How to setup and use VSE Health Checker

Retrieval of VSE system parameters
Rule-based evaluation

Last formatted on: Friday, November 19, 2010

Joerg Schmidbauer
jschmidb@de.ibm.com

Dept. 3229
VSE Development
IBM Lab Böblingen
Schönaicherstr. 220

D-71032 Böblingen
Germany

Disclaimer

This publication is intended to help VSE system programmers setting up infrastructure for their operating environment. The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk. Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment. Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of other companies:

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.
Microsoft, Windows, Windows XP, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

Contents

Changes:

October 2010 – initial version.

1 Introduction

This paper describes the setup and use of the VSE Health Checker. The VSE Health Checker is a system diagnosis utility to retrieve, display, and analyze performance relevant data from a VSE system. It can be used to detect VSE configuration problems, which often result in decreased performance or even system outages. Gathered data can be exported and imported in XML format. A health check is performed on the basis of a snapshot of the configuration of a given VSE system. There is no automatic measuring of the system status in fixed time intervals.

VSE data is retrieved by sending console commands, submitting VSE/POWER jobs, downloading VSE Librarian members, and invoking CICS transactions. There is no prerequisite to any vendor tools. The output is transferred to the workstation, parsed, and displayed in the GUI for further analysis.

While on the one hand you could retrieve all data also manually, the advantages of using the Health Checker are:

- Native textual output from VSE can be displayed using graphics
- Less important output from native commands can be omitted
- Output from multiple VSE commands can be displayed together
- Additional information can be calculated from several native outputs

The Health Checker does not change any system parameters on VSE. All actions only read data from VSE. Although the whole process of getting data from VSE can take up to several minutes, this is just elapsed time. CPU overhead is minimal.

The following software has been used in the test setup.

- z/VSE 4.2.2
- TCP/IP for VSE/ESA 1.5F
- Java 1.6.0_18 from Sun Microsystems
- VSE Connector Client on the workstation side
- VSE Health Checker, update from xx/2010 on the workstation

2 Prerequisites

Although the Health Checker in theory works with systems from VSE/ESA 2.6 with PQ88809 / UQ88864 or higher, we recommend using it with z/VSE 4.1 or later. Some functionality (e.g. security related data) is only available with z/VSE 4.2 or later. So the prerequisites are:

- z/VSE 4.1 GA or higher
- TCP/IP running on VSE
- VSE Connector Server running on VSE (job STARTVCS)
- VSE Connector Client installed on workstation side
- Java 1.5 or higher
- The STAT transaction must be defined in order to obtain CICS TS status data
- The CHKT transaction must be defined in order to obtain a list of TS queues

3 Restrictions

Currently the following restrictions apply.

- CICS/VSE partitions cannot be fully supported, because it's not possible to invoke transactions via the operator console, e.g. MSG F2,DATA=CEMT I SYSTEM. You can specify CICS/VSE partitions when defining a host in the GUI, but data will be incomplete.

- The TCP/IP plugin only supports parameters of the Connectivity Systems, Inc. TCP/IP stack. Other VSE TCP/IP stacks can be used to connect to VSE and retrieve data, but parameters cannot be displayed.

- The Health Checker relies on parsing the output of console commands and VSE/POWER jobs. When using OEM products like MLOG, which modify the output of commands, the Health Checker is no longer able to parse the output. See section MLOG considerations on page 21 for how to solve the problem.

- Security Manager related parameters are only supported for the IBM provided Basic Security Manager (BSM). External Security Managers (ESM), like CA TopSecret, or BIM Alert are not supported.

- When list queue output is automatically spooled to a VM/CMS user, the Health Checker can't access this output for parsing. Examples for this are the CICS statistics and the CICS TS queue list. The LST class parameter on the host configuration box has been introduced to overcome this problem, see section “Initial Health Checker setup” Introductionon page 5 for details.

- The list of programs loaded into the DSA is taken from the CICS statistics output. Therefore it only shows programs that were loaded during the CICS statistics interval. Make sure to set the interval to the maximum possible value (23h 59min).

4 Installing the prerequisite programs

The VSE Health Checker requires the installation of the VSE Connector Client. You can download the most current versions of the VSE Connector Client and VSE Health Checker from

http://www.ibm.com/systems/z/os/zvse/downloads/

After downloading and unpacking the zip-file, execute one of the contained install scripts:

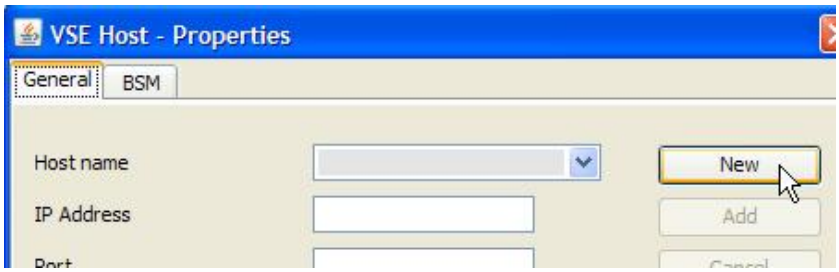
- Setup.bat – for Windows
- Setup.cmd – for Windows NT
- Setup.sh – for Unix / Linux

