4D Server Health – Automatic Reporting

By Thomas Maul, General Manager, 4D Germany.
TN 07-05

Introduction

This Technical Note demonstrates how to build an automatic 4D Server "health" reporting system. A sample database is included that shows how to create daily reports that can be sent via email or served as a Web page. This system allows the database administrator to detect possible problems in advance as well as periodically monitor the state of the server.

Note that the example code uses functionality that is already built into 4D to gather the report information. The only plug-in used is 4D Internet Commands, for emailing.

The Server Health Report

This section of the Technical Note describes the various report sections that are created using the example code. The example report data presented here was gathered on a Windows machine.

Server Overview

4D Version	F0010805
Last start	2006-10-10 18:08:10
Operating System	Windows™ XP 5.1.2600 Service Pack 2
Last reboot	20061013 03:00
Physical memory	2.062.764 KB
Phy mem free	850.416 KB
Virtual memory	1.837.456 KB
Virt mem free	23.244 KB
Last backup	13.11.2006 02:00:00
Backup Status	No detected error.
Next backup	14.11.2006 02:00:00

This section shows information about the 4D version in use (from the command **Application version**) as well as when 4D Server was last started (in this example October 10, 2006, 6pm; this particular date is always in international format).

Additionally this section shows information about the operating system in use (including Service Pack information), the last reboot reported from OS, and memory conditions.

The last line shows the 4D backup status (date formatted on a German system).

Critical errors in last 48 hours

Computer	User		Log on
SALES-XP	John Smith		2006-11-10 16:01:14
	John	Windows™ XP	5867
	History - previous Crashes		
	SALES -XP	John Smith	2006-11-10 14:35:43
	SALES -XP	John Smith	2006-11-08 14:41:57
	SALES -XP	John Smith	2006-11-03 14:10:38
	SALES -XP	John Smith	2006-11-02 10:17:52

This next section is generated if any critical errors have been detected. If the 4D Server machine or any of the Client computers did not quit the application using the quit command, it is added to the critical errors. This section only appears if the critical error happened in the last 2 days. If a computer crashes, a history of the last problems is also included.

The example above shows that the computer with the computer name (from the operating system) "SALES-XP", with operating System Windows-XP build 5687 connected to the server, using the Windows operating system account name "John", and with the 4D user account "John Smith" (this could be from the 4D password system or your own) did not quit using the quit command. Note that the log only shows the time of the last successful login, not of the crash itself.

This example also shows that the same user on the same computer had another crash shortly before, and also had several problems the last few days.

In this example, because the report does not show crashes from any other computers, it is obvious that the problems are either related to the computer itself or to the user (maybe “John Smith” is using a special part of the application nobody else is using). If crashes from multiple clients had been logged, or for the server itself, then it can be surmised is that the problem is on the server side.

Hard disk usage

Name	Total	Free	Yesterday	Last month	Last year
C:System	244.187.968	224.300.208			
E: Backup	244.187.968	185.682.268			-5%

This section lists the connected hard disks, their total space and the currently free space, in Kilobytes. The historical columns “Yesterday”, “Last month”, and “Last year” are only used if the difference from the current value is 1% or more.

Table Usage

Table	Records	Used space	Yesterday	Last month	Last year
autologbook	14,528				10%
Code_Data	71			5%	7%
Forums	307				11%
Messages	117,424			1%	23%
ServerHealth	124			1%	33%
URL_Encoder	165				34%
URL_Logbuch	11,832			5%	219%
Word_list	202,584				12%

This section lists all tables of the database with the current count of records. The column "Used space" is the ratio max record to current records; fully grey is 100% used. If there are records deleted, the address table contains empty space, which is shown here. If the table contains many records and there was a major delete operation, some operations will become slower (like ALL RECORDS), so using 4D Tools to compact the data should be considered. This information can also be used to detect large delete operations, which may happen accidentally.

The historical columns show the increase/decrease of the tables.

Process usage

Process	CPU-Time	Memory	Yesterday	Last month	Last year
System Idle Process	3123:11:40	28	75%	75%	75%
4DServer.exe	517:47:43	469.552	-4%	-5%	-2%
explorer.exe	49:07:55	18.388	-18%	-27%	55%
retroclient.exe	40:50:31	4.936	3%	3%	38%
System	23:44:29	236	11%	11%	68%
services.exe	08:17:24	9.036	54%	54%	139%
csrss.exe	05:02:05	5.424	8%	6%	60%
winlogon.exe	02:57:05	13.500	26%	22%	40%
lsass.exe	00:57:18	5.112	3%	2%	2%
wmiprvse.exe	00:26:03	8.320	2%	2%	4%
smax4pnp.exe	00:13:01	6.036			
cscript.exe	00:07:49	9.588	2%	2%	-10%
VTTrayp.exe	00:07:49	4.320			
svchost.exe	00:05:13	6.124	5%	5%	243%
logon.scr	00:05:13	2.028	3%		12%
wscntfy.exe	00:05:13	2.716			
smss.exe	00:02:36	556	5%	3%	63%
spoolsv.exe	00:02:36	7.716	6%	6%	101%
remotsvc.exe	00:02:36	1.504	-41%	-41%	254%
alg.exe	00:02:36	4.608	2%	3%	270%

