

CA Nimsoft Monitor

Probe Guide for E2E Application Response Monitoring

e2e_appmon v2.2 series



Copyright Notice

This online help system (the "System") is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This System may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This System is confidential and proprietary information of CA and protected by the copyright laws of the United States and international treaties. This System may not be disclosed by you or used for any purpose other than as may be permitted in a separate agreement between you and CA governing your use of the CA software to which the System relates (the "CA Software"). Such agreement is not modified in any way by the terms of this notice.

Notwithstanding the foregoing, if you are a licensed user of the CA Software you may make one copy of the System for internal use by you and your employees, provided that all CA copyright notices and legends are affixed to the reproduced copy.

The right to make a copy of the System is limited to the period during which the license for the CA Software remains in full force and effect. Should the license terminate for any reason, it shall be your responsibility to certify in writing to CA that all copies and partial copies of the System have been destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS SYSTEM "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS SYSTEM, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The manufacturer of this System is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2014 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Legal information on third-party and public domain software used in this product is documented in the *Third-Party Licenses and Terms of Use* (http://docs.nimsoft.com/prodhelp/en_US/Library/Legal.html).

Contact CA

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback about Product Documentation

Send comments or questions about CA Technologies product documentation to nimsoft.techpubs@ca.com.

To provide feedback about general CA Technologies product documentation, complete our short customer survey which is available on the support website at <http://ca.com/docs>.

Contents

Chapter 1: E2E Application Response Monitoring (e2e_appmon) Overview v2.2	7
About This Guide	8
Related Documentation	9
Preconfiguration Requirements	9
Supported Platforms	9
Chapter 2: E2E Application Response Monitoring (e2e_appmon) Configuration Details v2.2	11
e2e_appmon Node	11
Profiles Node	14
Script Node	16
Variables Node	18
Configure a Node	19
Manage Profiles	19
Create a Deployable Script Package	20
Chapter 3: e2e_appmon QoS Threshold Metrics	23
e2e_appmon QoS Metrics	23
e2e_appmon Alert Metrics Default Settings	23
Chapter 4: E2E Application Response Monitoring (e2e_appmon) Troubleshooting v2.2	25
NimRecorder is not Deployed or Installed	25
Scripts not Working Fine after Upgrade	26
Scripts not Working due to Synchronization	26

Documentation Changes

This table describes the version history for this document.

Version	Date	What's New?
2.2	November 2014	<ul style="list-style-type: none">■ Updated the Overview section.■ Updated the NimRecorder to version 5.1.
2.2	June 2014	Documentation on the initial Admin Console GUI version of this probe. (Previous versions of this probe are configured using Infrastructure Manager).

Chapter 1: E2E Application Response Monitoring (e2e_appmon) Overview v2.2

The E2E Application Response Monitoring probe runs the precompiled NimRecorder scripts for monitoring the response time and other performance parameters of an application. The probe runs scripts at specified intervals to monitor the application availability and total run time of each script for generating QoS and alarms messages.

You can use the NimRecorder (shipped with e2e_appmon_dev probe) to write your own scripts. The *e2e_appmon API* lets you include checkpoints within your scripts. The scripts must be compiled on those machines where they are intended to execute.

Important! Do not run other applications or tasks on the monitoring system, as it disrupts the application monitoring and the measurement process of the probe.

Note: The probe has certain limitations to use Optical Character Recognition (OCR) in the scripts. So, instead of using OCR, you can use the bitmap synchronization (synchronization on an image) to use the text logo.

The other related things of the E2E Application Response Monitoring probe are as follows:

- **Sample Script:** The probe is deployed with a sample script. Activate the probe and run the sample script. Compile the script on the target computer before executing it. If the script is executed before it is compiled, an error message is thrown. For example, the error message is: **Error at line 348: Impossible to load the module.**

A compiled script is not delivered with the probe because the script is platform-dependent. Therefore, the compiled script can run only on similar platforms.

- **Probe Editions:** The e2e_appmon probe is available in two different editions, which are as follows:
 - The **Runtime Edition**, which runs the precompiled scripts.
 - The **Developer Edition**, the **e2e_appmon_dev**, which lets you write your own scripts and include checkpoints within those scripts. The e2e_appmon_dev probe can measure the time interval between each process with the total runtime of the script. In addition, some help files for writing the scripts are also included.