5 Initial Health Checker setup

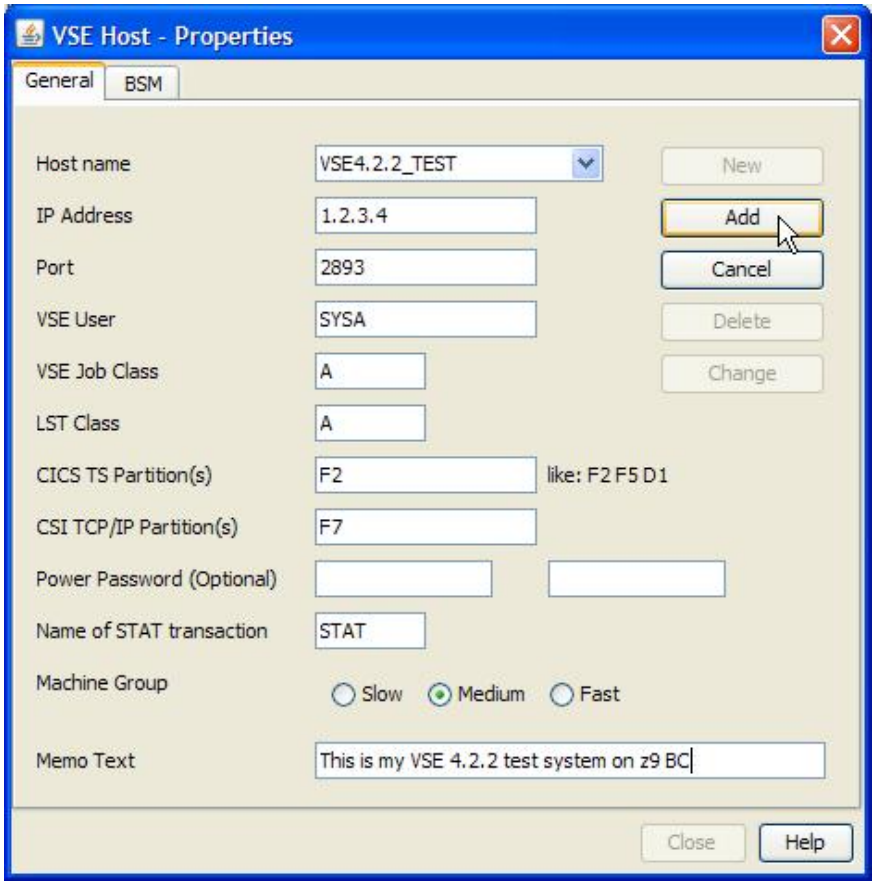
Before you can retrieve any data from your VSE system you must make it known to the Health Checker.

When starting the Health Checker the first time, the GUI comes up with no VSE system defined. To define your first VSE system, click on the “VSE Host properties” button.

On the "VSE Host properties" box press **New** to add a new VSE system.



Then enter the properties of your VSE system.

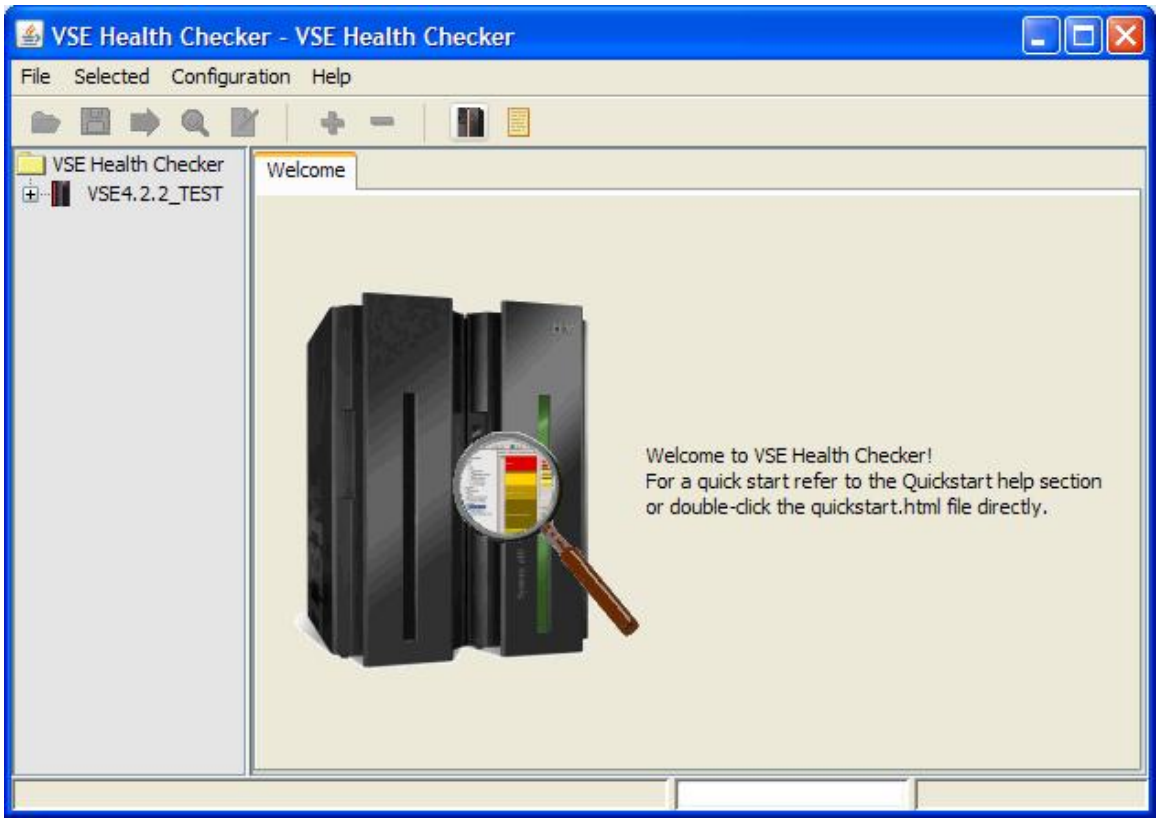


Press **Add**.

Note: if you are spooling LST queue output to another system, e.g. VM/CMS, use the "LST class" parameter to specify a LST class that is not started for transferring output. This allows keeping output from jobs submitted by the Health Checker in the LST queue, where they get processed by the Health Checker.

Then press **Close**.

The Health Checker main window now displays a new tree node for your VSE system.



The Health Checker is now ready for use. However, to get the full functionality, some additional programs and transactions must be installed and configured on the VSE side.

6 Installing and configuring additional programs

The VSE Health Checker uses additional external programs in order to retrieve and display information. This section shows how they are installed and configured.

