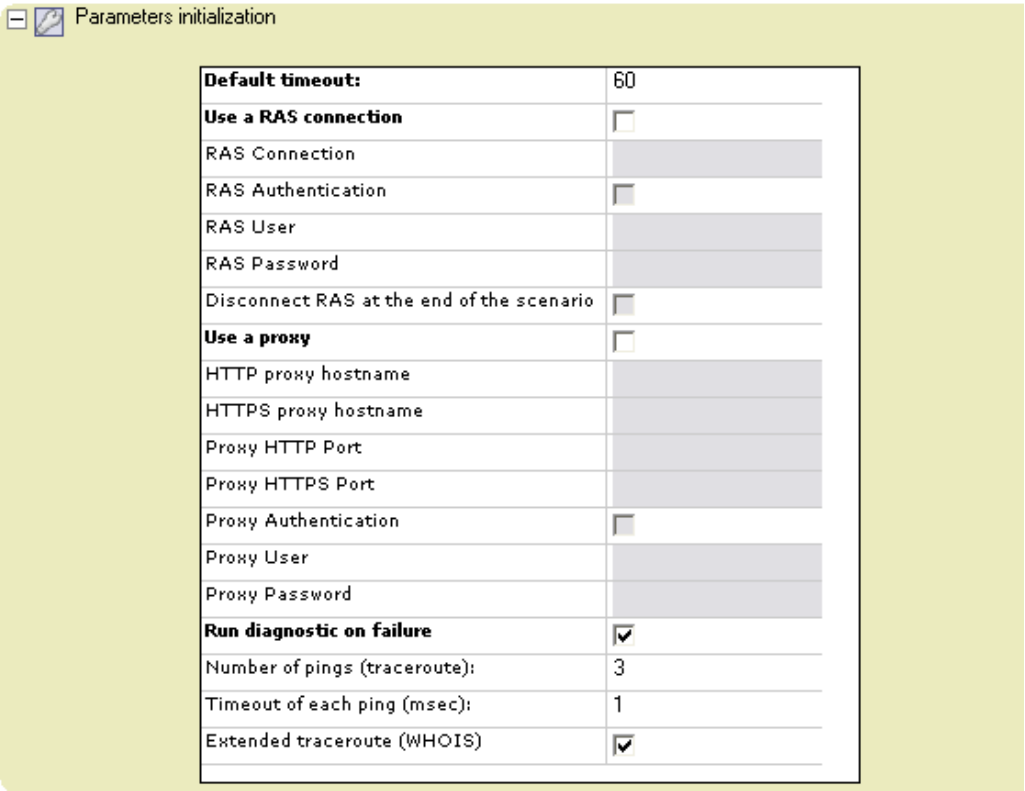
You can edit the following parameters of IP Analyser requests in the wizard:

- **Think time**: reflection time after request execution *(default: 2 seconds)*
- **Timeout** for request execution *(default: 10 seconds)*
- **Exit Type**: action to take in the event of request unavailability (*continue*, *error*, or *abort*)
- **Error Message**: text of the error message in the event of request unavailability

The main parameters of each type of request can be edited by expanding the request information interface by clicking ⊞:

To access and edit all of a request's settings, double-click the row in which the request is located.

The **Parameter initialization** panel offers access to the scenario's main default parameters, as well as connection and authentication options.



If you make changes to this section, they will also be applied in **Connection settings** or **Scenario options**, depending on the case.

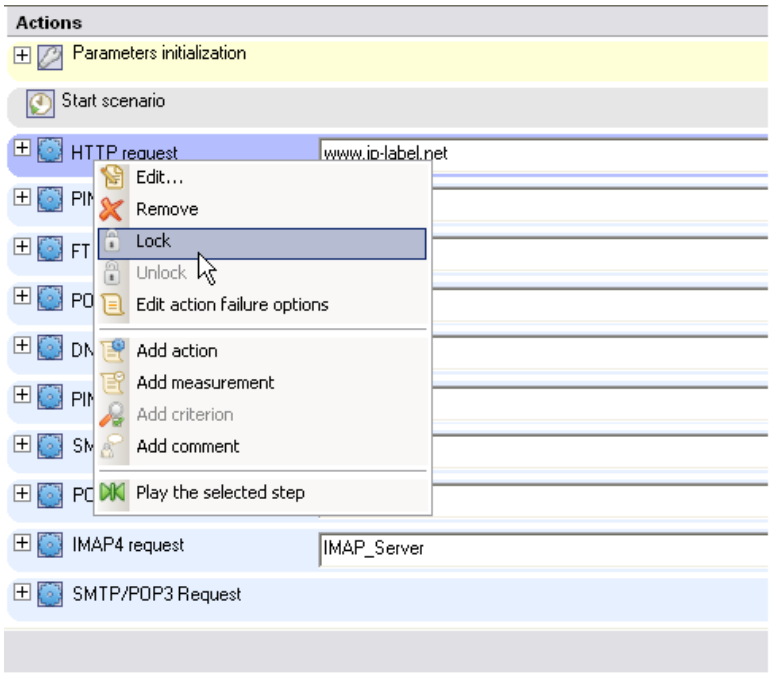
5.3.3.1 FEATURES OF SCENARIO ACTIONS

You can reorganize a scenario's actions by moving them with the mouse. To do so, click on the action you want to move, and drag it to its new position in the script.

Note: if you place an action outside the **Start scenario** and **Stop scenario** limits in the actions grid, the execution time of this action will not be counted in the overall execution time of the transaction.

If you move an action to a spot that is not permitted, the following symbol indicates that you cannot perform this move:  . When it is allowed to place the action where you intend, the following symbol appears:  .

By right-clicking an action in the list, you can choose to **Lock** this action so that it cannot be moved. You can also **Unlock** an action:



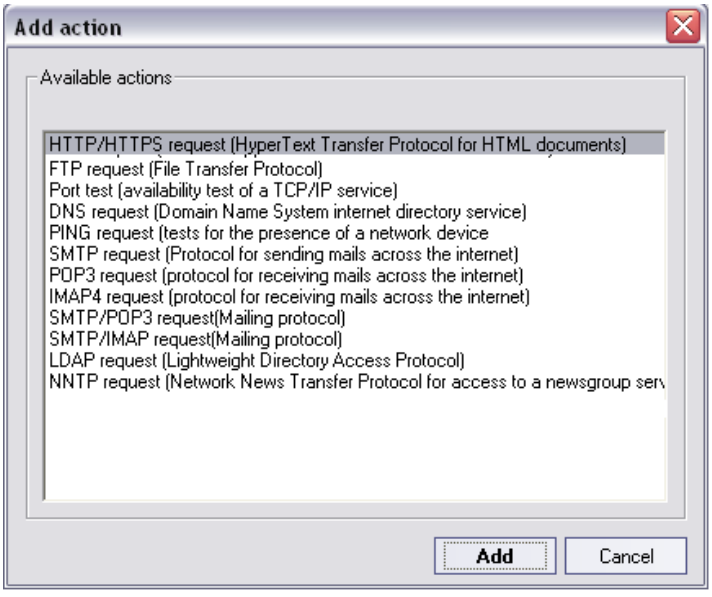
This can be useful when scenarios are very long, so that you can organize it and avoid errors in the order of the series of actions.

5.3.3.2 WIZARD TOOL BAR

In the upper part of the wizard screen, command buttons allow you to edit and run scenario requests:

 **Add action**: click to add a request to the list

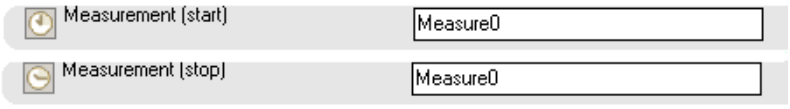
Choose the type of request to add from the selection provided:



 not available in the IP Analyser client environment

Add measurement: click to add a measurement to the list of actions.

The wizard then generates an automatically indexed measurement name as well as the measurement's start and stop actions:

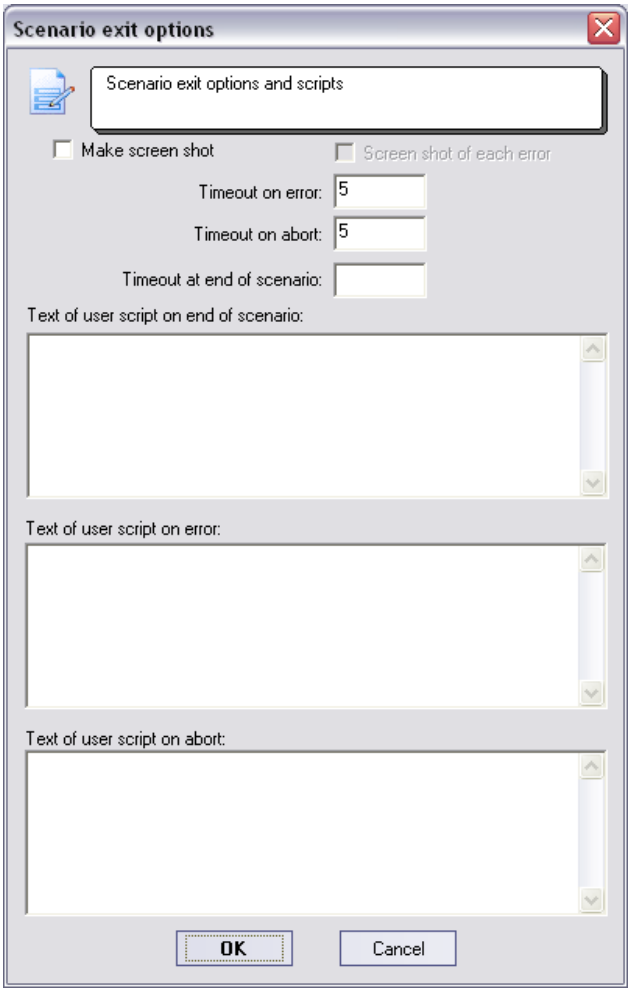


You can rename the measurement by typing a new one in the field; you may also move actions by dragging and dropping them with the mouse.

Remove selection: click to delete the selected objects from the list

Edit: click to edit the selected object (access to request settings)

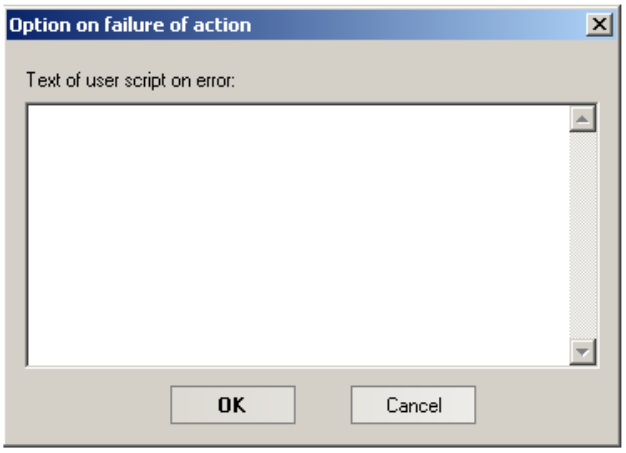
Edit scenario exit option: click to display scenario exit options:



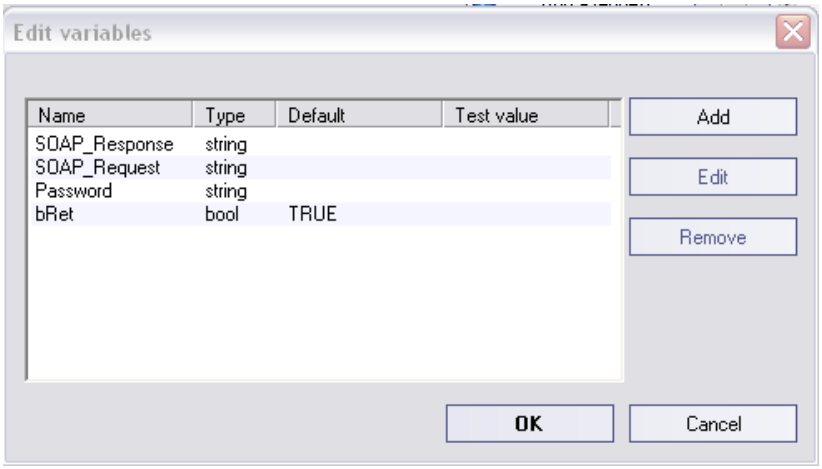
- **Make screenshot**: performs a screenshot for exits on error. Depending on the scenario's protocol type, this option may not be available, for instance when there is no visual element directly usable for execution diagnostics for this type of scenario.
- **Screenshot of each error**: select this option if you want to generate a screen shot for each successive error.

- **Pause on error**: timeout (Wait) before exit on error
- **Pause on abort**: timeout (Wait) before exit on abort
- **Pause at end of scenario**: timeout (Wait) before exit on a normal end of scenario
- **Text of user script on end of scenario**: code of the script to run prior to the normal end of scenario
- **Text of user script on error**: code of the script to run prior to the end of a scenario in error
- **Text of user script on abort**: code of the script to run prior to the end of a scenario in abort

 **Edit action failure option**: click to enter the code of a script to run in case of error in the selected step



 **Edit list of variables**: click to edit the list of configurable variables in the scenario script

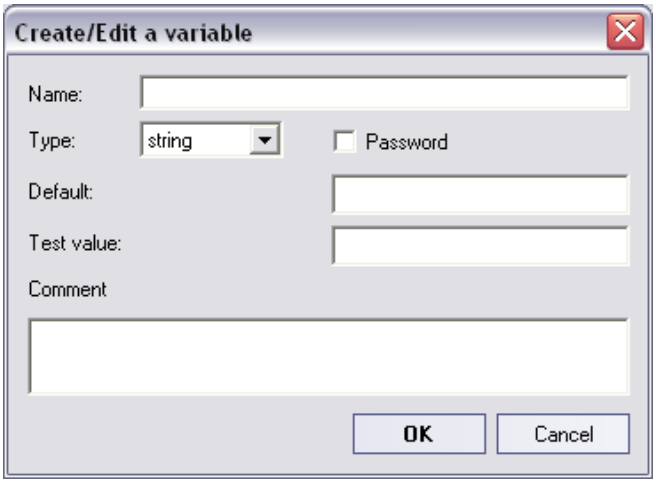


The table shows the variables that are present in the script and provides the following information about each one:

- **Name**: name of the variable
- **Type**: type of variable
- **Default**: default value for this variable (initialization)
- **Test Value**: test value of this variable for running the script in the wizard

The buttons in the interface allow you to act on variables in the script:

- **Add**: click to add a new variable to the list



  - **Name**: name of the new variable
  - **Type**: type of the new variable
  - **Password**: select this option of the variable will contain a password so that it will be encrypted in the script (function GetPassword)
  - **Default**: default value for this variable at the start of the script
  - **Test Value**: test value of this variable for running the script in the wizard
  - **Comment**: comments area to add to the initialization section of the script for this variable.

- **Edit**: to edit the parameters of the selected variable. This option is available only for the variables added to the initial list.
- **Remove**: to delete the selected variable. This option is available only for the variables added to the initial list.

 **Play all steps**: to play all the steps in the scenario

 **Next step**: to run the next step in the scenario

 **Stop execution**: to stop running the scenario

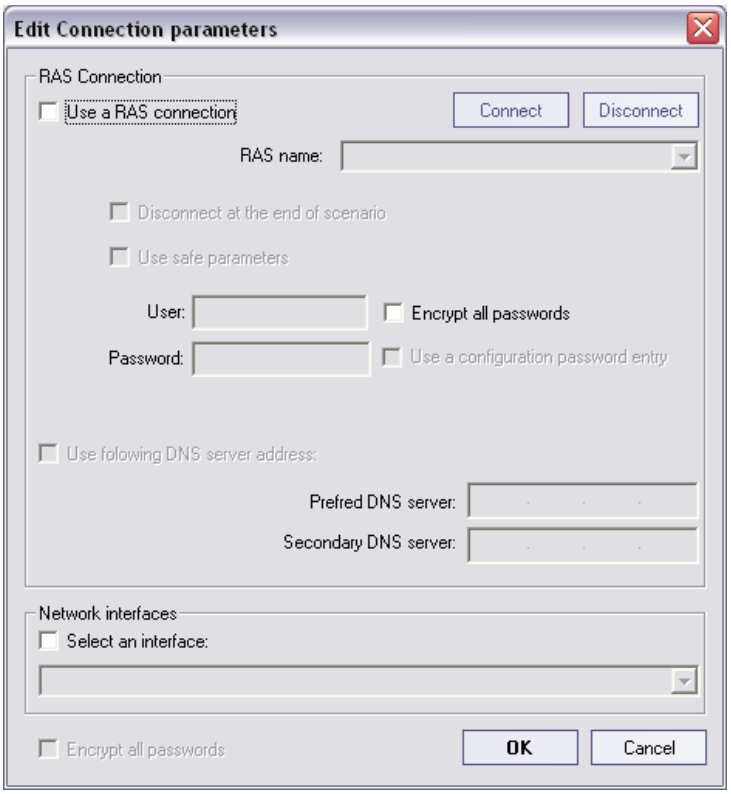
 **Play the selected step**: to run only the steps you selected in the assistant

5.3.4 CONFIGURING THE COMMUNICATION MODE

Click the button **Connection settings** to set the scenario's communication mode.

On this page, indicate whether the scenario is designed to operate on a local area network (LAN) or by remote access using a modem (RAS).

You may, when applicable, indicate the DNS servers (primary and secondary) you are using for a RAS connection, and specify the network interface to use.



Supply the following information:

- RAS connection
  - **Use a RAS connection**: to enable management of the RAS connection.

    Select the name of the RAS entry in the Windows configuration panel (the drop-down list containing the entries defined)
  - **Disconnect at the end of scenario**: provokes RAS disconnect at the end of the scenario's execution
  - **Use safe parameters**: for specifying a user login and password for secure access
  - **Encrypt all passwords**: to encrypt the password in the script
  - **Use a configuration password entry**: to use the password set in Newtest Transaction Builder's *Password* tab page
  - **Use following DNS server address**: select this option to specify the DNS servers to use when the RAS connection has been established.
  - **Preferred DNS server** / **Secondary DNS server**: enter the address or name of the primary and secondary DNS servers to take into account
- Network interface
  - **Select an interface**: select this option to specify the network card that the scenario will use (the drop-down list shows all the cards installed on the system)

5.3.4.1 MEASUREMENTS ASSOCIATED WITH THE RAS CONNECTION

Description of response time measurements (1/100th of a second)

Measurement	Description
RasConnection	Time it takes to connect to the RAS server (to the 1/100th of a second)
Authenticate	RAS authentication time (to the 1/100th of a second)
SetIPCP	IP address attribution time (to the 1/100th of a second)

Description of numeric measurements

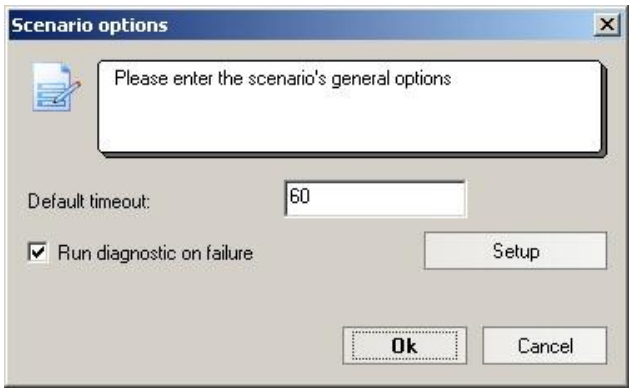
Measurement	Description
RasCode	Newtest RAS error code
RasCodeEx	RAS raw error code
RasSpeed	Negotiated speed (bps)

Description of string measurement

Measurement	Description
RasErrorString	RAS error label

5.3.5 CONFIGURING THE SCENARIO'S GENERAL OPTIONS

The **Scenario options** fields offer access to the interface for setting the scenario's general options.



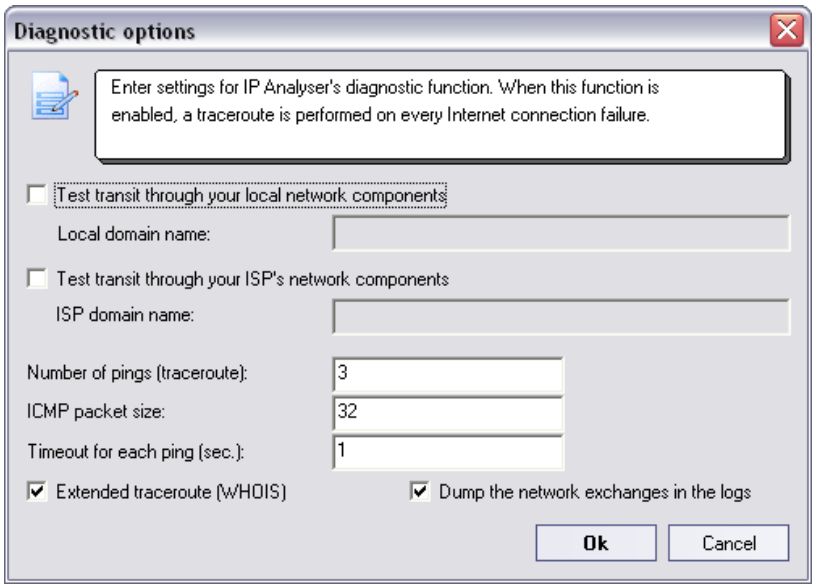
Please supply the following information:

- **Default timeout**: default timeout is 60 seconds
- **Run diagnostic on failure**: select this option to define diagnostic settings

  Click the **Setup** button to set diagnostic options:

  - **Test transit through your local network components**: select this option to check whether one of the components in the local network is the cause of disruptions
  - **Local domain name**: type the name of the local network domain

- **Test transit through your ISP's network components**: select this option to check whether one of the components of your ISP's network is the cause of disruptions
- **ISP domain name**: type of name of your ISP's domain
- **Number of pings (traceroute)**: enter the number of pings the diagnostic function should issue per traceroute analysis command
- **ICMP packet size**: indicate the size of the packet transmitted by the ping requests *(default is 32 bytes)*
- **Timeout for each ping (sec.)**: indicate the timeout for ping requests *(default is 1000 ms)*
- **Extended traceroute (WHOIS)**: to enable extended traceroute (identification of the devices transited by the traceroute command)
- **Dump the network exchanges in the logs**: select this option to record the log of TCP packets upon each launch of the request (advanced analysis). These appear in blue in the NTBR message area and are recorded in the SCHEDULE.TRC log file on robots.



5.4 STRUCTURE OF IP ANALYSER GENERATED SCRIPTS

IP Analyser scenario scripts that are generated by the wizard have a conventional structure. They are divided into distinct sections (Diagnostic, RAS, and the request). Each part is introduced by a comments section. As is the case with any NEWTEST scenario, a script header opens the script and a script footer appears at the end.

<declaration block>

<Diagnostic configuration section>

<RAS connection section>

<request section>

<script footer>

5.4.1 AVAILABLE FUNCTIONS

FUNCTION	DESCRIPTION
AddHttpHeader	Ajout d'un entête spécifique pour le protocole HTTP
ConnectRas	Opens a RAS connection
DisConnectRas	Disconnects a current RAS session
DnsRequest	Sends a DNS request to check whether the DNS server is functioning correctly (connection to a server and resolution of the server name)
FtpRequest	Sends a request to check for the availability of a file on an FTP server
GenerateUniqueMailSubject	Generates a unique message subject to be used for the SMTP request
GenerateUniqueNntpSubject	Génération d'un sujet unique destiné aux fonctions NNTP
GetDnsRequestResult	Returns the IP address resolved by the DNS request
GetDnsRequestTime	Returns the overall time of the DNS request
GetDnsServerTime	Returns the time of the DNS request itself
GetDnsTime	Returns the DNS resolution time for the DNS request
GetFtpAuthenticationTime	Returns FTP server authentication time
GetFtpConnectTime	Returns the time to connect to the FTP server or to the proxy server
GetFtpDnsTime	Returns the DNS resolution time for the FTP request
GetFtpOpenDataTime	Returns the time it takes to open an FTP data channel for the fransfer
GetFtpRequestTime	Returns the overall time of an FTP request
GetFtpTransferRate	Returns the transfer rate for the FTP request
GetFtpTransferTime	Returns the downloading time of the file for the FTP request
GetHttpConnectTime	Returns the time to connect to the HTTP server or to the proxy server.
GetHttpDnsTime	Returns the DNS resolution time for an HTTP request
GetHttpDownloadRate	Returns the transfer rate of the HTTP request
GetHttpDownloadTime	Returns a downloading time of the document for the HTTP request.
GetHttpFirstByteTime	Returns the time elapsed between the start of the sending of the

	request and the reception of the first byte of data
GetHttpRequestTime	Returns the overall time of the HTTP request
GetImap4AuthenticationTime	Returns the IMAP4 server authentication time
GetImap4ConnectTime	Returns the time to connect to the server
GetImap4DnsTime	Returns the DNS resolution time for the IMAP4 request
GetImap4ReceiveRate	Returns the message's rate of transfer to the client computer (in bytes/second)
GetImap4ReceiveTime	Returns the time to receive the message on the client computer
GetImap4RequestTime	Returns the overall time of the IMAP4 request
GetImap4WaitMailTime	Returns the time to find a message on an IMAP4 server
GetLdapBindTime	Returns the time to connect (bind) to the server
GetLdapDnsTime	Returns the DNS resolution time for the LDAP request
GetLdapRequestTime	Returns the overall time of the LDAP request
GetLdapSearchTime	Returns the time to search for the value associated with a given attribute
GetNntpArticlesNb	Retourne le nombre d'articles dans un NewsGroup
GetNntpAuthenticationTime	Returns the NNTP server authentication time
GetNntpCancelArticleTime	Retourne le temps de suppression d'un article
GetNntpConnectTime	Returns the time to connect to the server
GetNntpDnsTime	Returns the DNS resolution fime for the NNTP request P
GetNntpGetArticleRate	Retourne le taux de transfert de l'article (en octet/seconde)
GetNntpGetArticleTime	Retourne le temps de téléchargement d'un article
GetNntpGroupsTime	Returns the time to verify the existence of the newsgroup
GetNntpListArticlesTime	Retourne le temps de lister tous les articles
GetNntpListNewsGroupsTime	Retourne le temps de lister tous les NewsGroups
GetNntpNewsGroupsNumber	Retourne le nombre des NewsGroups
GetNntpPostArticleRate	Retourne le taux de transfert de l'article (en octet/seconde)
GetNntpPostArticleTime	Retourne le temps d'envoie d'un article
GetNntpRequestTime	Returns the overall time of the NNTP request
GetNntpSearchArticleTime	Retourne le temps nécessaire pour chercher un article dans un newsgroup
GetNntpWaitArticleTime	Retourne le temps nécessaire pour trouver un article
GetPasswordEx	Reads an encrypted password
GetPingAvgTime	Returns the average time of a series of ping roundtrips
GetPingDnsTime	Returns the DNS resolution time for the PING request
GetPingLost	Returns the number of pings lost in a series of pings
GetPingLostPercent	Returns the percentage of lost pings in a series of pings
GetPingMaxTime	Returns the maximum time in a series of ping roundtrips
GetPingMinTime	Returns the minimum time in a series of ping roundtrips
GetPingReceived	Returns the number of pings received in a series of pings
GetPingSent	Returns the number of pings sent in a series of pings
GetPingTotalTime	Returns the overall time of the PING request
GetPop3AuthenticationTime	Returns the POP3 server authentication time

GetPop3ConnectTime	Returns the time to connect to the server
GetPop3DnsTime	Returns the DNS resolution time for the POP3 request
GetPop3ReceiveRate	Returns the message's rate of transfer to the client computer (in bytes/second)
GetPop3ReceiveTime	Returns the time to receive the message on the client computer
GetPop3RequestTime	Returns the overall time of the POP3 request
GetPop3WaitMailTime	Returns the time to find a message on an POP3 server
GetPortConnectTime	Returns the time to connect to the server
GetPortDnsTime	Returns the DNS resolution time for the PORT request
GetPortRequestTime	Returns the overall time of the PORT request
GetRasAuthenticationTime	Returns RAS authentication time
GetRasConnectTime	Returns RAS connection time
GetRasErrorCode	Returns the Newtest RAS error code
GetRasErrorLabel	Returns the Microsoft error code label for RAS
GetRasExtErrorCode	Returns the Microsoft raw error code for RAS
GetRasIpCpTime	Returns the time it takes to establish the RAS IP address
GetRasLocalIpAdress	Returns the local IP address assigned by the RAS server
GetRasServerIpAdress	Returns the gateway's IP address
GetRasSpeed	Returns RAS negotiated speed
GetSmtpAuthenticationTime	Returns SMTP server authentication time
GetSmtpConnectTime	Returns the time to connect to the server
GetSmtpDnsTime	Returns the DNS resolution time for the SMTP request
GetSmtpRequestTime	Returns the overall SMTP request time
GetSmtpSendRate	Returns the message's rate of transfer to the SMTP server (in bytes/second)
GetSmtpSendTime	Returns the message's sending time to the SMTP server
HttpRequest	Sends an HTTP request to check the availability of a specified URL on an HTTP server
Imap4Purge	Deletes one or more messages having a given subject
Imap4Request	Sends an IMAP4 request to check the availability of an IMAP4 server
InitEnvironment	Reinitializes the options set by the SetOption function
InitModem	Reinitializes the modem(s)
IsRasConnected()	Indicates whether the connection is already open ("Always on" connections)
LdapRequest	Sends an LDAP request to check the availability of an LDAP server (connection to and search for a value associated with a given attribute)
MobileParamExtrac	Fonction utilisée pour extraire les informations des messages en provenance d'un BlackBerry
NntpCancelArticles	Supprimer un article
NntpConnect	Etablir une connexion vers un serveur NNTP
NntpDisConnect	Déconnexion d'un serveur NNTP
NntpGetArticles	Télécharger un article

NntpListArticles	Lister tous les articles
NntpListNewsGroups	Lister tous les NewsGroups
NntpPostAndWaitArticle	Envoyer puis attendre un article dans un NewsGroup
NntpPostArticles	Envoyer un article dans un Newsgroups
NntpRequest	Sends an NNTP request to check for the availability of an NNTP server (connects and verifies the existence of a given newsgroup)
NntpWaitArticle	Attendre l'apparition d'un article dans le NewsGroup
PingCommand	Sends a series of pings to a network device.
PingRequest	Sends a ping request to check for the presence of a network device.
Pop3Purge	Deletes one or more messages having a given subject
Pop3Request	Sends a POP3 request to check the availability of a POP3 server
PortRequest	Sends a request to a TCPIP port of a specified server to check the connection
SetOption	Sets protocol, RAS connection or diagnostic options
SmtpRequest	Sends an SMTP request to check the availability of an SMTP server
TcpClose	Fermeture d'une connexion TCP
TcpConnect	Ouverture d'une connexion TCP
TcpRecv	Récupération de données sur une socket TCP.
TcpRecvFile	Récupération de données sur une socket TCP dans un fichier.
TcpSend	Envoi de données sur une socket TCP.
TCPSendFile	Sends a file to a TCP socket.
TraceRoute	Performs a TraceRoute on a network device. This function can be used independently of the IP Analyser client environment.

5.4.2 BASIC SCRIPT STRUCTURE (PORT REQUEST)

The example below shows the general structure of a script. A PORT request is used here for the example. The aspects specific to each different protocol are described in subsequent sections.

```
///
/// Scenario: Ipanalyser_port
/// Author: Patrick
/// Last generation time: 03/05/09 10:58:07
///

//@DECLARATION
DIM Response AS STRING
DIM User AS STRING
DIM Password AS STRING
DIM bRet AS BOOL
DIM PORT_Test AS ORDER(TRESP)
DIM PORT1_Dns AS ORDER(NUM,"msec")
DIM PORT2_Connect AS ORDER(NUM,"msec")
//@

//@BEGIN
// Begin program
   bRet = TRUE
   If bRet = TRUE Then
      //
      // Init IP
      //
      InitEnvironment()

      SetOption("DIAG_ENABLED","",1)
      SetOption("DIAG_PINGS_NB","",3)
      SetOption("DIAG_PACKET_SIZE","",32)
      SetOption("DIAG_LOCAL_DOMAIN","",0)
      SetOption("DIAG_ISP_DOMAIN","",0)
      SetOption("DIAG_PING_TIMEOUT","",1)
      SetOption("DIAG_EXTENTED_TRACEROUTE","",1)
      SetOption("DIAG_IP_TRACE","",-1)

   ENDIF
   If bRet = FALSE Then
      RET ErrorLine("Selector")
      GOTO ERROR
   ENDIF
   //@

   //@ACTION id=BVRFI4BH4jj4CQcz2tAyBn0a
      bRet = TRUE
      wait(2)
      RET SetScenarioStartTime()
      If bRet = TRUE Then
         // PORT
         // —
         SetOption("PORT_SERVER","server1",0)
         SetOption("PORT_PORT","",12009)
         SetOption("PORT_TIMEOUT","",10)
```

```
        // PORT request
        // ————
        bRet = PortRequest()


        // PORT Measurements
        // ————————
        IF bRet=TRUE THEN
          GetPortRequestTime(PORT_Test)
          RET PORT_Test
          GetPortDnsTime(PORT1_Dns)
          RET PORT1_Dns
          GetPortConnectTime(PORT2_Connect)
          RET PORT2_Connect
        ENDIF
      ENDIF
      If bRet = FALSE Then
        RET ErrorLine("Error3")
        GOTO ERROR
      ENDIF
      RET SetScenarioStopTime()
  //@

  //@END
  Wait(5)
// END program
END
//@

//@ERROR
ERROR:
  Wait(5)
END
//@

//@CANCEL
CANCEL:
  Wait(5)
  RET SetScenarioCanceled()
END
//@
```

5.4.3 STRUCTURE OF THE HTTP REQUEST SECTION

```
// HTTP
// —

SetOption("WEB_PROXY","",0)
SetOption("HTTP_USESSL","",0)
SetOption("HTTP_FOLLOW_REDIRECT","",0)
 TmpStr = UnescapeString("ip-label.newtest,ip-label")
SetOption("HTTP_CONTROLSTRING",TmpStr,0)
TmpStr = UnescapeString("")
 SetOption("HTTP_POST_DATA",TmpStr,0)
 SetOption("HTTP_URL","www.ip-label.net",0)
 SetOption("HTTP_PORT","",80)
 SetOption("HTTP_TIMEOUT","",10)

        // HTTP request
         // ———
         bRet = HttpRequest()

         // HTTP Measurements
         // ————
         IF bRet = TRUE THEN
           GetHttpRequestTime(HTTP_Test)
           RET HTTP _Test
           GetHttpDnsTime(HTTP 1_Dns)
           RET HTTP 1_Dns
           GetHttpConnectTime(HTTP 2_Connect)
           RET HTTP 2_Connect
           GetHttpFirstByteTime(HTTP 3_Firstbyte)
           RET HTTP 3_Firstbyte
           GetHttpDownloadTime(HTTP 4_GetTime)
           RET HTTP 4_GetTime
           GetHttpDownloadRate(HTTP 5_GetRate)
           RET HTTP 5_GetRate
         ENDIF
```

5.4.4 STRUCTURE OF THE FTP REQUEST SECTION

```
// FTP
        // —
        SetOption("PROTOCOL_USER","newtest@test.net",0)
        Password=""

GetPassword("#|067E85A8AB0447FAF48D4A5378244A1172BBB1B9B8D2905EB496FF931864A53D
A3DC9C98323DF954322A777941D5D4CD", Password)
        SetOption("PROTOCOL_PASSWORD",Password,0)
        SetOption("FTP_SERVER","ftp.test.net",0)
        SetOption("FTP_PORT","",21)
        SetOption("FTP_USEPASSIVEMODE","",0)
        SetOption("FTP_REQUEST_TYPE","get",0)
        SetOption("FTP_LOCALFILE","",0)
        SetOption("FTP_DISTANTFILE","/demo.pdf",0)
        SetOption("FTP_CONTROLSIZE","",0)
        SetOption("FTP_FILETYPE","binary",0)
        SetOption("FTP_DISABLE_TRANSFER_CMDS","",0)
        SetOption("FTP_TIMEOUT","",20)
```

```
// FTP request
// ———
bRet = FtpRequest()

// FTP Measurements
// ————
IF bRet=TRUE THEN
  GetFtpRequestTime(FTP_Test)
  RET FTP_Test
  GetFtpDnsTime(FTP1_Dns)
  RET FTP1_Dns
  GetFtpConnectTime(FTP2_Connect)
  RET FTP2_Connect
  GetFtpAuthenticationTime(FTP3_Auth)
  RET FTP3_Auth
  GetFtpOpenDataTime(FTP4_OpenData)
  RET FTP4_OpenData
  GetFtpTransferTime(FTP5_GetTime)
  RET FTP5_GetTime
  GetFtpTransferRate(FTP6_GetRate)
  RET FTP6_GetRate
ENDIF
```

5.4.5 STRUCTURE OF THE DNS REQUEST SECTION

```
 // DNS
// —
SetOption("DNS_USE_UDP","",0)
SetOption("DNS_SERVER","server1",0)
SetOption("DNS_PORT","",53)
SetOption("DNS_HOST","www.ip-label.net",0)
SetOption("DNS_TIMEOUT","",10)

// DNS request
// ———
bRet = DnsRequest()

IF bRet=TRUE THEN
  GetDnsRequestTime(DNS_Test)
  RET DNS_Test
  GetDnsTime(DNS1_Dns)
  RET DNS1_Dns
  GetDnsServerTime(DNS2_Server)
  RET DNS2_Server
  GetDnsRequestResult(Response)
ENDIF
```

5.4.6 STRUCTURE OF THE PING REQUEST SECTION

```
// PING
        // —
        SetOption("PING_HOST","server1",0)
        SetOption("PING_NB","",4)
        SetOption("PING_SIZE","",1)
        SetOption("PING_TIMEOUT","",10)

        // PING request
        // ———
        bRet = PingRequest()

        // PING Measurements
        // ————
        IF bRet = TRUE THEN
          GetPingRequestTime(PING_Test)
          RET PING_Test
          GetPingDnsTime(PING1_Dns)
          RET PING1_Dns
          GetPingMaxTime(PING2_MaxTime)
          RET PING2_MaxTime
          GetPingMinTime(PING3_MinTime)
          RET PING3_MinTime
          GetPingAvgTime(PING4_AvgTime)
          RET PING4_AvgTime
          GetPingSent(PING5_NbSent)
          RET PING5_NbSent
          GetPingReceived(PING6_NbRcvd)
          RET PING6_NbRcvd
          GetPingLost(PING7_NbLost)
          RET PING7_NbLost
          GetPingLostPercent(PING8_LostPct)
          RET PING8_LostPct
        ENDIF
```

5.4.7 STRUCTURE OF THE SMTP MAIL REQUEST SECTION

```
        // SMTP
        // —

        Subject = "Newtest message"
        Body = UnescapeString("This is a test message from Newtest Robot")

        Password=""

GetPassword("#|067E85A8AB0447FAF48D4A5378244A11012539602C5E361F3AEE165A8AE779699
3A5A1927F1C61B89E3521A9DE03F8CB", Password)
        SetOption("PROTOCOL_USER","user",0)
        SetOption("PROTOCOL_PASSWORD",Password,0)
        SetOption("SMTP_FROM","iplabel@ip-label.com",0)
        SetOption("SMTP_TO","mailbox@ip-label.com",0)
        SetOption("MAIL_SUBJECT",Subject,0)
        SetOption("SMTP_ATTACHFILE","$(SCENARIO_DIR)\test.txt",0)
        SetOption("SMTP_MESSAGESIZE","",41)
        SetOption("SMTP_SERVER","smtpserver",0)
```

```
SetOption("SMTP_PORT","",25)
SetOption("SMTP_USESSL","",0)
SetOption("SMTP_TIMEOUT","",10)
SetOption("MAIL_BODY",Body,0)

// SMTP request
// ————
bRet = SmtpRequest()

// SMTP Measurements
// ————————
IF bRet= TRUE THEN
  GetSmtpRequestTime(SMTP_Test)
  RET SMTP_Test
  GetSmtpDnsTime(SMTP1_Dns)
  RET SMTP1_Dns
  GetSmtpConnectTime(SMTP2_Connect)
  RET SMTP2_Connect
  GetSmtpAuthenticationTime(SMTP3_Auth)
  RET SMTP3_Auth
  GetSmtpSendTime(SMTP4_SendTime)
  RET SMTP4_SendTime
  GetSmtpSendRate(SMTP5_SendRate)
  RET SMTP5_SendRate
ENDIF
```

5.4.8 STRUCTURE OF THE POP3 MAIL REQUEST SECTION

```
 // POP3
// ——

Subject = "Test message"
        Sender = "newtest@iplabel.net"
        Body = "Test email"
        MailDate =""


        Password=""

GetPassword("#|067E85A8AB0447FAF48D4A5378244A115D5284004C5D120A8DA7617EEE946F9D
D075F783F2247F938BC7E817C9E38742", Password)
        SetOption("PROTOCOL_USER","POP_user",0)
        SetOption("PROTOCOL_PASSWORD",Password,0)
        SetOption("MAIL_SUBJECT",Subject,0)
        SetOption("MAIL_FROM",Sender,0)
        SetOption("MAIL_REGEXP","",0)
        SetOption("POP3_SERVER","POP3.server",0)
        SetOption("POP3_PORT","",110)
        SetOption("POP3_USESSL","",0)
        SetOption("POP3_TIMEOUT","",10)

        // POP3 request
        // ————
        bRet = Pop3Request()

        IF bRet= TRUE THEN
```

```
    Pop3Purge()

    Subject = GetStrOption("MAIL_SUBJECT")
    Sender  = GetStrOption("MAIL_FROM")
    MailDate = GetStrOption("MAIL_DATE")
    Body = GetStrOption("MAIL_BODY")


    GetPop3RequestTime(POP_Test)
    GetPop3DnsTime(POP1_Dns)
    GetPop3ConnectTime(POP2_Connect)
    GetPop3AuthenticationTime(POP3_Auth)
    GetPop3WaitMailTime(POP4_WaitMail)
    GetPop3ReceiveTime(POP5_ReceiveTime)
    GetPop3ReceiveRate(POP6_ReceiveRate)

    RET POP_Test
    RET POP1_Dns
    RET POP2_Connect
    RET POP3_Auth
    RET POP4_WaitMail
    RET POP5_ReceiveTime
    RET POP6_ReceiveRate
  ENDIF
```