VTTimer.exe	00:02:36	2.104			
SMax4.exe	00:02:36	3.064			

This section lists all processes running on the Server machine. The content depends on the operating system and installed software. This information can help the administrator get an idea of what other software is running on the 4D Server machine. One particularly useful piece of information is it shows how much CPU time and memory is used from other tasks. Note that the CPU time is the cumulated time since last reboot of the computer, while the used memory is the current usage, which may change during the day/usage. While the CPU time gives a good impression how the machine is used, the memory only gives an idea about the current situation while the snapshot is taken. Still it allows a quick way to see if 4D Server got enough memory assigned or if other applications are using a huge amount of memory.

The historical columns compare the memory situation; they show the values in red if there is a major increase compared to the previous day.

Database segment usage

Segment	Size	Yesterday	Last month	Last year
C:\4D Online\4D_Online.4DD	682.944			

This table gives an idea of the size of the database. All database segments will be listed here. The historical columns give an idea of the growth speed of the database. The database of this example only had a minor growth, so no values are displayed.

Example database

To test the reporting functionality with the example database, first launch with a single user 4D application (e.g. 4th Dimension). A prompt should appear and a subsequent dialog that can be used to enter email server and user data. You need an email user account that can send and receive to test this application (you can use the same account for both). The example supports SMTP Authorization, if this is required from your mail server.

After entering the required email information, quit the single user 4D application and launch the example database with 4D Server. For testing purposes you may wish to use some clients to connect and disconnect to the server. You may also "crash" one (use force quit/end task), then login with the same client again. Note that this example does not use a password system (to make testing easier), so the logged user names do not make real sense.

You can aslo start the web server on the server machine and test using a browser. The URL will be something like:

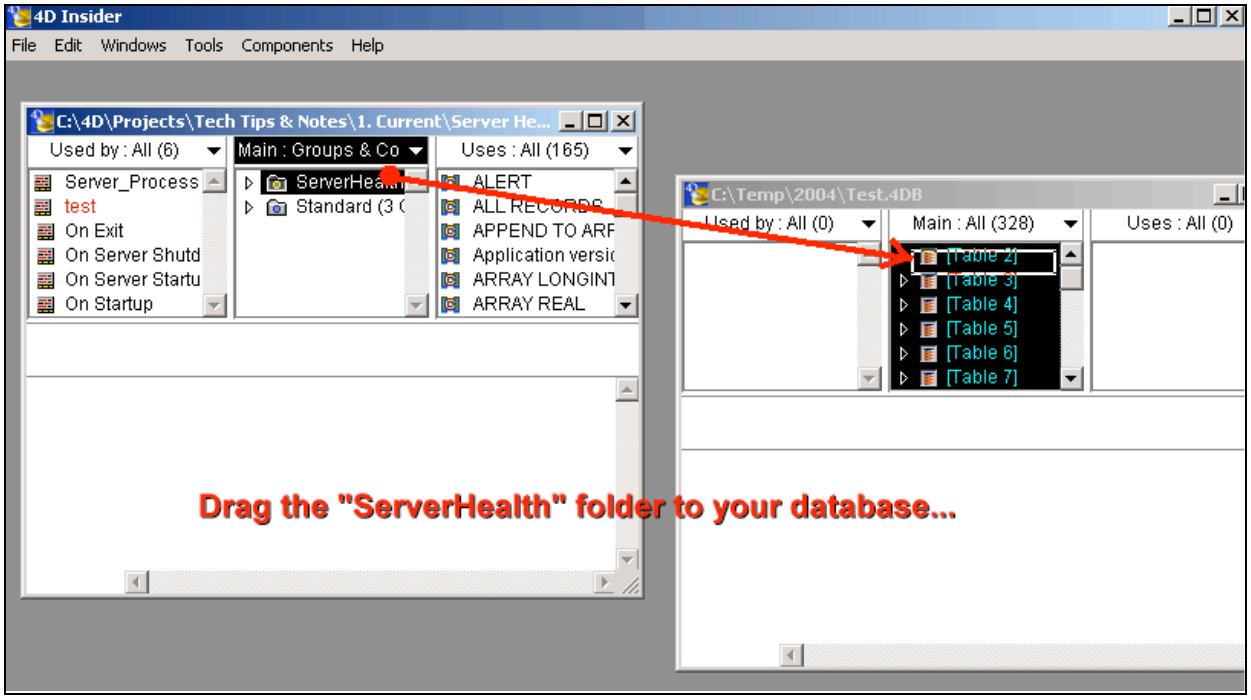
http://<IP of Server>/4Daction/serverhealth

Let the server run over night to test the nightly email. If you do not want to wait, execute the method "test" from one of the 4D Clients (or you can execute this method with the database running in 4D single-user).