6.1 CICS statistics

In order to get CICS TS statistics data, the Health Checker calls the STAT transaction via a console command, like

```
MSG F2,DATA=STAT
```

To define the STAT transaction, you can use skeleton DFH0STAT in ICCF library 59.

6.2 BSTXREF utility

Since May 2009, the Health Checker can use the BSTXREF utility to retrieve data related to the Basic Security Manager (BSM). The tool is intended to help administrators control the profile definitions in the BSM control file. In particular when you delete a user ID, you can use it to ensure that you have removed the user ID from all access lists and groups. This tool applies only to users who have migrated to the new security concept that came with z/VSE V3.1.1.

You can download the BSTXREF utility from

http://www.ibm.com/systems/z/os/zvse/downloads/tools.html

After installing the tool on VSE, no further configuration is necessary.

6.3 CHKTSEN program

The CHKTSEN program allows building a list of the CICS Temporary Storage queues (TS queues). A link job is part of the Health Checker installation. You must define a CICS transaction for program CHKTSEN. The Health Checker then calls the program like:

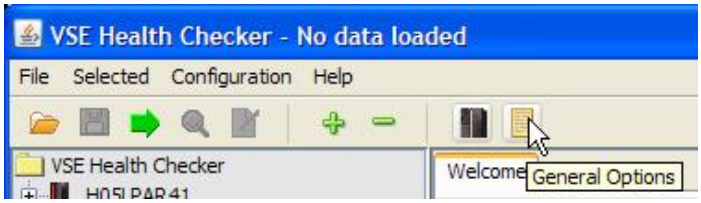
```
MSG F2,DATA=CHKT
```

The next sections show the steps to install and configure CHKTSEN on VSE.

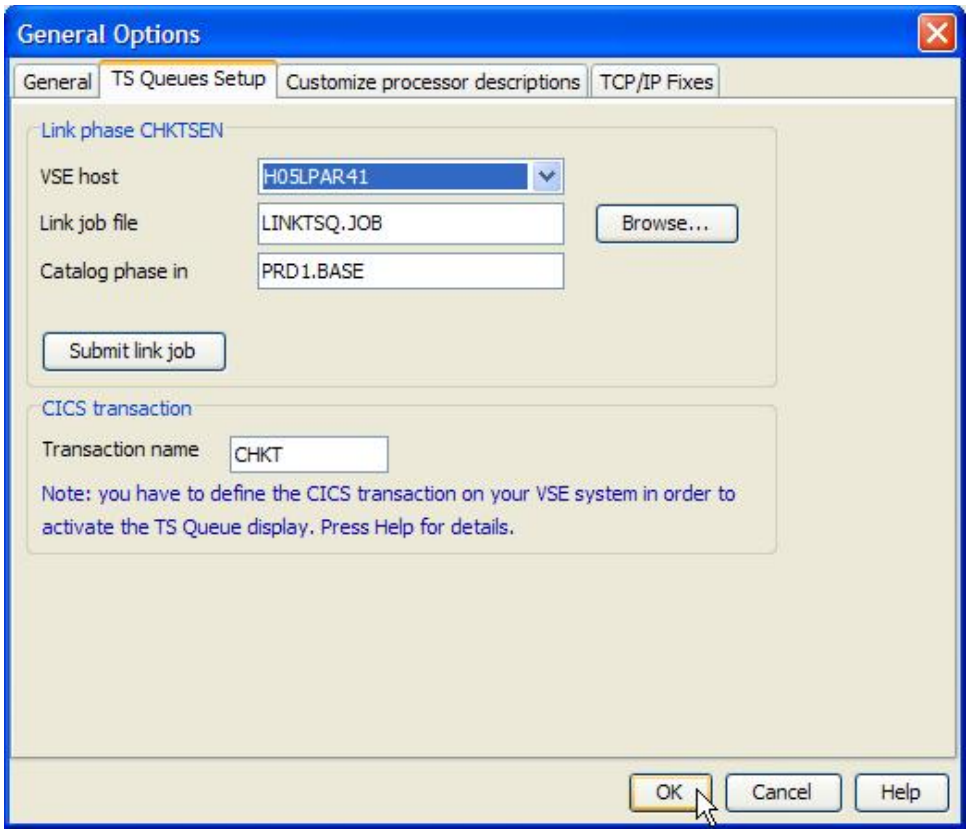
6.3.1 Submit job LINKTS.JOB

Submit the LINKTS.JOB, which is located in the Health Checker installation directory. You can submit the job from the Health Checker GUI; click on “General Options”.

Note: you must have defined your VSE system before you can use this function. Refer to section



On the “General Options” box select tab “TS queues setup”. Here, select your VSE system and press Submit.



The next step is to define the CHKT transaction.

6.3.2 Define transaction CHKT

Enter dialog MAINTAIN SECURITY PROFILES and add a new transaction.

```
IESADMBSLE                  MAINTAIN SECURITY PROFILES
BSM RESOURCE CLASS:    TCICSTRN                          ACTIVE
START....                                          (CASE SENSITIVE)
OPTIONS:   1 = ADD         2 = CHANGE      5 = DELETE      6 = ACCESS LIST

   OPT     PROFILE NAME                      DESCRIPTION       UNIVERSAL AUDIT
                                                                  ACCESS VALUE
    1      ftp                               IBM SUPPLIED                  22
    _      iccf                              IBM SUPPLIED                  12
    _      lpr                               IBM SUPPLIED                  12
    _      newc                              IBM SUPPLIED                  12
    _      ping                              IBM SUPPLIED                  12
    _      ropc                              IBM SUPPLIED                  12
    _      teln                              IBM SUPPLIED                  12
    _      AADD                              IBM SUPPLIED                  12
    _      ABRW                              IBM SUPPLIED                  12
    _      ACCT                              IBM SUPPLIED                  12
    _      ACEL                              IBM SUPPLIED                  12
    _      ACLG                              IBM SUPPLIED                  12

PF1=HELP                    3=END
PF7=BACKWARD   8=FORWARD    9=PRINT
```