- **NimRecorder:** The NimRecorder is used for writing and compiling scripts, which is included in the probe package. The NimRecorder options can be launched from **Start > All Programs > Nimsoft Monitoring > E2E Scripting:**
 - Compile *
 - Help *
 - Open Script *
 - Run Script
 - Script Editor *
 - Spy *
 - Uninstall NimRecorder

Note: The options marked with an asterisk (*) are available only in the **Developer Edition** or after installing the NimRecorder manually.

The probe is now available with **NimRecorder 5.1** for writing and executing scripts for 64-bit platforms. The NimRecorder 5.1 also supports browser-based scripting for Internet Explorer, Mozilla Firefox, and Google Chrome. Refer the NimRecorder 5.1 help for supported platforms and browser versions with other relevant information.

Access the **NimRecorder Help** from the **Start > All Programs > Nimsoft Monitoring > E2E Scripting > Help** location. This **Help** file is also available from the **Help** menu of the NimRecorder application window.

This section contains the following topics:

[About This Guide](#) (see page 8)

[Related Documentation](#) (see page 9)

[Preconfiguration Requirements](#) (see page 9)

About This Guide

This guide is for the CA Nimsoft Monitor Administrator to help understand the configuration of the E2E Application Response Monitoring probe and provides the following information:

- Overview of the E2E Application Response Monitoring probe and related documentation for previous probe versions.
- Configuration details of the probe.
- Field information and common procedures for configuring the probe.

Important! Description for the intuitive GUI fields is not included in the document.

Related Documentation

Related Documentation

Documentation for other versions of the e2e_appmon probe

The [Release Notes](#) for the e2e_appmon probe

[User documentation for the Admin Console](#)

Monitor Metrics Reference Information for CA Unified Infrastructure Management Probes

(http://docs.nimsoft.com/prodhelp/en_US/Probes/ProbeReference/index.htm)

Preconfiguration Requirements

The preconfiguration requirements of the E2E Application Response Monitoring probe are as follows:

- All the users included in the probe profile must have administrative privileges (read and write permissions) to the e2e_appmon and the e2e_appmon_dev directory.
- The **WTBho** Class, a NimRecorder add-on, must be enabled on supported web browser for executing the web-based scripts.

Supported Platforms

Refer to the [Compatibility Support Matrix](#) for the latest information about supported platforms. See also the [Support Matrix for Probes](#) for more information about the e2e_appmon probe.

Chapter 2: E2E Application Response Monitoring (e2e_appmon) Configuration Details v2.2

The E2E Application Response Monitoring probe is configured for creating the application monitoring profiles. Each monitoring profile can execute multiple scripts. You can also define thresholds for generating alarms when the script execution time exceeds the specified limit. The QoS messages are also configured for saving the response time of the application.

This section contains the following topics:

[e2e_appmon Node](#) (see page 11)

[Configure a Node](#) (see page 19)

[Manage Profiles](#) (see page 19)

[Create a Deployable Script Package](#) (see page 20)

e2e_appmon Node

The e2e_appmon node lets you view the probe information and configure the properties which are applicable to all monitoring profiles of the probe.

Navigation: e2e_appmon

Set or modify the following values as required:

e2e_appmon > Probe Information

This section provides information about the probe name, probe version, probe start time, and the probe vendor.

e2e_appmon > Log Level Configuration

This section is used to configure log properties of the probe.

- Log Level: specifies the detail level of the log file.
Default: 0 - Normal
- Log File Size (KB): defines the maximum size of the log file.
Default: 100

e2e_appmon > Run as User

This section is used for configuring the user details for providing necessary privileges to execute the script.

- Name: defines the user name of the target system where the probe runs the script. The script has the same access level as the user.

Default: administrator

- Password: defines the password for authorizing the specified user name.

- User Check to Prevent Scripts to be Run from the Wrong User: prevents the script execution by any other user, except the one that is specified in the probe.