5.4.9 STRUCTURE OF THE IMAP4 MAIL REQUEST SECTION

```
    // IMAP4
    // ——

Subject = "newtest imap test"
    Sender = "newtest@iplabel.net"
    Body = ""
    MailDate =""
    Folder = "inbox"

    Password=""

GetPassword("#|A568E6FDED0C338617E4A9E18CD6AF89277CAD1321E8DEA08AA10C85FDBD0EDB
DA700D079191CA95896F330BB2F724D1", Password)
    SetOption("PROTOCOL_USER","user",0)
    SetOption("PROTOCOL_PASSWORD",Password,0)
    SetOption("MAIL_SUBJECT",Subject,0)
    SetOption("MAIL_FROM",Sender,0)
    SetOption("MAIL_REGEXP","",0)
    SetOption("IMAP4_SERVER","IMAP_Server",0)
    SetOption("IMAP4_FOLDER",Folder,0)
    SetOption("IMAP4_PORT","",143)
    SetOption("IMAP4_USESSL","",0)
    SetOption("IMAP4_TIMEOUT","",10)

    // IMAP4 request
    // ————
    bRet = Imap4Request()

    IF bRet= TRUE THEN
```

```
  IMAP4Purge()

  Subject = GetStrOption("MAIL_SUBJECT")
  Sender  = GetStrOption("MAIL_FROM")
  MailDate = GetStrOption("MAIL_DATE")
  Body = GetStrOption("MAIL_BODY")
  Folder = GetStrOption("IMAP4_FOLDER")

  GetImap4RequestTime(IMAP_Test)
  RET IMAP_Test
  GetImap4DnsTime(IMAP1_Dns)
  RET IMAP1_Dns
  GetImap4ConnectTime(IMAP2_Connect)
  RET IMAP2_Connect
  GetImap4AuthenticationTime(IMAP3_Auth)
  RET IMAP3_Auth
  GetImap4WaitMailTime(IMAP4_WaitMail)
  RET IMAP4_WaitMail
  GetImap4ReceiveTime(IMAP5_ReceiveTime)
  RET IMAP5_ReceiveTime
  GetImap4ReceiveRate(IMAP6_ReceiveRate)
  RET IMAP6_ReceiveRate
ENDIF
```

5.4.10 STRUCTURE OF THE SMTP/POP3 MAIL REQUEST SECTION

```
// SMTP
// —

Subject = "_{72B34C66-9E70-4B63-9CE1-FC8B5814DECB}"

Password=""

GetPassword("#|E0D31748BB14A5FA68793AD0AAD32E2A7212CCF073C9BB77074E63F903097E94
C99C8B07DCCDE7601B8C05A71F9FD55B", Password)
SetOption("PROTOCOL_USER","user",0)
SetOption("PROTOCOL_PASSWORD",Password,0)
SetOption("SMTP_FROM","newtest@iplabel.net",0)
SetOption("SMTP_TO","test@iplabel.net",0)
SetOption("MAIL_SUBJECT",Subject,0)
SetOption("SMTP_ATTACHFILE","c:\test.txt",0)
SetOption("SMTP_MESSAGESIZE","",32)
SetOption("SMTP_SERVER","smtp_server",0)
SetOption("SMTP_PORT","",25)
SetOption("SMTP_USESSL","",0)
SetOption("SMTP_TIMEOUT","",10)

// SMTP request
// ———
bRet = SmtpRequest()

// SMTP Measurements
// ————

IF bRet = TRUE THEN

  GetSmtpRequestTime(SMTP_Test)
```

```
        RET SMTP_Test
        GetSmtpDnsTime(SMTP1_Dns)
        RET SMTP1_Dns
        GetSmtpConnectTime(SMTP2_Connect)
        RET SMTP2_Connect
        GetSmtpAuthenticationTime(SMTP3_Auth)
        RET SMTP3_Auth
        GetSmtpSendTime(SMTP4_SendTime)
        RET SMTP4_SendTime
        GetSmtpSendRate(SMTP5_SendRate)
        RET SMTP5_SendRate

        // POP3
        // ---
        Password=""

GetPassword("#|E0D31748BB14A5FA68793AD0AAD32E2A7212CCF073C9BB77074E63F903097E94
C99C8B07DCCDE7601B8C05A71F9FD55B", Password)
        SetOption("PROTOCOL_USER","user",0)
        SetOption("PROTOCOL_PASSWORD",Password,0)
        SetOption("MAIL_SUBJECT",Subject,0)
        SetOption("POP3_SERVER","pop_server",0)
        SetOption("POP3_PORT","",110)
        SetOption("POP3_USESSL","",0)
        SetOption("POP3_TIMEOUT","",10)

        // POP3 request
        // -------
        bRet = Pop3Request()

        IF bRet = TRUE THEN

                GetPop3RequestTime(POP_Test)
                RET POP_Test
                GetPop3DnsTime(POP1_Dns)
                RET POP1_Dns
                GetPop3ConnectTime(POP2_Connect)
                RET POP2_Connect
                GetPop3AuthenticationTime(POP3_Auth)
                RET POP3_Auth
                GetPop3WaitMailTime(POP4_WaitMail)
                RET POP4_WaitMail
                GetPop3ReceiveTime(POP5_ReceiveTime)
                RET POP5_ReceiveTime
                GetPop3ReceiveRate(POP6_ReceiveRate)
                RET POP6_ReceiveRate

                // POP3 PURGE
                Pop3Purge()
        ENDIF
      ENDIF
    ENDIF
    If bRet = FALSE Then
      RET ErrorLine("mail system error")
      GOTO ERROR
    ENDIF
```

5.4.11 STRUCTURE OF THE SMTP/IMAP4 MAIL REQUEST SECTION

```
        // SMTP
        // ——

Subject = "_{3F531B7E-3AF8-4E47-9279-89D60862436C}"

        Password=""

GetPassword("#|1AFFA90F6DD57A0FA92F528C697DEB5F8F1D03FD3EFF83271F354B24DC89CD429
02AFA96E8ADB87A6E69476A2AEA3F80", Password)
        SetOption("PROTOCOL_USER","user",0)
        SetOption("PROTOCOL_PASSWORD",Password,0)
        SetOption("SMTP_FROM","newtest@iplabel.net",0)
        SetOption("SMTP_TO","test@iplabel.net",0)
        SetOption("MAIL_SUBJECT",Subject,0)
        SetOption("SMTP_ATTACHFILE","c:\test.txt",0)
        SetOption("SMTP_MESSAGESIZE","",32)
        SetOption("SMTP_SERVER","smtp_server",0)
        SetOption("SMTP_PORT","",25)
        SetOption("SMTP_USESSL","",0)
        SetOption("SMTP_TIMEOUT","",10)

        // SMTP request
        // ———————
        bRet = SmtpRequest()

        // SMTP Measurements
        // ————————

          IF bRet = TRUE THEN
          GetSmtpRequestTime(SMTP_Test)
          RET SMTP_Test
          GetSmtpDnsTime(SMTP1_Dns)
          RET SMTP1_Dns
          GetSmtpConnectTime(SMTP2_Connect)
          RET SMTP2_Connect
          GetSmtpAuthenticationTime(SMTP3_Auth)
          RET SMTP3_Auth
          GetSmtpSendTime(SMTP4_SendTime)
          RET SMTP4_SendTime
          GetSmtpSendRate(SMTP5_SendRate)
          RET SMTP5_SendRate

          // IMAP4
          // ——
          Password=""

GetPassword("#|1AFFA90F6DD57A0FA92F528C697DEB5F8F1D03FD3EFF83271F354B24DC89CD429
02AFA96E8ADB87A6E69476A2AEA3F80", Password)
          SetOption("PROTOCOL_USER","user",0)
          SetOption("PROTOCOL_PASSWORD",Password,0)
          SetOption("MAIL_SUBJECT",Subject,0)
          SetOption("IMAP4_SERVER","IMAP_server",0)
          SetOption("IMAP4_PORT","",143)
          SetOption("IMAP4_USESSL","",0)
          SetOption("IMAP4_TIMEOUT","",10)
```

```
        // IMAP4 request
        // ———————
        bRet = Imap4Request()

        IF bRet = TRUE THEN

                GetImap4RequestTime(IMAP_Test)
                RET IMAP_Test
                GetImap4DnsTime(IMAP1_Dns)
                RET IMAP1_Dns
                GetImap4ConnectTime(IMAP2_Connect)
                RET IMAP2_Connect
                GetImap4AuthenticationTime(IMAP3_Auth)
                RET IMAP3_Auth
                GetImap4WaitMailTime(IMAP4_WaitMail)
                RET IMAP4_WaitMail
                GetImap4ReceiveTime(IMAP5_ReceiveTime)
                RET IMAP5_ReceiveTime
                GetImap4ReceiveRate(IMAP6_ReceiveRate)
                RET IMAP6_ReceiveRate

                // IMAP4 PURGE
                Imap4Purge()
        ENDIF
      ENDIF
    ENDIF
    If bRet = FALSE Then
      RET ErrorLine("Mailing system Error")
      GOTO ERROR
    ENDIF
```

5.4.12 STRUCTURE OF THE LDAP REQUEST SECTION

```
        // LDAP
        // —

        Password=""

GetPassword("#|1AFFA90F6DD57A0FA92F528C697DEB5F8F1D03FD3EFF83271F354B24DC89CD429
02AFA96E8ADB87A6E69476A2AEA3F80", Password)
        SetOption("PROTOCOL_USER","user",0)
        SetOption("PROTOCOL_PASSWORD",Password,0)
        SetOption("LDAP_DN","ou=experimental, o=University of Michigan, c=us",0)
        SetOption("LDAP_ATTRIBUTE","ObjectClass",0)
        SetOption("LDAP_VALUE","OrganisationalUnit",0)
        SetOption("LDAP_SERVER","LDAP_server",0)
        SetOption("LDAP_PORT","",389)
        SetOption("LDAP_TIMEOUT","",10)

        // LDAP request
        // ———————
        bRet = LdapRequest()


        // LDAP Measurements
        // ————————
```

```
    IF bRet = TRUE THEN
      GetLdapRequestTime(LDAP_Test)
      RET LDAP_Test
      GetLdapDnsTime(LDAP1_Dns)
      RET LDAP1_Dns
      GetLdapBindTime(LDAP2_Bind)
      RET LDAP2_Bind
      GetLdapSearchTime(LDAP3_Search)
      RET LDAP3_Search
    ENDIF

  ENDIF
  If bRet = FALSE Then
    RET ErrorLine("LDAP Error")
    GOTO ERROR
  ENDIF
```

5.4.13 STRUCTURE OF THE NNTP REQUEST SECTION

```
// NNTP
    // —
    Password=""

    SetOption("PROTOCOL_USER","",0)
    SetOption("PROTOCOL_PASSWORD",Password,0)
    SetOption("NNTP_NEWSGROUP","alt.news.iplabel",0)
    SetOption("NNTP_SERVER","NNTP_server",0)
    SetOption("NNTP_PORT","",119)
    SetOption("NNTP_TIMEOUT","",10)

    // NNTP request
    // ———
    bRet = NntpRequest()

    // NNTP Measurements
    // ————————
    IF bRet = TRUE THEN
      GetNntpRequestTime(NNTP_Test)
      RET NNTP_Test
      GetNntpDnsTime(NNTP1_Dns)
      RET NNTP1_Dns
      GetNntpConnectTime(NNTP2_Connect)
      RET NNTP2_Connect
      GetNntpAuthenticationTime(NNTP3_Auth)
      RET NNTP3_Auth
      GetNntpGroupsTime(NNTP4_GetGroups)
      RET NNTP4_GetGroups
    ENDIF

  ENDIF
  If bRet = FALSE Then
    RET ErrorLine("NNTP Error")
    GOTO ERROR
  ENDIF
```

6 SCENARIOS BUILT WITH THE JAVA APPLET HTTP-BASED CLIENT/SERVER ENVIRONMENT

Important: starting with version 7.5 the NewtRIA client environment replaces the Java client environment.

7 SCENARIOS BUILT WITH THE NEWTESTRIA CLIENT ENVIRONMENT

7.1 INTRODUCTION TO THE NEWTESTRIA CLIENT ENVIRONMENT

The NewtestRIA (Rich Internet Application) client environment is designed for rich web applications; it replaces the Java Applet client environment. As is true of Java-type scenarios, those built with NewtestRIA allow you to automate a series of actions that an user would perform on the graphical interface of a web browser, and more particularly on the user interface of an internet rich application (Java, Flash component, Silverlight, etc.).

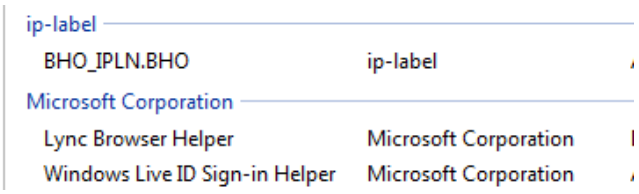
Unlike the Java Applet client/server environment, which operates solely with an Internet Explorer browser, the NewtestRIA client environment offers a wizard that enables you to set up either Internet Explorer or Firefox. It also provides improved diagnostics and analysis of web sites.

7.2 CONFIGURING THE ENVIRONMENT

The NewtRIA environment requires the 2 following browser plug-ins which are delivered with the install package:

- BHO (Browser Helper Objects)
- HttpWatch

You can check whether these modules are present with the browser's **add-ons manager**:



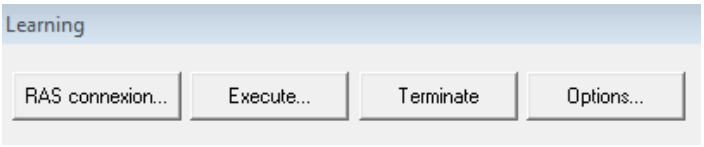
The basic version of HttpWatch (v9) that is provided supports IE11. To benefit from the latest fixes, you can upgrade it to the latest version at www.httpwatch.com.

7.3 SPECIFICS OF CREATING A NEWTESTRIA SCENARIO

7.3.1 CONFIGURING A NEWTESTRIA SCENARIO

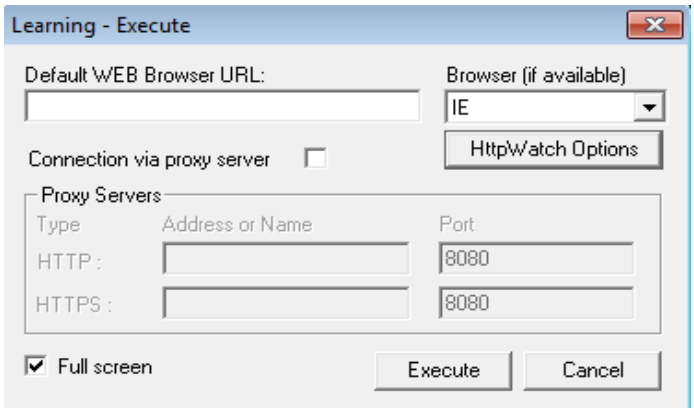
Define scenario settings as described in *Programming scenarios (Developer's Manual vol. 1)*, selecting **NewtestRIA** as the client environment.

7.3.2 LEARNING MODE – NEWTESTRIA SCENARIO

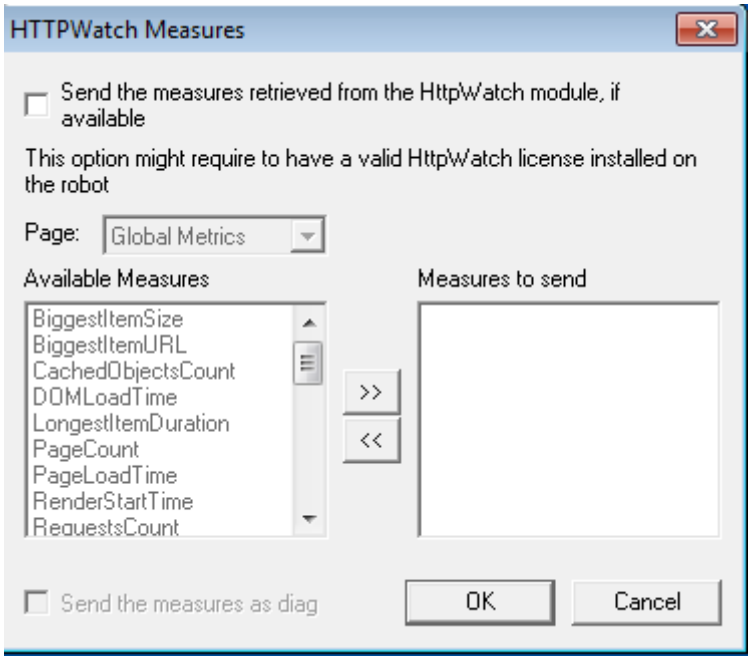


Click **Execute** to begin a learning session.

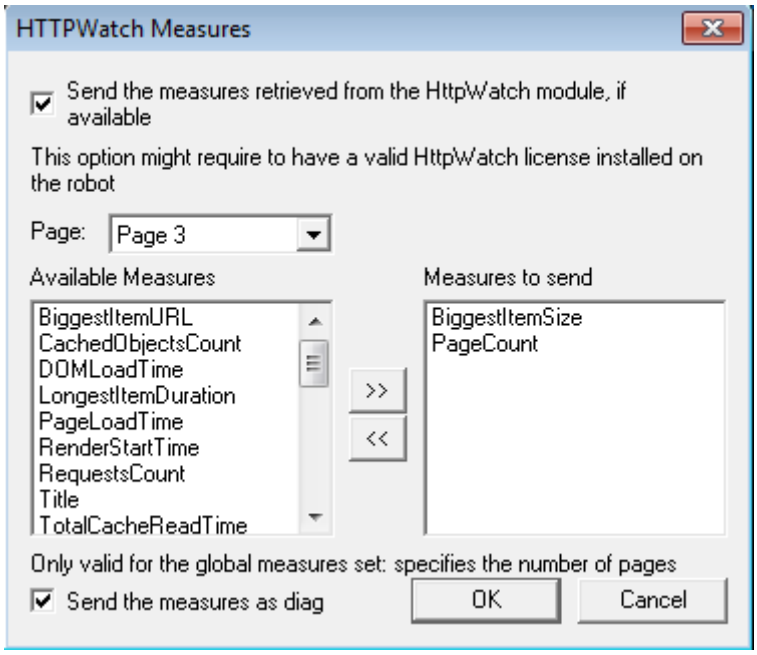
The Learning window then appears, and offers the choice of browser to use for the script (Internet Explorer or Firefox):



When you click **HttpWatch Options** the following window appears:



You can request the extraction of specific HttpWatch metrics. To receive them you must have HttpWatch Professional Edition.



Select the metrics to send by page, select the page number, and select the measurements to extract and send to Newtest Management Console.

The metrics listed below are available depending on the type of license:

- **hwl**: breakdown of page loading in diagnostic format
- **PageCount**: numeric; number of pages
- **Title**: log appli; page title
- **TotalTime**: response time in ms; total loading time
- **DOMLoadTime**: response time in ms; DOM loading time
- **RenderStartTime**: response time in ms; render time
- **PageLoadTime**: response time in ms; page loading time
- **TotalDNSLookupCount**: numeric; number of DNS requests
- **TotalDNSLookupTime**: response time in ms; DNS lookup time
- **TotalServerWaitTime**: response time in ms; server waiting time
- **TotalCacheReadTime**: response time in ms; time spent reading the cache
- **TotalSize**: numeric; total size in bytes
- **TotalImagesSize**: numeric; total size of images in bytes
- **TotalImagesCount**: numeric; total number of images
- **TotalFlashSize**: numeric; total size of Flash objects in bytes
- **TotalFlashCount**: numeric; total number of Flash objects
- **TotalDataSize**: numeric; total size of data in bytes
- **TotalDataCount**: numeric; total number of data objects
- **TotalCssSize**: numeric; total size of css in bytes
- **TotalCssCount**: numeric; total number of css
- **TotalJSSize**: numeric; total size of JavaScripts in bytes

- **TotalJSCount**: numeric; total number of JavaScripts
- **RequestsCount**: numeric; total number of requests
- **BiggestItemSize**: numeric; size of the largest item in bytes
- **BiggestItemURL**: log appli; URL of the largest item
- **LongestItemDuration**: response time ; time of the item that takes the longest to load
- **TTFB**: first byte response time

7.3.3 SAMPLE NEWTESTRIA GENERATED SCRIPT

```
// ————————————————————————————
//   Test_ria.SCE - 21/02/2012 14:51:18
// ————————————————————————————

DIM Global_time AS ORDER(TRESP)
DIM Launch_time AS ORDER(TRESP)
DIM TimeOut AS INT
DIM bRet AS BOOL
TimeOut = 90


bRet=ActorExec()
TopChrono (Global_time)
IF bRet=FALSE THEN
        RET ErrorLine("Err001")
        GOTO ERROR
ENDIF
Wait(2)

SetOption("BROWSER","IE",0)
SetOption("FULLSCREEN","",1)
SetOption("USEPROXY","",0)
SetOption("HTTP_PROXY","",0)
SetOption("HTTP_PROXYPORT","",8080)
SetOption("HTTPS_PROXY","",0)
SetOption("HTTPS_PROXYPORT","",8080)
bRet=AppExec("www.ip-label.net")
TopChrono (Launch_time)
IF WaitWnd(":t:Measuring QoE for all digital services - Windows Internet Explorer",TimeOut)=FALSE THEN
        RET Error(Launch_time)
        GOTO ERROR
ENDIF
IF WaitAppStrOcr("Discover",0,2,0,TimeOut,426,220,133,37,0)=FALSE THEN
        RET Error(Launch_time)
        GOTO ERROR
ELSE
        RET LogTps(Launch_time)
ENDIF
Wait(2)

IF WaitWnd(":t:Measuring QoE for all digital services - Windows Internet Explorer",TimeOut)=FALSE THEN
        RET ErrorLine("Err002")
        GOTO ERROR
ENDIF
ActivateWnd(3,-1,-1)
```

```
MouseClick(0,279,199)
IF WaitWnd(":t:End User Monitoring  QoE Solutions - Windows Internet Explorer",TimeOut)=FALSE THEN
	RET ErrorLine("Err003")
	GOTO ERROR
ENDIF
IF WaitAppBitmap("$(SCENARIO_NAME)001.bmp",TimeOut,198,391,124,152,0)=FALSE THEN
	RET ErrorLine("Pb image not found")
	GOTO ERROR
ENDIF
Wait(2)

IF WaitWnd(":t:End User Monitoring  QoE Solutions - Windows Internet Explorer",TimeOut)=FALSE THEN
	RET ErrorLine("Err004")
	GOTO ERROR
ENDIF
ActivateWnd(1,-1,-1)
SendKey("%{F4}")
Wait(2)

AppClose()
AddGenDiagFile("$(HTTPW_RESULTS)","","webrunner_digest.xml",false,1)
AddGenDiagFile("$(HTTPW_HWL_PATH)","","digest.hwl",false,0)
SendGenDiag("")
RET LogTps(Global_time)
Wait(2)

Wait(5)
END

ERROR:

HardCopy(1)
AppClose()
Wait(5)
```

7.4 DOM INSPECTION WIZARD

The DOM function can be added by right-clicking in the generated script, and then selecting 'Insert a DOM action':

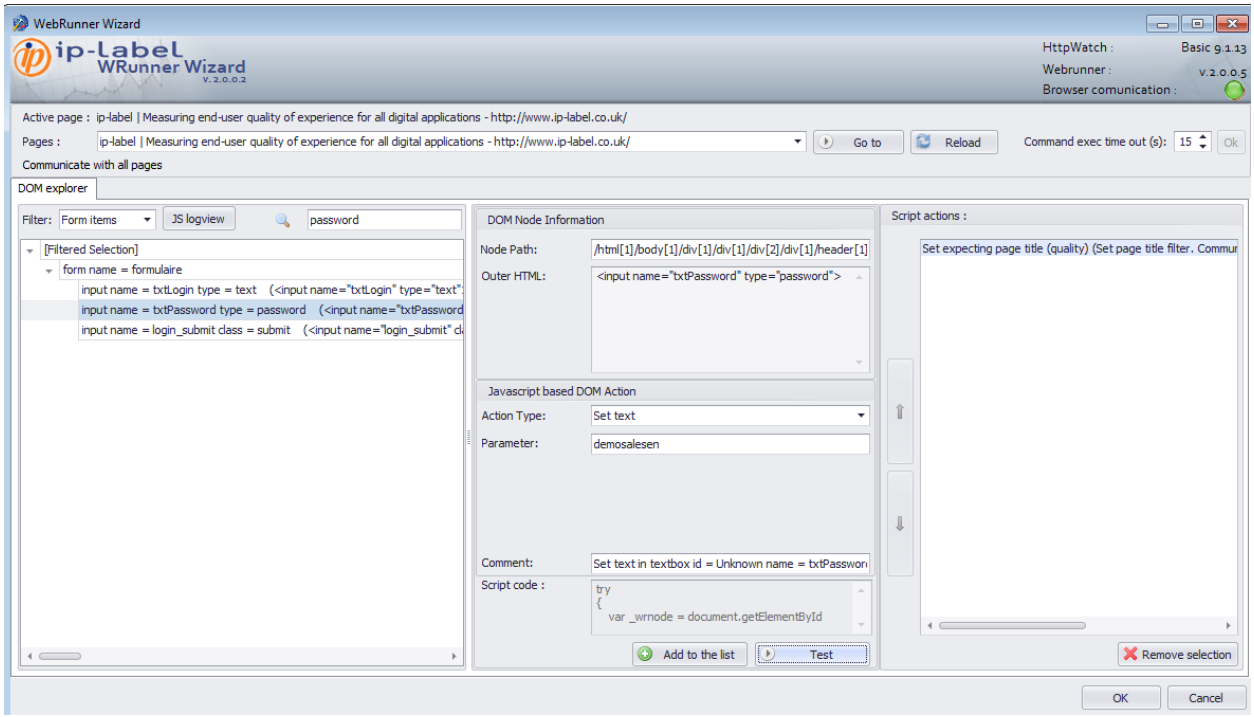
The wizard prompts you to fill in the URL and browser:

Note: to function correctly this tool requires the installation of ip-label plug-ins in the browsers (HBO-IPLN plug-ins visible in the add-ons).

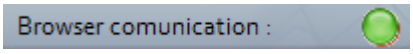
Note: the webrunner can also be opened in a command line:

[NTBRDIR]\EXE\Webrunner www.ip-label.com

The main window opens:



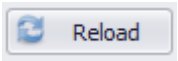
The green icon in the upper right indicates that the ip-label control module is communicating with the browser:



Version information for the 'webrunner' above the green icon indicates that the graphical plug-in is communicating with the ip-label control module:

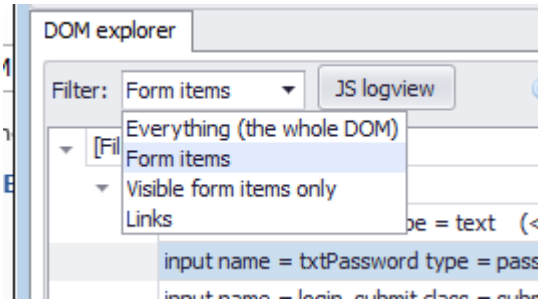
The HttpWatch version is also confirmed (v9.1.13, not available for Chrome).

The 'reload' button resets the communications chain:

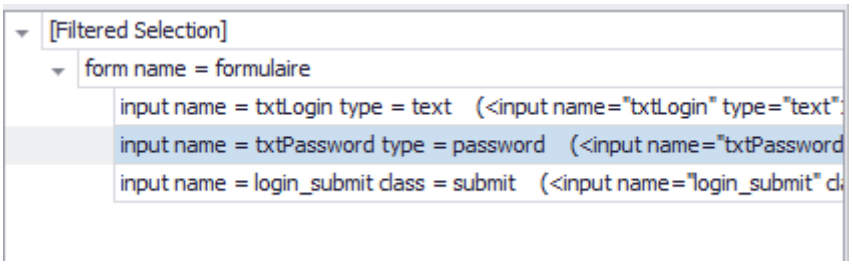


The DOM explorer pane (on the left) is for specifying the HTML item on which an action is to be performed.

It is possible to filter by HTML category:

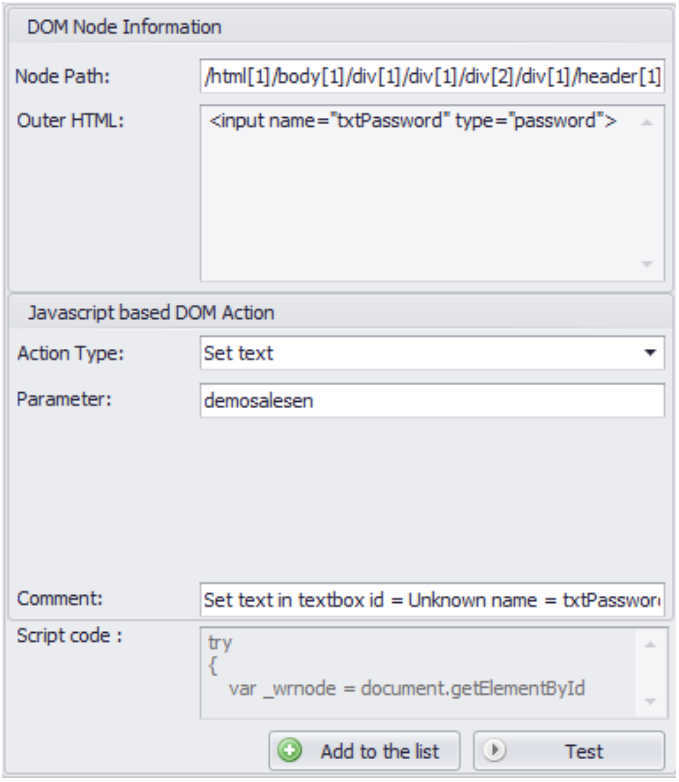


By way of example, for www.ip-label.com the 'Form items' filter selects login and password fields of the HTML authentication form:

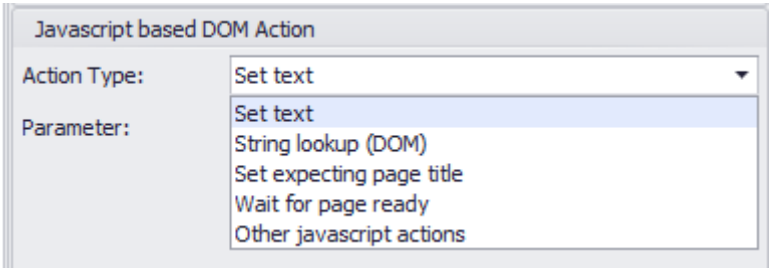


The choice 'Visible form items only' limits the choice to the fields visible to the user.

The center pane displays the XPath of the object selected and allows you to assign an action to it.

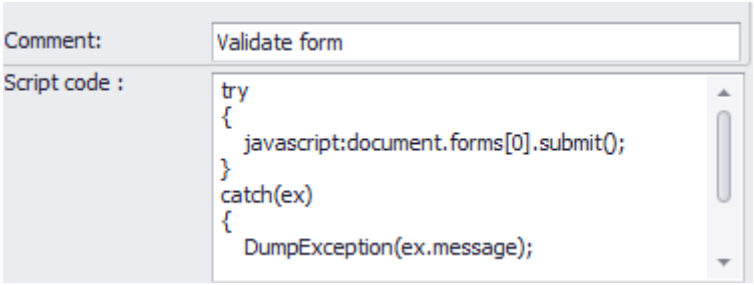


The actions available in the drop-down menu depend on the type of item selected (form field, link, etc.):



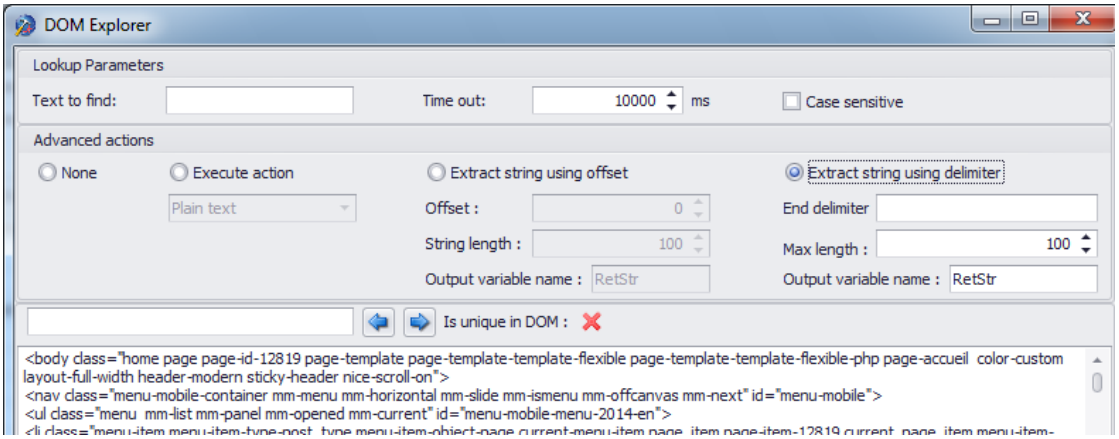
Actions that all DOM objects have in common:

- **String lookup (DOM)**: checks a character string that is to be specified. More than one string can be checked using the operator 'OR' in the following form: string1||string2.
- **Set expected page title**: finds the title of the page (may be partial) and selects the window with which communication is to be established. More than one string can be checked using the operator 'OR' in the following form: string1||string2.
- **Wait for page ready**: waits for the page to be stable in terms of communication between the browser and the ip-label tool (browser in 'ready' state). The function used is WaitForWebPageCommunication(Timeout).
- **Other javascript actions**: inputs custom javascript code. For instance for entry in the first form of the page: document.forms[0].submit(); or to close a pop-up (window.close)



To check for character strings (DOMlookup), a parameter wizard offers a number of search choices:

- Simple search (find a text)
- Search and action (click, etc.)
- Find and extract a string with a start delimiter plus the length of the string to extract (offset)
- Find and extract a string with a start/end delimiter. The maximum length avoids returning a string that is too long in case of an error in the choice of delimiter.



Returning to the main wizard window, the following actions on links are available:

- **Click**: performs a click on the link.
- **Load URL**: opens the URL of the link with a JavaScript document.open() type call

Actions on form fields:

- **Set Text**: inputs text
- **Check this option**: for radio buttons or checkboxes
- **Change Selection**: chooses from a drop-down list
- **Selected**: to select a radio button type of option
- **Submit**: clicks the Submit button

Note: when a page reloads, make sure you always wait until it is fully loaded by using the action 'wait for page ready'.

The actions are then inserted into the script.

Note: to automatically retrieve instructions each time the DOM explorer opens, right-click in the script at the spot located in the DOM block of code that is identified with the tags:

//@WEBRUNNERWIZARD id = {4F94E6E5-2C16-4052-B0CE-E7114E1B7D8A}

...

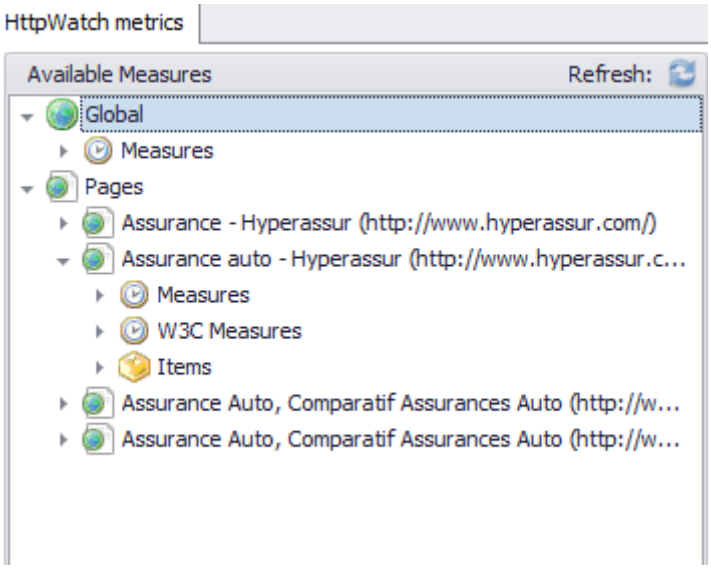
// @

Retrieval is also possible on another workstation following export/import.

Caution: changes made outside of the plug-in are lost when the plug-in reopens.

7.5 EXTRACTING WEB METRICS

In addition to HttpWatch metrics, W3C 'Navigation timing' measures conducted by the browser itself (starting with IE9; see http://caniuse.com/#feat=nav-timing) can be extracted. Display the pop-up menu by right-clicking. Next select 'NewtRIA actions' and then 'Insert measures extractions':



Note: right-click at the end of the script so that all of the steps appear.

The available metrics:

- **Measures**: HttpWatch (depending on your license; see preceding section)
- **W3C Measures**: performance timing measures carried out by the browser in line with W3C standards. For details go to: http://www.w3.org/TR/navigation-timing/

The generated script takes the following form:

```
//@WEBRUNNERWIZARD id = {37FBB7DB-4DC1-404C-9C15-E3D921109338}
// Measures
RET
ExtractMeasureFromXml("$(HTTPW_RESULTS)","webrunner_results/page[0]/W3C/loadEventEnd/text()",
0,"loadEventEnd","tresp")
RET
ExtractMeasureFromXml("$(HTTPW_RESULTS)","webrunner_results/page[0]/TTFB/text()",0,"TTFB","tresp"
)
RET
ExtractMeasureFromXml("$(HTTPW_RESULTS)","webrunner_results/page[0]/PageLoadTime/text()",0,"Pa
geLoadTime","tresp")
//@
```

8 SCENARIOS BASED ON THE EMULATOR (W/API) CLIENT ENVIRONMENT

8.1 INTRODUCTION TO EMULATOR SCENARIOS

This section discusses the specifics of implementing emulator scenarios as a supplement to information given on the general method of creating scenarios, and addresses issues specific to emulator client environments. You must therefore be familiar with the general method (see *Programming scenarios, Developer's Manual vol. 1*) before reading this section.

8.2 TYPOGRAPHICAL CONVENTIONS

[<basic1>] Indicates that the paragraph applies only to applications respecting the condition <basic1>.

<base>\user\actors Represents a disk tree structure path, where:
<base> is the NEWTEST base directory
<year> is the current year (4 digits)
<month> is the current month (2 digits)
<day> is the current day (2 digits)

Thus <base>\result\<year>\<month>\<day> can represent the disk path C:\NEWT50\RESULT\2000\01\22.

8.3 NOTIONS RELEVANT TO EMULATOR CLIENT ENVIRONMENTS

An **emulator** is an application based on a client workstation that simulates the display and commands of a passive terminal connected to one or more mainframe hosts. The part of the **terminal** corresponding to character display (usually 25 lines of 80 columns) is called the **PS (Presentation Space)**.

There are many types of terminals on the market, standard or proprietary. A client environment for the types of terminals listed below is provided with the NEWTEST product (the list is not exhaustive):

TERMINAL TYPE	STANDARD/MANUFACTURER
VT100, VT110	Standard associated with UNIX; character mode
VT220, VT320	DEC
3270, 5250	IBM
VIDEOTEX	French "Minitel" standard
DKU 7107	BULL
PEP, UTS	UNISYS

To provide access to the presentation space of their emulators, and to enable operations to be carried out, some software publishers provide a standardized **programming interface** or **API**. This interface can be used by applications such as NEWTEST for access to functions which cannot be obtained otherwise (move or determine the position of the cursor in the presentation space, obtain keyboard status, etc.).

The principle is that Windows instructions (keyboard input, mouseclicks, etc.) are sent to the emulator's DLL, which converts them into the protocol that enables communication with the host. The host responds in the same protocol. The DLL then converts that the other way to enable the response to be displayed in an 80-column, 25-row window.

Newtest supports two emulator API standards:

- the **UVTI** standard (developed by BULL),
- the **HLLAPI** standard (developed by IBM).

Note that although access with or without an API is completely equivalent in terms of operation and measurement quality, access without an API prevents the use of certain functions (control of cursor position, session status, search for strings in fixed positions, string extractions, error traces).

A list of emulators on the market is given below, showing the APIs they support. This list is indicative only.

NAME	VENDOR	NEWTEST MULTI-PROBE	EMULATIONS	API STANDARDS	NOTES
Access	DIALOGIC		3270, 5250	HLLAPI	32-bit version available
Affinity	BULL		Vtxxx, 7107	UVTI	
Extra	ATTACHMATE	YES	3270, 5250	HLLAPI	Intended for the 3270 32-bit version available
Irma / Select client	ATTACHMATE		3270	HLLAPI	Warning: the title of the emulation window changes at each session.
Ns3270 Ns5250 (Elite)	NETSOFT		3270 5250	HLLAPI	The 5250 emulator connection procedure is difficult to automate
Olicom	OLITEC		Videotex	-	Impossible to recognize character strings (bitmap display)
PathWork	DEC		Vt320	-	
Pathway			Vt320	-	
PC3270 / PCSWS	IBM			HLLAPI	
Reflexion			VTxxx, 3270	-	
Rumba	WALL-DATA		5250	HLLAPI	Recommended for the 5250 32-bit version available
Rumba	WALL-DATA		3270, Vtxxx	HLLAPI	32-bit version available
VTDX	JOG Informatique		Videotex, VTxxx, DKU	UVTI	32-bit version available
Timtel				-	Impossible to recognize character strings (bitmap display)
Wincom	ICOM	YES	Vtxxx, Videotex, 7107	UVTI	Recommended for Videotex, Vtxxx, and DKU. 32-bit version available

All emulators listed above interface with Newtest Probe. Newtest Multi-Probe, however, interfaces only with Extra and Wincom 32-bit emulators.

