

Failover Solutions in WhatsUp Professional 2006

Whitepaper

Unlike previous versions of WhatsUp, WhatsUp Professional/Premium 2006 has failover solutions built into the functionality of the base application. While these solutions may require some knowledge of SQL and the more advanced features of the WhatsUp application, the information provided in this paper should help solve any issue that may arise while finding a solution that fits your needs.

Introduction

When monitoring business critical devices, you cannot afford to be 'in the dark' when it comes to the state of those resources. If and when something goes down, you need to know about the problem before you can fix it. As someone who uses or is finding out more about WhatsUp Professional, you recognize the need for network monitoring, and know how important that need is. So what happens when the problem device on your network is the computer running WhatsUp Professional itself? Even though you've set everything up correctly, if that computer crashes or loses its network connection, you will be completely in the dark again. That's why a Failover/Fault Tolerance solution is important.

The terms Failover and Fault Tolerance refer to second-tier or backup network monitoring. When the primary installation of WhatsUp Professional is no longer responsive on the network, a second installation on another computer takes over the monitoring for the primary. In the past, WhatsUp FT was purchased and installed with WhatsUp Gold to provide this solution. Device Maps and configuration data were transferred from the primary to the secondary on a regular basis, and when the primary went down, the secondary would use those maps and data files to take over monitoring. This approach does not work with WhatsUp Professional, as WhatsUp Professional uses a relational database, MSDE or Microsoft SQL Server, to store all configuration and device data.

WhatsUp Professional 2006 has Fault Tolerance solutions built into the functionality of the application. This paper describes two possible solutions that should provide 24/7 monitoring for most WhatsUp Professional configurations. Each solution provides a different type and amount of coverage, but each is built on the same foundation.

The solutions here are simple solutions that all users should be able to configure and use. These are not 'complete' failover solutions, in that no databases are exchanged, and monitor data is not synchronized between systems. These solutions are to give 24/7 monitoring coverage to your devices, since a failover state should always be a temporary situation.

Running Multiple Installs

For both of the solutions in this paper, you must be running two copies of WhatsUp Professional concurrently. To do this, you will need to purchase two activation codes from our sales department. If you state that the second copy will be used in the failover role, you may qualify for a discount on the second activation code.

Create the Foundation

Once you have completely configured your primary system, you are ready to build your failover foundation. This foundation is made up of a regularly scheduled automatic backup of your primary WhatsUp Professional database. Since the database is the heart of the application's functionality, it is necessary to make sure the data stored there is as up-to-date and safe as possible. Imagine what you would lose if the hard drive the database is stored on failed to load. You would lose all of your configuration information, all of your historical data, and any customized feature you created during the setup and operation of the application.

- 1. Decide how often you want to back up your database. The real decision is to decide how long you are willing to go without data. If you only back up the database once a day, you have the potential of losing 23 hours worth of data.
- 2. Create a .bat file to store the backup osql script. The batch file should be located in the MSDE program installation directory. The default location is C:\Program Files\Microsoft SQL Server\80\Tools\Binn.

10 Maguire Road Lexington, MA 0242⁻ (781) 676-5700

Zekeringstraat 17 1014 BM, Amsterdar The Netherlands





The following is an example of how the osql script works with the default WhatsUp Professional installation: OSQL -E -n -D WhatsUp -Q "BACKUP DATABASE WhatsUp TO DISK = 'C:\Program Files\Microsoft SQL Server\MSSQL\$WHATSUP\Data\WhatsUp.dat' WITH INIT"

3. Create a recurring action to execute the .bat file. Go to the Action Library and configure a Program action to execute the .bat file. Then go to the Recurring Action dialog and schedule that action to execute based on the frequency of your choosing.

Now that the foundation has been established, you are ready to take the next step in configuring your failover solution.

Configuring Your Solution

The following solutions offer different amounts of failover coverage, with different levels of difficulty in setting up.



In each solution, there is a device on the primary dedicated to monitoring the secondary, and a device on the secondary monitoring the primary. These devices are the keys to the entire failover system, and must be configured properly to ensure complete coverage in the case of the failure of the primary.