Default: Not selected

- Reset Registry Settings Right After the User is Logged on: resets the registry settings after the user logs in successfully. Enabling this option makes the target system less vulnerable to malicious attacks over a network. Also, when the remote desktop connection (RDP) is used, a legal notice is displayed. However, ensure the presence of an automatic login to the system through the registry settings.

Default: Not selected

Note: This option aborts the login process on a slow processing system.

e2e_appmon > Run Properties

This section is used for configuring the execution properties of the script.

- Default Run Interval (Seconds): defines the time interval between each script execution process.

Default: 33

- Default Maximum Run Time (Seconds): defines the time limit for script execution. The probe can generate alarm when this limit is breached.

- Command: defines the path of the directory where the **TaskExec.exe** file is located. This TaskExec.exe file is used for executing the compiled scripts on both 32-bit and 64-bit platforms.

Default: \program files (x86)\Nimsoft\e2e_scripting\bin

- .ROB File Directory: defines the path of the directory where the compiled script files are stored.

Default: \program files (x86)\Nimsoft\e2e_scripting\scripts

Note: The default relative paths of the **Command** and **ROB File Directory** fields are not correct. Remove these default paths and configure the absolute path manually.

e2e_appmon > Alarm on Start Error

This section is used for configuring the alarm when the script fails to start on the target system.

- Publish Alarm: enables the probe for generating alarms when the script fails to start on the target system.
- Alarm on Start Error: specifies the alarm message name, which determines the alarm text and severity.

e2e_appmon > Alarm on Interval Breach

This section is used for configuring an alarm when the start time of script breaches the threshold limit for the start time.

e2e_appmon > Alarm on Process Kill

This section is used for configuring the alarm when the probe has to kill the script execution process. This probe kills the process because the script is not completed within the time limit.

e2e_appmon > Alarm on Disable

This section is used for configuring an alarm when the probe has to disable the script execution due to any reason. One such reason can be a script causing some potential security risk on the target system.

- Disable After: defines the time limit in seconds after which the probe disables the script execution.

e2e_appmon > Message Properties

This section displays a list of alarm messages that are defined in the E2E Application Response Monitoring probe. You can select and view the message details but cannot add or edit messages to the list. These alarm messages are specified for various alarm conditions, for example, when the script fails to start.

e2e_appmon > Profile Properties

This section displays a list of monitoring profile with their execution status in a table. Select a monitoring profile from the table and view its details.

- Name: identifies the monitoring profile name.
- ROB File: identifies the compiled script file name, which the profile executes for monitoring.
- Last Started: identifies the date and time value when the script was last executed.
- Running: indicates the script status, whether the script is currently executing.
- Times Used (Last Run in Seconds): identifies the time consumed during last script execution.
- Return Code (Last Run): identifies the return code of the script after last execution.

- Time Started (Seconds): identifies the number of successful script executions.
- Time Killed: identifies the number of times the probe has killed the script.
- Time Failed to Start: identifies the number of failed script executions.
- Maximum Run Time: identifies the maximum run time of the script among all executions.

Profiles Node

The **Profiles** node displays all the monitoring profiles which are created in the probe. This node does not have a section or field for configuration.

<Profile Name> Node

The *profile name* node represents the actual monitoring profile of the probe. This node lets you configure the profile-specific properties for monitoring.

Note: The profile name is user-configurable and is referred to as *profile name* in this document.

Navigation: e2e_appmon > Profiles > *profile name*

Set or modify the following values as required:

***profile name* > Profiles**

This section lets you activate or deactivate the monitoring profile.

***profile name* > Run Properties**

This section lets you configure the script and its execution properties.

- Compiled Script (ROB File): specifies the compiled script file, which the profile runs. The path of the **ROB Files Directory** field in the **Run Properties** section of the **e2e_appmon** node is used for fetching list of compiled scripts.
- Arguments: defines the parameters for executing the scripts.
- Maximum Run Time (Seconds): defines the maximum execution time of the script. The probe generates an alarm when the execution time exceeds this limit.

profile name > On Timeout

This section lets you configure the actions when the script execution time exceeds the limit.

- **Dump Screen on Timeout:** captures the snapshot of the application when the timeout occurs.