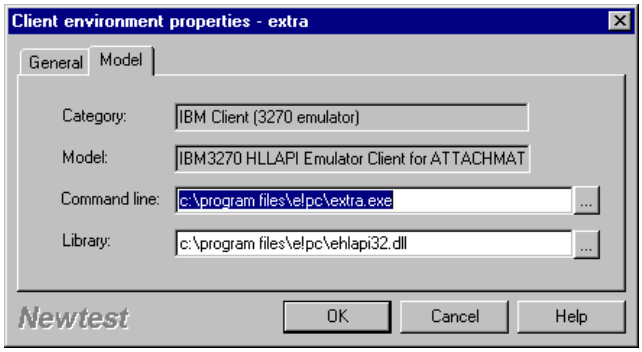
To display a presentation space, it is necessary to open an **emulation session**. The emulation session is generally contained in configuration file that defines it (path to the **mainframe**, display mode, network address, etc.).

Emulations for the 3270 and 5250 have an **OIA Zone,** a special display line (25th/bottom row of the screen) that is not part of the presentation space. This zone contains system information such as keyboard status (which can be blocked), and error information including the state of the connection (IBM codes). This information can be retrieved for processing using the Newtest Basic *GetOIAStr()* function.

8.4 SPECIFICS OF CREATING AN EMULATOR (API) SCENARIO

8.4.1 CONFIGURING AN EMULATOR CLIENT ENVIRONMENT

This section is a supplement to the general method of configuring client environments (see *Programming scenarios, Developer's Manual vol. 1*).



1. In the **Expert Mode** tab page, open the Client environments folder.

2. In the IBM Client (3270 emulator) folder, select the client environment model that corresponds to your emulator. (If your emulator does not appear in the list, see below.) Right-click to access the client environment's Properties.

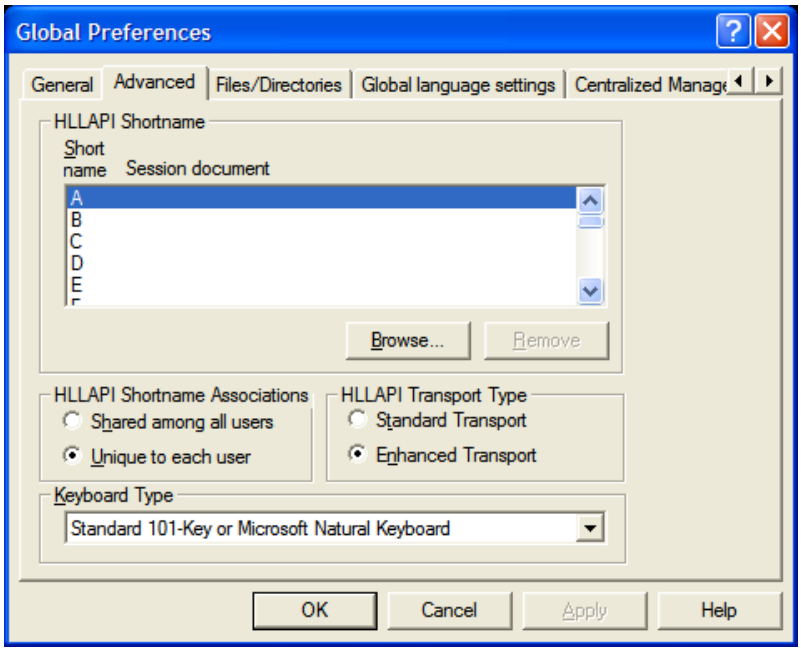
3. In the **Model** tab page, enter the path to the emulator. The Command line is used by the *ActorExec* function to launch the emulator. This field is unavailable if it is not necessary to specify a command line.

4. Enter the full path for the emulator interface (API) library in the **Library** field.

5. Finish configuring the client environment by clicking **OK**.

6. Create a session profile on your emulator.

[HLLAPI] Each session profile must be associated with an upper case letter of the alphabet (A-Z), which is called the **short session name**. If two sessions can be opened at the same time, they must be given distinct short names (in Global Preferences for Extra, or in the API configuration of any other emulator).



8.4.1.1 WHAT IF MY EMULATOR IS NOT ON THE LIST?

Find out whether it has an API. If so, its directory (or a sub-directory) should contain a file with one of the names below:

[UVTI]	UVTI32.DLL
[HLLAPI]	HLLAP32.DLL EHLLAPI32.DLL

After determining what type of API is involved, select either **All makes - HLLAPI** or **All makes - UVTI** from the list of models.

8.4.1.2 WHAT IF MY EMULATOR DOES NOT HAVE AN API?

Use the Windows client environment. The emulator is then treated in the same way as any other application. Refer to *Programming scenarios (Developer's Manual vol. 1)* on how to create scenarios. Nonetheless, the conditions specified given below in *Constraints* do apply.

8.4.2 LEARNING MODE – EMULATOR CLIENT ENVIRONMENT

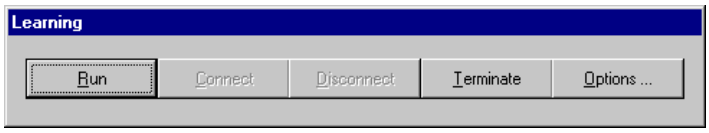
8.4.2.1 CONSTRAINTS

- **Scenarios must take account of occasional variations from the normal behavior of an application**, such as the presence of a page of information. If these variations are not considered, false unavailabilities may arise.
- **A measurement scenario must always leave the application in its initial state**. This constraint is essential to the cyclical nature of monitoring. To apply this rule, you will probably need to have in-depth knowledge of the application and the system.

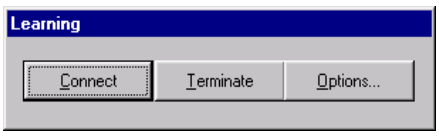
- **Plan a restoration procedure to return the system to its initial state**. For certain systems (particularly IBM), this may require considerable effort. Procedures for a clean exit may vary according to the area in which an error occurs. There is no general solution for this type of problem. Procedures for a return to the initial state may either be included in each script (just after the *Hardcopy()* instruction under the *ERROR* label), or in restoration scenarios, which are detailed in *Editing a scenario* under *Restoration*.

8.4.2.2 SPECIFICS OF THE EMULATOR SCRIPT GENERATION WIZARD

The following dialog box appears at the beginning of the learning phase (see *Script Generation Wizard*): Options differ vary by emulator type:



HLLAPI Emulator

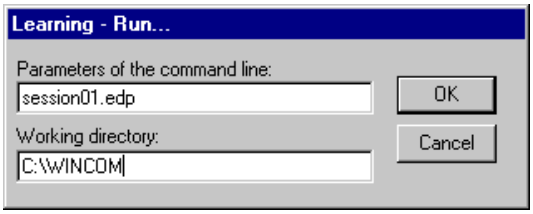


UVTI Emulator

- The **Run** button runs and initializes the emulator to carry out learning.
- The **Connect** button creates a link between NEWTEST and the emulator presentation space.
- The **Disconnect** button breaks the link between NEWTEST and the emulator presentation space.
- The **Terminate** button ends learning mode pass No. 1.
- The **Options** button displays a list of filters for the actions generated by the learning mode.

A. [HLLAPI]

1. Click the Run button.

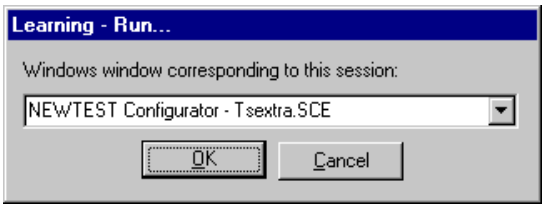


HLLAPI Emulator

2. If required, enter a complement for the command line. This will be concatenated with the command line defined in the client environment to form the command line actually used. You must therefore not forget to add a space at the very beginning of the line.

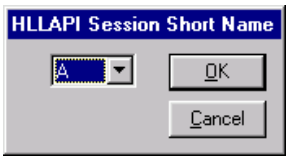
3. Check the working directory for the application.

4. Click OK.



5. Check the title of the Emulator window in the combo box and click OK.

6. Click the Connect button. The following box appears:



7. Enter the short session name associated with the open session. (The default name is generally the right one.)

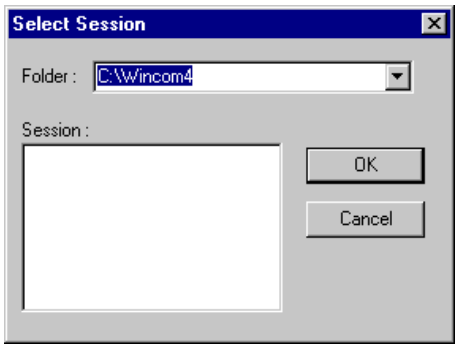
8. Carry out the necessary operations for the definition of the required response time measurements.

9. Click the Disconnect button.

10. Click Terminate. The script is generated automatically.

B. [UVTI]

1. Click the Connect button and the following box appears (it may vary depending on the emulator you are using):



2. Select the session to open in the list. It opens and establishes the physical connection.

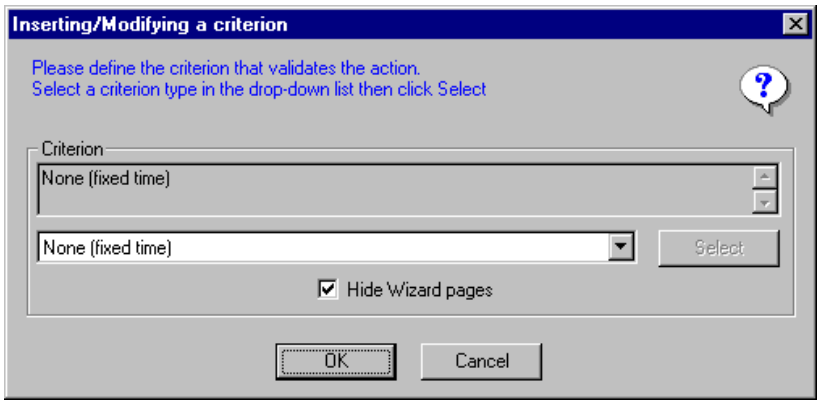
3. Carry out the necessary operations for the definition of response time measurements of your scenario.

4. Click Terminate; the open session closes and the script is generated automatically.

8.4.2.3 CONFIGURING SPECIFIC CRITERIA – EMULATOR (API) CLIENT ENVIRONMENT

PRESENCE OF A STRING

In most cases, the correct running of a command can be identified by a word on the screen. You thus have to use this criterion in this case. The function is *WaitStr*.



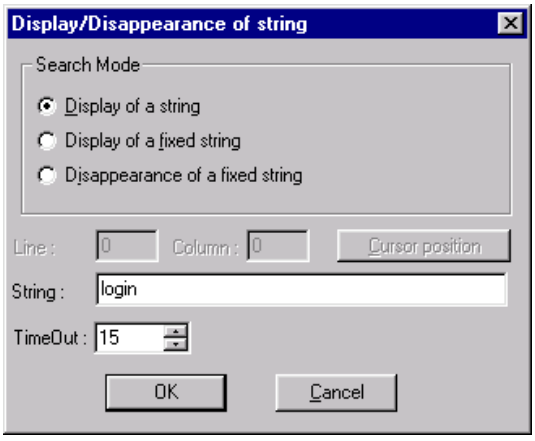
Select **Display/Remove** string in the drop-down list.

Warning: The UVTI standard does not recognize accents or other diacritical marks.

Select strings that are not present on the screen between the action and the arrival of the resulting display, so as to avoid a false synchronization or measurement.

Tip: With asynchronous terminals, search for character strings that appear **towards the bottom of the screen**, to take into account screen scrolling time.

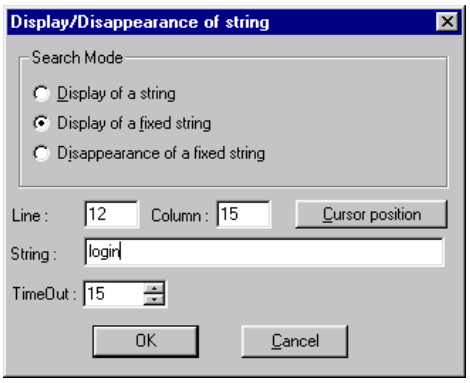
Click the **Select** button to display the next dialog box:



Using this window, indicate whether you want to detect **the presence of a mobile string** (independently of its position in the emulator screen), a **fixed string** (with X,Y coordinates entered in the emulator screen), or the **disappearance of a fixed string**.

PRESENCE/DISAPPEARANCE OF A STRING AT A FIXED POSITION

These criteria can be used when the result of a command is the appearance or disappearance of a word on the screen at a particular position. The word and its position must be specified. Avoid choosing a word that is already at the chosen position before the command is run. The function used is *WaitStr*.



The **Line** and **Column** fields indicate cursor position in the presentation space.

The **Cursor position** button updates the values of the Line and Column fields.

The **String** field indicates the string to be detected.

The recommendations given for the previous criterion (*Presence of a string*) are equally valid here.

8.4.2.4 EMULATOR SCENARIO GENERATED SCRIPTS

In the restoration procedures described above, plan a sequence of keystrokes to unlock the keyboard and a sequence allowing a return to the initial state without leaving a **ghost connection** in the system or application, e.g. a CICS connection.

A possible (radical) solution is to include commands in the script for **Disconnection** followed by **Connection** of your emulator. Unfortunately, this solution does not always avoid phantom connections, but it at least enables the session and communication layers to be reinitialized.

Note that disconnection of the API (*Disconnect*) must be placed before the sequence of physical disconnection by menu (*ClickMenu*) of your emulator.

Simple example of error recovery within a scenario:

```
ERROR:                          // error label generated
Hardcopy()                      // Screen shot
SendKey("{ESC}")                // Unlock keyboard
Wait(2)                         /// Key sequence for a blind exit
Sendkey("{F3}")                 //
Wait(5)                         //
Sendkey("{F3}")                 //
Wait(5)                         //
Sendkey("{F12}")                //
Wait(5)                         //
Disconnect()                    // HLLAPI disconnection (obligatory)
WaitWnd                         // search for a window
(0, "EXTRA! A - Session1", TimeOut, Window)

ClickMenu("Connection# Deactivate ")    // Physical disconnection by menu
Wait(5)                         // Arbitrary wait
ClickMenu("Connection#Activate")        // Physical reconnection by menu
END                             // End
```

Notes: Never remove the timeout instruction (*wait(5)*) generated at the end of the script. The delay can be reduced, but never below 2 seconds.

For emulators with API, **keep the** *Disconnect()* **functions** generated at the end of the script, before and after the *ERROR* label.

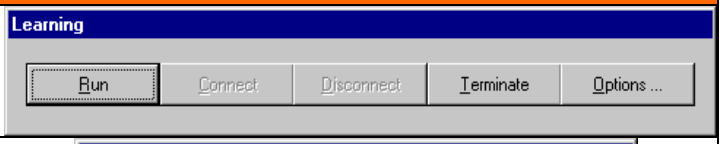
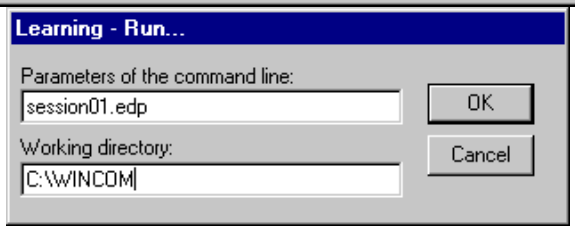
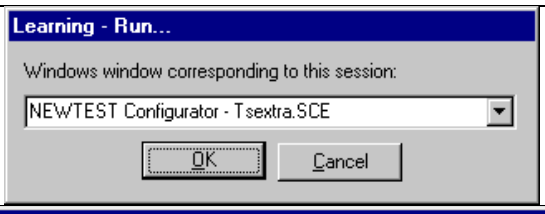
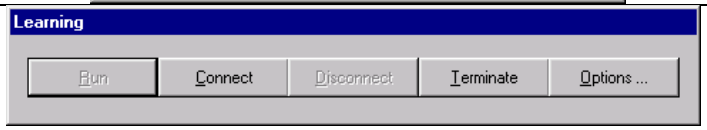
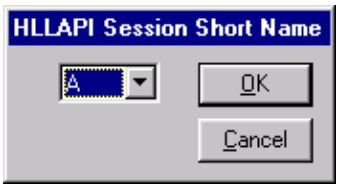
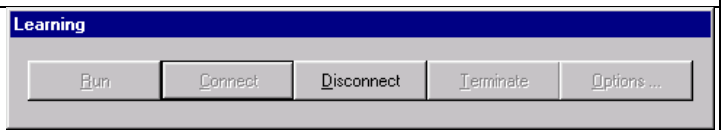
Beware of windows that change their name during execution (Irma emulator, for example)**.** Check that they always have the same name as during learning mode. If not, modify the *WaitWnd* instruction so as to use only the fixed part of the title.

8.4.3 AVAILABLE FUNCTIONS

FUNCTION	ENVIRONMENT(S)	DESCRIPTION
ActorExec	HLLAPI, UVTI	Initializes the client environment and launches the associated executable.
Connect	HLLAPI, UVTI	Connects to the presentation space (PS) of an emulator
DisConnect	HLLAPI, UVTI	Disconnects from the presentation space (PS) of an emulator
GetCursorPS	HLLAPI, UVTI	Returns the current position of the cursor in the presentation space (PS) of an emulator.
GetOIAStr	HLLAPI	Retrieves the emulator OIA zone (HLLAPI standard only).
GetStr	HLLAPI, UVTI	Retrieves a character string from a specified position in the emulator presentation space (PS).
ReceiveFile	UVTI	Receives a file transferred from a central system to the local workstation.
SearchStr	HLLAPI, UVTI	Locates a specified character string in the emulator presentation space (PS).
SendFile	UVTI	Transfers a file from the local workstation to a central system.
SendKey	HLLAPI, UVTI	Simulates a keystroke by a user.
SetCursorPS	HLLAPI, UVTI	Positions the cursor in the emulator presentation space.
WaitStr	HLLAPI, UVTI	Awaits the appearance (or disappearance) of a specified character string in the emulator presentation space. Searches the whole screen or a specific position (row/column).

8.5 EXAMPLE: CREATING A HLLAPI EMULATOR SCENARIO

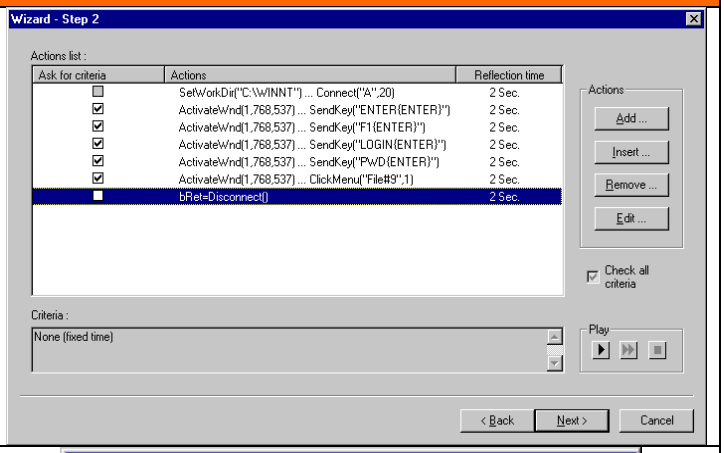
Assuming that the client environment has been configured (configuration of sessions and session-HLLAPI short name association), the example below illustrates all actions required to create an emulator scenario. The scenario opens a session and verifies that the session is correctly open.

Wizard step 1: Learning	
Click the **Run** button.	
Enter the name of the session profile file in the Parameters of the command line field (remember to enter a space at the beginning). Check that the working folder is correct. Click **OK**.	
Use the drop-down list to search for the window name **containing** the emulator session. Click OK.	
[HLLAPI] The session opens. Click the Connect button.	
[HLLAPI] Enter the **short session name** in upper case letters. **Two sessions with the same short name** cannot be opened simultaneously. Example: "A", "B", etc. [UVTI] Enter the name of the configuration file associated with the desired session. Example: c1.vtx, sess0.dku, pad.220 or other Click OK.	
Click the **Disconnect** button Click **Terminate**.	

Wizard step 2: defining criteria

Go to step 2 of *script generation wizard*. This dialog box contains the list of actions recorded in step 1. Check that all your actions are listed.

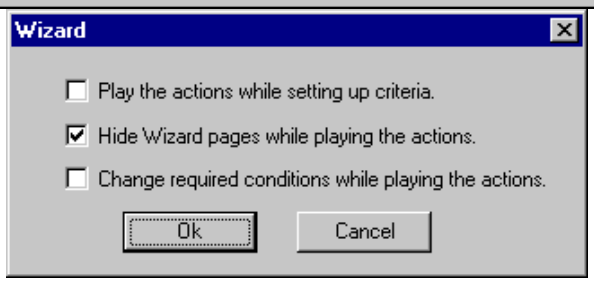
Go to the next step by pressing Next >.



If you choose not to play back the actions, NEWTEST proceeds to the step in which you to choose criteria (useful when you made a mistake concerning one or more criteria).

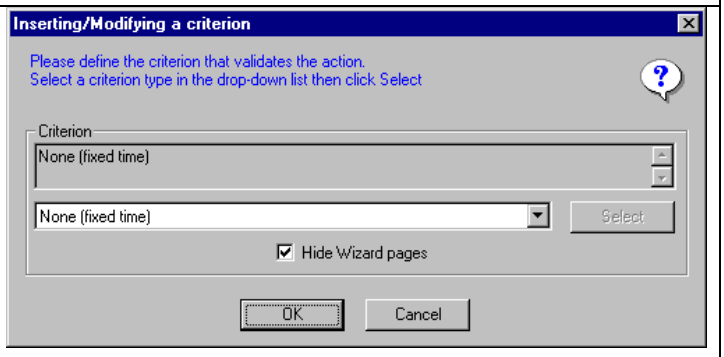
You can hide the wizard page while defining criteria so that it does not mask the application.

The last option, when available, allows you to select the current object by one of its attributes, for example when the name of the object is variable.



The choice of criteria dialog box appears whenever an action is played. In the case of emulators, choose Display/Remove String and select a string that was not present in the emulator BEFORE the current action.

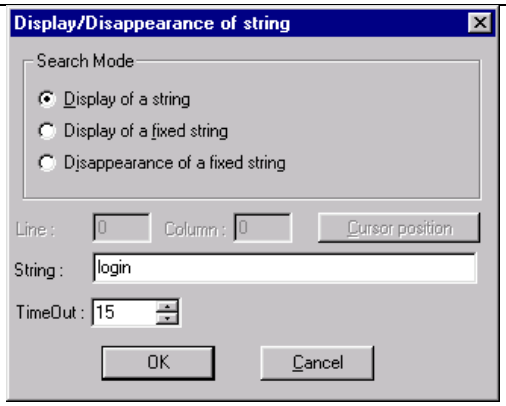
Whenever you want, you can call up or mask the wizard to identify the current action and select the correct criterion as a function of the latter.

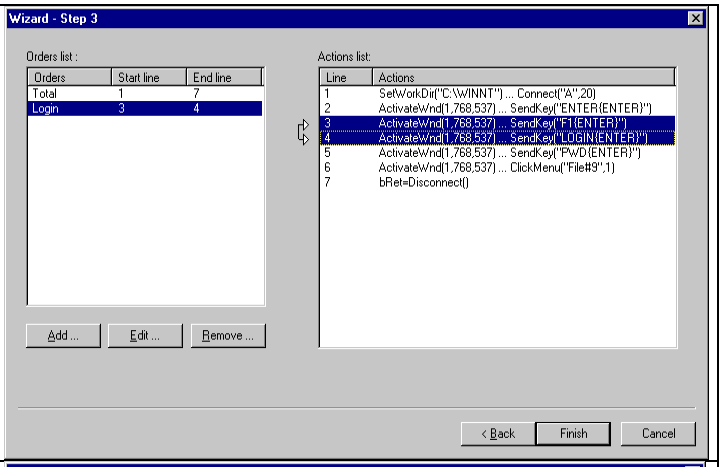
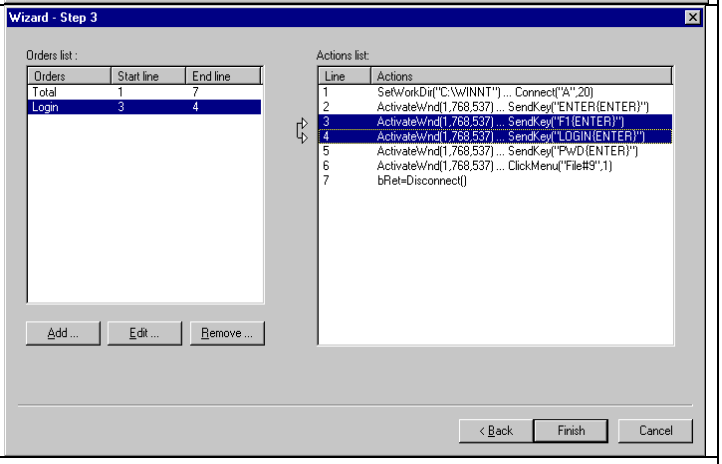


When you click Select, a box appears to enable you to select a string. If you select a fixed string, place the cursor at the beginning of the string in the presentation space and click the Cursor position button recover its coordinates.

Do not forget to reset the cursor to its initial position before continuing the operations.

The TimeOut text field is used to specify the timeout for the current criterion in the script (in seconds).



After selecting all criteria, step 3 of measurement configuration appears. Enter the measurements that will be done by this scenario. Click Finish when the operation is over.	
Test your script step by step or in one pass (see *Editing scenario script*). In case of error (e.g., incorrect criterion), the script can be modified directly using the full-page editor. Recompile and test again until the script is error-free.	
Add a procedure for exit in case of error at the end of the script (or prepare a restoration scenario).	

8.5.1 EMULATOR SCENARIO GENERATED SCRIPT

```
// ————————————————————————————
//    Ibm3270.SCE - 04/10/2001 15:33:20
// ————————————————————————————
DIM Total AS ORDER(TRESP)
DIM Login AS ORDER(TRESP)
DIM TimeOut AS INT
DIM bRet AS BOOL
TimeOut = 15
// Emulator Connection
//————————————
TopChrono (Total)
SetWorkDir("D:\PRD_GEN\EPC")
ActorExec(" session01.edp", "SESSION01 - EXTRA! Personal Client")
Connect("A",20)
IF WaitStr(0,"adapts",0,0,TimeOut)=FALSE THEN
        RET ErrorLine("Err001")
        GOTO ERROR
ENDIF
Wait(2)
```

```
// Sub-System Selection
//————————————
IF WaitWnd(":t:SESSION01 - EXTRA! Personal Client",TimeOut)=FALSE THEN
        RET ErrorLine("Err002")
        GOTO ERROR
ENDIF
ActivateWnd(1,616,462)
SendKey("b{ENTER}")
IF WaitStr(0,"ENTER",0,0,TimeOut)=FALSE THEN
        RET ErrorLine("Err002")
        GOTO ERROR
ENDIF
Wait(2)
//Login Sub-System
//————————
TopChrono (Login)
IF WaitWnd(":t:SESSION01 - EXTRA! Personal Client",TimeOut)=FALSE THEN
        RET ErrorLine("Err003")
        GOTO ERROR
ENDIF
ActivateWnd(1,616,462)
SendKey("{ENTER}")
IF WaitStr(0,"Password",0,0,TimeOut)=FALSE THEN
        RET ErrorLine("Err003")
        GOTO ERROR
ENDIF
Wait(2)
IF WaitWnd(":t:SESSION01 - EXTRA! Personal Client",TimeOut)=FALSE THEN
        RET ErrorLine("Err004")
        GOTO ERROR
ENDIF
ActivateWnd(1,616,462)
SendKey("auditec{TAB}paucitec{ENTER}")
IF WaitStr(0,"Company",0,0,TimeOut)=FALSE THEN
        RET ErrorLine("Err004")
        GOTO ERROR
ENDIF
RET LogTps(Login)
Wait(2)
//Sub-System exit
//————————
IF WaitWnd(":t:SESSION01 - EXTRA! Personal Client",TimeOut)=FALSE THEN
        RET ErrorLine("Err005")
        GOTO ERROR
ENDIF
ActivateWnd(1,616,462)
SendKey("{F3}")
IF WaitStr(0,"logoff",0,0,TimeOut)=FALSE THEN
        RET ErrorLine("Err005")
        GOTO ERROR
ENDIF
Wait(2)
IF WaitWnd(":t:SESSION01 - EXTRA! Personal Client",TimeOut)=FALSE THEN
        RET ErrorLine("Err006")
        GOTO ERROR
ENDIF
ActivateWnd(1,616,462)
```

```
SendKey("logoff{ENTER}")
IF WaitStr(0,"navigate",0,0,TimeOut)=FALSE THEN
        RET ErrorLine("Err006")
        GOTO ERROR
ENDIF
Wait(2)
bRet=Disconnect()
IF bRet=FALSE THEN
                RET ErrorLine("Err007")
        GOTO ERROR
ENDIF
RET LogTps(Total)
Wait(2)
Wait(5)
END
ERROR:
HardCopy()
Disconnect()
```

8.6 CREATING GROUPS FOR EMULATOR SCENARIOS

8.6.1 OPTION 1: CONNECTION/DISCONNECTION FOR EACH SCENARIO

In this classic plan, a connection is made at the start of each scenario, and disconnected at the end. This is simple and makes the scenarios completely independent of each other (provided that the quitting functions in case of error are properly designed, needless to say).

Scenario 1	**Scenario 2 …**	**Scenario n**
Open session	Open session	Open session
.	.	.
.	.	.
.	.	.
Close session	Close session	Close session

8.6.2 OPTION 2: CONNECTION/DISCONNECTION AT THE START AND END OF THE DAY

This has the advantage of being more like the behavior of a real user who would connect at the beginning of the day and disconnect at the end. It has the disadvantage of being more difficult to implement.

Two service scenarios must be created, one to open all sessions at the start of the day, the other to close them at the end.

CONNEX Scenario (8:00 service)

APP 1 Scenario (user 08.10 – 18.00 every 10 minutes)

APP 2 Scenario (user 08.10 – 18.00 every 10 minutes)

APP 3 Scenario (user 08.10 – 18.00 every 10 minutes)

DECONNEX Scenario (18:10 service)

This scenario is to be used with some precautions:

- For the Connection Scenario (CONNEX), check **start if time exceeded** in the **Sequence** group of options. In case of an unplanned restart (security **reboot**, power failure, etc.), this causes the Connection Scenario to run even if its run time (08:00) has elapsed.
- In USER Scenarios, do not remove the *ExecEmul* and *Connect* instructions. They are obligatory (because of links with the emulators and API), although they have no visible effect.
- Pay close attention to use of correct algorithms for exits in case of error (restoration scenario).

8.7 FREQUENTLY ASKED QUESTIONS

A. During the creation of the scenario, the **emulator does not start**

- See if you can start it without Newtest.
- Make sure you have chosen the correct client environment model.
- Verify that the requirements for Configuration of the API-Emulator links have been met.
- Check that a space has been inserted before the name of the session file.
- Consult the daily error log file (NEWTHLL.TRC or NEWTUVTI.TRC, depending on the type of client environment).

These files are in text format and are stored in the directory <base>\Result\<year>\<month>\<day>, e.g. C:\NEWTEST\RESULT\2001\01\22.

B. The emulator **does not start or does not connect** during the development phase

Close it and then close the configuration module (a faulty manipulation may have occurred).

If this doesn't solve the problem, shut down Windows and then reboot.

Warnings:

[UVTI] Never stop the emulator (application) during the script development phase (step by step running), or between the two phases of learning (definition of actions and selection of criteria). Stop it only when you have progressed from the construction phase (Configurator) to the operating phase (Executor).

[UVTI] Never close / open an emulator session manually (*File/Close*) during one of the creation or development phases (if you do, it can no longer be opened by the script). Always go to the end of the script so that the session can be correctly closed by the Disconnect() instruction.

C. Certain key actions are replayed without effect

[UVTI] Certain keystrokes may have no effect when replayed by the script (because of the keyboard management of the emulator). Reconfigure the emulator keyboard (see the emulator documentation) to associate the desired function with another key, e.g. a function key.

[HLLAPI] The IBM keyboard contains **special keys** (Clear, ATTN, PFxx, etc.). The emulator carries out a translation between the PC keyboard and these special keys. A table allows the API to carry out the same translation. The NEWTEST key codes (see *SendKey* function) are correlated with the HLLAPI mnemonics; NEWTEST is supplied with a preconfigured table (section [KeyMap] in the file *<base>\user\actors\hll.cf*).

If you wish to add to or modify these codes, the table below gives an exhaustive list of equivalences. The syntax for entries in table **hll.cf** is *<NEWTEST Key code corresponding to the PC keyboardPC>=< IBM mnemonic >.*

Save a copy of the table below before making changes!

HLLAPI MNEMONICS	IBM SPECIAL TERMINAL KEYS
@B	Backtab
@C	Clear
@D	Delete
@E	Enter
@F	Erase EOF
@I	Insert
@J	Jump
@L	Cursor Left
@N	New Line
@P	Print
@R	Reset
@T	Tab
@U	Cursor Up
@V	Cursor Down
@Y	Caps Lock
@Z	Cursor right
@s	Scroll Lock
@t	Num Lock
@@	@ symbol
@$	Alternate Cursor
@<	Backspace Erase
@0	Home
@1 - @9	PF1 - PF9

@a - @o	PF10 - PF24
@x - @z	PA1 - PA3
@A@F	Erase Input
@A@H	System Request
@A@J	Cursor select
@A@[Q10.]	Attn (Attention)
@A@T	Print screen
@A@b	Underscore
@A@y	Forward word tab
@A@z	Backward Word Tab
@S@x	Dup
@S@y	Field Mark

9 SCENARIOS BUILT WITH THE VOICE CLIENT ENVIRONMENT

9.1 INTRODUCTION

The proposed solution consists of simulating the consultation of a voice application that is accessible from a standard telephone.

A “voice” scenario is defined as a series of actions that enable the emission of DTMF frequencies (telephone touch-tone frequencies), the transmission of phrases synthesized from a text (depending on the option selected), and the recognition of phrases or DTMF signals generated on the monitored voice server.

The solution functions with telephony devices (voice boards, modem, etc.) that support audio stream, and whose drivers are TAPI-compatible with support for TSP (Telephony Service Provider) and MSP (Media Service Provider) functionalities. The following devices have been validated:

- **Analog**: analog Intel-Dialogic board (ref. D/41JCT-LS) (Windows 2000 PRO only)
- **ISDN**: BRI / 4 BRI Diva Server board from Dialogic with lastest generation drivers (Windows 2000 PRO only)

Other devices may have operational limits depending on the extent of their drivers’ TAPI compatibility.

The solution can also operate with QX2000/100-4L voiceboards from NMS EUROPE S.A. (PCI bus) with one of more dedicated analog telephone lines (Windows 2000 PRO and Windows XP).

IMPORTANT: retrieval of scenarios to version 7.2.0 R0 of the Voice Client Environment

Voice Client Environment scenarios (speech recognition) developed in previous Newtest versions are not compatible with the client environment in Newtest 7.2.1 R0 and later. You should expect a compilation error for such scenarios when Newtest is upgraded. You will need to fully regenerate these scenarios with the script generation wizard.

9.2 INSTALLING THE QX2000 VOICE BOARD FROM NMS

When the voiceboard has been installed in the computer and connected to the telephone network, install the software by carrying out the following steps:

- Open a local administrator session.
- If the window for Windows recognition of the voiceboard appears, select “Cancel”. Unselect the card “unknown device” (NT++) or “Multimedia controller” (XP) in the Windows control panel for multimedia devices so that the system does not re-detect the card at startup.
- Launch the program Setup NMS 7100-77073 located in the \Tools\QX2000\ directory of the Newtest CD. Specify the directory where you want to uncompress the contents.
- Launch setup.exe from the temporary uncompress directory.
- Select: Install product / NMS Product.
- Choose only QX.
- Installation directory (keep the default “NMS”).
- Select the country.
- Installation “Full”.
- Allow the setup program to modify the environment variables (Modify settings).
- Open a command line (CMD), and go to the directory NMS\QX\LOAD.
- Launch “QXLOAD” with the parameter “–CFG”: QXLOAD –CFG.
- When the execution is complete, restart the computer.

IMPORTANT: Starting with Newtest 7.3 R2, ensure the Voice client environment’s compatibility with the QX2000 voice board by editing the Windows registry to *delete the following entry*:

HKEY_CURRENT_USER\Software\NEWTEST\NewtVocal\TTSBufferSize

9.3 INSTALLING TAPI-COMPATIBLE TELEPHONE DEVICES

For the other telephony devices corresponding to the specifications laid down in section I above, please follow the manufacturer’s installation instructions.

When you have a choice of control interfaces for the devices, always select TAPI (example : for Dialogic Diva Server cards, launch the configuration module “Diva Server Configuration Manager” and then, in the “Service” part, replace the interface CAPI with TAPI).

In addition, make sure that the latest drivers supplied by the manufacturer are installed.

9.3.1 CONFIGURING OPERATION WITH AN INTEL DIALOGIC BOARD

For D/41JCT-LS voice boards from Intel-Dialogic, some settings of the WAV drivers installed with the Dialogic application must be (re)configured correctly.

First install Intel Dialogic System Software with TAPI options. Refer to documentation relevant to the current valid version supplied by Intel Dialogic. Install the latest service packs available with the current version of the drivers.

Do not forget to set Dialogic services to start up in automatic mode using the Dialogic Configuration Manager (DCM). It is also vital to install the *Dialogic TAPI* drivers that are available in *Program Files\DIALOGIC\LIB* (*Oemsetupfile*) following installation of the software.

In the Windows *Control Panel* go to *Sounds and Audio Devices*. Select the *Hardware* tab, and then the row ***Dialogic WAVE driver 1.X.*** Click *Properties* and then select the *Properties* tab. Scroll down in the list and select ***Dialogic WAVE driver 1.X***. Click *Properties*. In the configuration dialog box that opens, assign the value 8192 to *Receive Buffer Threshold*. Click [OK] and then allow the computer to restart.

9.3.2 CONFIGURING OPERATION WITH A DIALOGIC BRI OR 4BRI BOARD

This section deals with the boards previously known as EICON BRI and EICON 4BRI. First, install DIVA SERVER for Windows with TAPI options. Refer to documentation relevant to the current valid version supplied by Dialogic. Install the latest service packs available with the current version of the drivers.

Use the program “Diva Server Configuration Manager” to enter settings for your ISDN interface. Delete the CAPI interface from the module and add the TAPI interface, if this was not done when the software was installed. Next, configure a “link” between the TAPI module and the Diva Server board in the interface of the configuration module. Refer to the documentation specific to Diva Server BRI or 4BRI for appropriate procedures. Do not forget to activate your changes before leaving the configuration module.

Check your ISDN line after installing and configuring the software using the program “Check ISDN line” in the folder “Diva Server for Windows” in Start menu/Programs.

9.4 TECHNICAL PRINCIPLES OF THE SOLUTION

The voice client environment simulates the actions of a user in three ways:

- Sends DTMF frequencies that correspond to the 12 keys of a telephone keypad ([0-9], # and *).
- Reads out words or phrases by voice synthesis (numeric simulation of a human voice). This feature makes it possible to test voice recognition servers or, in other words, servers that react to voice commands. This is an optional function of the client environment.
- Reads pre-recorded sound files (.WAV) “online” to simulate voice commands. This feature is accessible using the Newtest Basic advanced functions dedicated to this client environment.

The voice client environment reacts as follows to events while the scenario runs:

- It detects one or more words: a voice recognition engine allows Newtest to wait for a set amount of time (until the criterion’s timeout) for the word(s) that serve as the criterion for this step.
- It detects DTMF frequencies issued by the remote voice server.

The script generation wizard allows you to build scenarios offline (without being connected to the voice server). It is therefore advisable to prepare the scenario script by writing down on a piece of paper all the steps to simulate prior to launching the scenario. In particular, it is important to know, for each step, the exact words or phrases expected, and the DTMF frequencies or synthesized words to send to the voice server.

9.5 RECOMMENDATIONS FOR CHOOSING CRITERIA

During this phase of scenario script preparation, it is important to choose pertinent criteria for monitoring the voice service.

If the criteria selected are too complex or vary over time, Newtest will detect undue unavailabilities. On the other hand, if the criteria are too simple (short words or words pronounced systematically regardless of the voice site's message), Newtest may not be able to determine malfunctions of the voice server.

The best results are obtained by using:

- Phrases rather than single words (a phrase is a more discriminating criterion)
- Parts of sentences that are unquestionably invariable (take care, for instance, not to use words that vary by date or time of day, etc.).
- To the extent possible, phrases should be pronounced without background music, in order to limit the difficulty of recognizing words in a complex sound environment.

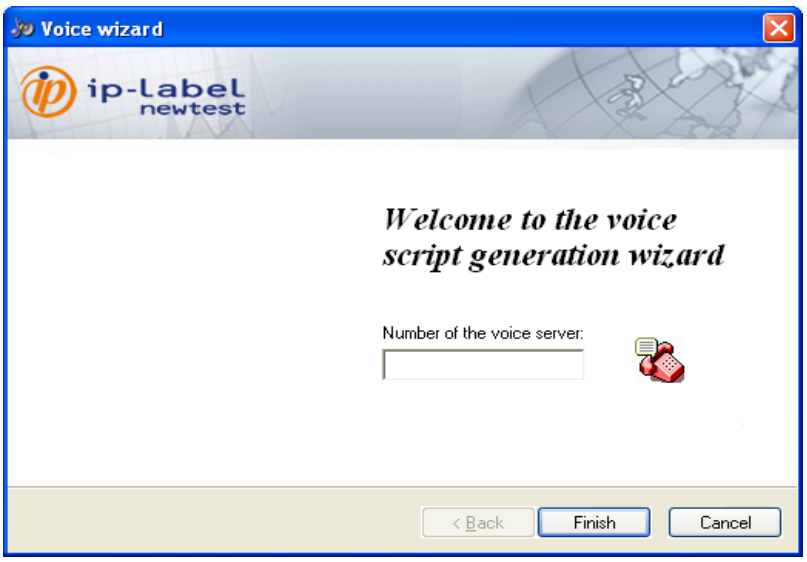
The voice client environment does, however, offer advanced settings to adjust the recognition mode to the complexity of the environment to manage on the voice server. These aspects are treated in the section about the wizard.