Define CHKT as shown below.

```
IESADMBSAE                 MAINTAIN SECURITY PROFILES
BSM RESOURCE CLASS:    TCICSTRN

Add Profile:

  PREFIX.............. _________         CICS region

  RESOURCE NAME...... CHKT
                                       Maximum length is  4 characters.
  GENERIC............. 2                (1=yes, 2=no)

  UNIVERSAL ACCESS... _                (_=None, 2=Read, 3=Update, 4=Alter)

  AUDIT-LEVEL 1 ..... 1                (_=None, 1=Failure, 2=Success, 3=All)
  ACCESS-LEVEL 1 .... 2                (2=Read, 3=Update, 4=Alter, _=default)

  AUDIT-LEVEL 2 .....                  (_=None, 1=Failure, 2=Success, 3=All)
  ACCESS-LEVEL 2 ....                  (2=Read, 3=Update, 4=Alter, _=default)
  DESCRIPTION........ IBM SUPPLIED        Optional remark
PF1=HELP                    3=END                    5=UPDATE

RESOURCE NAME FIELD IS CASE SENSITIVE. ENTER DATA AS REQUIRED.
```

Add CHKT to access list.

```
IESADMBSLE                 MAINTAIN SECURITY PROFILES
 BSM RESOURCE CLASS:    TCICSTRN                          ACTIVE
 START....  CHKt                                  (CASE SENSITIVE)
 OPTIONS:   1 = ADD         2 = CHANGE      5 = DELETE      6 = ACCESS LIST

    OPT     PROFILE NAME                      DESCRIPTION       UNIVERSAL AUDIT
                                                                   ACCESS VALUE
     6      CHKT                              IBM SUPPLIED                12
     _      CIEP                              IBM SUPPLIED                12
     _      CIRA                              IBM SUPPLIED                12


 PF1=HELP                    3=END
 PF7=BACKWARD   8=FORWARD    9=PRINT
```

Use option 1 ADD.

```
IESADMBSLA                 MAINTAIN ACCESS LIST
BSM    CLASS: TCICSTRN     PROFILE:  CHKT
START....                                NUMBER OF ENTRIES ON LIST:  00000
OPTIONS:   1 = ADD         2 = CHANGE      5 = DELETE

   OPT    NAME    ACC

    1


PF1=HELP                    3=END
PF7=BACKWARD   8=FORWARD
```

Now add the transaction to GROUP01 with access 2.

```
IESADMBSAA                  MAINTAIN ACCESS LIST
BSM    CLASS: TCICSTRN     PROFILE:  CHKT

Add Userid or Groupid:


  NAME.............. GROUP01          Userid or Groupid

  ACCESS............ 2                (_=None,
                                       2=Read, 3=Update, 4=Alter)


PF1=HELP                    3=END                     5=UPDATE
```

You must then rebuild the security tables via dialog 2.8.3

```
IESADMSL.IESEBSEC           SECURITY MAINTENANCE
                                                          APPLID: DBDCCIC
  Enter the number of your selection and press the ENTER key:

        1  BSM Resource Profile Maintenance
        2  BSM Group Maintenance
        3  BSM Security Rebuild
        4  Maintain Certificate - User ID List
        5  Define Transaction Security (DTSECTXN)


PF1=HELP                    3=END        4=RETURN                  6=ESCAPE(U
                            9=Escape(m)

==> 3                                              Path: 28
```

When now running the CHKT transaction, it will produce a list queue entry with name HEALTHTS. This entry is downloaded by the Health Checker, parsed and the content is displayed in the GUI.

7 Relationship to other VSE tools and downloadable packages

Besides the VSE Connector Client, the VSE Health Checker uses two additional Java class libraries that can also be used independently from the Health Checker.

The VSE Health Checker package always contains the actual versions of these class libraries. A separate installation of these libraries is not necessary.

7.1 VSESystem class library

The Java class library *VSESystem* can be used to retrieve the output from many VSE console commands and functions performed via VSE/POWER jobs. This allows using such information in any kind of Java programs.

The library is contained in file vsesystem.jar.

7.2 VSESecurity class library

The Java class library *VSESecurity* can be used to retrieve security related information from the VSE Basic Security Manager (BSM). Some of the functions require the BSTXREF utility; refer to section BSTXREF utility on page 8.

The library is contained in file vsesecurity.jar.

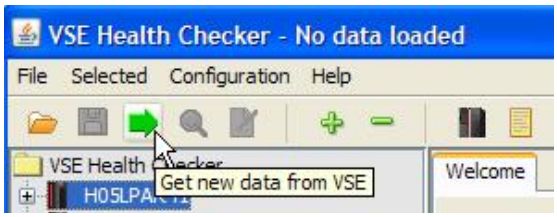
8 Using the VSE Health Checker

Using the VSE Health Checker simply consists of retrieving and evaluating data in regular time intervals. Our recommendation is performing a health check every month, except you changed system parameters.

At this point I assume you completed the initial setup and installed the additional programs.

8.1 Retrieving data

After defining your VSE system in the "Host configuration" box, you start the data retrieval by pressing the green arrow button. Make sure the VSE icon is selected. If you have multiple hosts defined, data retrieval starts for the selected host.



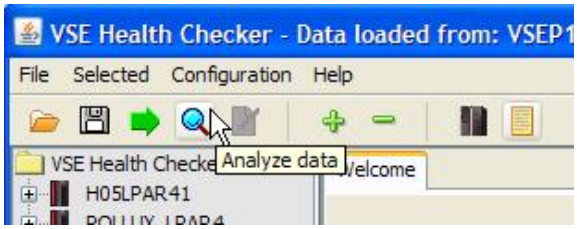
After the data retrieval has ended, you can browse through the Health Checker GUI and view the panels.

8.2 Cancelling data retrieval

You can stop data retrieval at any time by pressing the **ESC** key. The host icon must be selected at this time.