Solution 1

In this solution, the specific key devices are configured on the secondary system, and are only monitored when the primary system is down. When the primary system goes back up, the devices are no longer polled.

Use this solution if:

- · You only want to monitor key devices while in the failover state.
- · You only want to monitor key devices while in the failover state.

10 Maguire Road Lexington, MA 02421 (781) 676-5700

Zekeringstraat 17 1014 BM, Amsterdam The Netherlands





Pros: •Configuration difficulty is easy. •24/7 coverage on key devices. •Requires little bandwidth.

Cons: •Does not provide complete coverage of network. •Performance Data should be turned off.

How to configure this solution:

For 'how-to' information on specific topics, such as creating a device group, or configuring active monitors, please refer to the WhatsUp Professional Online Help.

- 1. Once your primary system has been configured for normal operation, install WhatsUp Professional on your secondary system.
- 2. On the secondary system, create a new device group called Key Devices.
- 3. In the Key Devices group, create new devices for each of the key resources you need to monitor when the primary goes down.
- 4. Configure those devices with the same monitors and actions you have in the primary.

Note: Be sure to disable Performance Monitoring on these devices during configuration. Since this solution uses the dependencies feature, you will constantly gather performance data, even when you are not actively using the secondary system for monitoring.

- 5. In the top level of the device list, create a device for the primary system.
- 6. Make sure that Ping is the only monitor that is configured on the primary system device.
- 7. Place an action on this device that alerts you when the primary goes down, and the failover state begins.
- 8. Place an action on this device that alerts you when the primary system is up again, and the failover state ends.

Device Properties : Primary	Pro 2006			X
Properties	Actions			
General Performance Monitors	C Apply this Action Polic	1.		
Active Manitors		OR		
Passive Monitors	Apply individual action	s.	-	1
Actions	State Change		Action to Perform	<u>A</u> dd
	Up		Primary Maintenance Mode Email on Up	<u>E</u> dit
Credentials				Remove
Polling				
Notes				
Menu Menu				
Attributes			OK Cancel	Help

10 Maguire Road Lexington, MA 02421 (781) 676-5700

Zekeringstraat 17 1014 BM, Amsterda The Netherlands



9. Right-click the device group (or groups) in the device list and select **Bulk Field Change > Down Dependency**.



10. On the Down Dependency dialog, select **Check only if the following device is down**, then use the browse (...) button to select the primary device created in step 6.

Bulk Field Change: Down Dependency		×
Check only if the following device is down:		ОК
Primary Pro 2006	<u> </u>	Cancel
		Help

Solution 2

In this solution, all of the key devices are in maintenance mode on the secondary system until the primary device goes down. When the device goes down, the Active Script action fires turning off maintenance mode to all of the devices. When the primary goes back up, maintenance mode is turned back on.

Use this solution if:

· You need to monitor performance data during the failover state.



Pros:

24/7 coverage on key devices. Requires little bandwidth. Performance Data can be collected while in Failover state.

Cons:

Does not provide complete coverage of network, unless you choose to configure all devices in the secondary.
Requires some scripting knowledge (or at least exposure.)

Lexington, MA 02421 (781) 676-5700

Zekeringstraat 17 1014 BM, Amsterdan The Netherlands





How to configure this solution:

For 'how-to' information on specific topics, such as creating a device group, or configuring active monitors, please refer to the WhatsUp Professional Online Help.

- 1. Once your primary system has been configured for normal operation, install WhatsUp Professional on your secondary system.
- 2. On the secondary system, create a new device group called Key Devices.
- 3. In the Key Devices group, create new devices for each of the key resources you need to monitor when the primary goes down.

Note: If you have more than one device group you want to monitor during failover, these groups must be placed in the Key Devices group.