9.6 USING THE SCRIPT GENERATION WIZARD

9.6.1 LAUNCHING THE WIZARD AND FIRST STEPS

When the wizard opens on an existing script, it offers to ***Modify the script*** or to ***Regenerate***. Regenerating causes the previous script to be overwritten. If you choose to modify the script, the wizard immediately displays a summary table of the steps of the scenario (as at the end of learning).

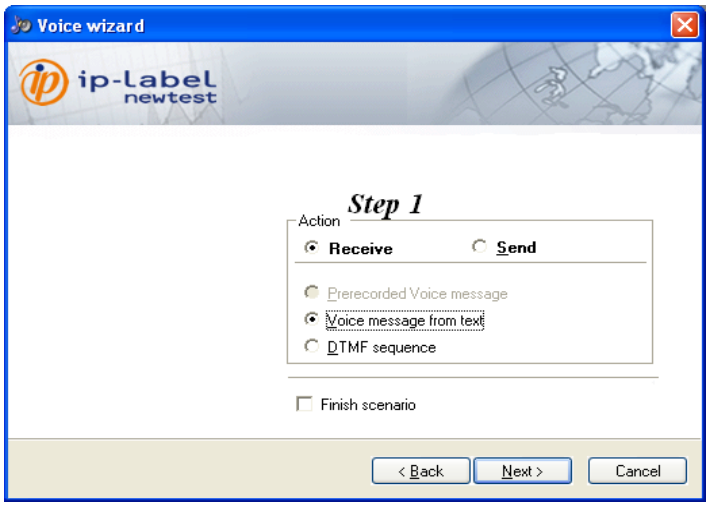
The first step in creating a scenario consists of entering the telephone number to dial:



Type this number into the field ***Number of the voice server***. Remember to precede it with the number that you need to dial for an outside line, as required by your particular autoswitch. If you need to insert a pause during dialing, use a comma (,). This corresponds to a more or less short pause depending on the parameters of your telephone equipment (default setting: 2 seconds). A dot or full stop (.) corresponds to a long pause (default: 5 seconds).

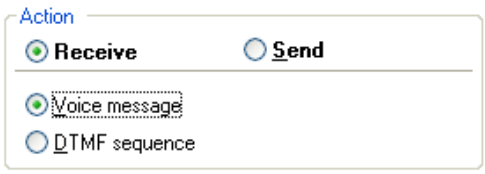
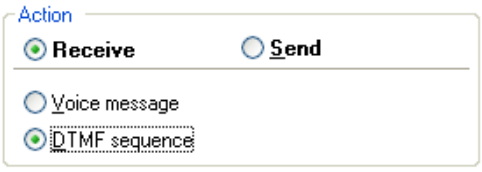
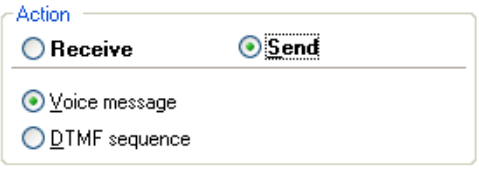
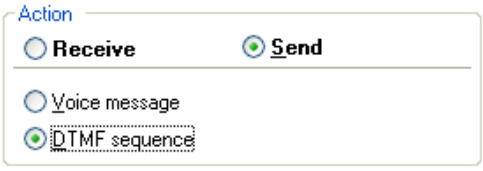
This number will not be dialed immediately, since the scenario script is being built offline.

Click **Next** to proceed to the following step:



Each step of the scenario starts out with this dialog box, which allows you to select the action to perform. To choose your action, click the appropriate radio buttons in the **Action** zone (4 combinations possible).

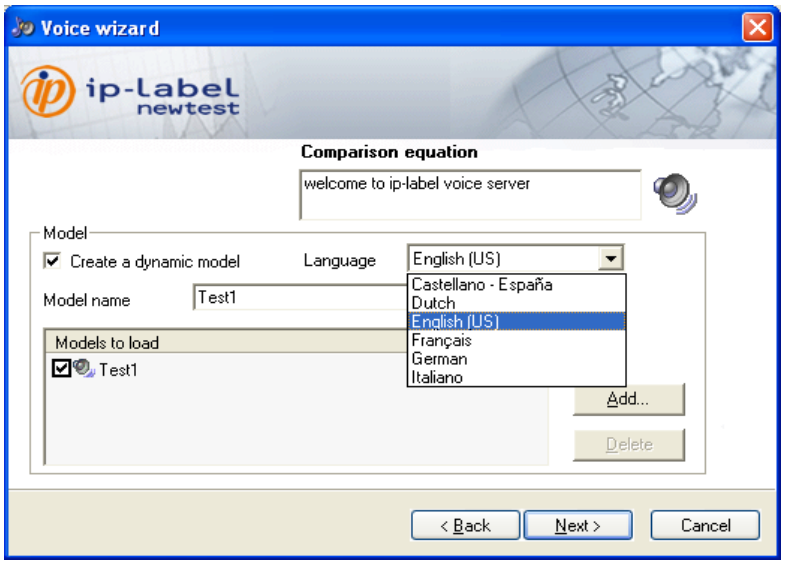
You can choose from 4 Newtest actions depending on the options allowed by the installed environment. The following table summarizes these 4 possibilities, and indicates whether it is a **criterion** (that validates an event) or a **user action** (that simulates an action of a voice service user).

Reception of a voice message	**Criterion**	
Reception of a DTMF frequency **(QX2000 board only)**	**Criterion**	
Sending of a voice message (depending on the options included in your client environment)	**User action**	
Sending of a DTMF frequency	**User action**	

After selecting the options, click **Next** to view the properties panel that corresponds to your choice.

9.6.2 RECEPTION OF A VOICE MESSAGE

This choice always corresponds to a criterion: Newtest will await the expected utterance for a set length of time. If Newtest does not hear it, an unavailability will be generated for this step of the scenario. Otherwise, Newtest will proceed to the next action.



In the field under **Comparison equation**, enter the password or phrase that Newtest is supposed to detect. Type the text without punctuation or special characters. You may use all the letters of the alphabet (including accented letters) and numbers from 0 to 9.

A complete phrase forms a WORD in the GRAMMAR of the MODEL used for the scenario. Thus, if you enter “Welcome to our voice site”, the criterion will be this entire sentence, and in no case only a part of it (such as “Welcome” or “voice site” which, taken alone, will not validate this criterion).

IMPORTANT ADVICE: A “silence” has often been noted on the voice boards used. This is a lapse of time of up to a second during which the device remains “deaf” to the speech flow sent by the server. This is technically normal and results from the load of the devices that enter into the capture of voice data, particularly on servers that allow for no delay between the user action and the sending of the following phrase. This phenomenon may be noted during scenario test runs with the speakers on. If you notice a silence in some steps at the beginning of the phrase pronounced by the voice server, set your criteria accordingly: Newtest hears the same thing you do when audio output is enabled.

There are 2 special characters that allow you to combine the criteria in a single step:

“ | “ stands for OR

ex: ”**hello|welcome to our voice site**“ means the criterion will be validated if Newtest hears “**hello** “ OR “**welcome to our voice site**”.

” & “ stands for AND

ex: ”**hello&welcome to our voice site**“ means the criterion will be validated if Newtest hears “**hello** “ AND “**welcome to our voice site**”.

CAUTION: the use of special characters (punctuation, symbols, etc.) other than the ones shown here in the comparison equation may result in a GRAMMAR generation error at the end of the wizard.

Language: in the drop-down list (languages installed with the Newtest product), select the speech recognition language to apply. When you have selected the language, all the other steps in the scenario’s voice recognition will use the same language (no subsequent selection is offered). You cannot perform speech recognition using more than one language per scenario.

The option **Create a dynamic model** in the wizard’s **Model** area is selected by default. It is advisable to keep this option activated in order to indicate to Newtest that it will have to build a list of words for voice recognition with respect to the **Comparison equation** that you enter for the scenario.

If you clear (deactivate) this option, you will have to load into the current scenario an existing model that includes the entirety of your strictly identical comparison equations (absolutely identical criteria syntax).

The **Models to load** area is for listing all the GRAMMAR models to use for the current scenario. Newtest’s default setting is to load, during the execution of the script, the model that it offers to create dynamically. If, however, you wish to load one or more existing models, you can use the buttons next to this area.

ADD: allows you to select a .JSG file in order to add an existing model to the scenario.

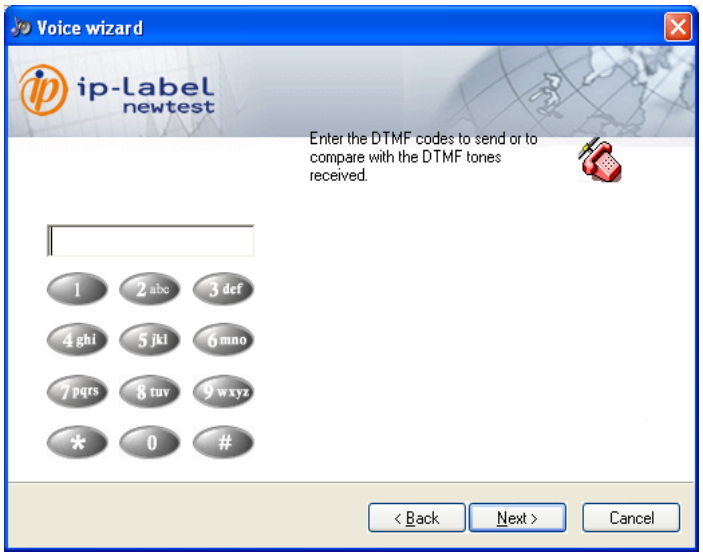
DELETE: to delete the selected model from the list.

NOTE: It is rarely necessary to modify the settings in the Grammar area at this point in the wizard. It is highly advisable to leave these options with their default values.

9.6.3 RECEPTION OF A DTMF FREQUENCY

This choice always corresponds to a criterion. This one makes Newtest wait until it receives (hears) a DTMF tone or series of tones.

It is rare that this type of signal is used to communicate with users of voice servers. It is possible, however, that this option is installed to allow simplified and automated navigation for running tests on pre-production or development servers. Newtest can in such cases identify the DTMF tones of the 12 keys of the telephone keypad: **[0-9], *, #**.



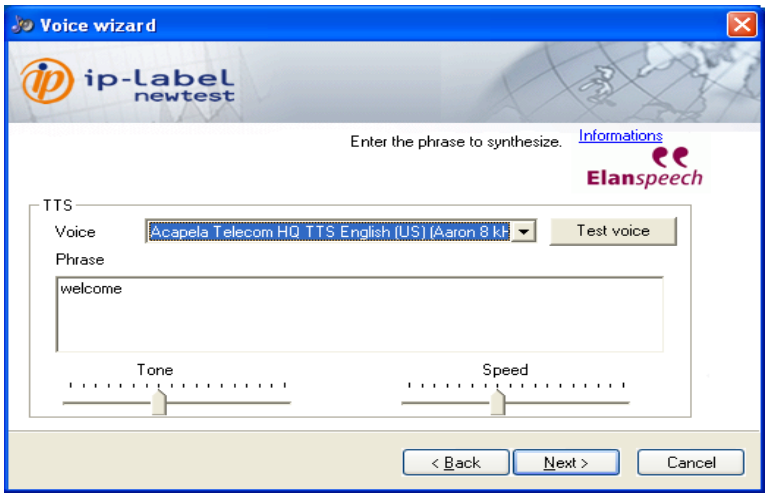
Using the virtual telephone keypad in the dialogue box or your keyboard, enter the DTMF tone or sequence of tones to detect.

Note: if you enter a series of tones, the criterion will not be validated until the tones are received in the order they were input. A single part of the DTMF tone series is not enough to validate the criterion

9.6.4 SENDING OF A VOICE MESSAGE

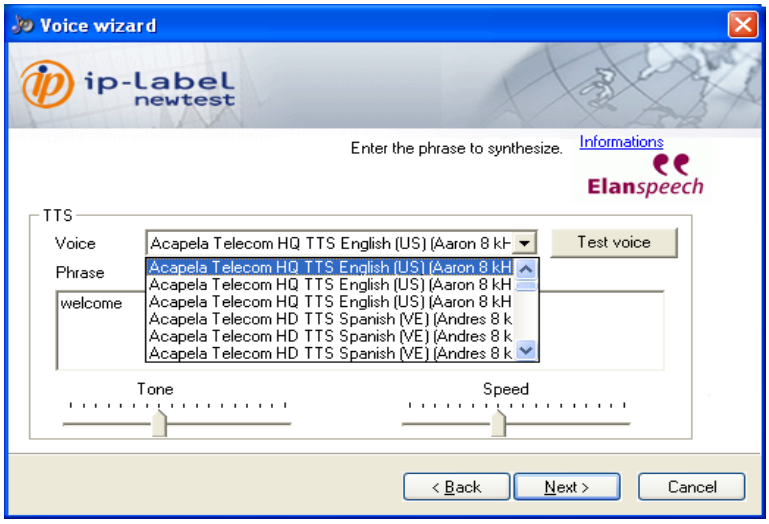
This function is available only if you purchased the option ELAN Text-to-Speech (TTS) with your Newtest voice client environment.

This is a user action. With a synthesized voice, Newtest can simulate the speech of a user. This option is very useful for testing voice servers with speech recognition.



In the ***Voice*** drop-down list, choose the type of synthesized voice to use. This choice is important because it determines the voice's sonority (male or female) and the way of pronouncing the text (the spoken language).

Note: it is important to choose the voice format which corresponds to the type of audio signal that is compatible with the telephone system in your country. In France, select only Alaw voices. In the United States, the appropriate type is MuLaw. If you do not select the correct format, the sound emitted by the speech synthesizer will be inaudible.

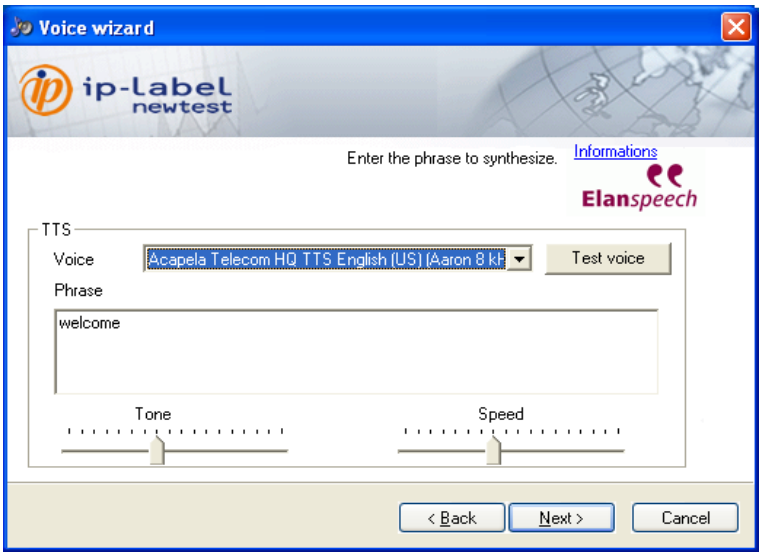


Most voice recognition services work best with a female voice. In most cases, it is advisable to use **Speech Cube US English SaySo(Laura 8khz Alaw)** for American English language voice servers.

Enter the text to synthesize in the ***Phrases*** field, using punctuation as you would naturally when writing a sentence. This will affect how the text is pronounced (rising intonation at the end of a sentences ending with an question mark, falling intonation for a sentence ending with a period or full stop, etc.).

You will need to make several trial runs before finding the optimum setting that will be correctly recognized by the voice server.

To test your text's voice synthesis, click **Test voice**. The wizard then appears, and pronounces the phrase according to your settings:



Fine-tune the pronunciation of your text with the following trackbars:

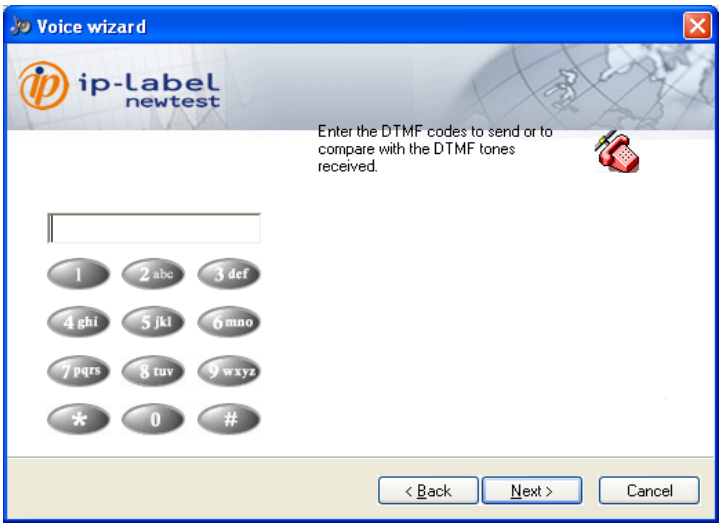
Tone: to make the voice lower (move the slider to the left) or higher (move the slider to the right)

Speed: to pronounce the text more slowly (move the slider to the left) or faster (move the slider to the right)

Advice on fine-tuning: a medium tone (with the **Tone** slider in the middle) spoken fairly slowly (**Speed** slider slightly to the left) facilitates recognition of the synthesized voice by the server's voice recognition software.

9.6.5 SENDING OF A DTMF TONE

This is the most frequently used action in voice scenarios. Most voice servers react to sequences of tones from a telephone keypad. Newtest can send any sequence that corresponds to the use of the 12 keys of the telephone keypad: **[0-9]**, ***** and **#**.

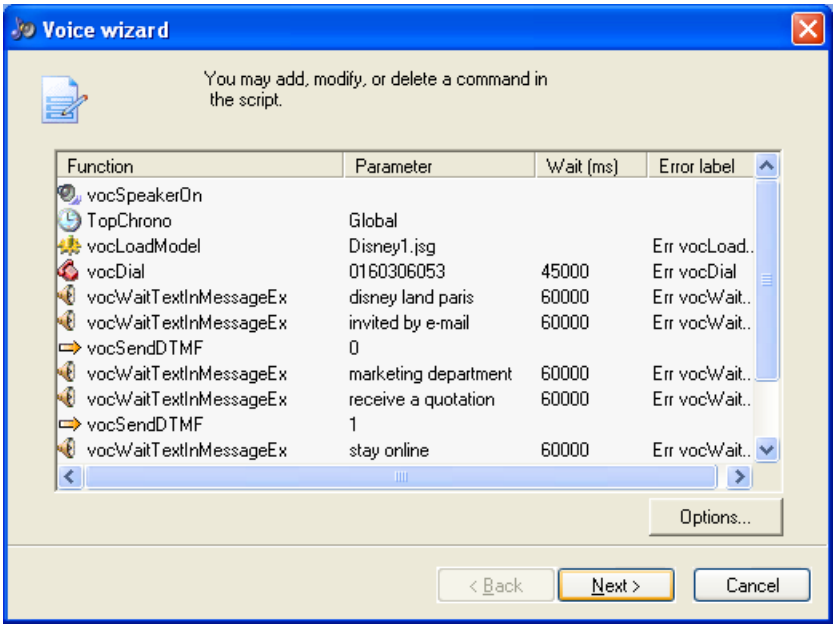


Using the virtual telephone keypad in the dialogue box or your keyboard, enter the DTMF tone or sequence of tones to send.

9.6.6 LEARNING OVER - SCENARIO SUMMARY

When you have finished the learning stage for the series of steps that make up your scenario, check the **Finish scenario** box at the bottom of the active window. This means that there will be no further steps, and that the scenario is over.

Click **Next** to display the final wizard step, the scenario summary:



You can at any time edit and modify the scenario using this interface by relaunching the script generation wizard.

The table shows the order of Newtest Basic instructions in the script that were generated by the wizard. Each row corresponds to a function in the script:

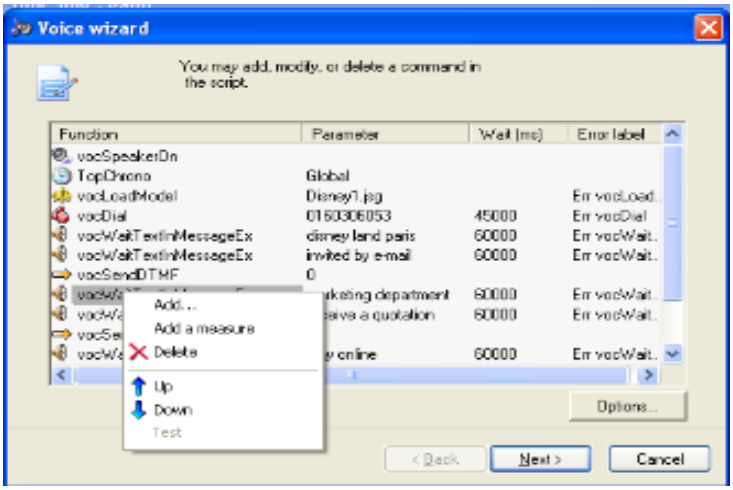
Function: name of the function (Newtest Basic order).

Parameter: value of the function's main parameter, if applicable. For functions with additional parameters, these are modified directly in the scenario script. You can edit this field and modify it by clicking in it.

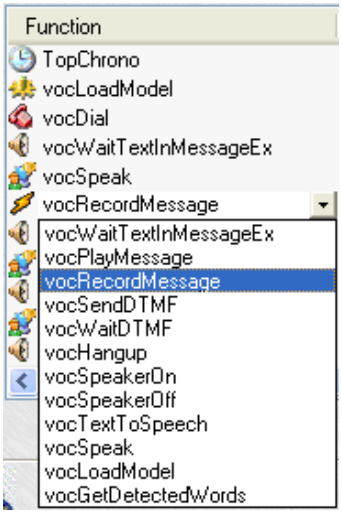
Wait (ms): value of the wait parameter if applicable to the function. For functions that include additional wait parameters, these are modified directly in the scenario script. You can edit this field and modify it by clicking in it.

Error label: text of the error to return in case of failure at the function. A standard, default text is set. It is highly recommended to set clear and precise error labels for each important step (primarily for criteria) as soon as you create the scenario, to facilitate analysis of the cause of unavailabilities later on. You can edit this field and modify it by clicking in it.

Right-click a row in the table to access editing functions for the script in progress, at the line currently selected:

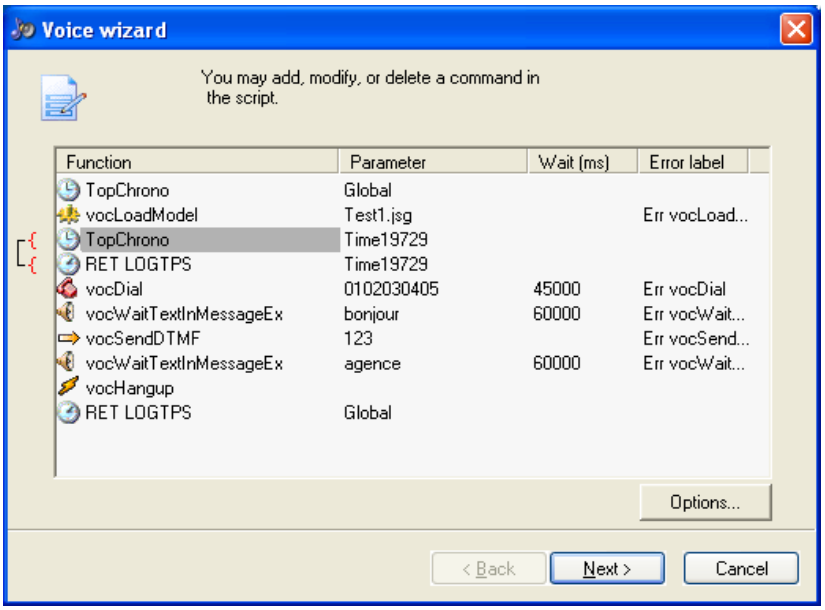


Add...: adds a Newtest Basic instruction to the script just after the selected line. Choose the instruction from the list.



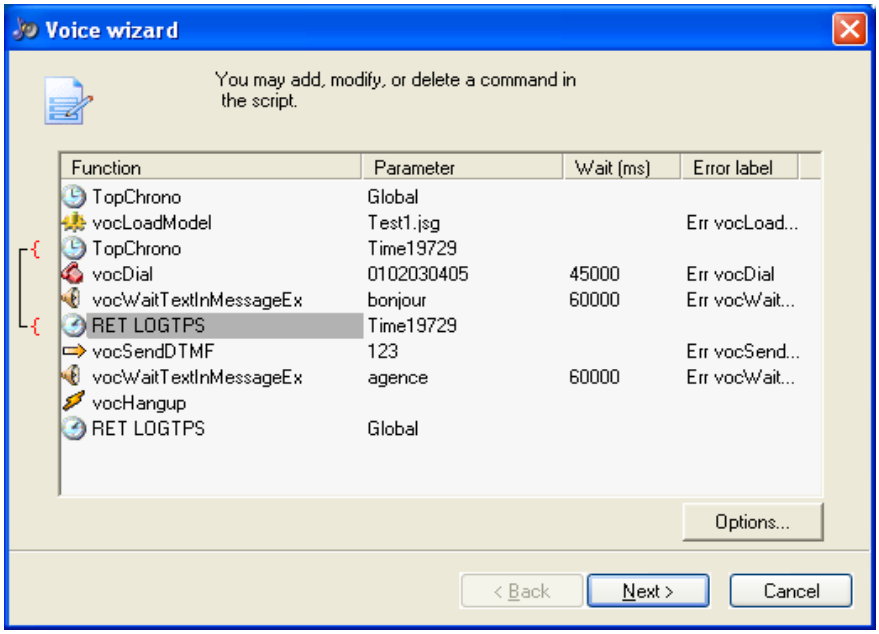
In the list, select an instruction to add. Then complete the parameters that apply to this function (default values are set automatically).

Add a measurement: add a measurement to the script (**Top Chrono** and **RET LOGTPS** instructions). The two instructions are placed immediately above the selected line:



Give the new measurement a name by editing its default parameter (Timexxxxx). You may modify either the parameter of the **TopChrono** instruction, or the parameter of **RET LOGTPS**. The name of the measurement is assigned automatically to the other instruction.

Lastly, using the mouse, drag and drop the **TopChrono** and **RET LOGTPS** instructions so that they surround the action(s) to be measured:



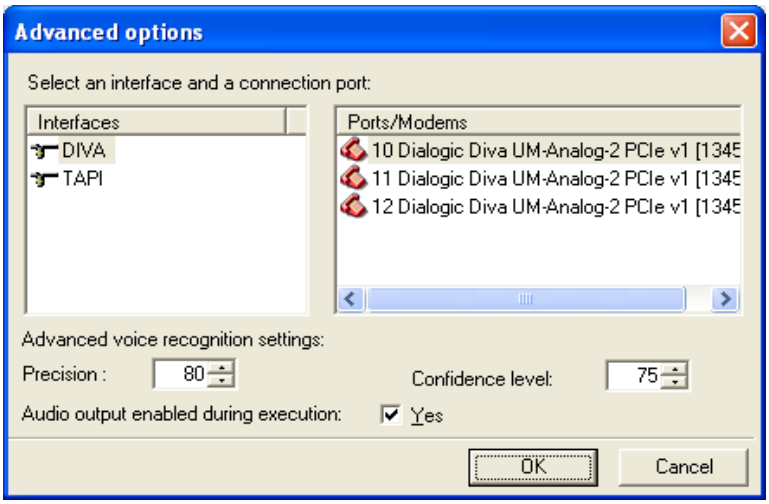
In the example above, the chronometer “Time4828” starts right after the called server picks up. It stops when it has detected the sentence “Hello Welcome to the voice server”.

Delete: deletes the selected instruction (shortcut: **Del** key)

Up: moves the selected instruction up one line (shortcut: drag and drop with the mouse)

Down: moves the selected instruction down one line (shortcut: drag and drop with the mouse)

OPTIONS: click this button to access the scenario’s advanced options.



Select an interface and a connection port: the ***Interfaces*** pane displays the list of telephone equipment available for connecting to the telephone network.

The ***Ports/Modems*** pane displays the connections (lines) available for the selected interface.

The TAPI interface appears when one or more audio stream compatible telephony devices are installed on the computer along with TAPI drivers.

The different lines (ports/modems) available on the device (interface) selected are displayed in numerical order with their logical name next to the icon . Select the line with which you want to connect to the telephone network (refer to your device's documentation).

Advanced voice recognition settings: these settings are for fine-tuning.

Precision: this setting controls the precision with respect to the speed of speech recognition.

The lower the value (0 being the lowest), the faster the recognition (quicker detection of criteria) to the detriment of recognition precision (lower possibility of correctly detecting the words pronounced).

The higher the value (up to 100), the greater the precision to the detriment of recognition speed. It is advisable to keep this setting at 80, the default value.

Confidence level: this setting controls the tolerance that you allow for the recognition of phrases. The lower the value (0 being the lowest), the more tolerance is allowed (meaning the confidence level is low) which increases the risk of recognizing words or phrases that were not pronounced.

The higher the value (up to 100), the less tolerance is allowed (the confidence level is high) which increases the risk of not recognizing words or phrases if the sound environment is complex (background noise, variation in the sound environment on the server, etc.). The default middle value works best in most cases.

However, if you encounter problems with a scenario's stability over time (failure to detect criteria), if may be useful to lower this setting in order to make the voice recognition system a little more tolerant of nuances in the phrases it has to detect. If, however, you find that Newtest tends to identify phrases or words that have not been pronounced, it is advisable to increase this value in order to increase the confidence level that you can allow for recognition.

Audio line out enabled during execution: select YES if you want the sound of the connection for this scenario to be audible at the QX 2000 board's audio output (refer to QX2000 documentation to locate the audio output).

9.6.7 CREATING THE GRAMMAR – COMPILING THE SCRIPT

When you quit the scenario summary by clicking the **Next** button, the process of creating the voice recognition grammar begins.

This step consists of phonetically translating the words used in your scenario's voice criteria equations.

The above screen appears following successful construction of the grammar.

However, if an error message results, you must check all wait criteria of voice messages and delete any special characters and punctuation that is not allowed.

To return to the preceding steps, use the **Back** buttons at the bottom of each wizard page.

When the grammar has been successfully created, Newtest offers to generate the scenario script.

Click **Finish**, and the scenario script is complete.

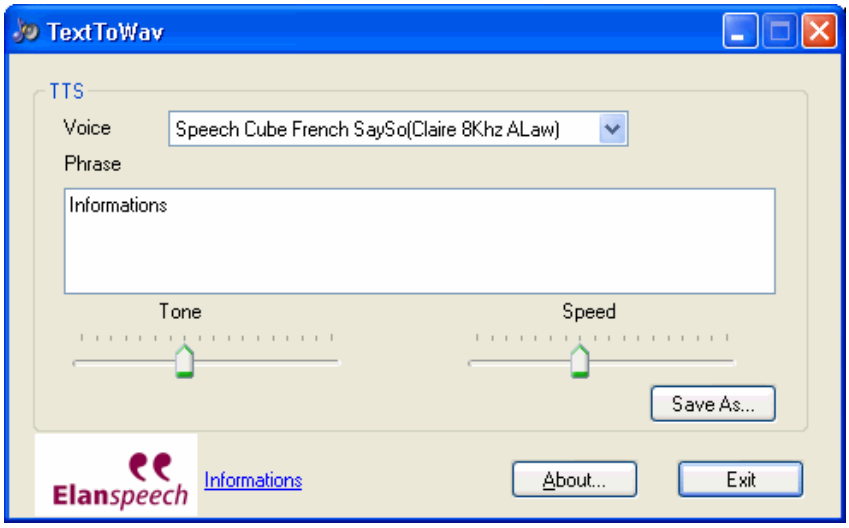
9.6.8 TEXT-TO-WAV UTILITY

If you do not have Elan Speech Cube 5.0 licenses on deployed Newtest robots, or if you are using scenario execution in different queues (Newtest Multi-Probes and Newtest Voice), you can use the Newtest development workstation to generate audio files (WAV) of synthesized phrases and send these audio flows using the command ***VocPlayMessage***. This way, a single Elan Speech Cube 5.0 license is needed on the computer that is used to develop the scenarios.

This utility serves only to facilitate the creation of WAV files. Functionalities for listening to generated messsages prior to saving are not available in this utility. It is therefore advisable to prepare and test the settings at first in the interface of the Newtest Voice script generation wizard.

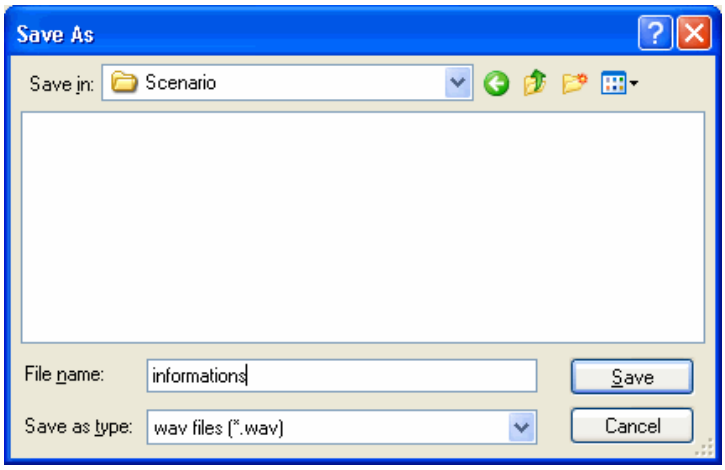
To configure a scenario this way, launch the TextToWav, which is installed with Newtest when you select the Voice Client Environment. Use the shortcut in the Newtest application group of the Start menu, or directly launch TextToWav.EXE which is located in Newtest's EXE directory.

Important: Elan Speech Cube 5.0 and ELAN ADO must be correctly installed on the computer (refer to the installation notice at the end of this chapter).



The interface is identical in every way to the script generation wizard. Select the type of **Voice**, enter the **Phrase** to pronounce, and set the speaking **Tone** and **Speed** so that they match the settings that you have already validated in the wizard.

Next, click **[Save As...]** to generate and save the resulting WAV file to the disk:



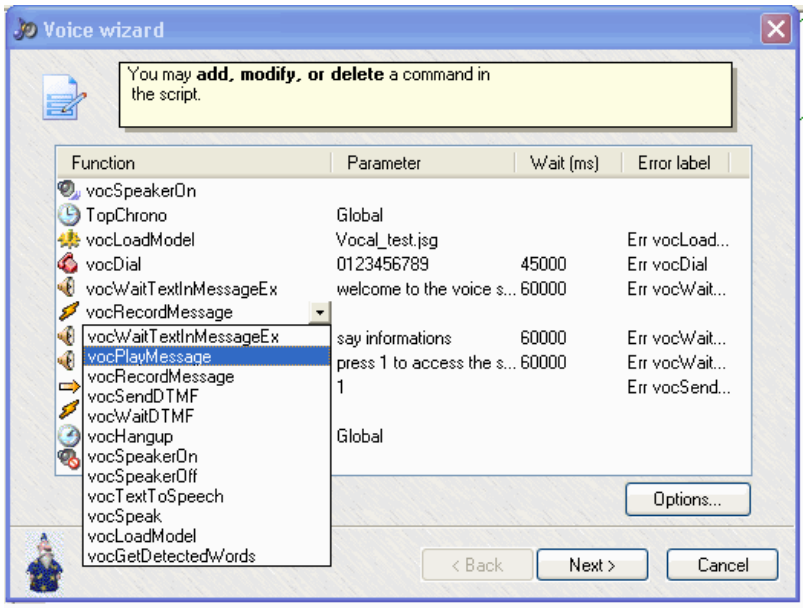
Select the location of the WAV file to generate and its name, and then click **[Save]**.

It is advisable to organize these files in a subdirectory under Newtest's USER directory to ensure that these files will be taken into account by Newtest's centralized management without further manual intervention. These files will henceforth be indispensable for proper operation of the scenario.

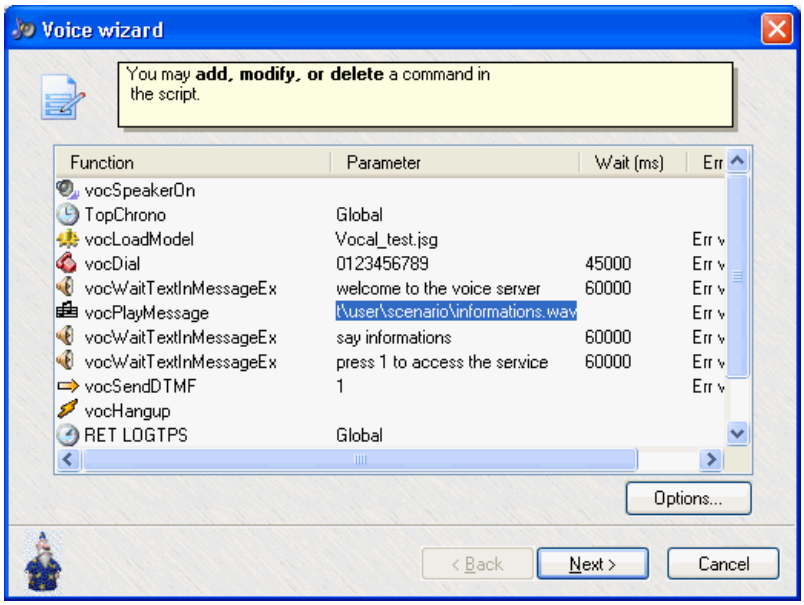
Repeat this operation as many times as necessary to obtain all the WAV files that your scenario needs. Test your audio files by using a WAV-file player (for instance, Windows Media Player).

Next create the scenario script using the voice wizard. Pass over each step that requires the sending of a voice message.

At the end of the script creation process, in the scenario's summary, you can add a ***VocPlayMessage*** commands to each step for which it is necessary to send a voice message:



Right-click the row below the one where you want to insert the transmission of a voice message. Next, select the command ***VocPlayMessage***.



Type the full path and name of the WAV file to play for this step.

Repeat this operation for all the steps involved in sending a voice message.

Your scenario is ready. Make sure that the WAV files designated in the scenario are in the location specified in the parameters of the vocPlayMessage functions on the robots deployed in production.

IMPORTANT: It may happen that the WAV file played as the scenario runs is not distinctly audible over the speakers. The sound is nonetheless being correctly emitted over the telephone line in accordance with the coding parameters you applied (A-Law or Mu-Law). The speakers' audio output is not a strict representation of this because speakers are dependent on the codecs installed on your computer and on Active-X parameters.

9.7 AUTOMATIC DIAGNOSTIC

A procedure for automatic creation of diagnostic files is included in the voice client environment.

When you replay the script in the Configurator or in the Newtest execution module, a WAV file (sound recording) is created each time Newtest does not detect the expected voice criterion. The audio file includes everything that was audible on the telephone line from the time Newtest began to wait for the criterion until the timeout for this criterion (end of wait).

The data in the file is in the following format:

<SCENARIO NAME>_<HOUR>_<MINUTE>_<SECOND>.

It is always saved to:

<Newtest installation directory>\RESULT\<Year>\<Month>\<Day>.

Example of a trace file:

TESTVOC_15_35_12.WAV

This filename means that an audio diagnostic was generated for the TESTVOC scenario at 15:32:12 on the date indicated by the filepath in the tree of the RESULT directory.

9.8 NEWTEST BASIC COMMANDS FOR THE VOICE ENVIRONMENT

NAME	DESCRIPTION
vocDial	Dials up
vocGetDetectedWords	Supplies a list of words detected by vocWaitTextInMessageEx
vocGetLastError	Returns the code of the latest error
vocGetLastLogFile	Returns the full path of the latest trace file (WAV) generated
vocGetStringError	Returns the label of the latest error
vocHangup	Hangs up and closes the context of the scenario
vocInitialize	Initialization of the Voice Client Environment
vocLoadModel	Loads voice recognition models (grammar)
vocPlayMessage	Plays a voice message (WAV file)
vocPreserveLogFile	Saves the trace file after the scenario runs. The file is automatically saved in the event the scenario status is “Unavailable/Failed”.
vocRecordMessage	Records a voice message
vocSendDTMF	Sends a DTMF sequence
vocSetAccuracyConfidence	Configures the accuracy of the speech recognition engine
vocSpeak	Sythesizes voice from text via the connected line (optional)
vocSpeakerOff	Deactivates QX 2000 voiceboard audio output
vocSpeakerOn	Activates QX 2000 voiceboard audio output
vocTextToSpeech	Sythesizes voice from text in a .wav file (optional)
vocWaitDTMF	Awaits an incoming DTMF sequence
vocWaitTextInMessageEx	Awaits a voice sequence (voice recognition)

Refer to Newtest online help for more information on these functions.

9.9 SAMPLE SCRIPT

The following script dials up voice server “123456” and waits for a welcome message containing the phrase “Please say which service you want”. Newtest then sends the DTMF frequency “ * “ to return to the menu.

Next, the phrase “Say the name of a service or say return” serves to validate that the menu was given. Newtest synthesizes, with a female voice, the word “Information”, i.e. the name of the department to test.

The rest of the scenario is a succession of recognitions of key phrases pronounced by the voice server, and the synthesizing of phrases simulating a human voice to test the operation of the voice recognition service of the server called.

The scenario activates the QX2000 voiceboard's audio output to allow real-time monitoring of its operation.