8.3 Rule-based evaluation

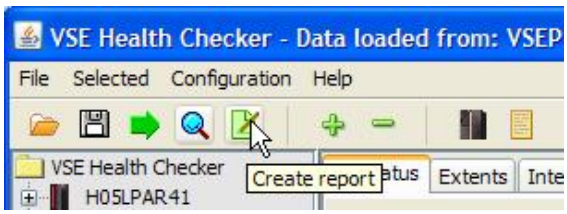
When the data retrieval has ended, toolbar button "Analyze data" becomes enabled.



By pressing the "Analyze data" button, retrieved data gets evaluated with the help of the rules database file. GUI elements, which are subject for evaluation, are then displayed either green (ok), yellow (warning), or red (error).

8.4 Viewing the HTML-based report

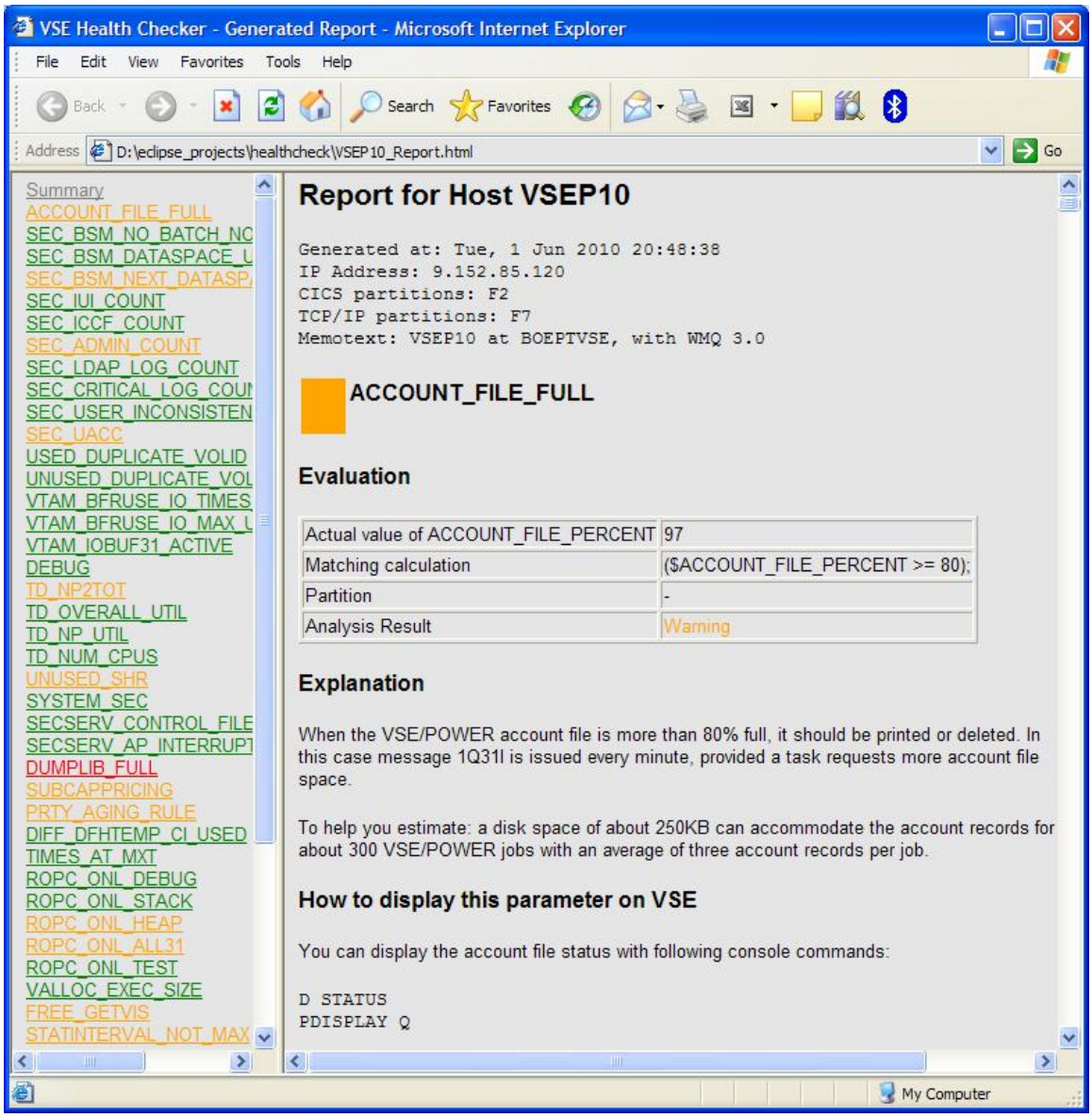
When the data analysis has ended, the "Create report" button becomes enabled.



By pressing the "Create report" button, an HTML-based report is created in the Health Checker install directory, a browser window is opened and the report is displayed.

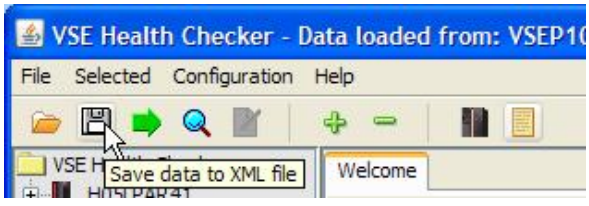
Note: you have to define a web browser to the Health Checker first.

Here is a sample screenshot of an HTML report.



8.5 Saving data to an XML-file

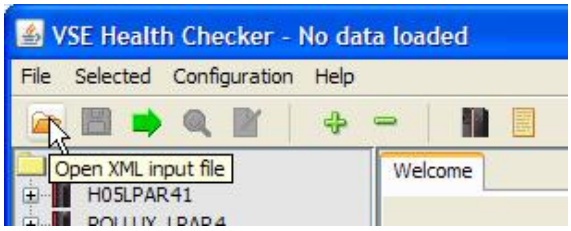
Retrieved data can be saved in XML format for further use. Just press the “Save data to XML file” button.



This opens a standard file dialog box where you can specify a file name and a location. The saved file can for example be used to compare the host status with another health check in the future.