- 4. Configure those devices with the same monitors and actions you have in the primary.
- 5. In the top level of the device list, create a device for the primary system.
- 6. Make sure that Ping is the only monitor that is configured on the primary system device.
- 7. Run the following osql command from a cmd window:

osql -E -D WhatsUp -Q "SELECT nDeviceGroupID FROM DeviceGroup WHERE sGroupName = 'Key Devices' "

This number is used at line 13 in the Active Script shown below, to identify the device group.

8. In the Action Library, create a new Active Script action that places the devices in the Key Devices group in or out of maintenance mode when executed.

dit Active Script Action		
Name:		
Primary Maintenance Mode		
Description:		
Maintenance Mode for Failover		
Imeout: Script tyge: 40 → (seconds) JScript ▼		
Script Text:		
<pre>// // Get our WhatsUp PRO DB, before using be sure it is NOT null // Get our WhatsUp PRO DB, before using be sure it is NOT null var oDb = Context.GetDB; // You can either hard code the group ID or we can get the value via an SQL statement // on the DeviceGroup table. Be sure your SQL works on more than just the name, you will need // to be sure of the parent group as well because groups could have the same name. For my // example I am hard coding, be sure to change this value to use the group ID in your setup. var nPrimaryGroupID = 6; // Get the trigger condition. We are interested in Device Up (32) and Device Down (64). // If we get the Device UP then we turn maintenance on for the group, on down we turn maintenance // ar DEVICE_UP = 32; var nTriggerCondition = Context.GetProperty(TriggerCondition"); var sErrorMessage = ""; // Ket Mathematic Mathemat</pre>		
/// Function returns the device ID's for a group. /// function getDeviceIDsInGroup(arrDeviceIDs, nGroupID) { // Validate the DB object if (null == oDb) {		ОК
return arrDeviceIDs;	>	Cancel Help

The complete syntax for this code can be cut and pasted from the following example. Once you have pasted it into the Active Script window, follow the directions commented in the code to make sure you are calling the correct device group name/ID. This file can be downloaded at:

http://www.ipswitch.com/support/whatsup_professional/guides/2006/primarymm.txt:

ekeringstraat 17 014 BM, Amsterdai de Netherlands

// This script will turn maintenance on or off based off the TriggerCondition property

//

11



```
// Get our WhatsUp PRO DB, before using be sure it is NOT null
var oDb = Context.GetDB;
     You can either hard code the group ID or get the value via an SQL
11
statement
     on the DeviceGroup table. The ID number is hard coded in this exam-
11
ple. Be sure to change this value to use the group ID in your setup. See
Solution 2 - step #6 in the White Paper for more information on how to do
this.
var nPrimaryGroupID = 6;
11
      Get the trigger condition. We are interested in Device Up (32) and
Device Down (64).
     If we get the Device UP then we turn maintenance on for the group,
11
on down we turn maintenance
11
     off.
var DEVICE UP = 32;
var DEVICE DOWN
                   = 64;
var nTriggerCondition = Context.GetProperty("TriggerCondition");
var sErrorMessage = "";
11
11
     Function returns the device ID's for a group.
11
function getDeviceIDsInGroup(arrDeviceIDs, nGroupID)
{
      11
            Validate the DB object
      if (null == oDb)
       {
             return arrDeviceIDs;
       }
      var sSql = "SELECT bDynamicGroup FROM DeviceGroup WHERE
nDeviceGroupID = " + nGroupID;
      var oRs = oDb.Execute(sSql);
       if ( !oRs.EOF )
       {
             if(oRs("bDynamicGroup") == 1)
             return arrDeviceIDs;
       }
      sSql = "SELECT DISTINCT Device.nDeviceID "
             + "FROM Device "
             + "INNER JOIN PivotDeviceToGroup ON Device.nDeviceID =
PivotDeviceToGroup.nDeviceID "
             + "WHERE PivotDeviceToGroup.nDeviceGroupID = " + nGroupID;
```