Default: Not selected

- **Kill Process on Timeout:** terminates the script execution process.

Default: Selected

profile name > On Error Return Alarm

This section lets you view the alarm details when the script generates an error after execution.

profile name > On Error Return

This section lets you configure the alarm, when the script generates an error after execution.

- **Expected Return Values=0:** expects 0 as the return code of the script. The probe generates the alarm, otherwise.

Default: Not selected

- **Dump Screen on Error:** captures and saves the snapshot of the application when the error occurs.

Default: Not selected

- **Alarm Message:** specifies the alarm message name, which identifies the alarm text and severity.

profile name > Cleanup on Timeout or Error Return

This section lets you specify another script, which runs when the profile script times out or returns an error. This script is used to delete erroneous actions for running other scripts.

profile name > Source Used for Quality of Service and Alarm Messages

This section lets you override the QoS and alarm source, which is configured in the script.

- **Override With:** enables the **Source** field for specifying the custom QoS and Alarm source.

Default: Not selected

- **Source:** defines the custom QoS and Alarm source.

profile name > Quality of Service Message

This section lets you activate the QoS on script execution time. You can also configure dynamic and static thresholds on this QoS.

profile name > Alarm on Start Error

This section lets you view the alarm details when the script fails to start on the target system.

profile name > Alarm on Interval Breach

This section lets you view the alarm details when the start time of script breaches the maximum start time threshold limit.

profile name > Alarm on Process Kill

This section lets you view the alarm details when the probe has to kill the script execution process.

profile name > Alarm on Disable

This section lets you view the alarm details when the probe has to disable the script execution due to any reason.

Note: The **Alarm on Start Error**, **Alarm on Interval Breach**, **Alarm on Process Kill**, and **Alarm on Disable** are configurable through the **e2e_appmon** node. You can only view the alarm details at profile level.

Script Node

The **Script** node lets you create independent and deployable script packages. These packages can be deployed on the target robot (similar to other probe packages) for monitoring. You can add multiple scripts and their dependent files to a deployable package.

Important! Refer to the **Limitations** section, which provides information to create and deploy the script packages.

Navigation: e2e_appmon > Script

Set or modify the following values as required:

Script > Scripts

This section lets you define the script package details.

- Name: defines a unique script package name. This name must be unique.

Note: Use the **Validate Package Name** option from the **Actions** drop-down list and verify the package name that meets the naming conventions.

- Version: defines the script package version.

Default: 1.0

- Description: defines a short description about the package functionality.

- List of Scripts: lets you select and move a script from the **Available** list to the **Selected** list for creating the script package. The list is available only when the probe is activated. You can add more than one script to the package file.

The script execution settings are inherited from the associated profile. If more than one profile is associated to a script, the script execution settings are inherited from the first monitoring profile. If there is no monitoring profile, the default execution settings are used.

Script > Path/Dependent Files

This section lets you add the dependent files, which you want to deploy on the target system with the script package. For example, the script is referring a .dll file during the execution. Use the **New** button for adding more than one file to the list.

- Rob File: specifies the ROB file for which you can define a dependent file.
- Path/Dependent Files: defines the dependent file name and path. Use the browse button for navigating to the correct path.

Note: Use the **Save** option from the **Actions** drop-down list for saving the dependent file details.

Limitations

The limitations of creating and deploying the script packages are as follows:

- Script that is recorded on a particular OS can only be deployed on that OS. For example:
 - If recorded on Windows 7, the script does not work on Windows Server 2012 R2 or Windows 8.
 - If recorded on the 64-bit robot, the script can only be deployed on the 64-bit robot.
 - If recorded on OS with IE9, the script can only be deployed on OS with IE9.
 - If recorded on OS, where a dependent file is on the E:\ drive, the target host must also have an E:\ drive present.
 - If recorded with the NimRecorder 3.9, the target host must also have the NimRecorder 3.9.
- The absolute path of dependent files is provided while recording the scripts.
- To create a script package, a profile is created for the scripts that are included in the script package. If there is no profile, a default profile is created with the **ROB** file name in the configuration file after deploying the package.