The generated script is as follows:

```
////////////////////////////////////////////////////////////////////////////////
// Name: 'Serv3000'
// Date: 04/11/2003
//
////////////////////////////////////////////////////////////////////////////////

DIM bRet AS BOOL
DIM iErr AS INT
DIM sError AS STRING
DIM sResponse AS STRING
DIM Global AS ORDER(TRESP)

bRet = vocInitialize ( "QX",80,50 )                    // Initializes the voice environment with QX2000 voiceboard
IF bRet = FALSE THEN                                   // Sets speech recognition parameters (Precision: 80,
                                        GOTO ERROR                confidence level: 50)
ENDIF

vocSpeakerOn ()                                        // Activates the QX 2000 voiceboard's audio output
TopChrono ( Global )                                   // Starts the chronometer for the GLOBAL measurement
bRet = vocLoadModel ( "Serv3000.jsg" )                 // Loads the voice recognition grammar
IF bRet = FALSE THEN
                                        iErr = vocGetLastError ()              // Retrieves the error code in case of Timeout
                                        sError = vocGetStringError ( iErr )    // Retrieves the error label in case of Timeout
                                        RET ErrorLine ( "Err vocLoadModel" )
                                        GOTO ERROR
ENDIF

bRet = vocDial ( "0","123456",45000 )                  // Dials up the voice server "123456"
IF bRet = FALSE THEN
                                        iErr = vocGetLastError ()
                                        sError = vocGetStringError ( iErr )
                                        RET ErrorLine ( "Err vocDial" )
                                        GOTO ERROR
ENDIF

sResponse = vocWaitTextInMessageEx ( "Please say which service you want",60000 )      // Speech recognition criterion. Waits
for
IF sResponse = "" THEN                                                                // the phrase for 1 min.
                                        RET ErrorLine ( "Error Welcome" )
                                        GOTO ERROR
ENDIF

bRet = vocSendDTMF ( "*" )                                          // Sends a DTMF sequence, key " * "
IF bRet = FALSE THEN
                                        iErr = vocGetLastError ()
                                        sError = vocGetStringError ( iErr )
                                        RET ErrorLine ( "Err vocSpeak" )
                                        GOTO ERROR
ENDIF

sResponse = vocWaitTextInMessageEx ( "Say the name of a service or say return",60000)   // Speech recognition criterion.
IF sResponse = "" THEN                                                                  // Waits for the phrase for 1 min.
                                        RET ErrorLine ( "Error Service choice" )
                                        GOTO ERROR
ENDIF

bRet = vocSpeak ( "information","Speech Cube US English SaySo(Laura 8Khz ALaw)",113,100 )   // Speech synthesis of the
IF bRet = FALSE THEN                                                                        word "Information" with a
                                        RET ErrorLine ( "Err vocSpeak" )                    female voice, "Laura".
                                        GOTO ERROR
ENDIF

sResponse = vocWaitTextInMessageEx ( "takes your calls while you are out",60000 )   // Speech recognition criterion.
IF sResponse = "" THEN                                                              Waits for the phrase for 1 min.
                                        RET ErrorLine ( "Error information" )
                                        GOTO ERROR
```

```
ENDIF

bRet = vocSpeak ( "menu","Speech Cube US English SaySo(Laura 8Khz ALaw)",113,100 )    // Speech synthesis of the word
IF bRet = FALSE THEN                                                                  "menu" with a female
                                        RET ErrorLine ( "Err vocSpeak" )              voice, "Laura"
                                        GOTO ERROR
ENDIF

sResponse = vocWaitTextInMessageEx ( "voici la liste de choix disponibles",60000)  // Speech recognition criterion. Waits for
IF sResponse = "" THEN                                                                the phrase for 1 minute
                                        RET ErrorLine ( "Error index" )
                                        GOTO ERROR
ENDIF

vocHangup ()                                    // Hangs up the line and closes the voice environment
RET LOGTPS ( Global )                           // Returns the timer of the GLOBAL measurement
vocSpeakerOff ()                                // Closes the QX 2000 voiceboard's audio output

END

ERROR:
vocHangup ()
```

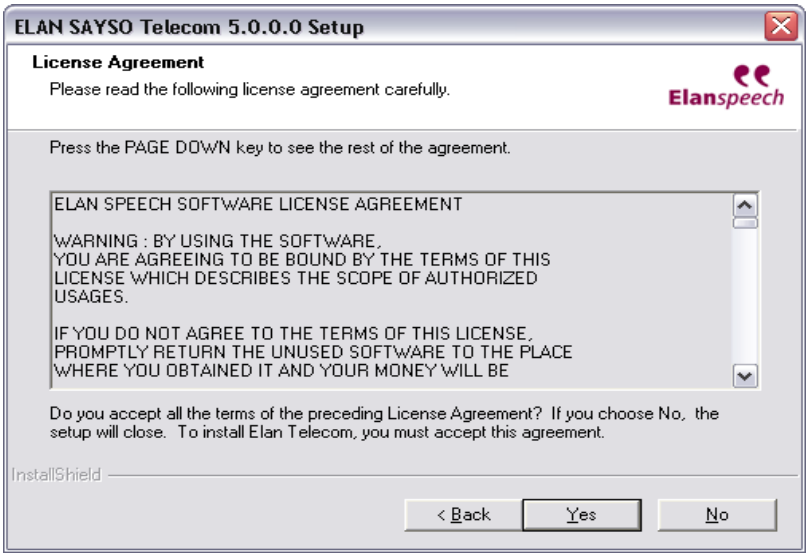
9.10 INSTALLING ELAN SPEECH CUBE 5.0

To use voice synthesis with the Newtest Voice Client Environment (and with the Text To WAV utility), ELAN SPEECH CUBE SAYSO 5.0 must be installed. If it is not, the action Send/Voice message will be unavailable (shaded) and this function will remain inaccessible.

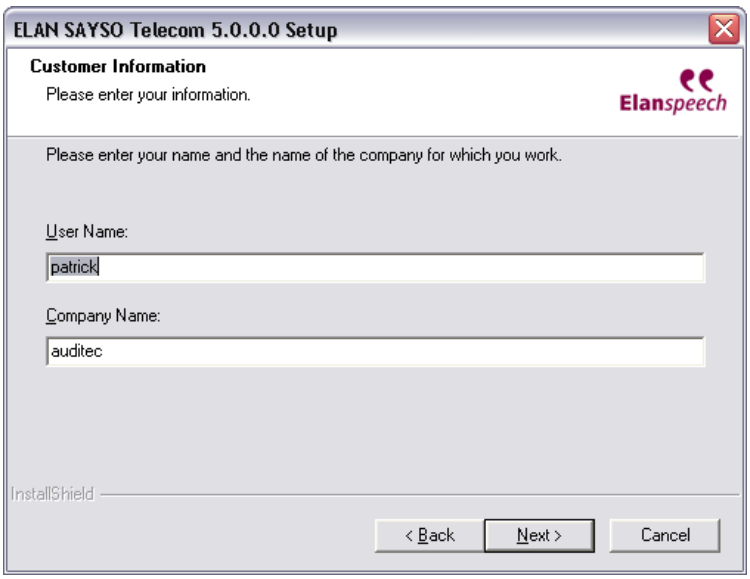
9.10.1 INSTALLATION PROCEDURE

Insert CD 1 of Elan Sayso Telecom Elanspeech V 5.0 into the CD drive.

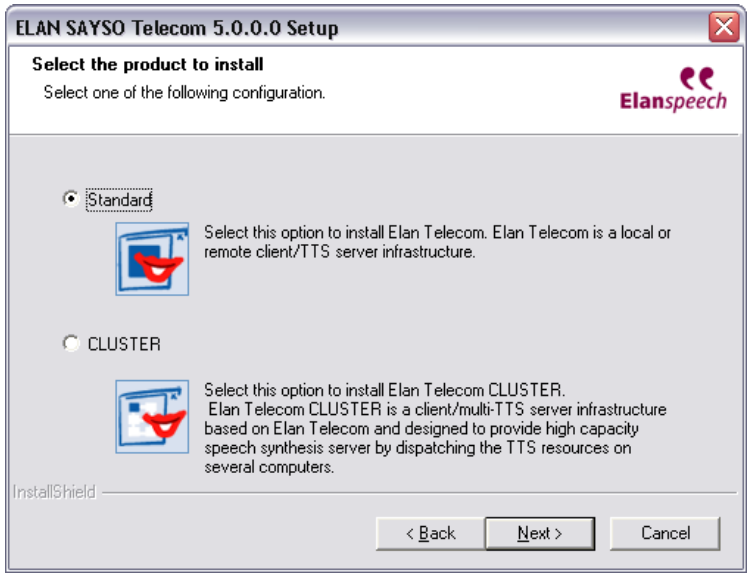
Click **Elan Telecom (Sayso/Tempo)**, and then click **Next** when prompted to launch the installation procedure.



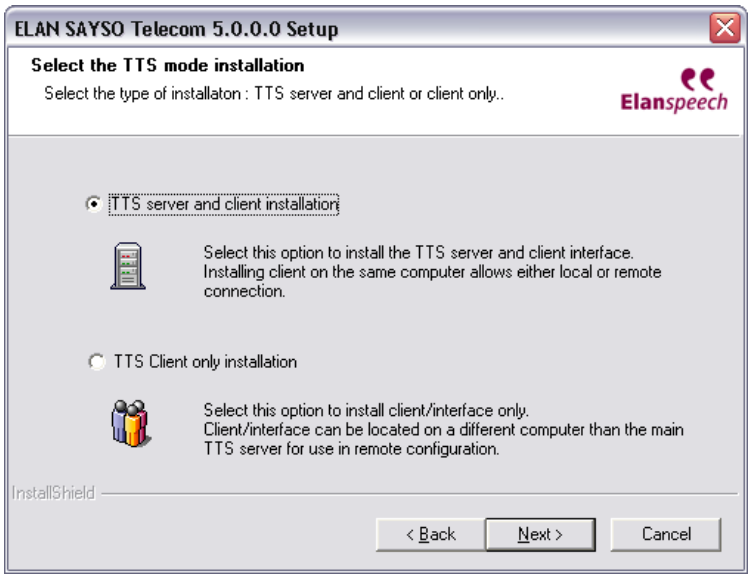
Click **Yes** if you accept the **ELANspeech** license agreement shown. If you do not agree, do not proceed with installation.



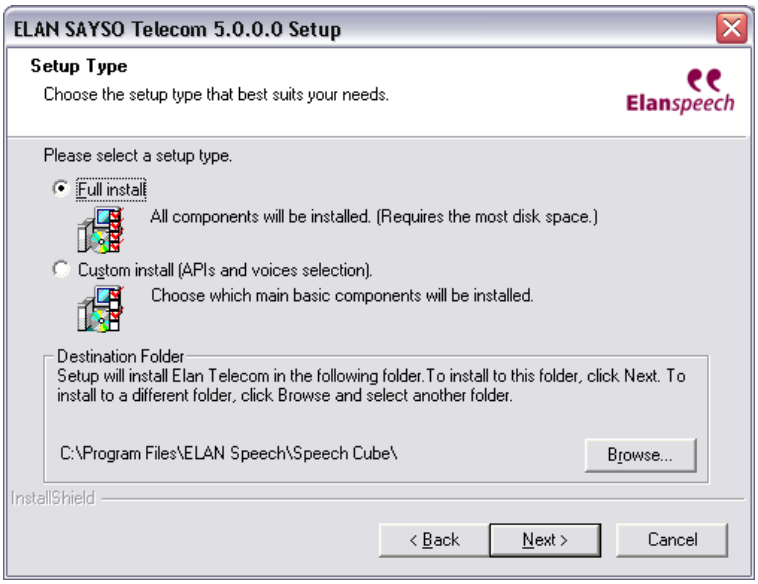
Enter the information requested and then click **Next**.



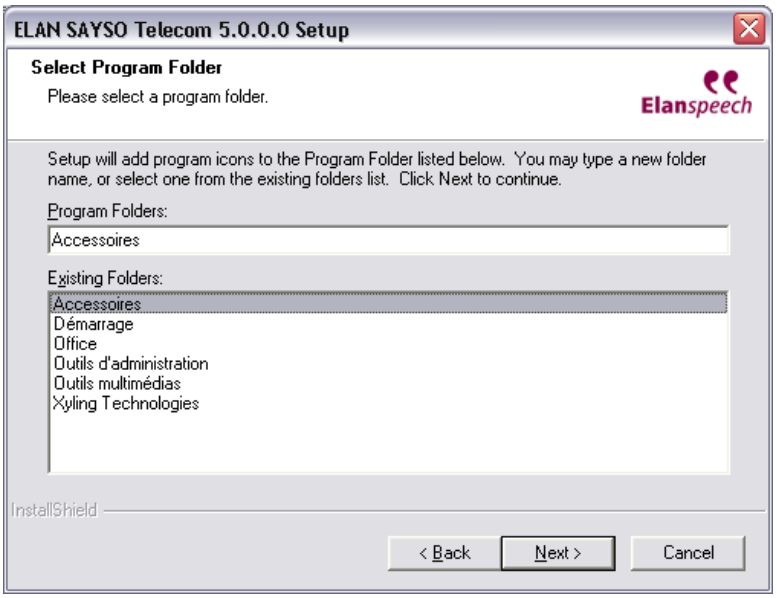
Select ***Standard*** and then click **Next**.



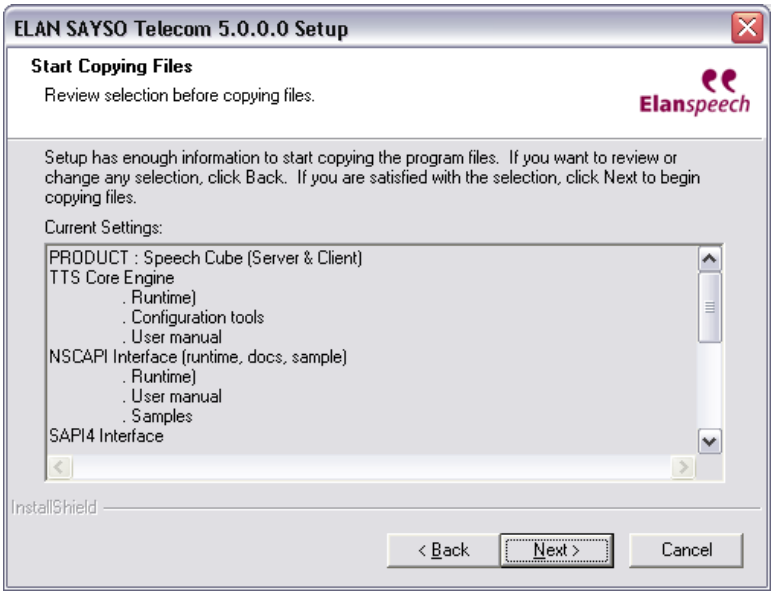
Select ***TTS server and client installation*** and then click **Next.**



Select ***Full install*** and then click **Next.**



Set the default location of the program in the Start menu, or accept the default location, and then click **Next**.

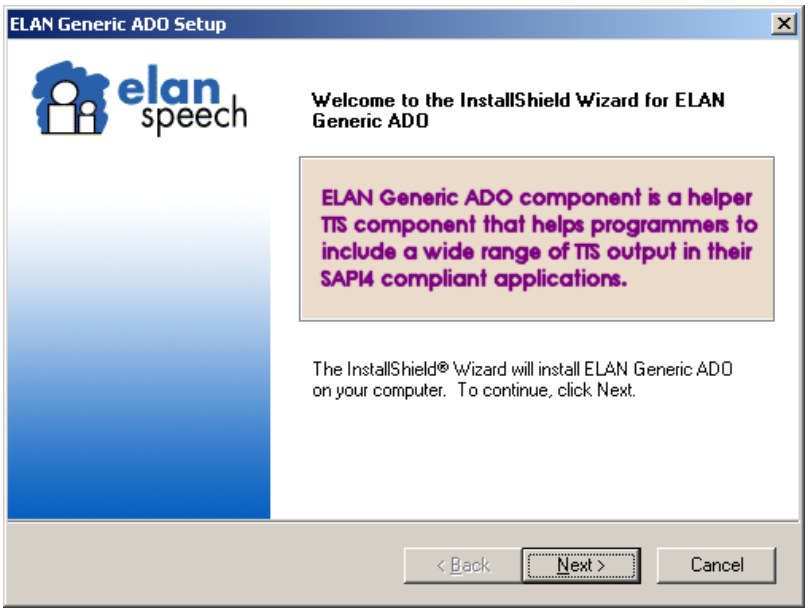


Click **Next** to confirm the information you entered.

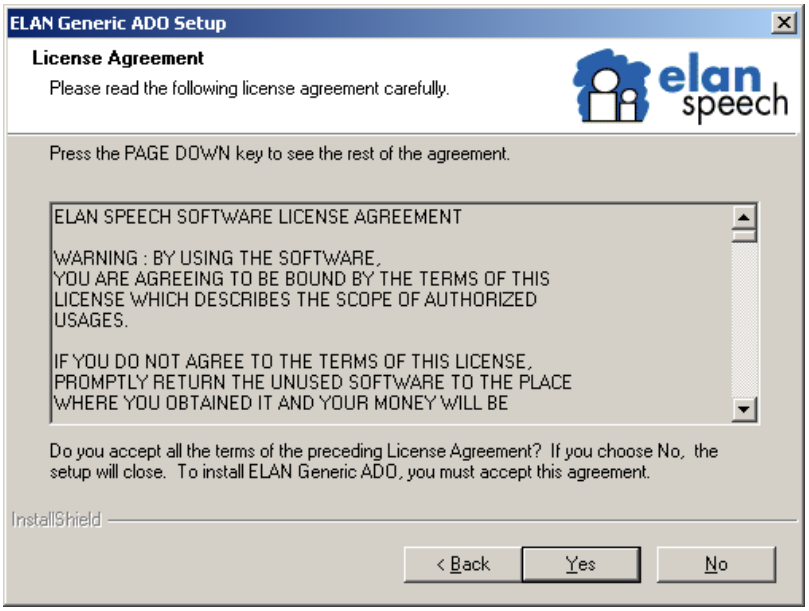
Allow installation to proceed.

When this part of the software has been installed, you need to add the module **ELAN-ADO**.

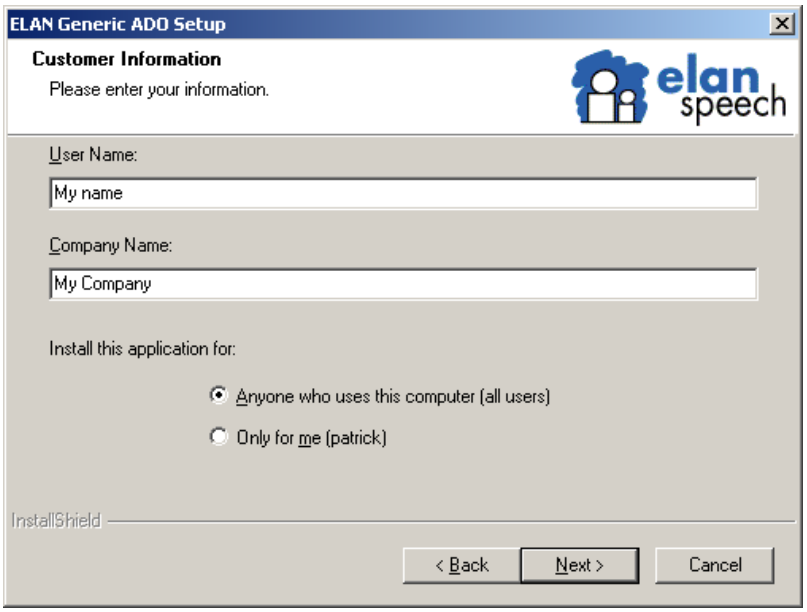
Click **Elan Generic ADO** in the installation welcome page after inserting CD 1 into the drive again.



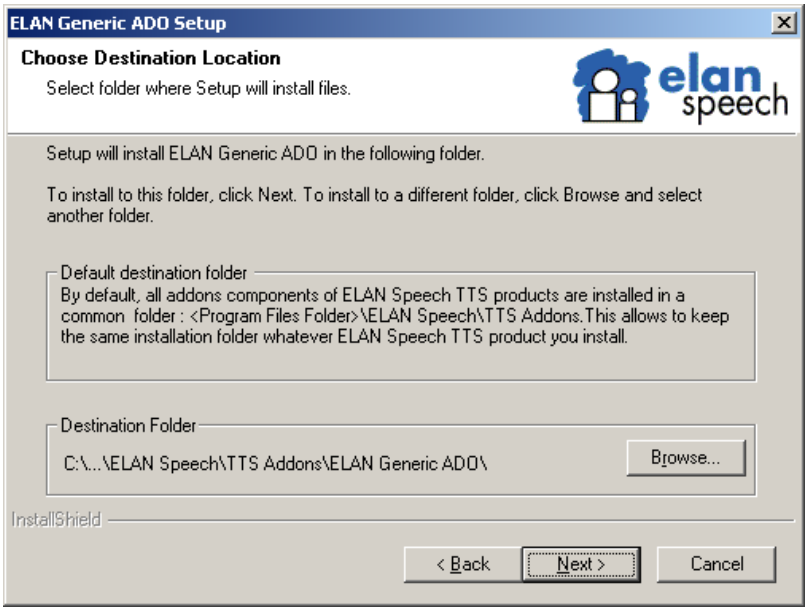
Click **Next**.



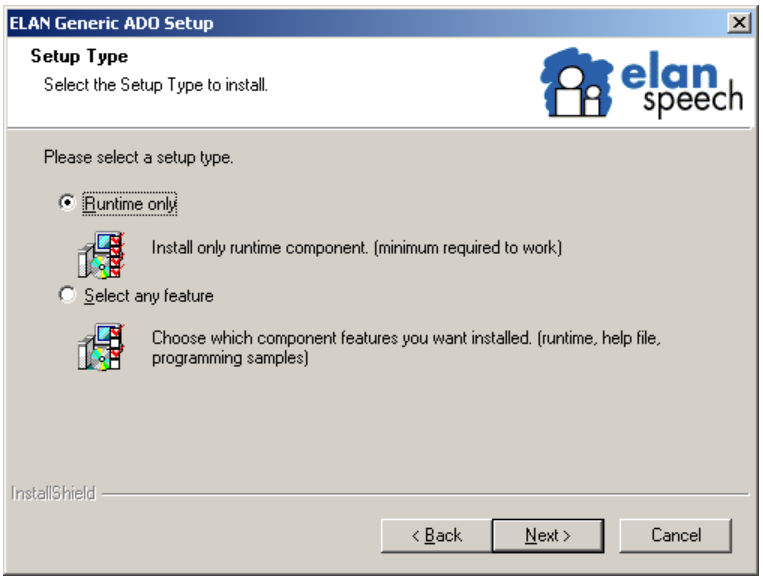
Click **Yes** if you accept the **ELANspeech** license agreement shown. If you do not agree, do not proceed with installation.



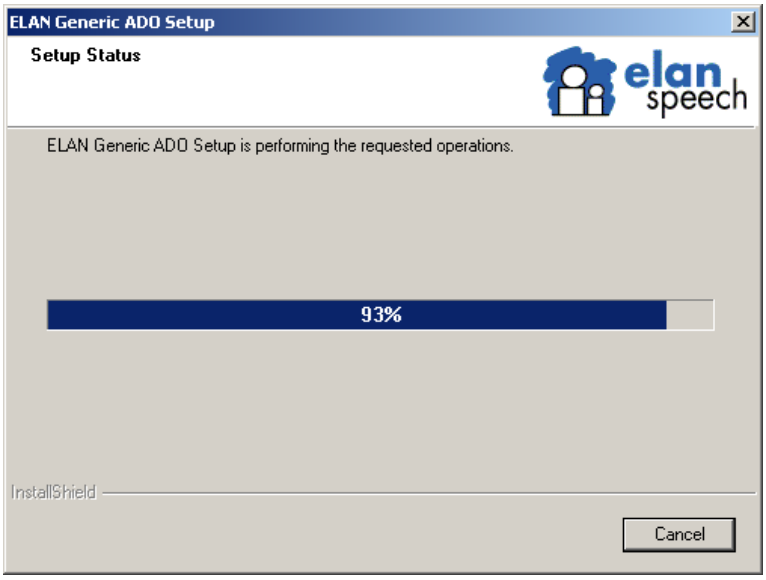
Enter the information requested and click **Next.**



Click **Next.**



Select ***Runtime only*** and then click **Next**.



Wait while setup finishes installing the module.

When both products have been installed, speech synthesis functions become available in the Newtest Voice Client environment.

9.10.2 PRODUCT LICENSE

The installation of **ELAN SPEECH CUBE** generates a default lending license that is valid for one month.

> If you plan to continue to use this product, you must purchase a license from the reseller of your choice or directly from **ElanSpeech**.

Version 5.0 of the product is required, as this is the only one validated to date for use with the Voice Client Environment.

10 SPECIFICS OF AUDIO RECORDING

10.1 CAPTURE FUNCTION

Audio recording makes it possible to capture a Windows sound source in a file for processing or to return as a diagnostic. It does not require a telephone voice board because it is carried out on standard Windows audio flows.

The primitive StartAudioCapture takes a destination file (.wav) for the recording, along with the following parameters:

- **Sampling**: CD quality sound is at 44.1 kHz, DVD at 48 kHz and telephone at 8 kHZ. The default sampling frequency can be set in system properties. You can nevertheless impose, for example, 8 kHz for telephony needs (PESQ has been validated for 8 and 16 kHz samplings).
- **Mono/stereo filter**: to select a single channel if necessary
- **Deletion of silences (trim)**: this feature must be activated when you want to detect an absence of audio signal. The duration of the recording is then null.

The primitive StartAudioCapture then returns:

- a file that can be exported in .wav format
- the recording duration. Combined with the deletion of silences (trim) parameter at the beginning and end, this value is indicative of an absence of sound when the duration is equal to 0..

```
bRet=StartAudioCapture(InputFile,113)
IF bRet=FALSE THEN
        RET ErrorLine("Err001: error starting audio capture")
        GOTO ERROR
ENDIF

//Record duration
Wait(20)
Duration=StopAudioCapture()
TraceinfoInt("Stop audio duration: ", Duration)
```

The duration of the recording can then be tested to generate an error if there is no sound:

```
IF Duration <= 0 THEN
  RET ErrorLine("Err002 StopAudioCapture: no sound")
  GOTO ERROR
ENDIF
```

10.2 AUDIO IN SESSION MODE

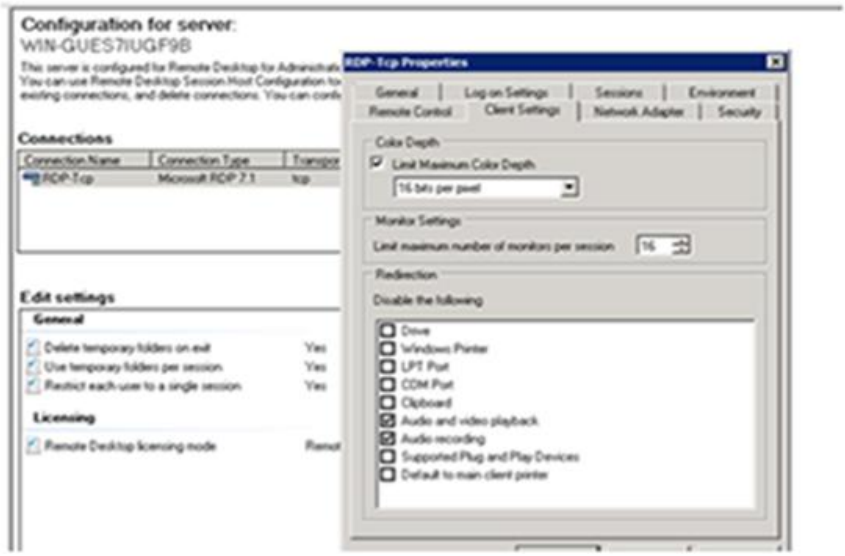
When the robot is operating in session mode on a server operating system, additional configuration may be necessary.

- *Configuring audio service*

Call the program *services.msc* on Windows 2008 & 2012, and set the *Windows Audio* service to start automatically.

- *Configuring audio on remote desktop sessions*

For Windows 2008, use the tool *tsconfig.msc* to disable the RDP-TCP *Audio and video playback* option.



11 SCENARIOS BUILT WITH THE COM PORT CLIENT ENVIRONMENT

11.1 INTRODUCTION

This client environment is designed to control devices, such as modems or mobile telephones, which connect to a serial port (COM) of a computer.

COM Port scenarios consist of a set of actions involving serial connection to the controlled device, transmission of character strings such as AT commands, and reception of strings returned by the hardware (criteria).

11.2 INSTALLING THE DEVICE

Make sure that the device to be monitored is connected to one of the computer's serial ports and that the connection is functioning (it may be helpful to use Windows Hyperterminal, for example, to make sure it is possible to establish a connection with the device).

Ensure that you have on hand all the configuration information for connections with devices.

11.3 TECHNICAL PRINCIPLES OF THE SOLUTION

The COM PORT client environment performs the following actions:

- establishes serial communication with the device
- sends a string to the device over the serial connection
- waits for (polls) strings received on the connected serial port for comparison with a reference pattern (criterion)
- memorizes, in working variables, the strings received
- closes the connection and frees the serial port

The script generation wizard builds a scenario made up of exchanges between the computer and the controlled device. This exchange can consist of either:

- a transmission of data and a wait for a response
- a wait for data

The data received is retrieved character by character and stored in the buffer until the awaited string is found, or until timeout is reached.

The wizard allows you to add measurement points (Newtest chronometer) to the sequence of such actions and also add functions for extracting character strings from the data received.

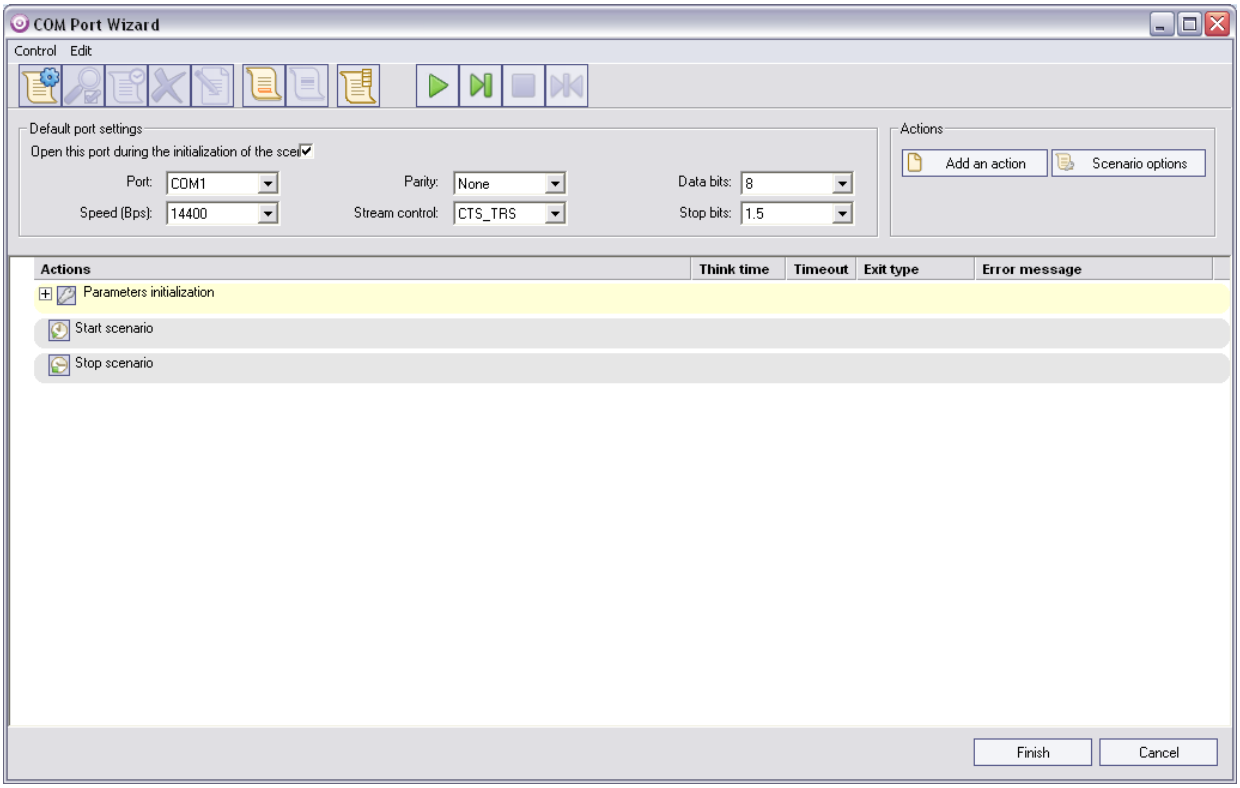
The sequence of exchanges whose parameters are set individually in the interface is what makes up the scenario script. It is also possible, using the COM Port client environment, to monitor the overall operation of one or more transactions involving a series of exchanges.

Example: a scenario runs the series of AT commands required by a mobile telephone to send an SMS, verifies that the acknowledgement receipt is receive, and oversees the purge of the telephone's local elements.

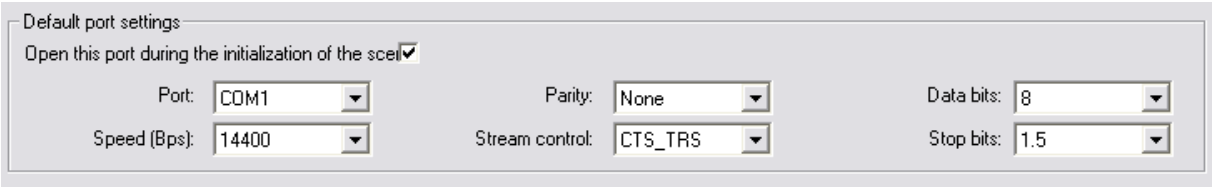
11.4 PROGRAMMING THE COM PORT SCENARIO

11.4.1 BUILDING A COM PORT SCENARIO WITH THE SCRIPT GENERATION WIZARD

The main window of the COM Port client environment wizard allows you to configure the actions to run as well as the parameters to use for monitoring devices connected to the computer's COM ports.



The **Default port settings** area allows you to prepare port parameters for the monitored device:



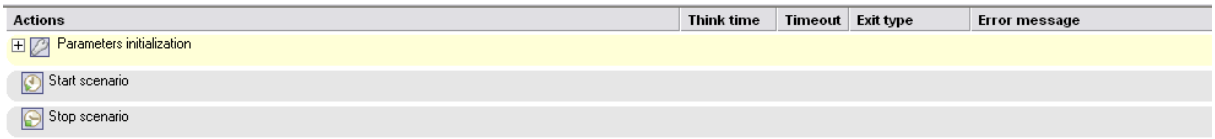
The COM port designated in this part of the interface will be open at the start of the script and closed at the end. The actions added to the script may open other ports if necessary. You can, therefore, keep a default communications port if it is the only one on which exchanges will take place during the scenario, or if it serves for communications in parallel with other actions to perform.

If it is not useful to open a port automatically, unselect the option **Open this port during the initialization of the scenario**.

Set the following communication parameters:

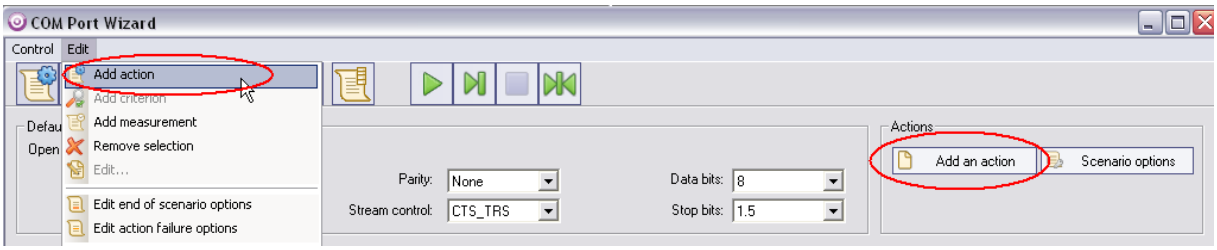
- **Port**: using the list, indicate the COM port (serial port) on the computer to which the device is connected
- **Parity, Data bits**, **Speed (Bps)**, **Stream control**, **Stop bits**: fill in each of the parameters for the type and speed of serial communication applicable to the device connected to the selected COM port. Please refer to the technical documentation of the specific device.

11.5ADDING ACTIONS TO THE SCENARIO

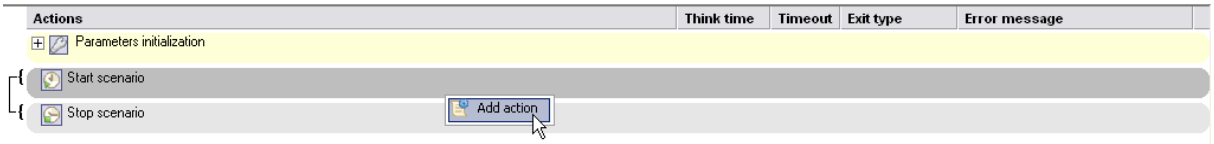


When the wizard in launched, the table shows only the scenario's start and end limits. The duration of the scenario's execution, which is automatically transmitted to Newtest Management Console, is calculated on the time it takes for the actions to run between these two limits.

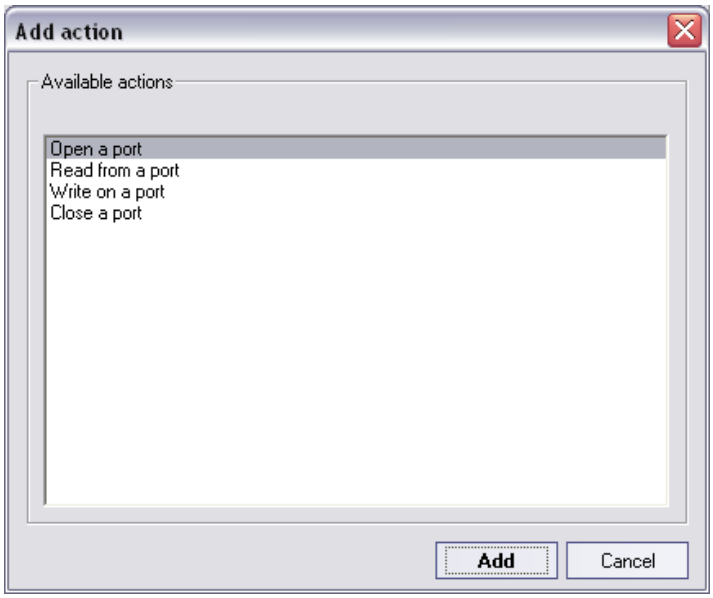
To add actions to the script, use the menu Edit, option Add action or click the button Add an action in the wizard:



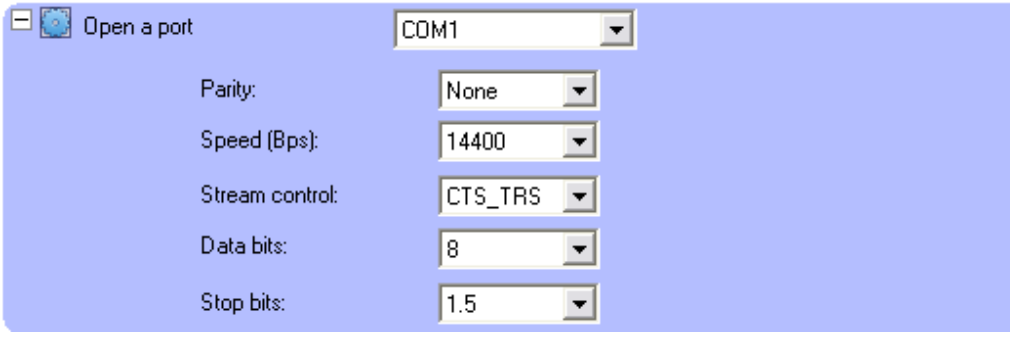
You can also use the pop-up menu by right-clicking in the table's action display section:



The dialogue box which appears next facilitates the choice of type of action to insert into the scenario script:



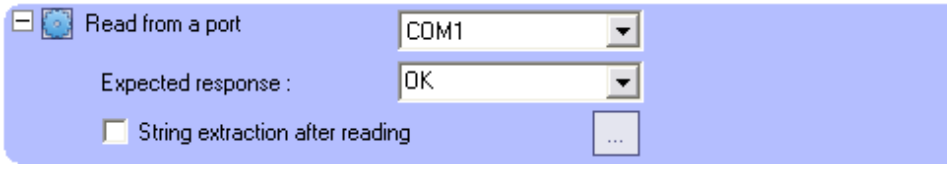
Open a port: add an action to open a COM port. Access the settings of the action by clicking ⊞ in order to specify the following elements:



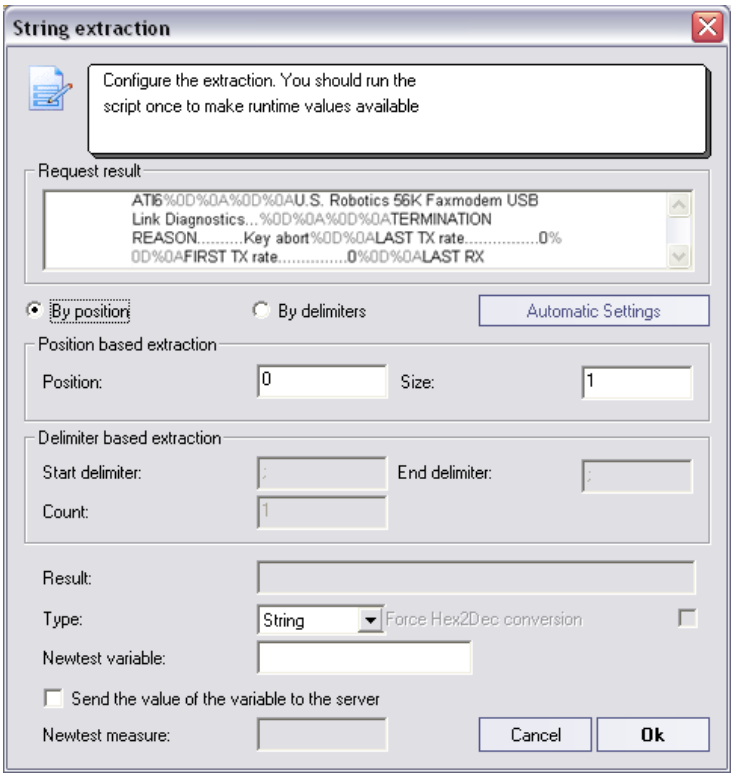
- **Open a port**: using the list, indicate the COM port (serial port) on the computer to which the device is connected.
- **Parity, Speed (Bps)**, **Stream control**, **Data bits**, **Stop bits**: fill in each of the parameters for the type and speed of serial communication applicable to the device connected to the selected COM port. Please refer to the technical documentation of the specific device.