8.6 Loading data from an XML-file

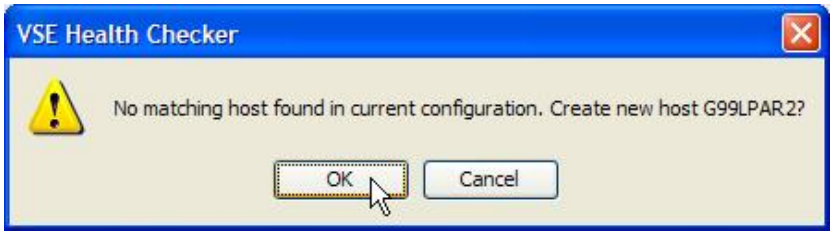
You can load your previously saved data via the folder icon in the toolbar.



You can also load data that was saved on another workstation from a VSE system you don't have defined in this Health Checker configuration. Each XML-file contains some information about the VSE system from which the data was retrieved. If you are now trying to load data from, let's say "VSE A", but you selected "VSE B", then you get prompted to select the right system:

Reply **OK** on this box to load the data for a host definition that matches the information contained in the XML-file. After loading the data, the related host is selected in the GUI.

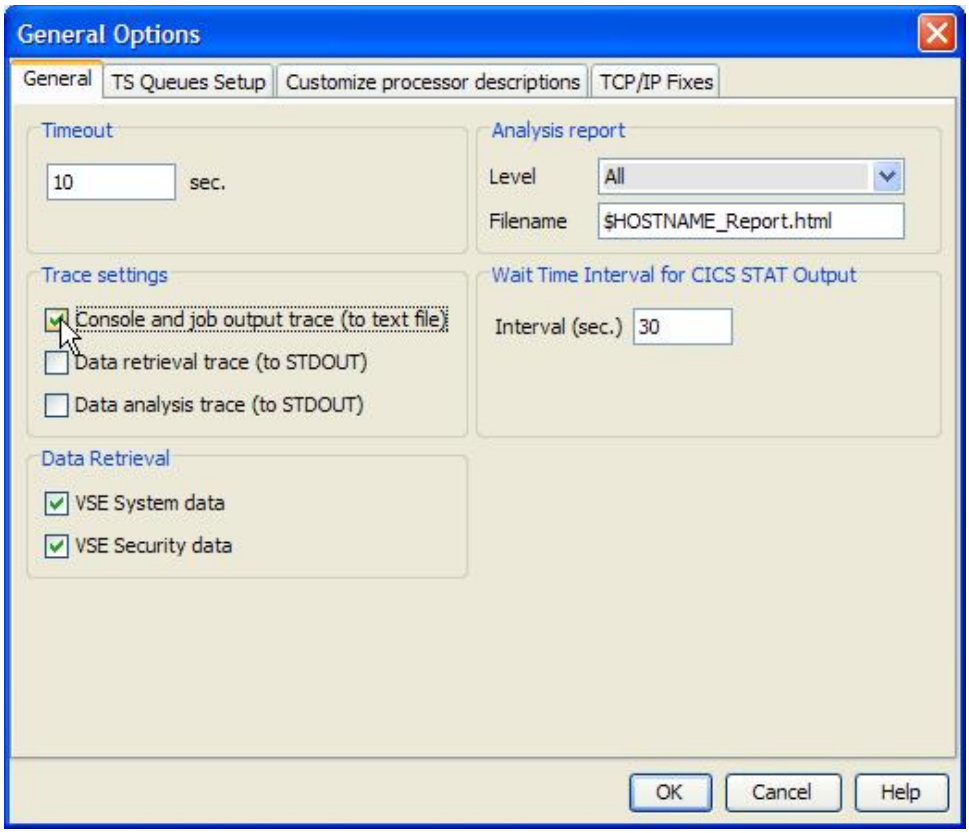
If there is no matching host defined in your Health Checker configuration, a new host is created from the data contained in the XML-file.



Press **OK** to create a new host definition for the data to be loaded.

8.7 Activating the trace

You can activate various types of trace via the "General Options" dialog box.



The following types of trace are available:

- Console and job output trace: writes the original output from submitted console commands and VSE/POWER jobs in a plain text file with a generated file name, like: `vsehealth_24.08.2005_14-14-31.txt.`
- Data retrieval trace: writes trace output from the data retrieval process to STDOUT (Java-console, DOS-Box)
- Data analysis trace: writes trace output from the data analysis function to STDOUT.

9 Maintaining the rule database

The Health Checker comes with a pre-defined rules database: VSEHealthKnowledgeBase.xml. The rules are coded in XML. Feel free to change them according to your needs.

Note: you can find more information about rules in article “Health Checking for z/VSE “, in zJournal issue June/July 2006, available online at:

http://www.zjournal.com/index.cfm?section=article&aid=399

9.1 Structure of the rules database file

The rules database file VSEHealthKnowledgebase.xml is separated into several sections:

- *Rules section:* contains the rules, which are separated into various categories.
    - Non partitioned – rules that apply independent of a partition
    - All partitions – rules that apply for all partitions
    - All static partitions – rules that apply for all static partitions
    - All dynamic partions – rules that apply for all dynamic partitions
    - All CICS partitions – rules that apply for all CICS TS partitions
    - All TCP/IP partitions – rules that apply for all TCP/IP partitions
    - Specific partitions – rules that apply for specific partitions (e.g. F1)
- *Machine group limits:* contains special rules that apply depending on the machine's speed.
- *Data pool mapping:* contains the mappings of XML variables to setter methods of the Health Checker Java API.
- *GUI mapping:* contains the mappings from rules to their related GUI elements. These GUI elements can be Labels (static text), or tables.