10 Maguire Road Lexington, MA 02421 (781) 676-5700



```
oRs = oDb.Execute(sSql);
      while(!oRs.EOF)
       {
             arrDeviceIDs[arrDeviceIDs.length] =
parseInt(oRs("nDeviceID"));
             oRs.MoveNext();
       }
      var oGroupRs = oDb.Execute(""
             + "SELECT DISTINCT nDeviceGroupID "
             + "FROM DeviceGroup "
             + "WHERE nParentGroupID = " + nGroupID
             );
      while(!oGroupRs.EOF)
       {
             arrDeviceIDs = getDeviceIDsInGroup(arrDeviceIDs,
parseInt(oGroupRs("nDeviceGroupID")));
             oGroupRs.MoveNext();
       }
      oGroupRs.Close();
      oRs.Close();
      return arrDeviceIDs;
}
function StartProcess()
{
       Context.NotifyProgress( "The trigger state is " +
nTriggerCondition);
             Only work on these two trigger states.
      11
      if (nTriggerCondition == DEVICE UP || nTriggerCondition ==
DEVICE DOWN)
       {
             var bMaintenance = true; // assume maintenance should be
turned on
             if (nTriggerCondition == DEVICE DOWN)
             {
                    bMaintenance = false;
             }
             var arrDeviceIDs = new Array()
             arrDeviceIDs = getDeviceIDsInGroup(arrDeviceIDs,
nPrimaryGroupID);
             sErrorMessage = "Processing " + arrDeviceIDs.length + "
devices";
```

if (arrDeviceIDs.length > 0)

IPSWITCH



```
{
                    11
                           Use the WhatsUp PRO helpers
                    var oUtility = new ActiveXObject("CoreAsp.Utility");
                    var oEventHelper = new
ActiveXObject("CoreAsp.EventHelper");
                    for(var i = 0; i < arrDeviceIDs.length; i++)</pre>
                     {
                           oUtility.SetDeviceMaintenanceMode(bMaintenance,
arrDeviceIDs[i]);
                           oEventHelper.SendChangeEvent(2, arrDeviceIDs[i],
1);
                    }
              }
       }
       else
       {
              sErrorMessage = "Trigger state " + nTriggerCondition + " was
not proceessed.";
       }
       return 0;
}
//
11
       Start of the process.
11
11
       Test the trigger to be sure if we are in the right state.
var nReturn = StartProcess();
Context.NotifyProgress( sErrorMessage);
Context.SetResult(nReturn, sErrorMessage);
```

- 9. Place an action on this device that alerts you when the primary goes down, and the failover state begins.
- 10. Place an action on this device that alerts you when the primary system is up again, and the failover state ends.
- 11. Assign the Primary Maintenance Mode action to the primary device, associating it with an up state.
- 12. Assign the Primary Maintenance Mode action to the primary device, associating it with a down state. You may want to use the Down for 5 minutes state to give you time between when you receive the down alert and when the failover state begins.

10 Maguire Road Lexington, MA 02421 (781) 676-5700

Zekeringstraat 17 1014 BM, Amsterdan The Netherlands



Properties	Actions			
General Performance Monitors	C Apply this Action Policy,]
Active Monitors		OR		
Passive Monitors	• Apply individual actions.			
Actions	State Change	Action to P	erform	<u>A</u> dd
Credentials	Down at least 5 min Down at least 5 min Up	Primary Mair Email on Do Primary Mair Email on Up	Primary Maintenance Mode Email on Down Primary Maintenance Mode	
Polling	- CP	Linal of op		
v				
Notes				

Other Solutions?

These solutions may not solve everyone's needs when it comes to failover and fault tolerance, but we know that given these instructions, many of our users will figure out ways to solve their particular problems with subtle tweaks of these solutions, or to invent new solutions on their own. We are currently working on other solutions ourselves, so as we come up with, or are informed of new solutions, we will test those and publish future papers.

10 Maguire Road Lexington, MA 02421 (781) 676-5700

Zekeringstraat 17 1014 BM, Amsterdam The Netherlands

Copyright © 2006, Ipswitch, Inc. All rights reserved. WhatsUp is a registered trademarks of Ipswitch, Inc. Other products or company names are or may be trademarks or registered trademarks and are the property of their respective holders.