Note: As of now, you cannot edit the default settings.
- A profile and a script cannot have the same name on the target robot, otherwise either the profile name, or the script name is overridden. The robot does not display any warning message while deploying the probe.

- The dependent files can only be added from the **C:** drive. This process is in line with how current script path is configured in the probe.
- All the files are copied to the folders according to the robot on which script is recorded, after deploying the package.
- The robots, on which script package is deployed, must be running the e2e_appmon probe. The **Distribution Manager** deploys the script package without any validation.
- You cannot deploy only one script from a package (in case the package contains more than one script) to the robot.

Variables Node

The **Variables** node lets you define variables, which are used in multiple scripts. For example, you can put a global user name and password in the variable value which can be used in multiple scripts.

Navigation: e2e_appmon > Variables

Set or modify the following values as required:

Variables > Variables

This section lets you view the list of variables in the grid. Select any variable from the list and edit the variable value. You can also select the **Delete** option of the grid to remove the variable from the list.

Note: Use the **Options** icon next to the **Variables** node in the navigation pane for adding variables.

Variables > Quality of Service Variables

This section is used for mapping the QoS name of the earlier wintask probe with the e2e_appmon probe QoS. Use this section while migrating from the earlier wintask probe to the e2e_appmon probe.

Configure a Node

This procedure provides the information to configure a section within a node. Each section within the node lets you configure properties of the probe for managing script execution properties.

Follow these steps:

1. Navigate to the section within a node that you want to configure.
2. Update the field information and click **Save**.

The specified section of the probe is configured.

Manage Profiles

A monitoring profile specifies the script and its execution properties, which the probe runs on the target system. Being a system administrator, you can create more than one profile and can monitor multiple applications response. The alarm properties are identical for each profile as defined in the **e2e_appmon** node of the probe.

Follow these steps:

1. Navigate to the **e2e_appmon** node of the probe.
2. Define the user credentials under the **Run as User** section for executing the script. These credentials provide necessary rights to execute the script.
3. Define the default execution properties of the script under the **Run Properties** section.
4. Define the appropriate alarms under the **Alarm on Start Error**, **Alarm on Interval Breach**, **Alarm on Process Kill**, and **Alarm on Disable** sections.
5. Click the **Options** icon next to the **Profiles** node in the navigation pane.
6. Select the **Add New Profile** option.
7. Define the **Profile Name** and click **Submit**.

The probe creates a monitoring profile and displays the success message. The profile name appears under the **Profiles** node in the navigation pane.

8. Select the *profile name* node in the navigation pane and specify the script under the **Run Properties** section.
9. Configure other necessary properties of the monitoring profile under the **On Timeout**, **On Error Return**, **Source Used for Quality of Service** and **Alarm Messages** sections.

10. Configure QoS and dynamic or static thresholds under the **Quality of Service Message** section.

11. Click **Save**.

The probe starts executing the script on the target system for generating alarms and QoS messages.

Create a Deployable Script Package

As an application administrator, you are required to monitor application response on multiple systems using the same set of scripts. The E2E Application Response Monitoring probe lets you create a deployable script package. This script package can have more than one script with defined execution and alarm properties. You can deploy this package on any other NMS Robot and start monitoring.

Follow these steps:

1. Navigate to the **Script** node of the probe.
2. Define the following package details in the **Scripts** section:
 - Name
 - Version
 - Description
 - List of Scripts
3. Select the scripts from the **Available** list to the **Selected** list.
4. Select the **Validate Package Name** option from the **Actions** drop-down list. This option prevents you from creating a duplicate package with the same name and version.

5. Go to the **Path/Dependent Files** section and add the files, which are necessary for executing the script.

Note: Leave the **Path/Dependent Files** section blank when there are no dependent files.

6. Select the **Publish to Archive** option from the **Actions** drop-down list for creating a package.

The probe creates a deployable package in the **Archive** directory of the NMS and displays a success message on the probe GUI.

Note: You can deploy this package to any other NMS Robot where the E2E Application Response Monitoring probe is running.