Read the port: adds the action for reading the designated COM port in order to listen on the port and wait for the character string specified in the **Expected response** field for a maximum length of time corresponding to the **Timeout**. If the string is not received within this time, the command will be considered unavailable.

Access the settings of the action by clicking ⊞ in order to specify the following elements:



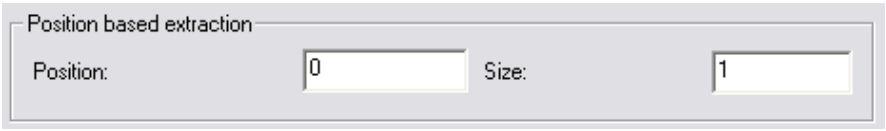
- **Read from a port**: using the list, indicate the COM port (serial port) on the computer from which you wish to read data.
- **Expected response**: select from the list or type in the character string awaited (criterion) on the designated COM port. If this response is not received, the step shows an error.
- **String extraction after reading**: select this option to extract to a variable a part of the content captures on the designated COM port. This option allows you to extract a result which varies with each scenario run. Click  to access the configuration panel:



Note: if you want to be able to work with the result elements retrieved on the COM port (recommended), you must play the port read action in the wizard prior to editing the extraction of a string. (See “automatic mode” below.)

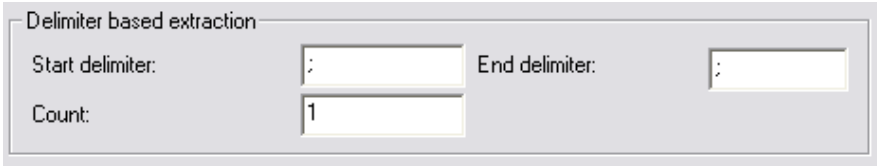
This panel allows you to configure the extraction of a string and its transmission, if desired, in a Newtest variable.

- **By position**: locates the string to extract with respect to its position within the string captured on the COM port



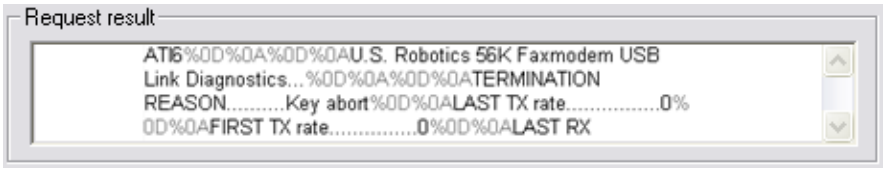
  - **Position**: position of the first character of the string searched for in the results of reading on the COM port
  - **Size**: number of characters read starting at the **Position**

- **By delimiters**: locates the string to extract with respect to a start and end delimiting character or string.

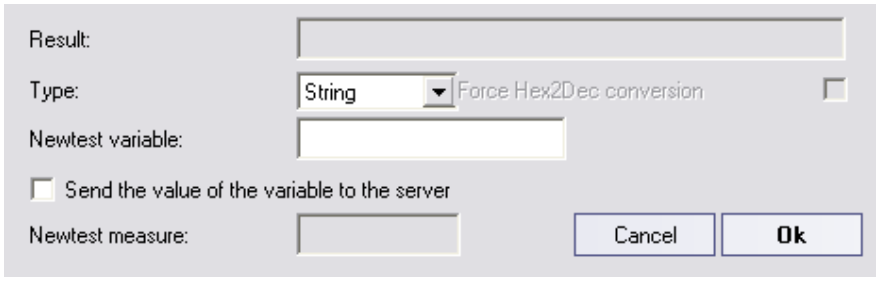


  - **Start delimiter**: character or string preceding the string to be retrieved
  - **Count** : occurrence of the start delimiter with respect to the start of the string to be retrieved on the COM port
  - **End delimiter** : character or string following the string to be retrieved

Automatic mode is available when the request has already been played in the wizard. In this case, the information resulting from reading the COM port appears in the field **Request result**:



Select the string corresponding to the result you with to retrieve on every execution. Next, select the method to use (by position or by delimiter), and then click the button **Automatic settings.** The wizard automatically configures the parameters for extracting the string and fills in the corresponding fields.



- **Result**: result retrieved from the currently programmed string extraction selection
- **Type**: in the drop-down list, select the type of variable (String or Int) to use to retrieve the data extracted. If the type INT is selected, the string retrieved will be converted into numbers. If the type is Int (integer), you can select the option Force Hex2Dec conversion to convert the base-16 numeric result into base-10 numeric.
- **Newtest variable**: name of the Newtest variable to receive the result of the extraction
- **Send the value of the variable to the server**: select this option to transmit the retrieved content to the server in a variable (*ORDER(NUM)* for numeric data; *ORDER(APPLI)* for a character string)
- **Newtest measurement**: name of the Newtest ORDER variable to receive the extracted data

IMPORTANT: in a command, Newtest stops reading the COM port when it has received the expected criterion. It receives data character by character. It is therefore important that the criterion be located AFTER the data to process if you want to extract and process data from the string received.

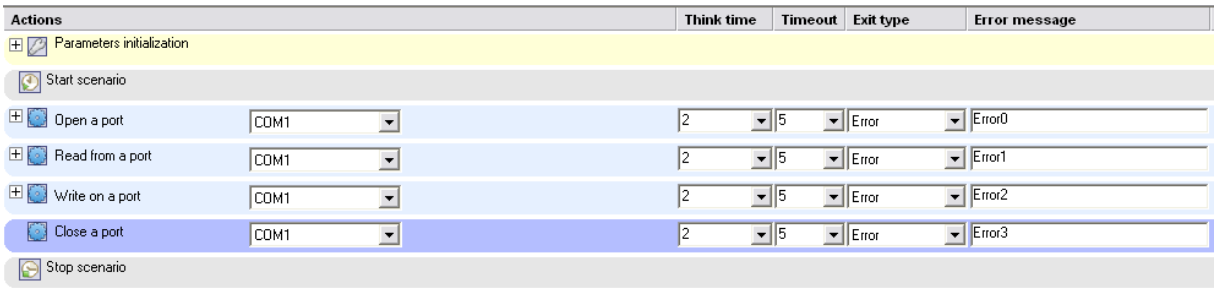
Example: if the expected response to a command is **abcdefOKghijk...** and you wish to process data by extracting the string located after **OK**, you will have to specify a string located after **OK**. Otherwise you will be able to work only with the part **abcdefOK**.

Write on a port: adds the action for writing to the designated COM port. Access the settings of the action by clicking ⊞ in order to specify the following elements:

- **Write on a port** : using the list, indicate the COM port on the computer to which you want to send a command
- **Command :** string to send to the designated port
- **Send CRLF after the command**: select this option to send a carriage return at the end of the string *Command*.

Close a port: adds the action for closing the designated COM port.

- **Close a port** : using the list, select the COM port to close



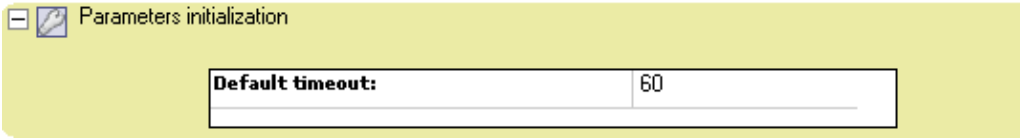
The script generation wizard automatically inserts all actions between the first and last actions, i.e. *Start scenario* and *Stop scenario*.

11.5.1 USING THE COM PORT WIZARD INTERFACE

You can edit the following parameters of COM port actions in the wizard:

- **Think time**: reflection time after request execution *(default : 2 seconds)*
- **Timeout** for request execution *(default: 60 seconds)*
- **Exit Type**: action to take in the event of request unavailability (*continue*, *error*, or *abort*)
- **Error Message**: text of the error message in the event of request unavailability

The **Parameter Initialization** panel offers access to the scenario's default parameter: the default timeout for each action on a COM port.



If you make changes to this section, they will also be applied in the general **Scenario options**.

Le timeout indiqué ici est pris en compte dans les actions qui utilisent le paramètre **Default** pour la valeur de **TimeOut.**

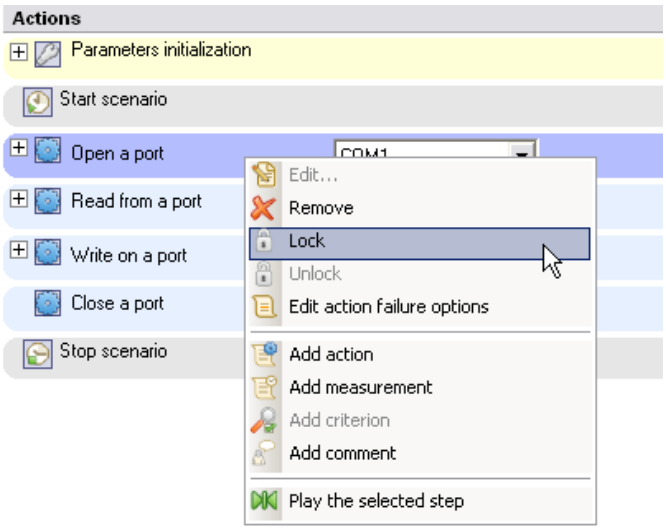
11.5.1.1 FEATURES OF SCENARIO ACTIONS

You can reorganize a scenario's actions by moving them with the mouse. To do so, click on the action you want to move, and drag it to its new position in the script.

Note: if you place an action outside the **Start scenario** and **Stop scenario** limits in the actions grid, the execution time of this action will not be counted in the overall execution time of the transaction.

If you move an action to a spot that is not permitted, the following symbol indicates that you cannot perform this move:  . When it is allowed to place the action where you intend, the following symbol appears:  .

By right-clicking an action in the list, you can choose to **Lock** this action so that it cannot be moved. You can also **Unlock** an action:



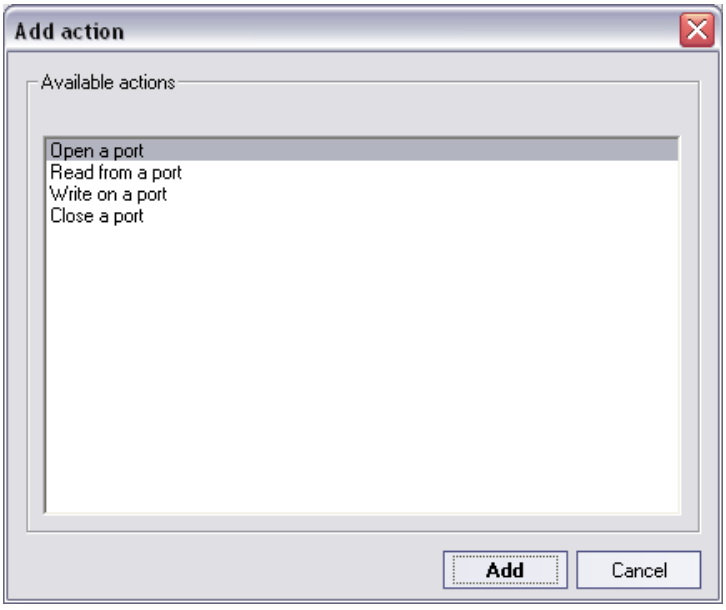
This can be useful when scenarios are very long, so that you can organize it and avoid errors in the order of the series of actions.

11.5.1.2 WIZARD TOOL BAR

In the upper part of the wizard screen, command buttons allow you to edit and run scenario actions (you can also access these actions in the pop-up menu by right-clicking in the wizard panel):

 **Add action**: click to add an action to the list

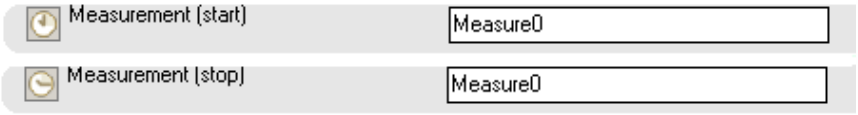
Choose the type of action to add from the selection provided:



 *not available in the COM PORT client environment*

 **Add measurement**: click to add a measurement to the list of actions.

The wizard then generates an automatically indexed measurement name as well as the measurement's start and stop actions:

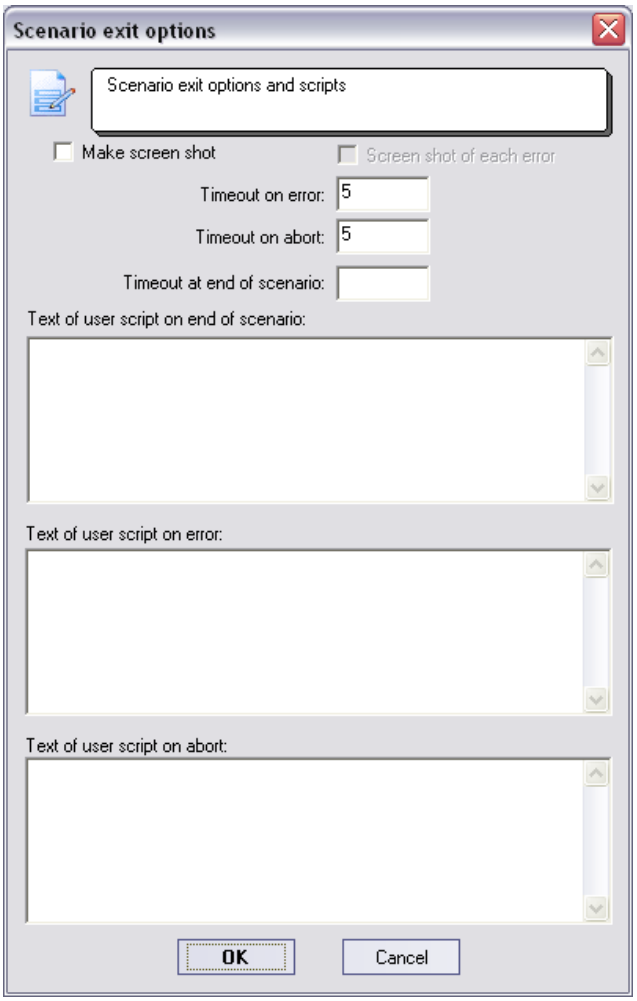


You can rename the measurement by typing a new one in the field; you may also move actions by dragging and dropping them with the mouse.

 **Remove selection**: click to delete the selected objects from the list

 **Edit**: *not available in the COM port client environment*

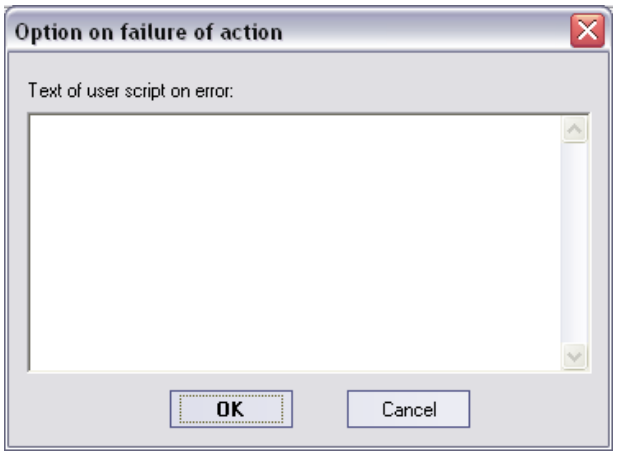
 **Edit scenario exit option**: click to display scenario exit options:



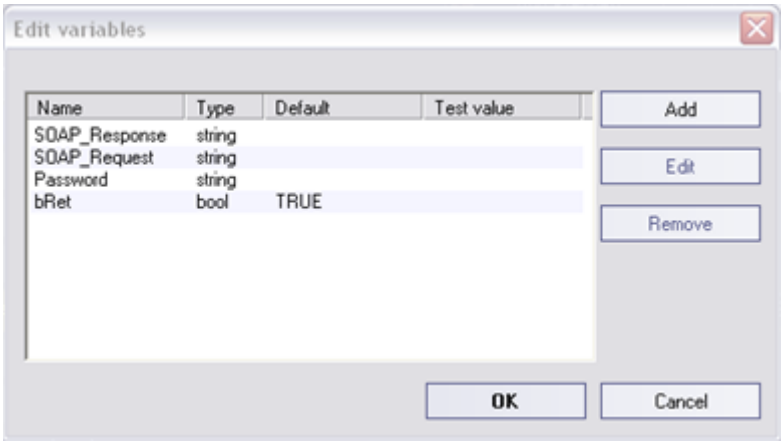
- **Make screenshot**: performs a screenshot for exits on error. In the case of a COM port scenario, this option is not available because there is no visual element directly usable for execution diagnostics for this type of scenario.
- **Screenshot of each error**: select this option if you want to generate a screen shot for each successive error.
- **Pause on error**: timeout (Wait) before exit on error

- **Pause on abort**: timeout (Wait) before exit on abort
- **Pause at end of scenario**: timeout (Wait) before exit on a normal end of scenario
- **Text of user script on end of scenario**: code of the script to run prior to the normal end of scenario
- **Text of user script on error**: code of the script to run prior to the end of a scenario in error
- **Text of user script on abort**: code of the script to run prior to the end of a scenario in abort

 **Edit action failure option**: click to enter the code of a script to run in case of error in the selected step:



 **Edit list of variables**: click to edit the list of configurable variables in the scenario script:

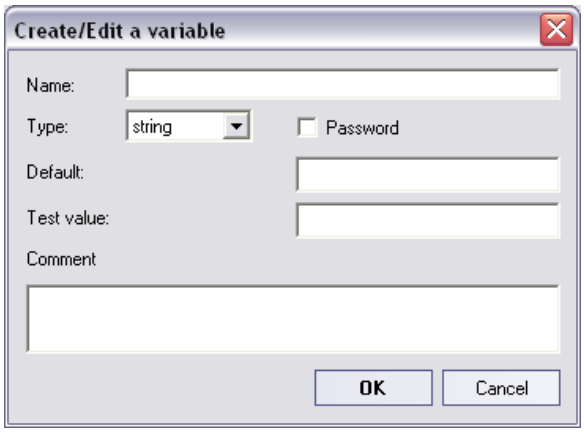


The table shows the variables that are present in the script and provides the following information about each one:

- **Name**: name of the variable
- **Type**: type of variable
- **Default**: default value for this variable (initialization)
- **Test Value**: test value of this variable for running the script in the wizard

The buttons in the interface allow you to act on variables in the script:

- **Add**: click to add a new variable to the list



  - **Name**: name of the new variable
  - **Type**: type of the new variable
  - **Password**: select this option of the variable will contain a password so that it will be encrypted in the script (function GetPassword)
  - **Default**: default value for this variable at the start of the script
  - **Test Value**: test value of this variable for running the script in the wizard
  - **Comment**: comments area to add to the initialization section of the script for this variable.

- **Edit**: to edit the parameters of the selected variable. This option is available only for the variables added to the initial list.
- **Remove**: to delete the selected variable. This option is available only for the variables added to the initial list.

 **Play all steps**: to play all the steps in the scenario

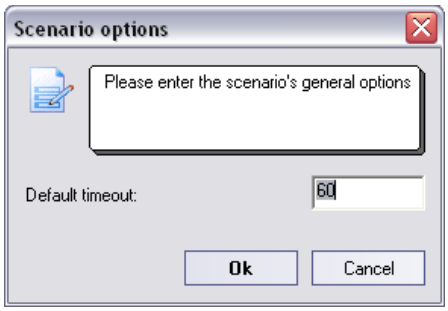
 **Next step**: to run the next step in the scenario

 **Stop execution**: to stop running the scenario

 **Play the selected step**: to run only the steps you selected in the assistant

11.5.2 CONFIGURING THE SCENARIO'S GENERAL OPTIONS

The **Scenario options** fields offer access to the interface for setting the scenario's general options.



Please supply the following information:

- **Default timeout**: default timeout is 60 seconds for each scenario action in the wizard table that uses the default value for the timeout

11.6STRUCTURE OF COM PORT GENERATED SCRIPTS

COM port scenario scripts that are generated by the wizard have a conventional structure. Each part of the script is delimited by a comments section. As with any Newtest scenario, a script header opens the script and a script footer appears at the end:

<script header>

<opening of COM port >

<requests to one or more COM ports>

<closing of COM port>

<script footer>

11.6.1 AVAILABLE FUNCTIONS

Function	Description
OpenPort	Opens a COM port
ClosePort	Closes a COM port
WritePort	Sends traffic to a COM port
WaitStrPort	Waits for a string in the incoming traffic of a COM port
SmsSend	Sends an SMS (text message)
SmsSearch	Searches for an SMS (text message) having the specified criteria. Supplies an index.
SmsDelete	Deletes an SMS (text message) from an index (sent by SMSsearch)
SmsGetInfo	Finds data in an SMS (text message)
GetMobileSN	Retrieves the series number (SN) of the mobile phone connected to the COM port
SetSndLineFeedOnly	Modifies the characteristics of the sending of a carriage return after issuing a command
SmsDeleteAll	Deletes all SMS (text messages) from a SIM card

11.6.2 SAMPLE COM PORT SCRIPT

```
/// Scenario: Portcom
/// Author: Patrick
/// Last generation time: 02/24/09 13:49:49
///

//@DECLARATION
DIM Response AS STRING
DIM ExtractedStr AS STRING
DIM TmpInt AS INT
DIM bRet AS BOOL
//@

//@BEGIN
// Begin program
   bRet = TRUE
   If bRet = TRUE Then
      //
      // Init PORT COM
      //

      bRet = OpenPort("COM1",14400,8,0,1,1)

   ENDIF
   If bRet = FALSE Then
      RET ErrorLine("Selector")
      GOTO ERROR
   ENDIF
   //@

   //@ACTION id=BX09ZJBDnJ6cAIJaWIBPYgAj
      bRet = TRUE
      wait(2)
      RET SetScenarioStartTime()
      If bRet = TRUE Then
         //
         // OpenPort Action
         //
         bRet = OpenPort("COM1",14400,8,0,1,1)
      ENDIF
      If bRet = FALSE Then
         RET ErrorLine("Error0")
         GOTO ERROR
      ENDIF
   //@

   //@ACTION id=Cy4uJQBGUkrYDegDuyC0q9WO
      bRet = TRUE
      wait(2)
      If bRet = TRUE Then
         //
         // Read
         //
         bRet = WaitStrPort("COM1","OK",Response, 5)
         SetArgString("", Response)

      ENDIF
```

```
    If bRet = FALSE Then
      RET ErrorLine("Error1")
      GOTO ERROR
    ENDIF
  //@

  //@ACTION id=DqCHU5BMn8brDn6jS6DnjL8z
    bRet = TRUE
    wait(2)
    If bRet = TRUE Then
      bRet = WritePort("COM1","",5,false)

    ENDIF
    If bRet = FALSE Then
      RET ErrorLine("Error2")
      GOTO ERROR
    ENDIF
  //@

  //@ACTION id=CP5dmQBlackLCEQFG$DV4O46
    bRet = TRUE
    wait(2)
    If bRet = TRUE Then
      //
      // Close COM1 port
      //
      ClosePort("COM1")
    ENDIF
    If bRet = FALSE Then
      RET ErrorLine("Error3")
      GOTO ERROR
    ENDIF
    RET SetScenarioStopTime()
  //@

  //@END
  // Roolback Selector
  //
  // Close COM1 port
  //
  ClosePort("COM1")
  Wait(5)
// END program
END
//@

//@ERROR
ERROR:
  // Rollback Selector
  //
  // Close COM1 port
  //
  ClosePort("COM1")
  HardCopy(0)
  Wait(5)
END
//@
```

```
//@CANCEL
CANCEL:
   // Rollback Selector
   //
   // Close COM1 port
   //
   ClosePort("COM1")
   Wait(5)
   RET SetScenarioCanceled()
END
//@
```

12 SCENARIOS BUILT WITH THE SOAP CLIENT ENVIRONMENT

12.1 INTRODUCTION

SOAP (Simple Object Access Protocol) is a standard developed by the W3C (World Wide Web Consortium). This protocol relies entirely on XML in defining the general structure of the messages exchanged between web services components.

As a supplement to information given on the general method of creating scenarios, this section discusses the specifics of implementing SOAP requests using the scenario wizard.

12.2 PREPARATION

Microsoft Internet Explorer 4.0 or higher must be installed.

If you are using a modem to connect to the internet, Remote Access Service (RAS) must be installed and configured.

12.3 TECHNICAL PRINCIPLES OF THE SOLUTION

The SOAP client environment generates SOAP requests to web services servers which host the applications to be monitored.

The principle is to use the configuration elements described in standardized WSDL files to select the SOAP methods to test, configure their variables, and select the criterion to validate their operation (expected SOAP response).

If the use of service description files (WSDL) is not possible, advanced mode allows the entire request and its parameters to be entered into an editor which presents the structure of the request in XML format.

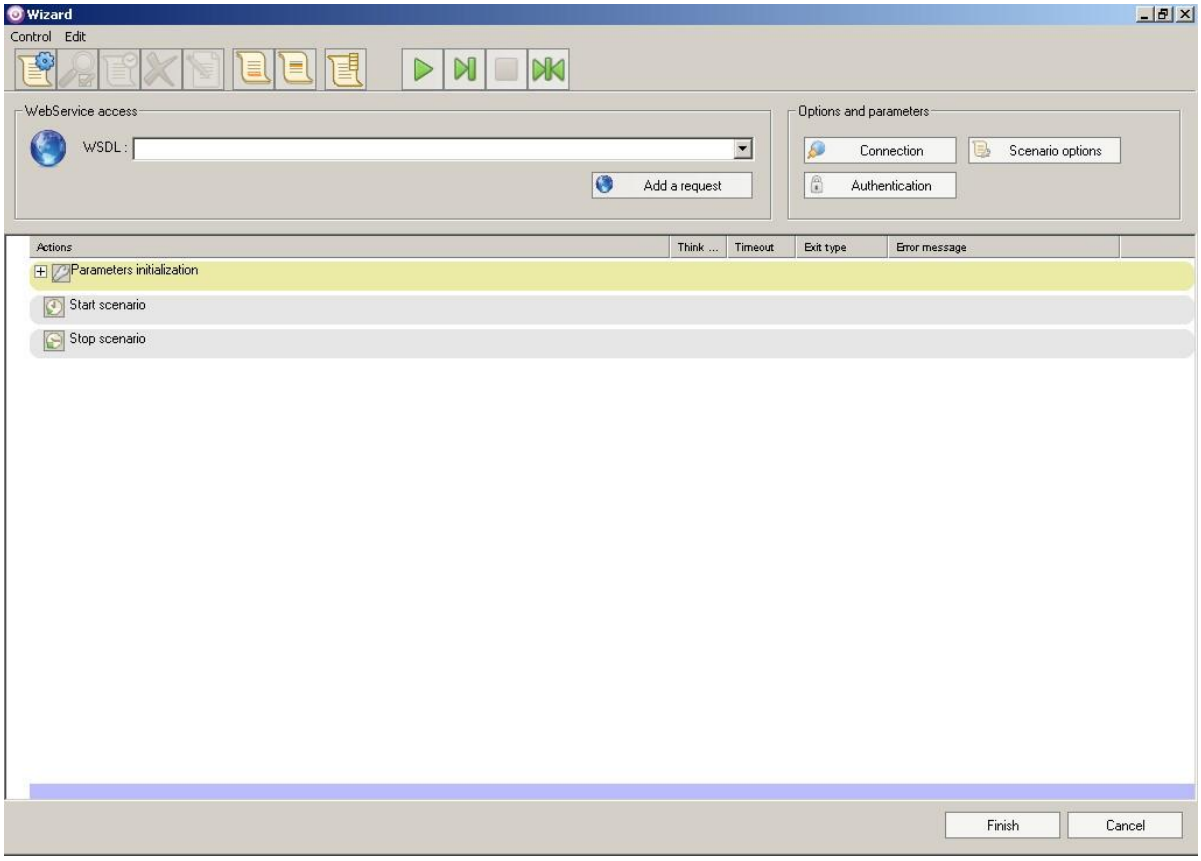
The wizard also provides for the extraction of data in the XML structure of the SOAP response in order to store it/process it in the script.

Running a series of SOAP requests alongside the processing of data retrieved makes it possible to build complex business transaction with a view to checking that the business service is operational.

12.4 PROGRAMMING THE SOAP SCENARIO

12.4.1 BUILDING A SOAP SCENARIO WITH THE SCRIPT GENERATION WIZARD

The main window of the SOAP client environment wizard allows you to configure the requests that you wish to run, as well as the parameters and options of the scenario.



The upper part of the window, “WebService access” is where you provide the URL of the WSDL (Web Services Description Language) document which describes the contents of the web service.

You can:

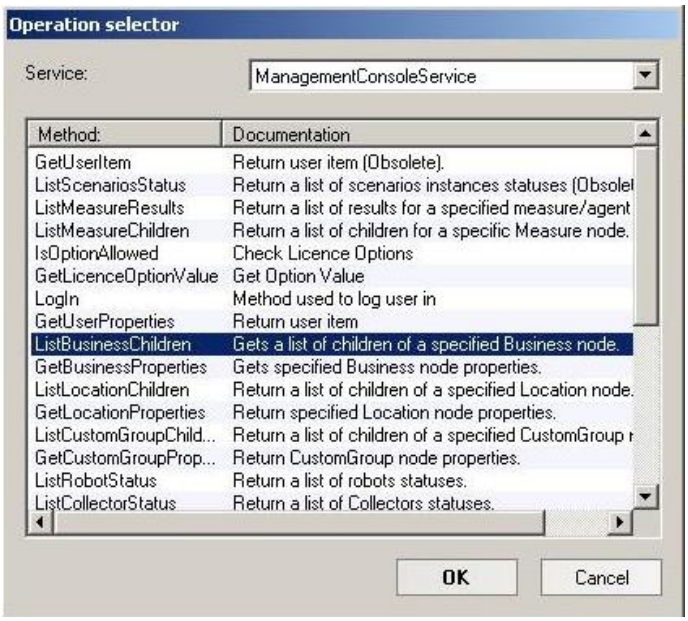
- enter the URL to load the WSDL document
- select a URL in the drop-down list (keeps the last 10 URLs entered in the scenario wizard)

12.4.2 CREATING A SOAP REQUEST

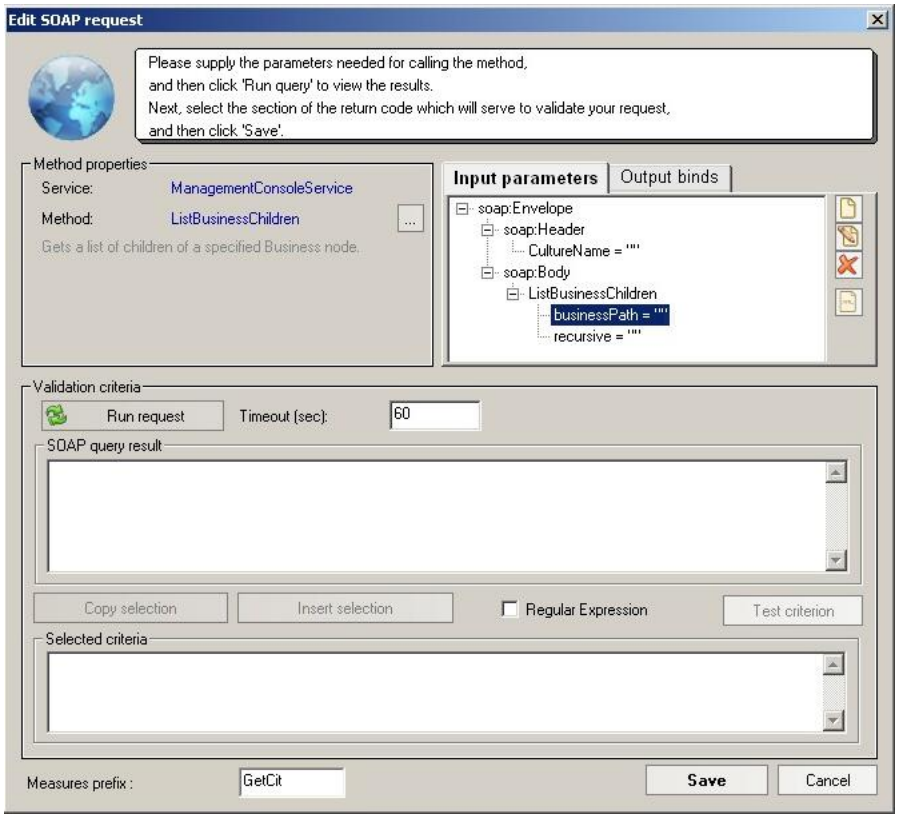
After supplying the URL of the WSDL document, click the button **Add a request** to view the list of available services and methods.

Select the service from the drop-down list to display the list of methods.

Next, select the method to use and then click **OK**:

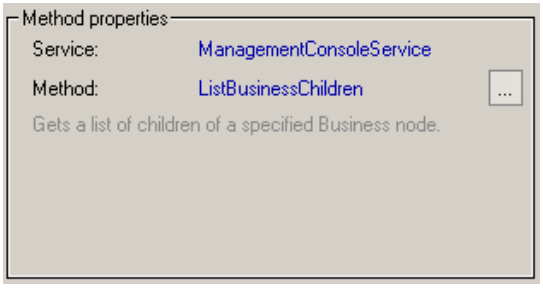


The following window allows you to provide the information the wizard needs to build the request:

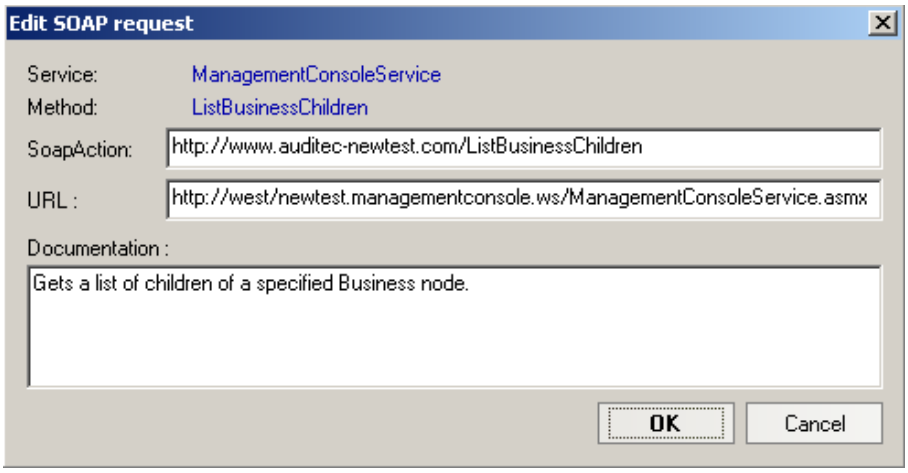


- Method properties:

This area shows the name of the selected service and method.

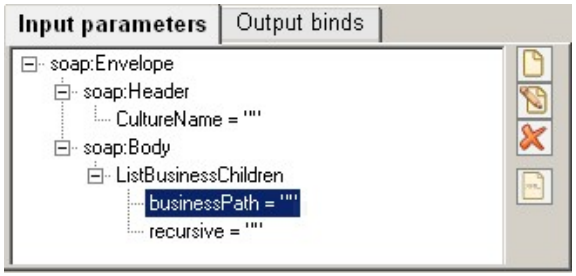


You can edit the general settings of the SOAP request by clicking the ... button.



- Input parameters tab

The tree shows the input parameters:



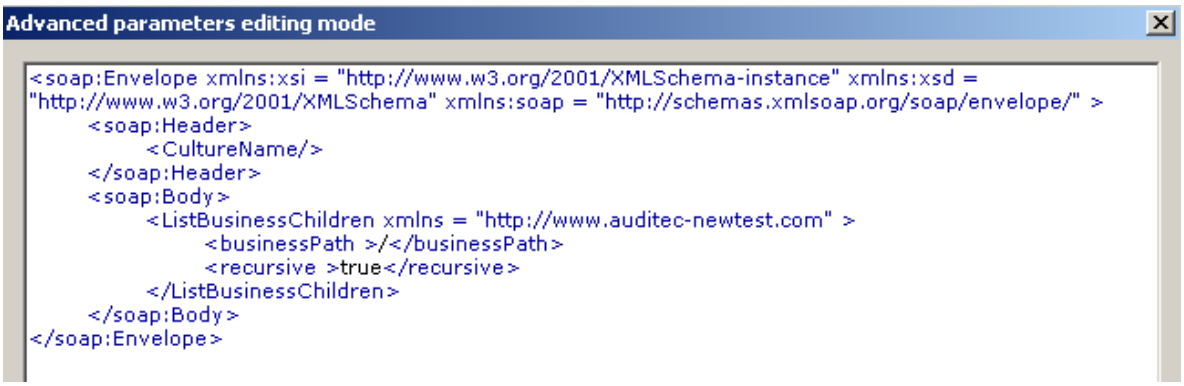
You can edit the list of parameters using the following command buttons:

 **Add a parameter** to add a parameter below the selected object.

 **Edit the selected parameter** to enter the name and value of the selected parameter.

 **Remove the selected parameter** to delete the selected parameter.

 **Advanced parameter editing mode** to edit the XML code of the input parameters (advanced mode).

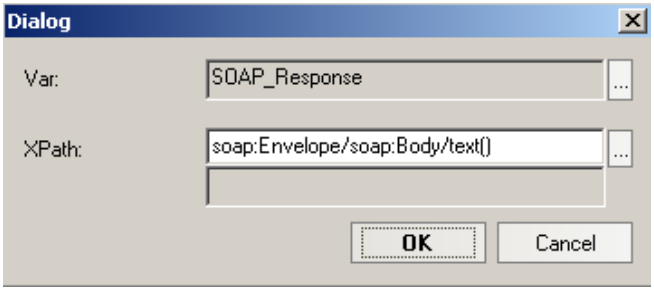


- Output binds tab

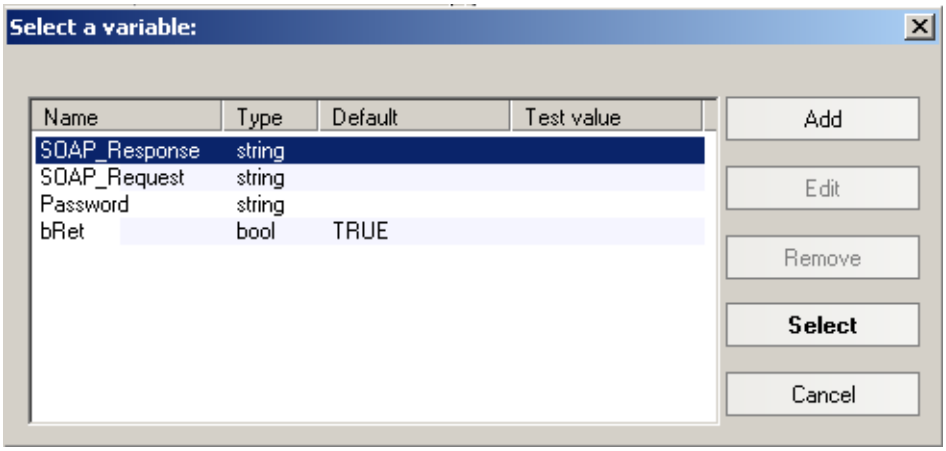
The list allows you to define the output binds.

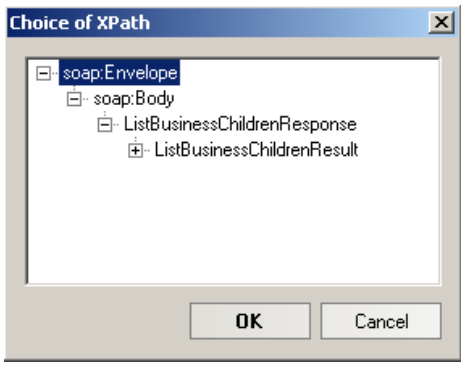
You can edit the list of binds using the following command buttons:

 **Add bind** to add a bind to the list. Next, enter the name of the variable and its path in XML structure:



The  buttons lead to a window for selecting the variable and XML path:





Edit bind to edit the selected bind.

Delete bind to delete the selected bind.

- Validation criteria

This area allows you to set validation criteria for the result of the SOAP request.

Click the **Run request** button to execute the request and obtain its result.

You may set a timeout for the request *(in seconds)*.

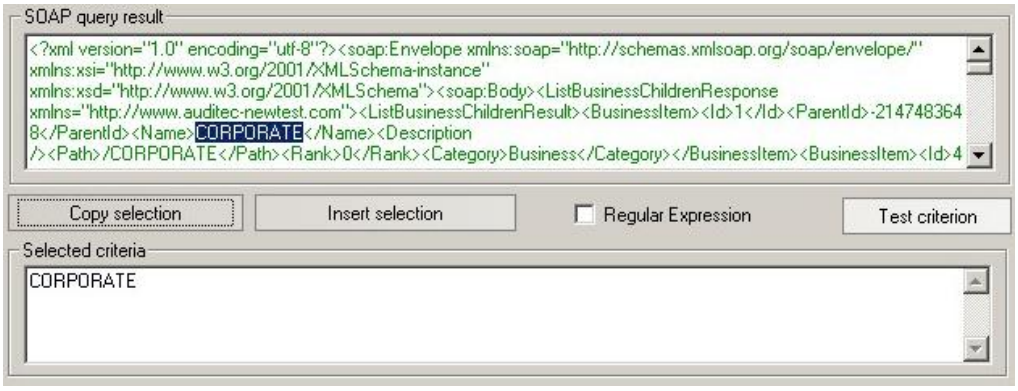
The result of the SOAP request appears in the **SOAP query result** pane.