9.2 Variables

XML variables are defined in the data pool mapping section.

```
<ACCOUNT_FILE_PERCENT_FULL>
  [PDisplayQData;AccountFilePercentFull]
</ACCOUNT_FILE_PERCENT_FULL>
```

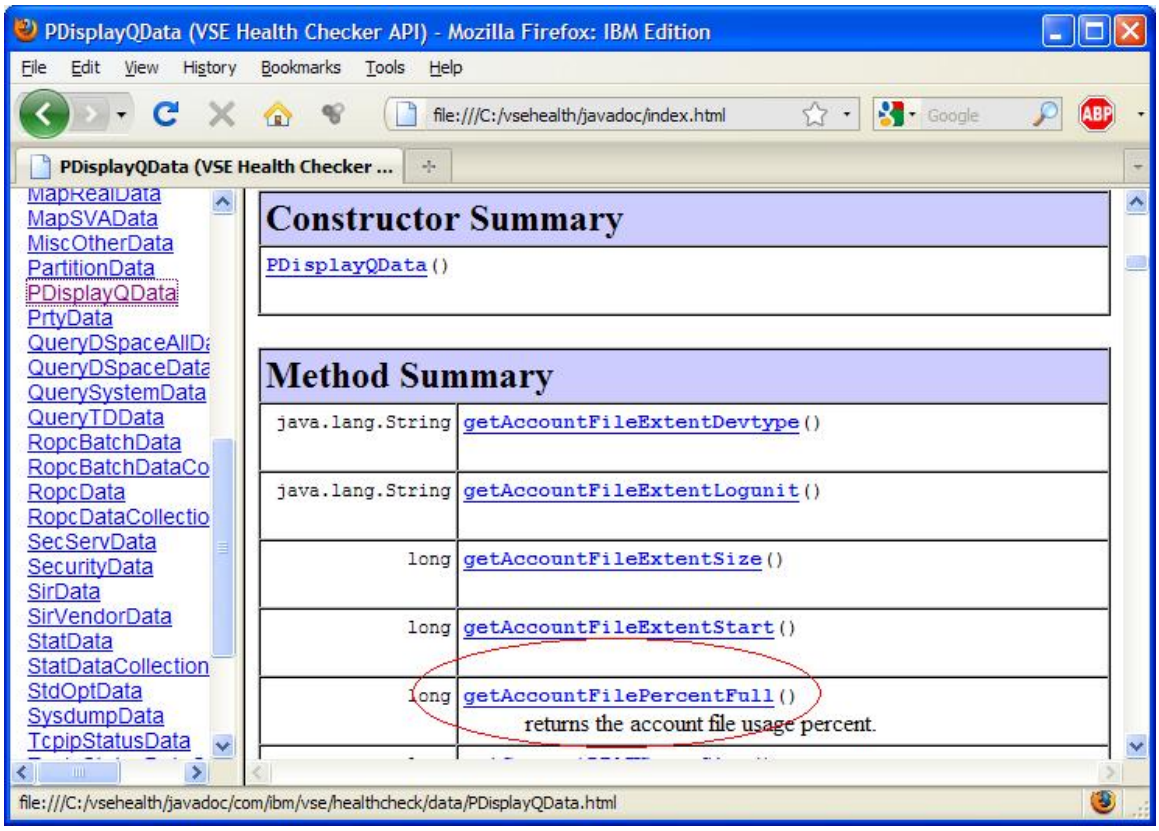
The above defines an XML variable `$ACCOUNT_FILE_PERCENT_FULL`, which gets its value from the Health Checker Java API via method *getAccountFilePercentFull()* in Java class *PDisplayQData*.

The syntax is self-explaining: the Java class name and the getter method name without the "get" prefix are separated by a semicolon. Both are enclosed by brackets.

You can find the method when opening the Javadoc for this class. The Javadoc files are located in folder Javadoc in the Health Checker install directory. On Windows platforms you can also access the Javadoc directly via the Programs menu.

9.3 Introducing new variables

The current rules file contains only those variables, which are in fact used by any rules. If you need more variables, just look into the Javadoc. If you can't find a related getter-method there, please send us an email to zvse@de.ibm.com and we will add a new method.



The following section shows how to use variables in rules.

9.4 Rules

A rule specifies up to three conditions, which are mapped to the colors red, yellow, and green. The sections may appear in any order, but they're evaluated in the given order (i.e., the last positive evaluation determines the color).

```
<ACCOUNT_FILE_FULL>
   <rule>
     <green>
       ($ACCOUNT_FILE_PERCENT_FULL &lt; 80);
     </green>
     <yellow>
       ($ACCOUNT_FILE_PERCENT_FULL &gt;= 80);
     </yellow>
   </rule>
   <explanation>
     explanations/account_file_full.html
   </explanation>
</ACCOUNT_FILE_FULL>
```

The rule grammar is shown in the Health Checker help. In the **General Help** click on **Help Index** and browse to the **Rule grammar** link. Note that special HTML characters, such as "<", ">", and "&", must be coded like "`<`", "`>`", and "`&`".

Note: unfortunately there is currently no process of automatically merging your changes in your current rules file into the rules file that comes with a new Health Checker update.

The explanation section refers to an HTML snippet, explaining the reason for this rule, how the related data in the Health Checker GUI can be displayed with native VSE commands, and where to find additional information. Together with variable HTML parts, showing the evaluation results, these snippets are copied together to get the HTML-based report, see section Viewing the HTML-based report on page 13.