Chapter 3: e2e_appmon QoS Threshold Metrics

Many probes ship with the default QoS threshold values. The default threshold values provide an idea of the type of values to be entered in the fields. These threshold values are not necessarily recommended values. To aid in tuning thresholds and reducing false-positive alarms, this section describes the QoS metrics and provides the default QoS thresholds.

This section contains the following topics:

[e2e_appmon QoS Metrics](#) (see page 23)

[e2e_appmon Alert Metrics Default Settings](#) (see page 23)

e2e_appmon QoS Metrics

The following table describes the checkpoint metrics that can be configured using the e2e_appmon probe.

Monitor Name	Units	Description	Version
QOS_E2E_EXECUTION	Milliseconds	Monitors the script execution time.	1.0

e2e_appmon Alert Metrics Default Settings

This section contains the alert metric default settings for the e2e_appmon probe.

Alarm Name	Warning Threshold	Warning Severity	Error Threshold	Error Severity	Description	Version
Disabled	None	None	None	Major	Profile is disabled by the probe.	1.0
Killed	None	None	None	Minor	Profile did not complete on time and had to be killed.	1.0
ReturnError	None	None	None	Minor	Script returned an unexpected value.	1.0
StartError	None	None	None	Minor	Profile is not able to start and execute the script.	1.0
TimeBreached	None	None	None	Warning	Profile does not start on time.	1.0

Chapter 4: E2E Application Response Monitoring (e2e_appmon) Troubleshooting v2.2

This section contains the following topics:

[NimRecorder is not Deployed or Installed](#) (see page 25)

[Scripts not Working Fine after Upgrade](#) (see page 26)

[Scripts not Working due to Synchronization](#) (see page 26)

NimRecorder is not Deployed or Installed

Symptom:

I am using the e2e_appmon probe deployed on my robot. However, the NimRecorder is not getting deployed to record new script and to edit existing scripts. The probe is deploying the script executor only.

Solution:

Instead of deploying the e2e_appmon probe, deploy the e2e_appmon_dev (1.91 or later) probe. The NimRecorder is deployed automatically.

OR

For the e2e_appmon probe (standard edition), install the NimRecorder manually.

Follow these steps:

1. Go to **Start > All Programs > Nimsoft Monitoring > E2E Scripting** and select the **Uninstall NimRecorder**.
The Script Executor is removed.
2. Go to **[Nimsoft Installation drive] > Nimsoft > probes > Application > e2e_appmon > Install** folder.
3. Double-click the **nimrecorder.msi**.

Note: The name of .msi installer file is based on the NimRecorder version. For example, it is **nimrecorder50.msi** for installing the NimRecorder 5.0.

4. Follow the instructions and install the NimRecorder.

The help file of the NimRecorder is accessible at **Start > All Programs > Nimsoft Monitoring > E2E Scripting > Help** after installing the NimRecorder.

Scripts not Working Fine after Upgrade

Symptom

My existing scripts have stopped working after upgrading the probe.

Solution

Recompile your existing scripts with new NimRecorder and make them compatible with new NimRecorder.

Drop an email to info@wintask.com for troubleshooting your script-related issues and support on NimRecorder 5.1.

Scripts not Working due to Synchronization

Symptom

My existing scripts have stopped or lose focus while running the script through NimRecorder because of synchronization of text.

Solution

The **UsePage** and **Web** functions are used for synchronization process. The Internet Explorer (IE) uses the window name instance for the bitmap synchronization and it varies with each IE execution. Therefore, the wintask recommendation is to use the **TopInstance** function of NimRecorder instead of providing a constant instance number.

For example:

```
Pause 10 secs until
  Bitmap("test.bmp")
  InWindow("IEXPLORE.EXE|Internet Explorer_Server|WinTask_x64 Help - Internet Explorer|1",2)
  InArea( 101, 172, 26, 41 )
```

For Topinstance, the constant instance number can be replaced with:

```
Pause 10 secs until
  Bitmap("test.bmp")
  InWindow("IEXPLORE.EXE|Internet Explorer_Server|WinTask_x64 Help - Internet Explorer|1",Topinstance())
  InArea( 101, 172, 26, 41 )
```

Drop an email to info@wintask.com for troubleshooting your script-related issues and support on NimRecorder 5.1.