Using the following command buttons, you can set validation criteria based on request results:

Copy selection to copy the text selected in the criterion text box.

Insert selection to insert the selected text into the criterion text box.

You can enable option **Regular Expression** to use regular expressions in the criterion text box *(recognized syntax detailed in the appendix)*.

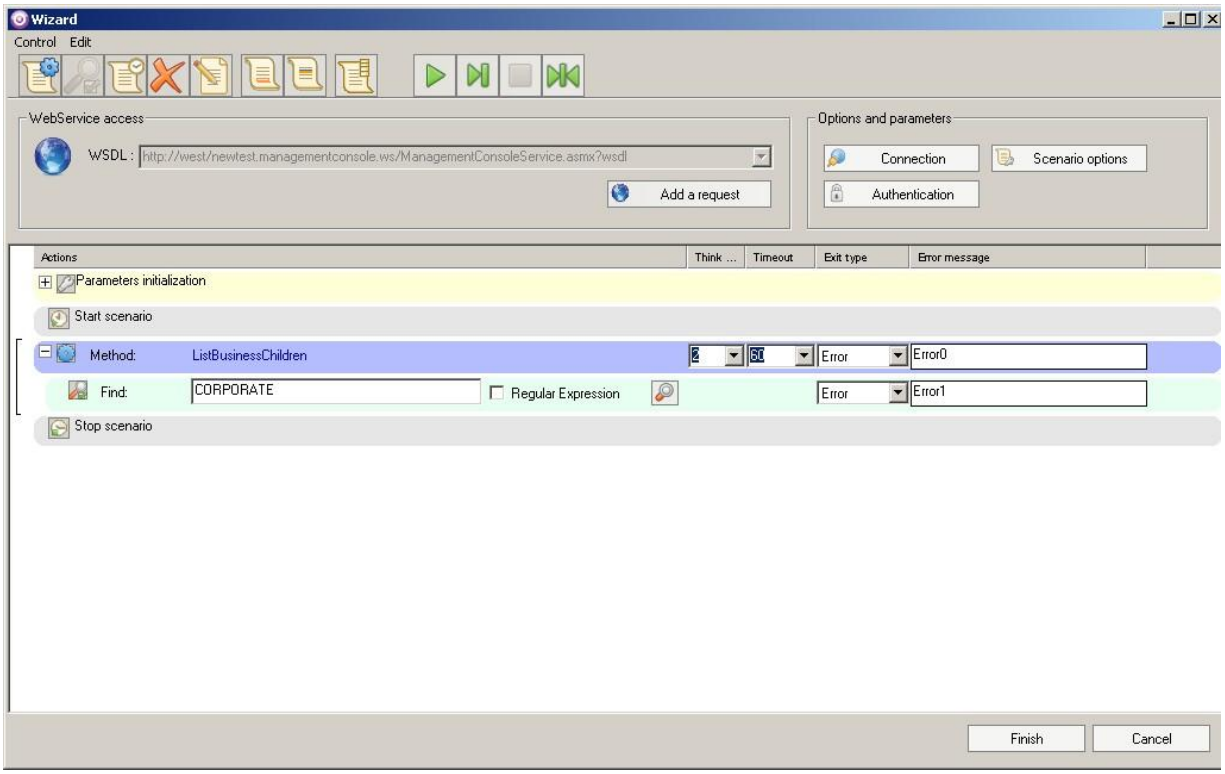


Click the button **Test criterion** to check for the presence of the validation criterion.

- **Measurement prefix**: allows you to set a prefix for the measurements made using this request in the scenario. A prefix is indispensable for identifying the measurements of a request uniquely with respect to those of other requests and transmitting the information in an identifiable way to NMC.
- The prefix you indicate will be added, followed by an underscore (_), to the name of each measurement. The maximum length of the prefix is 6 characters.

In the Edit SOAP request window, click **Save** to save the request.

The scenario wizard then inserts the request between the actions **Start scenario** and **Stop scenario**, as below:



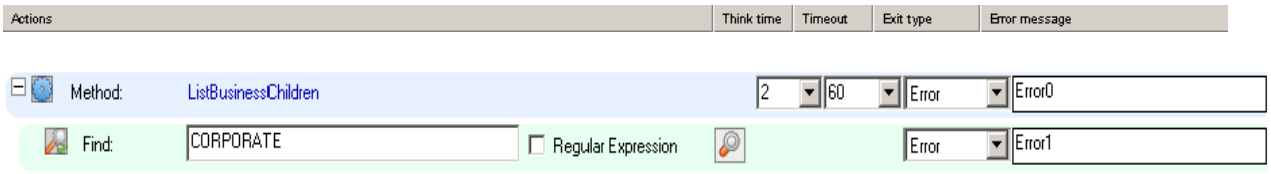
12.4.3 USING THE SOAP WIZARD INTERFACE

You can modify SOAP request parameters in the wizard:

- **Think time** : reflection time after request execution *(default value: 2 seconds)*
- **Timeout** of the execution of the request *(default: 10 seconds)*
- **Exit Type** : action in the event of request unavailability *(continue, error, abort)*
- **Error Message** : text of the error message in the event of request unavailability

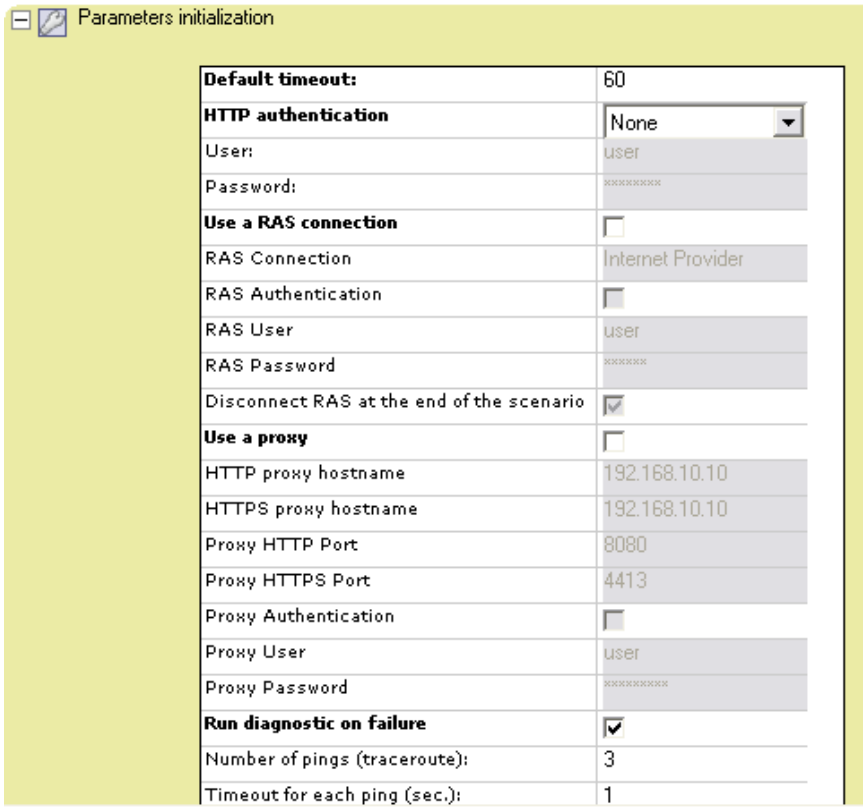
The following parameters may be modified in the criteria rows:

- text of the validation criterion (with the possibility of enabling the regular expression option)
- SOAP request execution timeout (default value: 60 seconds)
- action in the event of request unavailability (continue, error, abort)
- text of the error message in the event of request unavailability



Authentication: management of .p12 certificates

The **Parameter initialization** panel offers access to the scenario's main default parameters, as well as connection and authentication options.



If you make changes to this section, they will also be applied in **Connection** settings or **Scenario options**, depending on the case.

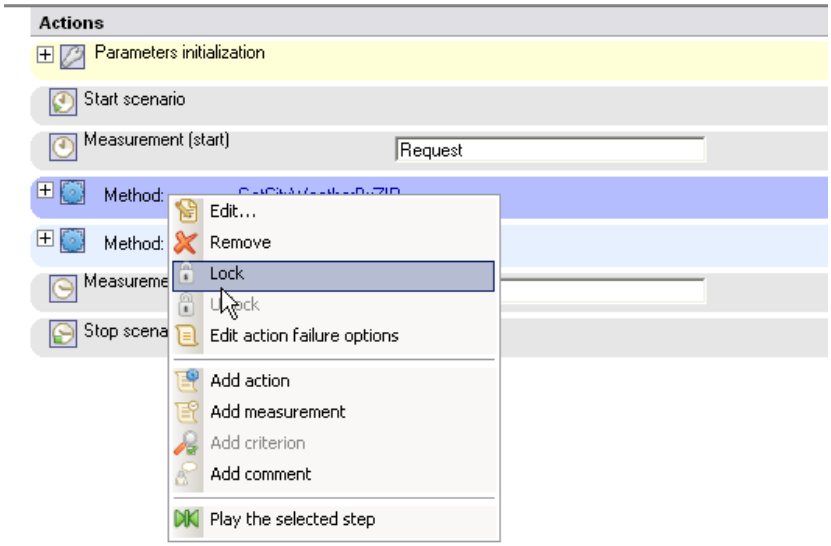
12.4.3.1 FEATURES OF SCENARIO ACTIONS

You can reorganize a scenario's actions by moving them with the mouse. To do so, click on the action you want to move, and drag it to its new position in the script.

Note: if you place an action outside the **Start scenario** and **Stop scenario** limits in the actions grid, the execution time of this action will not be counted in the overall execution time of the transaction.

If you move an action to a spot that is not permitted, the following symbol indicates that you cannot perform this move:  . When it is allowed to place the action where you intend, the following symbol appears:  .

By right-clicking an action in the list, you can choose to **Lock** this action so that it cannot be moved. You can also **Unlock** an action:



This can be useful when scenarios are very long, so that you can organize it and avoid errors in the order of the series of actions.

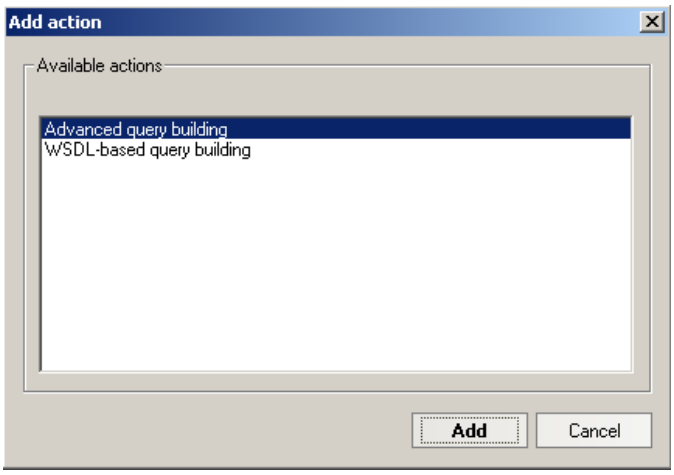
12.4.3.2 WIZARD TOOL BAR

Command buttons in the upper part of the wizard window allow you to edit and run scenario requests:

 **Add action** is for adding a request to the list.

Select a type of request definition wizard:

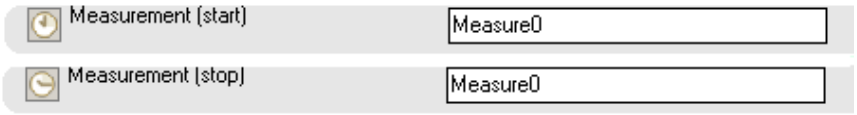
- **Advanced query building**: request parameters are set manually.
- **WSDL-based query building**: allows you to build the query from the list of methods supplied in the WSDL file.



not available in the SOAP client environment

Add measurement to add a measurement to the list of actions.

The wizard then generates an automatically indexed measurement name and the measurement's start and stop actions:

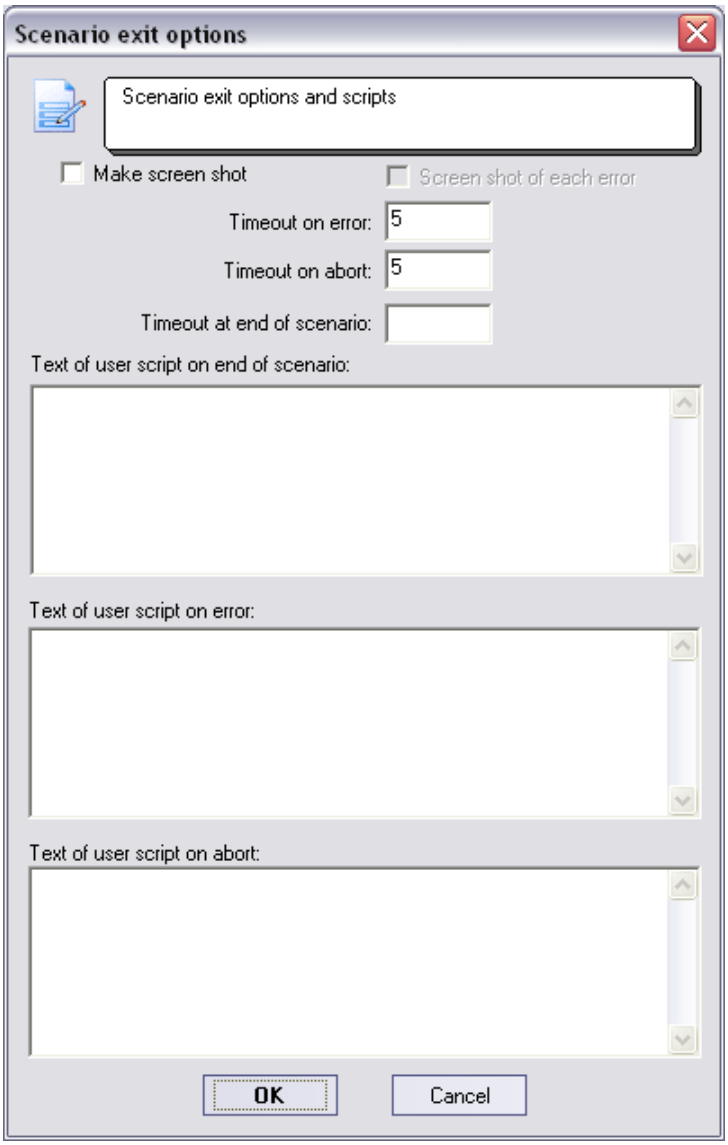


You can rename the measurement in the text box, as well as move the actions by dragging and dropping with the mouse.

Remove selection to delete the selected items from the list.

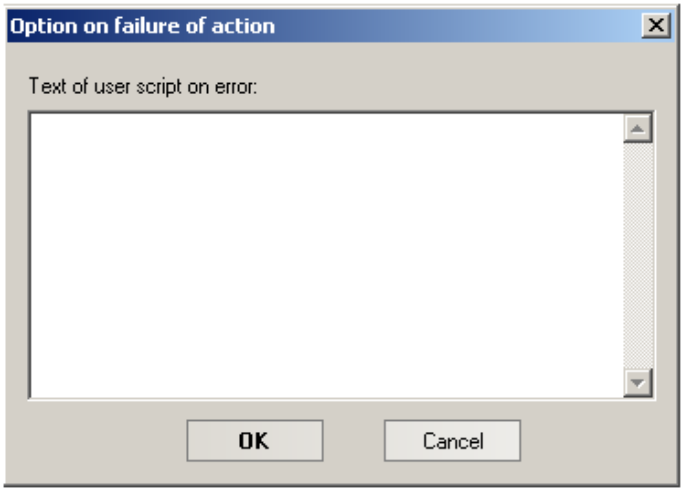
Edit to edit the selected item.

Edit scenario exit option: click to display scenario exit options:

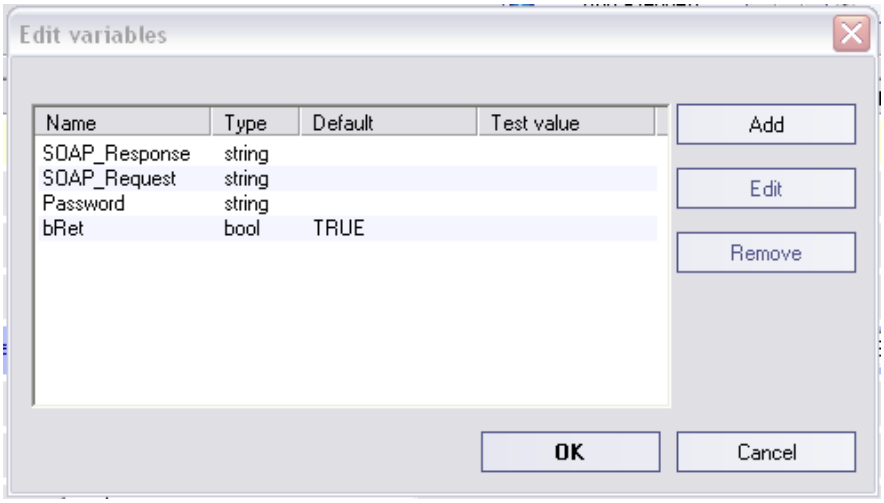


- **Make screenshot**: performs a screenshot for exits on error. In the case of a SOAP protocol scenario, this option is not available because there is no visual element directly usable for execution diagnostics for this type of scenario.
- **Screenshot of each error**: select this option if you want to generate a screen shot for each successive error.
- **Pause on error**: timeout (Wait) before exit on error
- **Pause on abort**: timeout (Wait) before exit on abort
- **Pause at end of scenario**: timeout (Wait) before exit on a normal end of scenario
- **Text of user script on end of scenario**: code of the script to run prior to the normal end of scenario
- **Text of user script on error**: code of the script to run prior to the end of a scenario in error
- **Text of user script on abort**: code of the script to run prior to the end of a scenario in abort

Edit action failure option to enter an error text to display in the event the step becomes unavailable.



Edit list of variables: click to edit the list of configurable variables in the scenario script

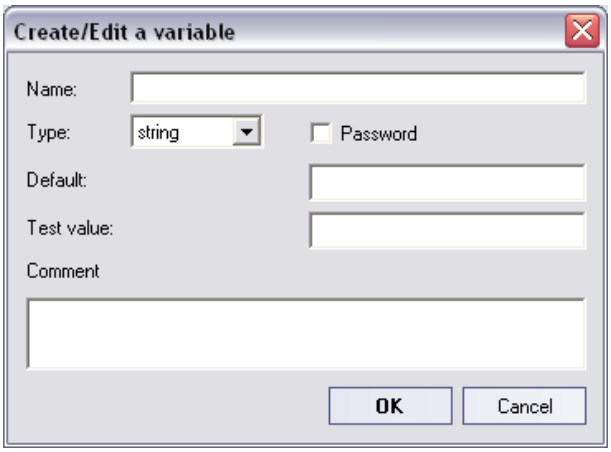


The table shows the variables that are present in the script and provides the following information about each one:

- **Name**: name of the variable
- **Type**: type of variable
- **Default**: default value for this variable (initialization)
- **Test Value**: test value of this variable for running the script in the wizard

The buttons in the interface allow you to act on variables in the script:

- **Add**: click to add a new variable to the list



- **Name**: name of the new variable
- **Type**: type of the new variable
- **Password**: select this option of the variable will contain a password so that it will be encrypted in the script (function GetPassword)
- **Default**: default value for this variable at the start of the script
- **Test Value**: test value of this variable for running the script in the wizard
- **Comment**: comments area to add to the initialization section of the script for this variable.

- **Edit**: to edit the parameters of the selected variable. This option is available only for the variables added to the initial list.
- **Remove**: to delete the selected variable. This option is available only for the variables added to the initial list.

 **Play all steps** to play all of the scenario's steps

 **Next step** to play the next step in the scenario

 **Stop execution** to interrupt the scenario's execution

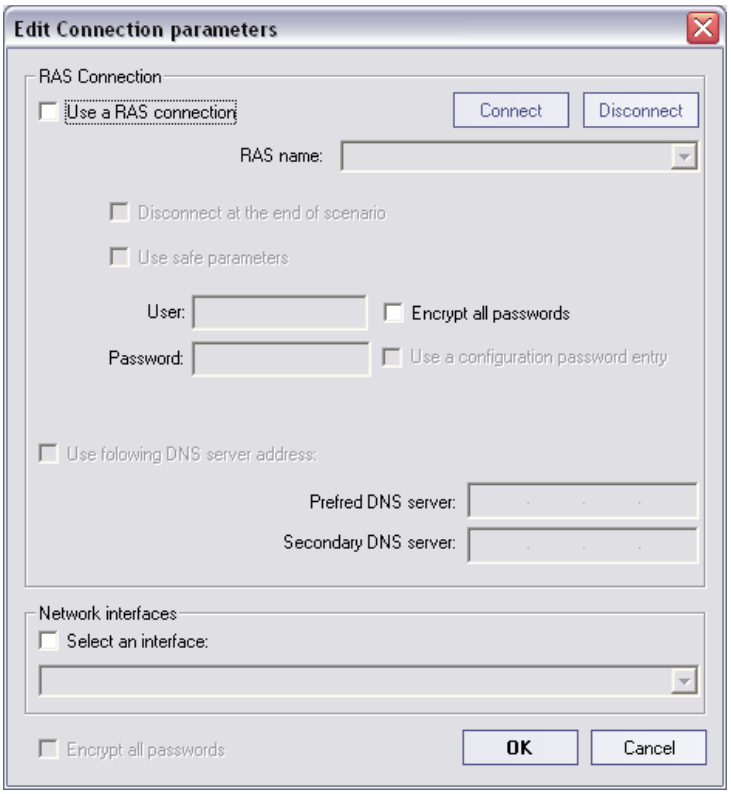
 **Play the selected step** to run only the steps selected in the wizard

12.4.4 CONFIGURING THE COMMUNICATION MODE

Click the button **Connection settings** to set the scenario's communication mode.

This page determines whether the scenario is designed to function in a local area network (LAN) or relies on the opening of a remote connection by modem (RAS).

You may specify the use of primary and secondary DNS servers if you are using RAS, and indicate the network interface to use:



Specify the following:

- for RAS connections
  - **Use a RAS connection** to enable RAS connections.

    Indicate the name of the RAS entry defined in the Windows control panel (the dropdown list contains the entries defined).
  - **Disconnect at the end of scenario** to force RAS disconnect at the end of the scenario run.
  - **Use safe parameters** to require a user login and password for secure access.
  - **Encrypt all passwords** to encrypt passwords in the script.
  - **Use a configuration password entry** to use the password defined on the Password tab page of Newtest Transaction Builder.
  - **Use following DNS server address**: select this option to specify the DNS servers to use when the RAS connection has been established.
  - **Preferred DNS server** / **Secondary DNS server**: enter the address or name of the primary and secondary DNS servers to take into account
- Network interface
  - **Select an interface**: select this option to specify the network card that the scenario will use (the drop-down list shows all the cards installed on the system)

Measurements associated with RAS connections

Description of response time measurements (1/100th of a second)

Measurement	Description
RasConnection	Time it takes to connect to the RAS server (to the 1/100th of a second)
Authenticate	RAS authentication time (to the 1/100th of a second)
SetIPCP	IP address attribution time (to the 1/100th of a second)

Description of numeric measurements

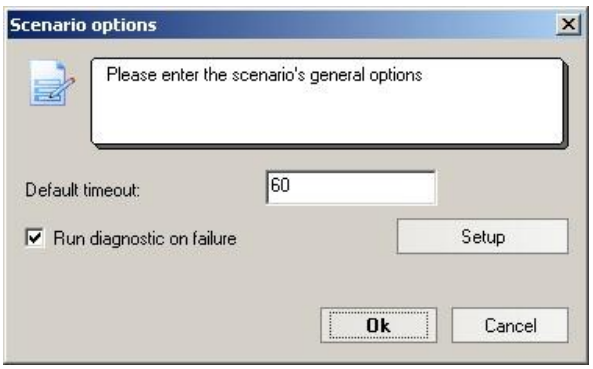
Measurement	Description
RasCode	Newtest RAS error code
RasCodeEx	RAS raw error code
RasSpeed	Negotiated speed (bps)

Description of string measurement

Measurement	Description
RasErrorString	RAS error label

12.4.5 CONFIGURING THE SCENARIO'S GENERAL OPTIONS

The **Scenario options** fields offer access to the interface for setting the scenario's general options.



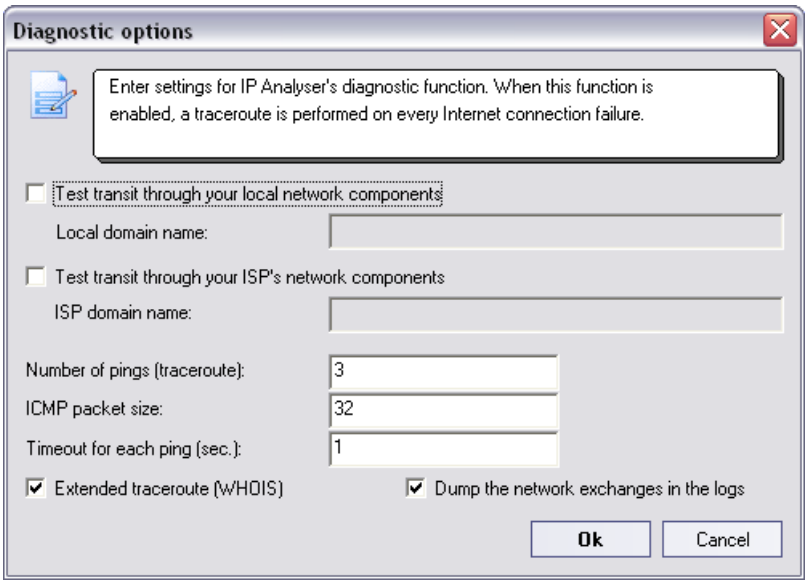
Specify the following:

- **Default timeout**: pre-set default timeout is 60 seconds.
- **Run diagnostic on failure**: select this option to set diagnostic options.

Click **Setup**:

- **Test transit through your local network components**: select this option to check whether the root of the anomaly lies in one of the local network components.
- **Local domain name**: enter the domain name of the local network.

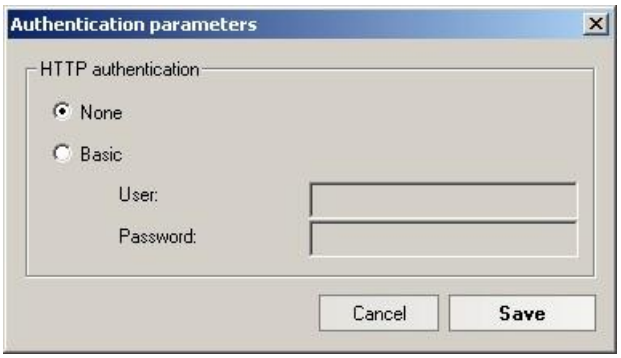
- **Test transit through your ISP's network components**: select this option to check whether the root of the anomaly lies in one of the components of your internet provider's network.
- **ISP domain name**: enter your internet provider's domain name.
- **Number of pings (traceroute)**: specify the number of pings that the diagnostic function should send when it receives an analyze traceroute command.
- **ICMP packet size**: specify the size of the packet transmitted by the pings *(default is 32 bytes)*
- **Timeout for each ping (sec.)**: indicate the timeout for pings *(default is 1000 ms)*
- **Extended traceroute (WHOIS)**: enables extended traceroute (identifies devices transited by the traceroute command)
- **Dump the network exchanges in the logs**: select this option to record the log of TCP packets upon each launch of the request (advanced analysis). These appear in blue in the NTBR message area and are recorded in the SCHEDULE.TRC log file on robots.



12.4.6 CONFIGURING AUTHENTICATION PARAMETERS

This page allows you to set authentication parameters for the HTTP connection:

- **None**: no HTTP authentication.
- **Basic**: enter the user login and password to use for HTTP authentication.



Note: authentication using a client certificate is also possible (cer, pfx or p12; Java Keystore is not supported).

This type of authentication can be configured using the instruction SetOption and the variable "HTTP_CLIENT_CERTIFICATE" :

SetOption("HTTP_CLIENT_CERTIFICATE", "c:\test\cert1.p12", 0)

12.4.6.1 GENERATING THE SCENARIO SCRIPT

Click **Finish** to generate the scenario script.

12.5 STRUCTURE OF SOAP GENERATED SCRIPTS

12.5.1 DEFINITION OF PROGRAMMING BLOCKS

The SOAP script generated by the wizard has a structure made up of several blocks:

<comments block>

<variables declaration block>

<request intialization block>

from 1 to n requests:

<SOAP request execution block>

<criteria validation block>

<script footer>

12.5.2 MEASUREMENTS GENERATED

Description of response time orders

Measurement	Description
soap_test	Overall time of the SOAP request (to the 1/100th of a sec.)

Description of numeric orders

Measurement	Description
soap1_dns	DNS resolution time (msec)
soap2_connect	time to connect to server (msec)
soap3_firstbyte	server processing time (msec)
soap4_gettime	download time (msec)
Soap5_getrate	transfer rate (bytes/second)

12.5.3 AVAILABLE FUNCTIONS

FUNCTION	DESCRIPTION
SetOption	sets protocol, RAS connection or diagnostic options
SetSoapXmlParameters	defines the XML parameters of the SOAP request
AddHttpHeader	defines the request's HTTP header
SoapRequest	runs a SOAP request
GetHttpRequestTime	returns the total time of the HTTP request
GetHttpDnsTime	returns the DNS resolution time of the HTTP request
GetHttpConnectTime	returns the time to connect to the HTTP server *(or to the proxy server)*
GetHttpFirstByteTime	returns the time to receive the first byte
GetHttpDownloadTime	returns the time to download the document
GetHttpDownloadRate	returns the transfer rate

12.5.4 SAMPLE GENERATED SCRIPT

```
///
/// Scenario: SOAP
/// Author: admin
/// Last generation time: 10/15/08 10:31:23
///

//@DECLARATION
DIM SOAP_Response AS STRING
DIM SOAP_Request AS STRING
DIM Password AS STRING
DIM bRet AS BOOL
DIM SOAP_Test AS ORDER(TRESP)
DIM SOAP1_Dns AS ORDER(NUM,"msec")
DIM SOAP2_Connect AS ORDER(NUM,"msec")
DIM SOAP3_Firstbyte AS ORDER(NUM,"msec")
DIM SOAP4_GetTime AS ORDER(NUM,"msec")
DIM SOAP5_GetRate AS ORDER(NUM,"byte/s")
//@

//@BEGIN
// Begin program
   bRet = TRUE
   If bRet = TRUE Then
      //
      // Init Soap: script generated using
http://myserver/newtest.managementconsole.ws/ManagementConsoleService.asmx?wsdl
      //
```

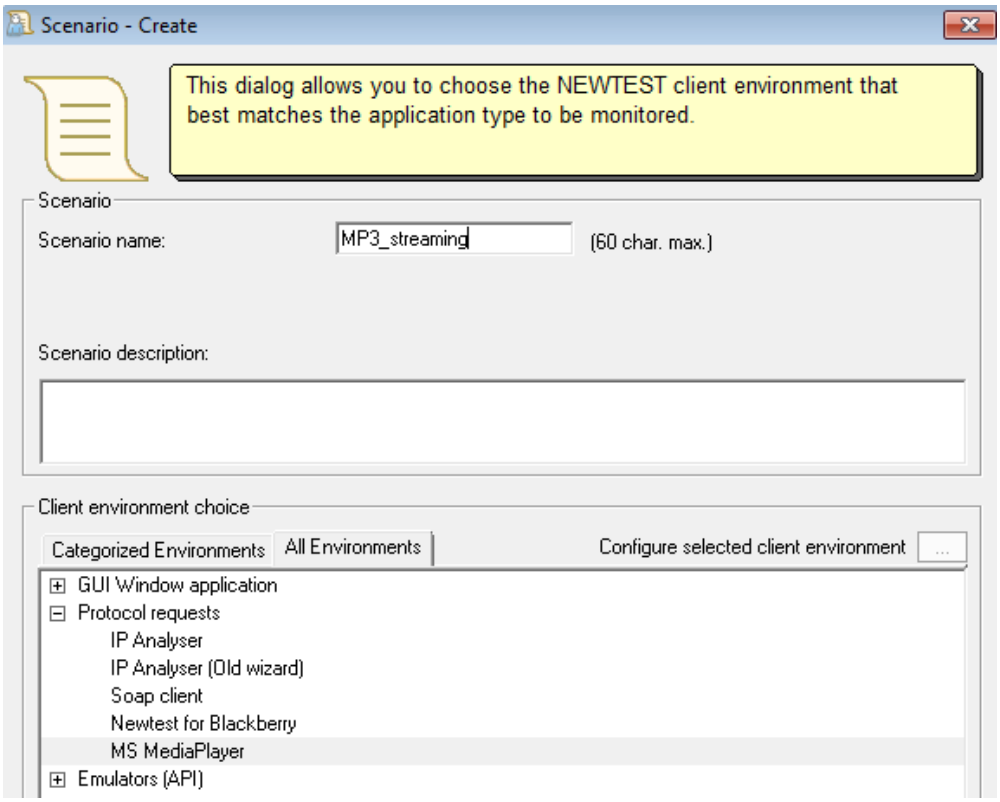
```
        SetOption("DIAG_ENABLED","",1)
        SetOption("DIAG_PINGS_NB","",3)
        SetOption("DIAG_PACKET_SIZE","",32)
        SetOption("DIAG_LOCAL_DOMAIN","",0)
        SetOption("DIAG_ISP_DOMAIN","",0)
        SetOption("DIAG_PING_TIMEOUT","",1)
        SetOption("DIAG_EXTENTED_TRACEROUTE","",1)

        SetOption("LAN_INTERFACE","",0)

    ENDIF
    If bRet = FALSE Then
        RET ErrorLine("Selector")
        GOTO ERROR
    ENDIF
    //@

    //@ACTION id=CeoegnBKIg8iCHJN66Dzz9oi
        bRet = TRUE
        wait(2)
        RET SetScenarioStartTime()
        If bRet = TRUE Then
            //
            // Soap Call :  ListBusinessChildrenSoapIn
            //
            //          Gets a list of children of a specified Business node.
            SOAP_Request = SetSoapXmlParameters("","<soap:Envelope xmlns:xsi =
""http://www.w3.org/2001/XMLSchema-instance"" xmlns:xsd =
""http://www.w3.org/2001/XMLSchema"" xmlns:soap =
""http://schemas.xmlsoap.org/soap/envelope/""
><soap:Header><CultureName/></soap:Header><soap:Body><ListBusinessChildren xmlns =
""http://www.auditec-newtest.com"" ><businessPath >/</businessPath><recursive
>true</recursive></ListBusinessChildren></soap:Body></soap:Envelope>")

SetOption("HTTP_URL","http://myserver/newtest.managementconsole.ws/ManagementConsoleService.a
smx",0)
            SetOption("HTTP_PORT","",80)
            SetOption("HTTP_TIMEOUT","",60000)
            SetOption("HTTP_POST_DATA" , SOAP_Request, 0 )

            AddHttpHeader("soapaction: http://www.auditec-newtest.com/ListBusinessChildren")

            bRet = SoapRequest(SOAP_Response)

            IF bRet = TRUE THEN

             // HTTP Measurements
             // ————————

             GetHttpRequestTime(SOAP_Test)
             RET SOAP_Test
             GetHttpDnsTime(SOAP1_Dns)
             RET SOAP1_Dns
             GetHttpConnectTime(SOAP2_Connect)
             RET SOAP2_Connect
             GetHttpFirstByteTime(SOAP3_Firstbyte)
             RET SOAP3_Firstbyte
             GetHttpDownloadTime(SOAP4_GetTime)
```

```
            RET SOAP4_GetTime
            GetHttpDownloadRate(SOAP5_GetRate)
            RET SOAP5_GetRate
          ENDIF
        ENDIF
        If bRet = FALSE Then
          RET ErrorLine("Error0")
          GOTO ERROR
        ENDIF
    //@

    //@CRITERIA id=BTqUB2BJgU53DvHhemANK$aN
        If bRet = TRUE Then
          // Soap search criteria

          IF Find(SOAP_Response, "CORPORATE")=0 THEN
            bRet = FALSE
          ENDIF

        ENDIF
        If bRet = FALSE Then
          RET ErrorLine("Error1")
          GOTO ERROR
        ENDIF
        RET SetScenarioStopTime()
    //@

    //@END
    Wait(5)
// END program
END
//@

//@ERROR
ERROR:
    Wait(5)
END
//@

//@CANCEL
CANCEL:
    Wait(5)
    RET SetScenarioCanceled()
END
//@
```

13 SCENARIOS BUILT WITH THE MS MEDIA PLAYER CLIENT ENVIRONMENT

For streaming audio or video you can use Windows Media Player to carry out quality measurements with the Newtest robot.

To do so, open the **MS Media Player** wizard under the category “All environments”:



After you test the URL that you provide, the script is generated in the following form:

```
//
*************************************************************************
*****
// * This script was generated by the NEWTEST - MEDIAPLAYER  Script Generation Wizard
// * 17/02/2014 11:54:33
//
*************************************************************************
*****


DIM MediaP_Play_Result AS ORDER(NUM)
DIM Ret_Play AS INT
DIM MediaP_lost AS ORDER(NUM)
DIM MediaP_received AS ORDER(NUM)
DIM MediaP_maxbw AS ORDER(NUM)
DIM MediaP_minbw AS ORDER(NUM)
DIM MediaP_avebw AS ORDER(NUM)
DIM MediaP_startbuf AS ORDER(NUM)
DIM MediaP_rebuf AS ORDER(NUM)
DIM MediaP_maxbitrate AS ORDER(NUM)
```

```
DIM MediaP_avframerate AS ORDER(NUM)


Ret_Play = PlayMediaPlayer("http://stream-hautdebit.frequence3.net:8000",10)
RET LOGNUM(MediaP_Play_Result, Ret_Play)

GetMdpLost(MediaP_lost)
RET MediaP_lost

GetMdpReceived(MediaP_received)
RET MediaP_received

GetMdpMaxBandWidth(MediaP_maxbw)
RET MediaP_maxbw

GetMdpMinBandWidth(MediaP_minbw)
RET MediaP_minbw

GetMdpAverageBandWidth(MediaP_avebw)
RET MediaP_avebw

GetMdpStartBufferDuration(MediaP_startbuf)
RET MediaP_startbuf

GetMdpReBuffering(MediaP_rebuf)
RET MediaP_rebuf

GetMdpMaxBitRate(MediaP_maxbitrate)
RET MediaP_maxbitrate

GetMdpAverageFrameRate(MediaP_avframerate)
RET MediaP_avframerate


LABEL_END:
END
```

14 SCENARIOS BUILT WITH THE BLACKBERRY CLIENT ENVIRONMENT

14.1 INTRODUCTION

The Newtest for BlackBerry solution is designed to check the availability and performance of BlackBerry services with an end-to-end view (from the device itself).

14.2 USAGE CONTEXT

14.2.1 ENVIRONMENT

BlackBerry functionalities have been validated for Newtest version 7.4.1 or higher.

The BBAgent application is installed on the BlackBerry agent. It interprets commands and extracts data from the BlackBerry.

Please refer to the agent's installation manual to implement the agent.

14.2.2 BASICS

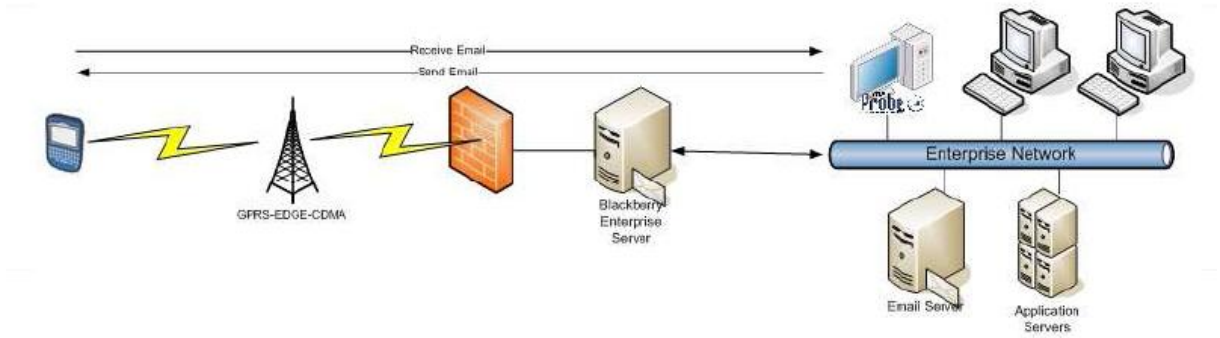
The principle of the transaction is to send an SMTP message in text format (not HTML or multipart) to the address associated with the test BlackBerry and to use the elements received in the reply.

The general operation is as follows: when the agent running on the BlackBerry is launched, it will await incoming mails and react if the mail received fulfills certain criteria. In response to those requests, it will send a mail to the address indicated in the field RESPONDTO or automatically reply to the request issuer.

Availability and response times

A Newtest robot sends "encoded" messages on a regular basis to one or more test-dedicated BlackBerry devices and to other BlackBerry terminals in the hands of testers. A specific program is implemented in each of the target BlackBerry devices. When it receives an e-mail, each BlackBerry returns a reply that makes it possible to check the general operation of the service (in accordance with the command received), transmission times, and the elements that adversely affect service.

This offer is based on the IP Analyser client environment in conjunction with the BBAgent software installed on each test BlackBerry.



14.3 PREREQUISITES

The examples given must be generated via Newtest Transaction Builder 7.4.1 and deployed to Newtest robots of the same or more recent version. In addition, the programs and methods must be implanted on the company's BlackBerry.

Implementation of Newtest BlackBerry methods requires BlackBerry OS version 4.2.1 or higher.