Installation

The example database is based on the 4D 2004 example "4D Invoice" (from the 4D Product Line CD). The 4D Server Health code has been added to this database.

To install the Server Health code in your application, simply open both the example database and your own database with 4D Insider. You can then drag the group "ServerHealth" into your database (or do it manually by copy & paste):



This will copy six project methods:

ServerHealth	to access the report through browser
ServerHealth_Create	to be called on startup/exit/night
ServerHealth_Report	to send the report as email
HTMLMail_Send	from Tech Note 05-37
Compiler_HTMLMail	Compiler directives
Compiler_ServerHealth	Compiler directives

One Choice List:

ServerHealth_Text	includes localizable text

One table (including standard forms):

ServerHealth	for historical data (12 months)

In addition you have to manually copy the content of these two folders (located next to the example structure file):

Extras	HTML templates for the report
Plugins	contains 4D Internet commands 2004.5

Please note that 4D Internet Commands 2004.5 or newer is mandatory.

Of course you can modify the HTML templates as needed to change the look and feel of the report templates to meet your requirements.

In addition some fields or variables to store the following preferences are needed:

Email Server Hostname	SMTP server used to send the report
Email Sender	the address from which to send the report
Email Sender password	the Sender's SMTP password
Email Receiver	the report recipient

If your application already includes email handling, you probably already store this information. Otherwise create the fields in your Preferences table, following the example table [Parameter] in the example database.

The next step is to add code to On Startup/On Exit and On Server Startup/On Server Exit, which will log each start/quit from an application. These changes allow the reporting code to detect client and/or server crashes. If there is a startup entry without an exit, the application did not correctly quit.

Method On Server Startup

```
ServerHealth_Create ("StartServer")
$p:=New process("Server_Process";256000;"Server Background Process")
```

Note: if you already have a background process running on the server machine, you do not need to start another process, simply modify your method to call the reporting code (see below).

Method On Server Shutdown

```
$p:=Execute on server ("ServerHealth_Create";64000;"ServerHealth";"QuitServer")
```

Note: this must be done in another process, because On Server Shutdown cannot

create new records.

Method On Startup

```
ServerHealth_Create ("StartClient";Current user;"Thomas on Windows XP")
```

If you use your own password system, enter the user name in the 2nd parameter. The 3rd parameter should give information about the used computer. You can use – if existing – your own code to receive that or the example code:

```
PLATFORM PROPERTIES($Plattform;$System;$Maschine;$language)
$OS_Version:=ServerHealth_Report ("$$ShowOSVersion")
$userdata:=Current machine+Char(9)+Current machine owner+Char(9)+$OS_Version+Char(9)+
String($Maschine)+Char(9)+String($language)
ServerHealth_Create ("StartClient";Current user;$userdata)
```

Method On Exit

```
ServerHealth_Create ("QuitClient";Current user)
```

This logs the quit event for the client.

Finally we need a method running on the server which collects periodically, usually during the night, the current status of the server machine. If you already have a background process, use that one. Otherwise create a method named "Server_Process" with these contents:

```
READ ONLY([Parameter])
DELAY PROCESS(Current process;60*10)  ` wait 10 seconds after server start before doing anything

$currentdate:=Current date
While (Not(Process aborted))  ` automatically stops on Quit
   If ($currentdate#Current date)  ` a new day!
      $currentdate:=Current date
      ServerHealth_Create ("CreateStatistic")
      ALL RECORDS([Parameter])
      ServerHealth_Report ([Parameter]Report_Receiver;[Parameter]SMTP_User;
-> [Parameter]SMTP_Server; "4D Server Report";[Parameter]SMTP_User;[Parameter]SMTP_User_Password)

   Else
      ` do any other needed statistics/works here.
      ` You can set the delay to a shorter value or longer value as needed.
      DELAY PROCESS(Current process;60*60)  ` check again in 60 seconds
   End if
End while
```

The method runs in an endless loop until the server quits. Once a day it creates the new statistically data and then sends it as an email.

Other Usages

Besides sending the report as email, it could also be accessed in other ways, like as Widget/Dashboard or via a Web browser.

The example database supports access through a web browser. Start the Web server and access it (on the local machine, else use your IP-address/host name) for example:

http://localhost/4daction/serverhealth

Also note that the emails sent from the example databse are only in HTML format, there is no plain text version.

Summary

The example database presented in this Technical Note demonstrates an excellent enhancement to any 4D Server application by providing a professional daily report of the server's health.