14.4 NEWTEST / BLACKBERRY METHODS

14.4.1 DESCRIPTION

The BBAgent software running on the BlackBerry analyzes the mail received and performs a number of tasks if certain criteria are met.

On the Newtest robot, an IP Analyser scenario using SMTP and POP3/IMAP4 requests provides for sending to / receiving from BlackBerry devices that are equipped with the module.

The main criterion for filtering messages is the message subject line. BBAgent processes only those incoming messages with [BBAGENT TEST] or [BBAGENT REQUEST] in the subject line. These messages are analyzed and interpreted, and then a reply is sent. Following processing, the received message is deleted to unclutter the list of messages on the BlackBerry.

When using BBAGENT REQUEST, the message to send must be encoded by the command BBAgentEncode. The reply from the BlackBerry is likewise encoded, and requires the command BBAgentDecode to be read.

You must add encoding and decoding commands to IP Analyser "BlackBerry" scripts manually. The following paragraphs provide details about these commands.

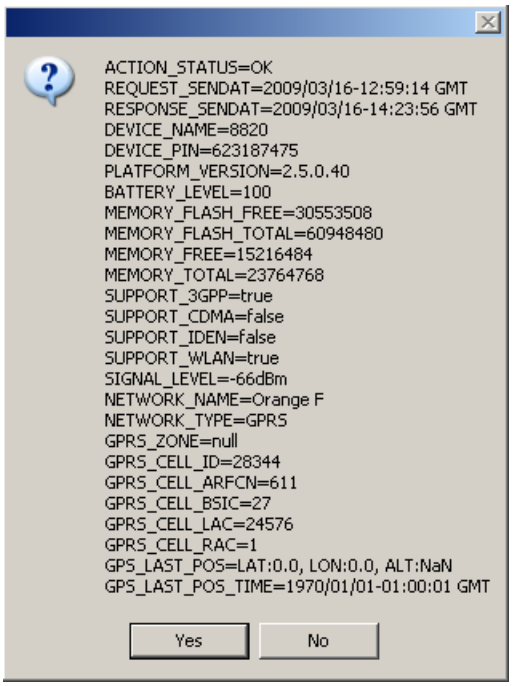
The course of action consists of systematically (unless otherwise indicated) encoding the message sent with IP Analyser, and decoding the message received by the BlackBerry.

In the examples below, the result which BBAgent returns, shown in the POP inbox, is displayed using the command DisplayMessage.

Any other message is considered as a normal message, and is not subjected to any special processing.

14.4.2 ATTRIBUTES

The values returned by Newtest for BlackBerry methods are displayed when you use the command INFOS (see section 12.4 above).



Values and descriptions:

ACTION_STATUS	: OK if the request succeeds, NOK if not.
DEVICE_NAME	: Retrieves the current device's product name.
DEVICE_PIN	: Retrieves the PIN number for this device.
PLATFORM_VERSION	: Retrieves the platform version.
BATTERY_LEVEL	: Retrieves the current battery level. (%)
MEMORY_FLASH_FREE	: Returns the amount of free flash. (bytes)
MEMORY_FLASH_TOTAL	: Returns the amount of flash needed by the VM. (bytes)
MEMORY_FREE	: Returns the amount of memory/storage still free and available for use.
MEMORY_TOTAL	: Total available memory (free + allocated)
SUPPORT_3GPP	: True if 3GPP Wireless Access Family is supported
SUPPORT_CDMA	: True if CDMA Wireless Access Family is supported
SUPPORT_IDEN	: True if iDEN Wireless Access Family is supported
SUPPORT_WLAN	: True if WLAN Wireless Access Family is supported
SIGNAL_LEVEL	: Retrieves current signal strength in dBm (typically between -121 and -40 dBm)
NETWORK_NAME	: Retrieve the name of the carrier network on which the device is currently registered.
NETWORK_TYPE	: Retrieves the type of the Network. CDMA, GPRS, UMTS, iDEN or Unknown
GPRS_ZONE	: Retrieves the time zone name.
GPRS_CELL_ID	: Retrieves the current cell ID.
GPRS_CELL_ARFCN	: Retrieves the Absolute Radio Frequency Channel Number.
GPRS_CELL_BSIC	: Retrieves the Base Station Identity Code.
GPRS_CELL_LAC	: Retrieves the Location Area Code.
GPRS_CELL_RAC	: Retrieves the Routing Area Code.

GPS_SUPPORTED	: False if the device doesn't support GPS. The following fields will not appear in this case.
GPS_LAST_POS	: Last known position. (INFO does not call a GPS sync.)
GPS_LAST_POS_TIME	: Retrieves the date/time of the last GPS synchronization
GPS_ERROR	: Exception thrown during GPS-related functions, if any.

14.4.3 PRESENCE CHECK [BBAGENT TEST]

Purpose

The purpose of this function is to perform a simple check of the BlackBerry's presence and the successful launch of its BBAgent.

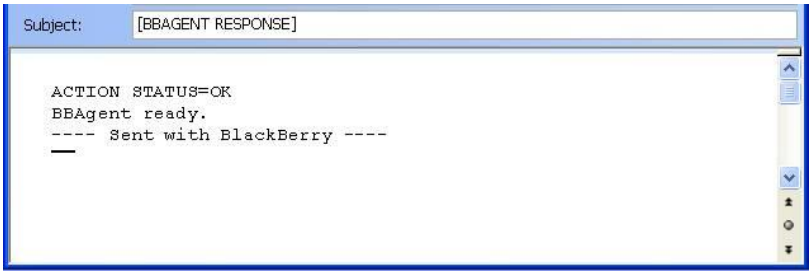
Use

The function should be placed in the message subject line. Sending an e-mail to the BlackBerry, with [BBAGENT TEST] in the subject line, will return the indicator "BBAgent Ready" in the body of the reply without any other message.

Neither the message sent nor its acknowledgement are encoded. This command does not allow values to be extracted or checked. It is simply a reply test. It is therefore possible to launch it from a classic e-mail client.

Example of use

The return expected when the test is positive is a reply with "BBAgent ready" in the body of the mail:



14.4.4 INFOS METHOD

Purpose

This function retrieves indicators specific to the tested BlackBerry, such as the signal level, battery, or the last known GPS position. This function, like all those that follow, waits for a message that has been encoded by the function BBAgentEncode, and because of this can be only be tested using a Newtest script.

Use

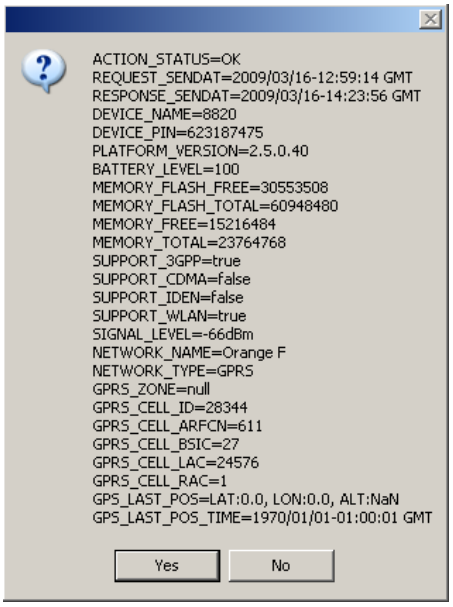
The function is called by a message whose title is [BBAGENT REQUEST]. It is inserted into the body of the message as follows:

ACTION=INFOS

RESPONDTO=my_mail_account@domain.com

The reply takes the form of a return message with [BBAGENT RESPONSE] in the subject line. The Newtest function BBAgentDecode decodes the message and makes it usable. If the RESPONDTO field is not filled in, the message is returned to the sender.

Sample valid return:



You can then check or extract values of these elements by using the command: MobileParamExtract ()

14.4.5 WGET METHOD

Purpose

This function launches a test of internet connectivity to check the availability of external access from the BlackBerry.

Use

The function is called by a message with [BBAGENT REQUEST] in the subject line. It is inserted into the body of the message as shown below:

ACTION=WGET

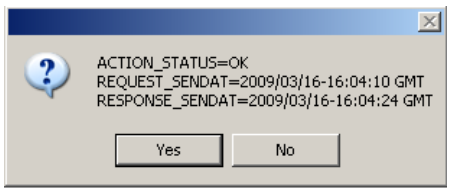
URL=http://mon_site_web.com

When it receives such a message, the BlackBerry attempts to load the page designated by the URL. The information returned consists of the server's return code, the time it took to connect, and the size of the page. In the event of a problem (limited internet connectivity, wrong parameters, invalid URL, etc.), the message returned contains a short description of the problem encountered.

The reply is transmitted in a return message with [BBAGENT RESPONSE] in the subject line.

The Newtest function BBAgentDecode decodes the message and makes it usable.

Sample valid return:



Restrictions

- The function called internally does not accommodate a timeout. Therefore the timeout is dependent on the BlackBerry (3 minutes).
- WAP/GPRS gateways appear to use their own DNS. Because of this some domains cannot be resolved; the function will therefore return a failure.

14.4.6 ECHO METHOD

Purpose

This function returns a timestamp, and in this way makes a measurement of the efficiency of the BlackBerry's message processing chain.

Use

The function is called by a message with [BBAGENT REQUEST] in the subject line. It is inserted into the message body as follows:

ACTION=ECHO

SENDAT=YYYY/MM/DD-HH:MM:SS

The BlackBerry places the contents of the SENDAT line in its reply (renaming it REQUEST_SENDAT), and adds the field RESPONSE_SENDAT to contain the timestamp of the moment the BlackBerry processed the command.

14.4.7 GPS METHOD

Purpose

This function initiates satellite synchronization of BlackBerry devices equipped with the GPS module.

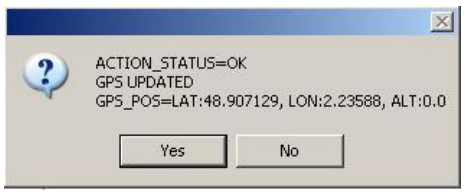
Use

Satellite synchronization can take some time (up to 15 minutes). This function should therefore be used with caution.

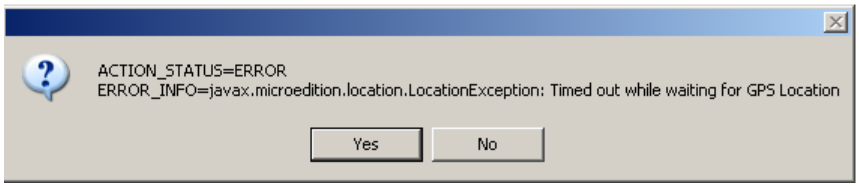
The function is called by a message with [BBAGENT REQUEST] in the subject line. It is inserted into the message body as follows:

ACTION=UPDATE_GPS

If the outcome is successful, the BlackBerry returns the latest position of the BlackBerry:



In case of failure, a brief description is returned:



Important: This function can impede processing insofar as the BlackBerry will not process any other request until synchronization has been completed.

14.5SAMPLE BLACKBERRY SCENARIO

The following example is based on the function WGET. The aim of the test is to check the availability of access to public internet from a BlackBerry. The URL to be tested: www.google.com

The requests are configured within the IP Analyser client environment:

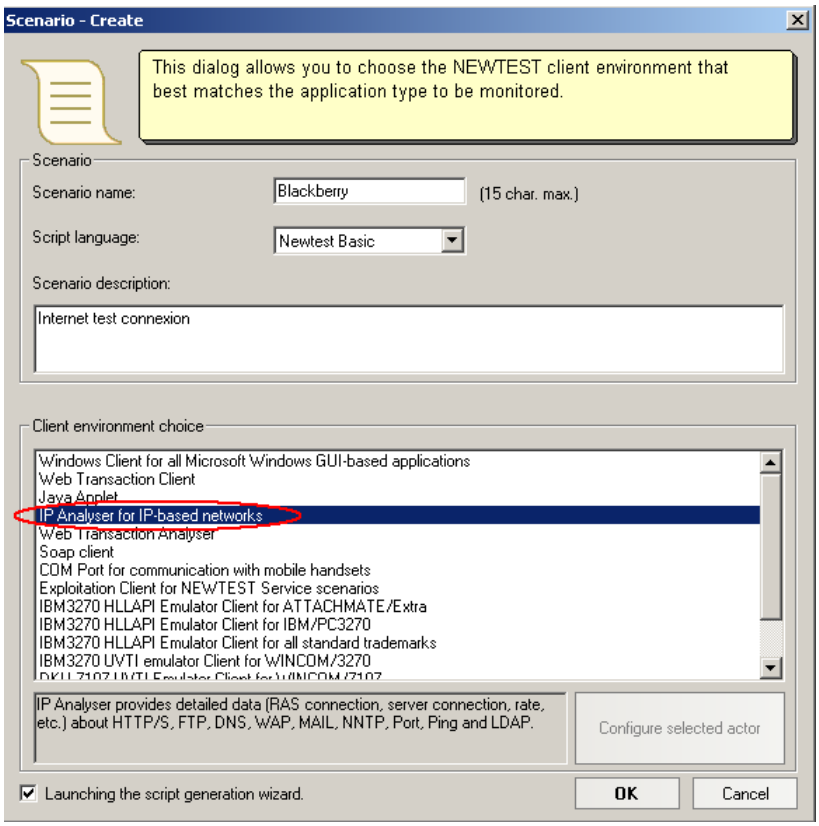
For the example, the transaction to replay consists of sending the BlackBerry a mail with the following properties:

- To: iplabel@blackberry.orange.fr
- From: testsmtp@ip-label.net
- Message subject: [BBAGENT REQUEST]
- Message body: ACTION=WGET\nURL=http://www.google.com\n

The expected return message is built as follows:

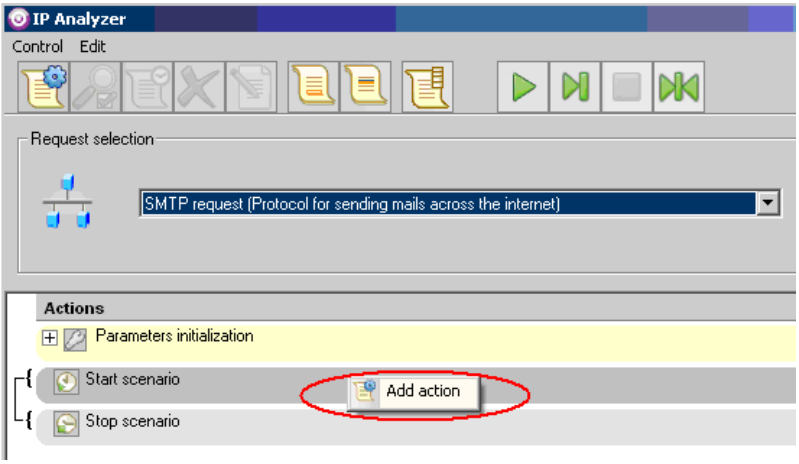
- To: testsmtp@ip-label.net
- From: iplabel@blackberry.orange.fr
- Message subject: [BBAGENT RESPONSE]
- Attribute tested: RESPONSE_CODE=302

A new scenario may now be created in the IP Analyser wizard:

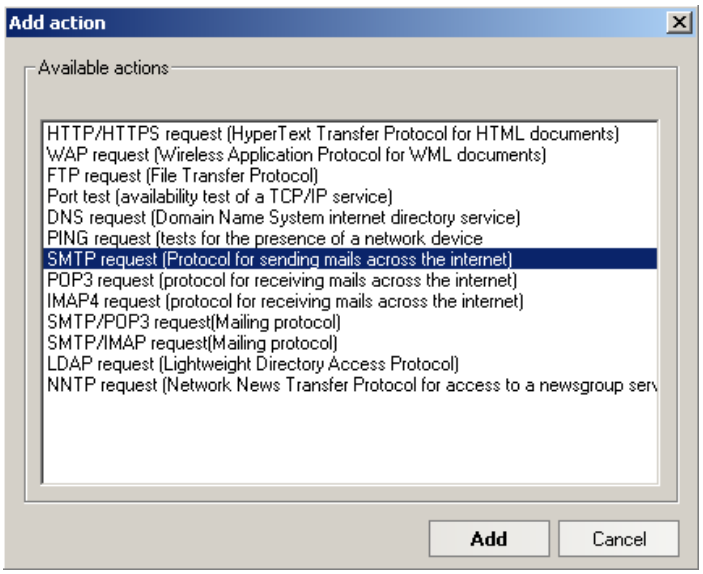


14.5.1 CREATING AN SMTP REQUEST

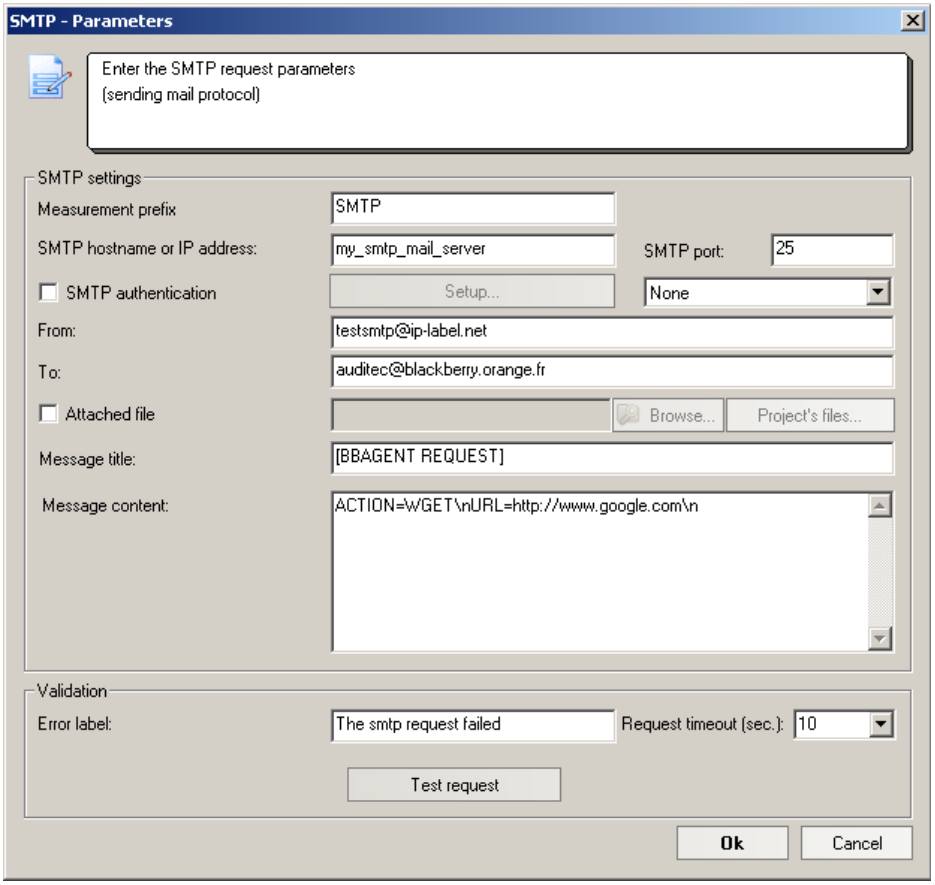
In the wizard, add an action by right-clicking in the work area:



Next select an SMTP-type request, and then click the “Add” button:

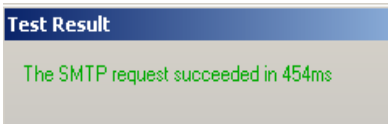

Add action
Available actions
HTTP/HTTPS request (HyperText Transfer Protocol for HTML documents)
WAP request (Wireless Application Protocol for WML documents)
FTP request (File Transfer Protocol)
Port test (availability test of a TCP/IP service)
DNS request (Domain Name System internet directory service)
PING request (tests for the presence of a network device
SMTP request (Protocol for sending mails across the internet)
POP3 request (protocol for receiving mails across the internet)
IMAP4 request (protocol for receiving mails across the internet)
SMTP/POP3 request(Mailing protocol)
SMTP/IMAP request(Mailing protocol)
LDAP request (Lightweight Directory Access Protocol)
NNTP request (Network News Transfer Protocol for access to a newsgroup serv
Add
Cancel

Fill in the fields of the SMTP request wizard with the information needed for the query:

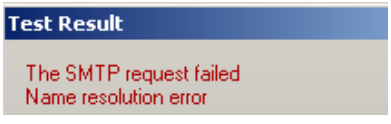


The button allows you to check that the SMTP request is issued correctly.

Successful test:



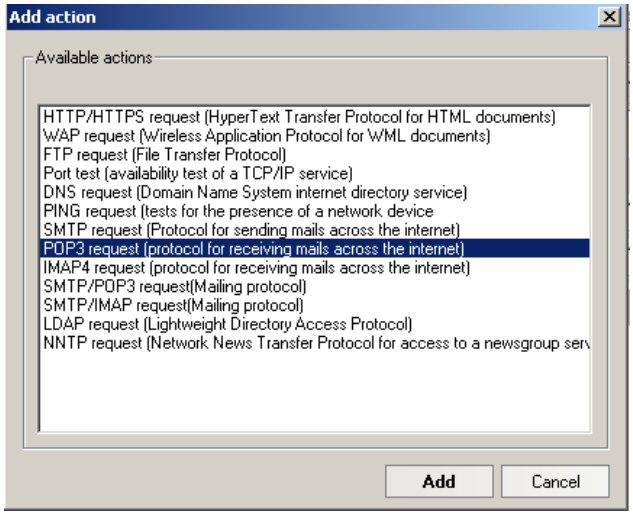
Failed test:



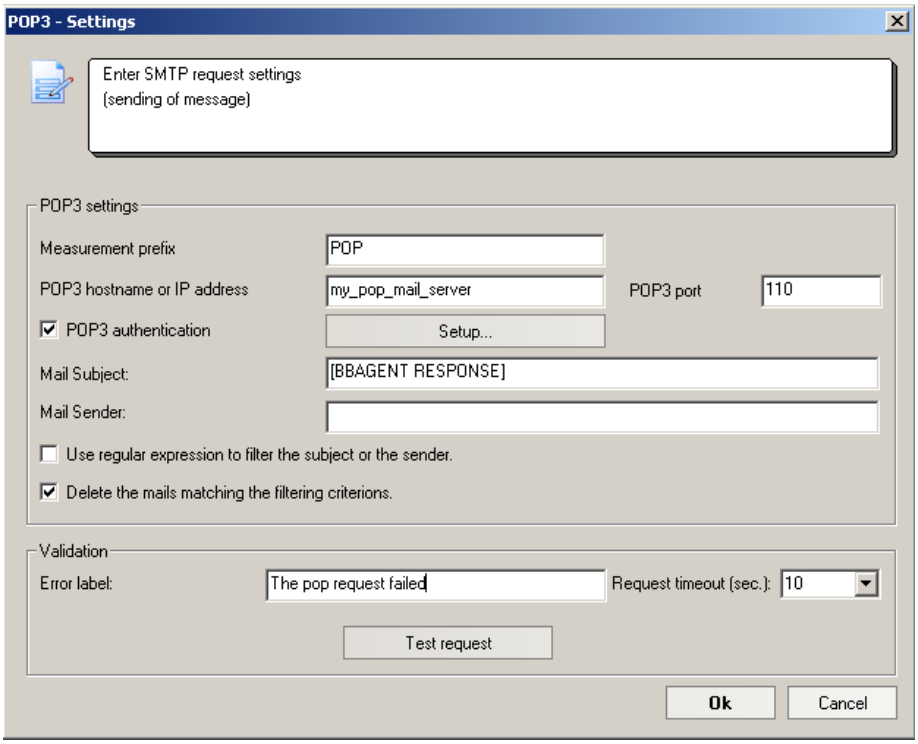
14.5.2 CREATING A POP REQUEST

The next step is to create the action that will extract the reply message from the POP server to which the BlackBerry's acknowledgement was sent.

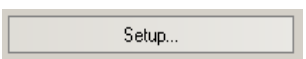
Select a POP3 type request:



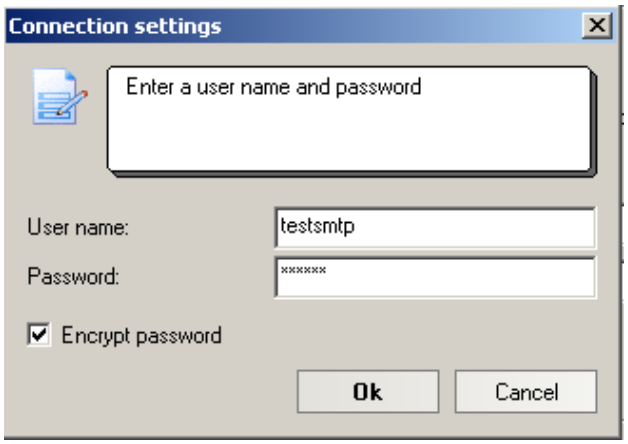
Fill in the fields of the SMTP request wizard with the information needed for the query:



Select the option.



The button is for setting the parameters for authentication on the server:



14.5.3 ENCODING THE MESSAGE BODY

It may be necessary to encode the body of the SMTP message and the contents to search for in the return message from the POP server. In this case, you must manually insert code into the script. The functions used in this case are BBAgentEncode and BBAgentDecode. The variable that contains the message body is passed to the parameter Body.

Changes to make:

- In the SMTP request, after assigning the variable Body:

```
63  //@ACTION id=BJCjTmBMuNJvDnDP25CoGiRz
64    bRet = TRUE
65    wait(2)
66    RET SetScenarioStartTime()
67    If bRet = TRUE Then
68      //Subject = GenerateUniqueMailSubject()
69      Subject = "[BBAGENT REQUEST]"
70      Body = UnescapeString("ACTION=WGET\nURL=http://www.google.com\n")
71      Body=BBAgentEncode(Body)
```

- In the POP request, after assigning the variable Body:

```
150       Pop3Purge()
151
152       Subject = GetStrOption("MAIL_SUBJECT")
153       Sender  = GetStrOption("MAIL_FROM")
154       MailDate = GetStrOption("MAIL_DATE")
155       Body = GetStrOption("MAIL_BODY")
156       Body=BBAgentDecode(Body)
```

It remains only to extract the value that you wish to set as your criterion. The Newtest command for this is MobileParamExtract().

In the example below, we want to ensure that the value of the entry RESPONSE_CODE is equal to 302.

```
162       Result = MobileParamExtract(Body,"RESPONSE_CODE")
163       IF Result="302" THEN
164         Goto NEXT
165       ELSE
166         Ret ErrorLine("302 response code not found. Internet access down")
167       ENDIF
```

14.6SOURCE CODE OF THE FINAL SCENARIO

```
//@DECLARATION
DIM Response AS STRING
DIM User AS STRING
DIM Password AS STRING
DIM Subject AS STRING
DIM Body AS STRING
DIM Sender AS STRING
DIM MailDate AS STRING
DIM bRet AS BOOL
DIM SMTP_Test AS ORDER(APPLI)
DIM SMTP1_Dns AS ORDER(APPLI,"msec")
DIM SMTP2_Connect AS ORDER(APPLI,"msec")
DIM SMTP3_Auth AS ORDER(APPLI,"msec")
DIM SMTP4_SendTime AS ORDER(APPLI,"msec")
DIM SMTP5_SendRate AS ORDER(APPLI,"byte/s")
DIM POP_Test AS ORDER(APPLI)
DIM POP1_Dns AS ORDER(APPLI,"msec")
DIM POP2_Connect AS ORDER(APPLI,"msec")
DIM POP3_Auth AS ORDER(APPLI,"msec")
DIM POP4_WaitMail AS ORDER(APPLI,"msec")
DIM POP5_ReceiveTime AS ORDER(APPLI,"msec")
DIM POP6_ReceiveRate AS ORDER(APPLI,"byte/s")
DIM Result AS STRING
//@


// Script generated by the IP Analyser wizard


//@BEGIN
// Begin program

bRet = TRUE
If bRet = TRUE Then
//
// Init IP
//
InitEnvironment()


SetOption("DIAG_ENABLED","",1)
SetOption("DIAG_PINGS_NB","",3)
SetOption("DIAG_PACKET_SIZE","",32)
SetOption("DIAG_LOCAL_DOMAIN","",0)
SetOption("DIAG_ISP_DOMAIN","",0)
SetOption("DIAG_PING_TIMEOUT","",1)
SetOption("DIAG_EXTENTED_TRACEROUTE","",1)


ENDIF
If bRet = FALSE Then
RET ErrorLine("Selector")
GOTO ERROR
ENDIF
//@
```

```
//@ACTION id=BJCjTmBMuNJvDnDP25CoGiRz
bRet = TRUE
wait(2)
RET SetScenarioStartTime()
If bRet = TRUE Then
   //Subject = GenerateUniqueMailSubject()
   Subject = "[BBAGENT REQUEST]"
   Body = UnescapeString("ACTION=WGET\nURL=http://www.google.com\n")

   // The command BBAgentEncode is added manually to encode the body of the message to send

   Body=BBAgentEncode(Body)


   // Continuation of the script generated by the IP Analyser wizard


   // SMTP
   // ---
   Password=""

   SetOption("PROTOCOL_USER","",0)
   SetOption("PROTOCOL_PASSWORD",Password,0)
   SetOption("SMTP_FROM","testsmtp@ip-label.net",0)
   SetOption("SMTP_TO","iplabel@blackberry.orange.fr",0)
   SetOption("MAIL_SUBJECT",Subject,0)
   SetOption("SMTP_ATTACHFILE","",0)
   SetOption("SMTP_MESSAGESIZE","",12)
   SetOption("SMTP_SERVER","my_smtp_mail_server",0)
   SetOption("SMTP_PORT","",25)
   SetOption("SMTP_USESSL","",0)
   SetOption("SMTP_TIMEOUT","",10)
   SetOption("MAIL_BODY",Body,0)


   // SMTP request
   // -------
   bRet = SmtpRequest()

   // SMTP Measurements
   // ---------
   IF bRet= TRUE THEN
         GetSmtpRequestTime(SMTP_Test)
         RET SMTP_Test
         GetSmtpDnsTime(SMTP1_Dns)
         RET SMTP1_Dns
         GetSmtpConnectTime(SMTP2_Connect)
         RET SMTP2_Connect
         GetSmtpAuthenticationTime(SMTP3_Auth)
         RET SMTP3_Auth
         GetSmtpSendTime(SMTP4_SendTime)
         RET SMTP4_SendTime
         GetSmtpSendRate(SMTP5_SendRate)
         RET SMTP5_SendRate
   ENDIF
ENDIF
If bRet = FALSE Then
```

```
   RET ErrorLine("The smtp request failed")
   GOTO ERROR
ENDIF
//@

//@ACTION id=D70l4nBFW$7_DJ4g2mDAIT5d
bRet = TRUE
wait(2)
If bRet = TRUE Then

   Subject = "[BBAGENT RESPONSE]"
   Sender = ""
   Body = ""
   MailDate =""


   // POP3
   // ——
   Password=""

GetPassword("#|DFE073AF0AD1905F42F4B9450FE8EF598C06612BEF672891DFD7C0B3F7314CD95
D123AC1A5647F2A894ED43063264797", Password)
   SetOption("PROTOCOL_USER","testsmtp",0)
   SetOption("PROTOCOL_PASSWORD",Password,0)
   SetOption("MAIL_SUBJECT",Subject,0)
   SetOption("MAIL_FROM",Sender,0)
   SetOption("MAIL_REGEXP","",0)
   SetOption("POP3_SERVER","<my_pop_mail_server",0)
   SetOption("POP3_PORT","",110)
   SetOption("POP3_USESSL","",0)
   SetOption("POP3_TIMEOUT","",10)


   // POP3 request
   // ————————
   bRet = Pop3Request()

   IF bRet= TRUE THEN

         Pop3Purge()

         Subject = GetStrOption("MAIL_SUBJECT")
         Sender  = GetStrOption("MAIL_FROM")
         MailDate = GetStrOption("MAIL_DATE")
         Body = GetStrOption("MAIL_BODY")

// The command BBAgentDencode is added manually to decode the body of the return message


         Body=BBAgentDecode(Body)

         // Manually added extraction and check of attribute value

         Result = MobileParamExtract(Body,"RESPONSE_CODE")
         IF Result="302" THEN
               Goto NEXT
         ELSE
               Ret ErrorLine("302 response code not found. Internet access down")
```

```
                GOTO ERROR
        ENDIF

// Continuation of the script generated by the IP Analyser wizard


        GetPop3RequestTime(POP_Test)
        GetPop3DnsTime(POP1_Dns)
        GetPop3ConnectTime(POP2_Connect)
        GetPop3AuthenticationTime(POP3_Auth)
        GetPop3WaitMailTime(POP4_WaitMail)
        GetPop3ReceiveTime(POP5_ReceiveTime)
        GetPop3ReceiveRate(POP6_ReceiveRate)

        RET POP_Test
        RET POP1_Dns
        RET POP2_Connect
        RET POP3_Auth
        RET POP4_WaitMail
        RET POP5_ReceiveTime
        RET POP6_ReceiveRate
   ENDIF

ENDIF
If bRet = FALSE Then
   RET ErrorLine("The pop request failed")
   GOTO ERROR
ENDIF
RET SetScenarioStopTime()
//@

//@END
Wait(5)
// END program
END
//@

//@ERROR
ERROR:
Wait(5)
END
//@

//@CANCEL
CANCEL:
Wait(5)
RET SetScenarioCanceled()
END
//@
```

14.7 NEWTEST BLACKBERRY FUNCTIONS

The list of functions below is specific to the IP Analyser client environment in the context of tests carried out on BlackBerry devices. The other functions, common to all IP Analyser tests, are not listed here.

Function	Description
BBAgentEncode	Encodes the data in SMTP messages send to the BlackBerry
BBAgentDecode	Decodes the data in messages received on the BlackBerry
MobileParamExtrac	Extracts information from messages issued by a BlackBerry

15 APPENDIX

15.1 CLIENT ENVIRONMENTS

The table below shows the client environment(s) concerned for each DLL. In the right-hand column, "Old" indicates old client environments that are still in use but are no longer delivered with the latest Newtest versions.

DLL	CLIENT ENVIRONMENT	STATUS
EXPLOIT	Operation	
NEWT16	Windows client environment	
NEWTACC	(General functions)	
NEWTALRM	(General functions) / Alarms	
NEWTCOM	COM port (SMS) client environment	
NEWTCTA	QX2000 voice board client environment	Old
NEWTCTRX	Citrix client environment	Old
NEWTGEN	(General functions)	
NEWTHLL	HLLAPI emulators client environment	
NEWTICMP	(General functions)	
NEWTJAVA	Java Applet client environment	
NEWTMDP	Media Player client environment	
NEWTNAV	Web Transaction Client environment	
NEWTNET	IP Analyser client environment (new)	
NEWTNT	(General functions)	
NEWTREAL	RealPlayer streaming client environment	Old
NEWTSAP	SAP client environment	Old
NEWTUVTI	UVTI emulators client environment	
NEWTWEB	Web Transaction Analyser environment	
NEWTWIN	Windows client environment	
NEWTX11	X-Window client environment	Old
VIEWIMG	General	
NEWTCFT	CFT	Old
NEWTCON	Console	Old
NEWTIE	IE4 and Netscape client environment	Old
NEWTIE5	IE5 client environment	Old
NEWTIP	TCP/IP client environment	Old
NEWTMRAS	MultiRAS IP Analyser client environment	Old
NEWTTCP	TCP network client environment	Old
NEWTTLCO	IP Analyser client environment (old)	Old
NEWTVOX	VOX voice board client environment	Old
NEWTWWW	Web client environment	Old

15.2 REGULAR EXPRESSION SYNTAX

A regular expression is a pattern of text that consists of ordinary characters (for example, letters a through z) and special characters, known as *metacharacters*. The pattern describes one or more strings to match when searching a body of text. The regular expression serves as a template for matching a character pattern to the string being searched.

The following table contains the complete list of metacharacters and their behavior in the context of regular expressions:

CHARACTER	DESCRIPTION
\	Marks the next character either as a special character, a literal, a backreference, or an octal escape. For example, 'n' matches the character "n". '\n' matches a newline character. The sequence '\\' matches "\" and "\(" matches "(".
^	Matches the position at the beginning of the input string.
$	Matches the position at the end of the input string.
*	Matches the preceding subexpression zero or more times. For example, zo* matches "z" and "zoo". * is equivalent to {0,}.
+	Matches the preceding subexpression one or more times. For example, 'zo+' matches "zo" and "zoo", but not "z". + is equivalent to {1,}.
?	Matches the preceding subexpression zero or one time. For example, "do(es)?" matches the "do" in "do" or "does". ? is equivalent to {0,1}
{*n*}	*n* is a nonnegative integer. Matches exactly *n* times. For example, 'o{2}' does not match the 'o' in "Bob," but matches the two o's in "food".
{*n*,}	*n* is a nonnegative integer. Matches at least *n* times. For example, 'o{2,}' does not match the "o" in "Bob" and matches all the o's in "foooood". 'o{1,}' is equivalent to 'o+'. 'o{0,}' is equivalent to 'o*'.
{*n*,*m*}	*m* and *n* are nonnegative integers, where *n* <= *m*. Matches at least *n* and at most *m* times. For example, "o{1,3}" matches the first three o's in "fooooood". 'o{0,1}' is equivalent to 'o?'. Note that you cannot put a space between the comma and the numbers.
?	When this character immediately follows any of the other quantifiers (*, +, ?, {*n*}, {*n*,}, {*n*,*m*}), the matching pattern is non-greedy. A non-greedy pattern matches as little of the searched string as possible, whereas the default greedy pattern matches as much of the searched string as possible. For example, in the string "oooo", 'o+?' matches a single "o", while 'o+' matches all 'o's.
.	Matches any single character except "\n". To match any character including the '\n', use a pattern such as '[.\n]'.
(*pattern*)	Matches *pattern* and captures the match. The captured match can be retrieved from the resulting Matches collection, using the **SubMatches** collection in VBScript or the **$0...$9** properties in JScript. To match parentheses characters ( ), use '\(' or '\)'.
(?:*pattern*)	Matches *pattern* but does not capture the match, that is, it is a non-capturing match that is not stored for possible later use. This is useful for combining parts of a pattern with the "or" character (|). For example, 'industr(?:y|ies) is a more economical expression than 'industry|industries'.
(?=*pattern*)	Positive lookahead matches the search string at any point where a string matching *pattern* begins. This is a non-capturing match, that is, the match is not captured for possible later use. For example 'Windows (?=95|98|NT|2000)' matches "Windows" in "Windows 2000" but not "Windows" in "Windows 3.1". Lookaheads do not consume characters, that is, after a match occurs, the search for the next match begins immediately following the last match, not after the characters that comprised the lookahead.
(?!*pattern*)	Negative lookahead matches the search string at any point where a string not matching *pattern* begins. This is a non-capturing match, that is, the match is not captured for possible later use. For example 'Windows (?!95|98|NT|2000)' matches "Windows" in "Windows 3.1" but does not match "Windows" in "Windows 2000". Lookaheads do not consume characters, that is, after a match occurs, the search for the next match begins immediately following the last match, not after the characters that comprised the lookahead.
x|*y*	Matches either *x* or *y*. For example, 'z|food' matches "z" or "food". '(z|f)ood' matches "zood" or "food".
[*xyz*]	A character set. Matches any one of the enclosed characters. For example, '[abc]' matches the 'a' in

	"plain".
[^*xyz*]	A negative character set. Matches any character not enclosed. For example, '[^abc]' matches the 'p' in "plain".
[*a-z*]	A range of characters. Matches any character in the specified range. For example, '[a-z]' matches any lowercase alphabetic character in the range 'a' through 'z'.
[^*a-z*]	A negative range characters. Matches any character not in the specified range. For example, '[^a-z]' matches any character not in the range 'a' through 'z'.
\b	Matches a word boundary, that is, the position between a word and a space. For example, 'er\b' matches the 'er' in "never" but not the 'er' in "verb".
\B	Matches a nonword boundary. 'er\B' matches the 'er' in "verb" but not the 'er' in "never".
\c*x*	Matches the control character indicated by *x*. For example, \cM matches a Control-M or carriage return character. The value of *x* must be in the range of A-Z or a-z. If not, c is assumed to be a literal 'c' character.
\d	Matches a digit character. Equivalent to [0-9].
\D	Matches a nondigit character. Equivalent to [^0-9].
\f	Matches a form-feed character. Equivalent to \x0c and \cL.
\n	Matches a newline character. Equivalent to \x0a and \cJ.
\r	Matches a carriage return character. Equivalent to \x0d and \cM.
\s	Matches any whitespace character including space, tab, form-feed, etc. Equivalent to [\f\n\r\t\v].
\S	Matches any non-white space character. Equivalent to [^ \f\n\r\t\v].
\t	Matches a tab character. Equivalent to \x09 and \cI.
\v	Matches a vertical tab character. Equivalent to \x0b and \cK.
\w	Matches any word character including underscore. Equivalent to '[A-Za-z0-9_]'.
\W	Matches any nonword character. Equivalent to '[^A-Za-z0-9_]'.
\x*n*	Matches *n*, where *n* is a hexadecimal escape value. Hexadecimal escape values must be exactly two digits long. For example, '\x41' matches "A". '\x041' is equivalent to '\x04' & "1". Allows ASCII codes to be used in regular expressions.
num	Matches *num*, where *num* is a positive integer. A reference back to captured matches. For example, '(.)\1' matches two consecutive identical characters.
n	Identifies either an octal escape value or a backreference. If *n* is preceded by at least *n* captured subexpressions, *n* is a backreference. Otherwise, *n* is an octal escape value if *n* is an octal digit (0-7).
nm	Identifies either an octal escape value or a backreference. If *nm* is preceded by at least *nm* captured subexpressions, *nm* is a backreference. If *nm* is preceded by at least *n* captures, *n* is a backreference followed by literal *m*. If neither of the preceding conditions exists, *nm* matches octal escape value *nm* when *n* and *m* are octal digits (0-7).
nml	Matches octal escape value *nml* when *n* is an octal digit (0-3) and *m* and *l* are octal digits (0-7).
\u*n*	Matches *n*, where *n* is a Unicode character expressed as four hexadecimal digits. For example, \u00A9 matches the copyright symbol (©).