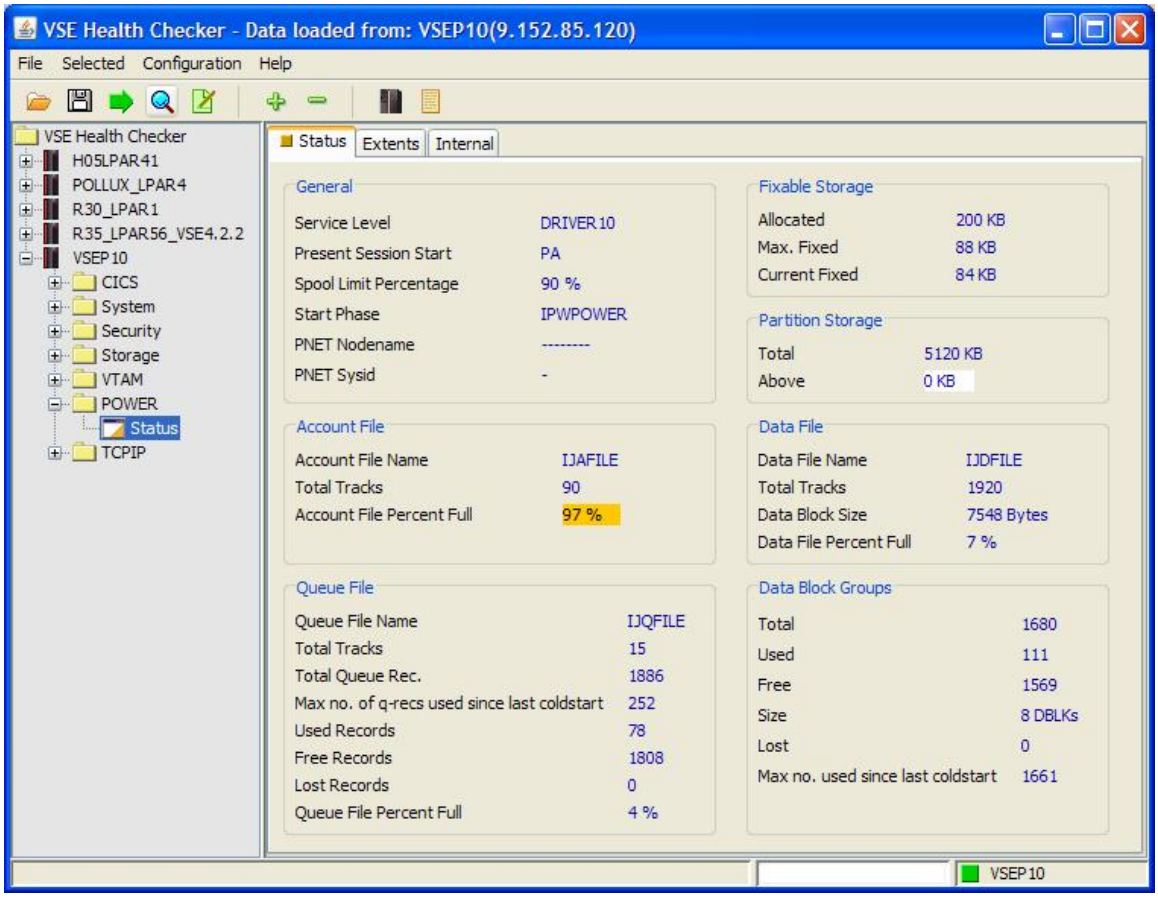
All explanation files are located in the explanations folder.

9.5 GUI mapping

If a rule gets evaluated, there must be a means to display the evaluation result in the Health Checker GUI. To allow this, the GUI mapping section in the rules database file contains mappings of rules to their related GUI elements.

```
<ACCOUNT_FILE_FULL>
  [POWER:Status;Account File Percent Full]
</ACCOUNT_FILE_FULL>
```

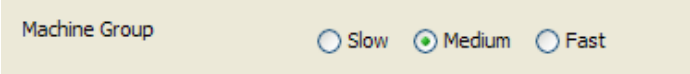
Above GUI mapping tells the Health Checker's data analyzing component that the evaluation result maps to GUI element named "Account File Percent Full" on the "Status" sub-node below the "POWER" tree node.



As a consequence, these strings are unique across all tabs related to one tree node.

9.6 Machine groups

The evaluation of some system parameters depends on the speed of the processor, for example the PRTYAGING parameter. When defining your VSE system in the Health Checker GUI, you had the choice to put your system into three performance categories: slow, medium, and fast. Of course, one particular processor might be considered to be fast at its GA time, but then move to medium and even slow as time goes by and new machines come to the market.



While rules normally compare the values of XML variables with static numbers, the rule related to PRTYAGING uses an indirection by introducing a second type of variable. This type of variable is prefixed with a "#" character and points to values defined in the machine group limits section.

```
<PRTY_AGING_RULE>
 <rule>
   <green>
       ($PRTYAGING &lt; #PRTYAGING);
   </green>
   <yellow>
       ($PRTYAGING &gt;= #PRTYAGING);
   </yellow>
 </rule>
 <explanation>
   explanations/prty_aging_rule.html
 </explanation>
</PRTY_AGING_RULE>
```

Basically, the actual value of the #PRTYAGING variable depends on the machine speed as defined in the machine group limits section.

```
<machine_group_limits>
  <allmachines>
     <PRTY_AGING_RULE>
       <PRTYAGING>
         5000            <- the value of #PRTYAGING is 5000 for any machine
       </PRTYAGING>
     </PRTY_AGING_RULE>
  </allmachines>
  <slow>
     <PRTY_AGING_RULE>
       <PRTYAGING>
         5000          <- the value is 5000 for slow machines
       </PRTYAGING>
     </PRTY_AGING_RULE>
  </slow>
  <medium>
     <PRTY_AGING_RULE>
       <PRTYAGING>
         2000         <-  the value is 2000 for medium speed machines
       </PRTYAGING>
     </PRTY_AGING_RULE>
  </medium>
  <fast>
     <PRTY_AGING_RULE>
       <PRTYAGING>
         1000          <-  the value is 1000 for fast machines
       </PRTYAGING>
```

```
    </PRTY_AGING_RULE>
  </fast>
</machine_group_limits>
```

There is always a value for “all machines” as a fallback for the case where there is a value missing for one of the categories (slow, medium, fast).

10 Known problems

This chapter describes some errors that showed up in our test setup.

10.1 Coding variables in the rules database

Each XML variable must start with a “$” character and must be delimited with at least one blank character from any following characters. Otherwise the analyze step will fail.

Symptom:

```
java.lang.StringIndexOutOfBoundsException: String index out of range: -65
       at java.lang.String.substring(Unknown Source)
       at om.ibm.vse.healthcheck.analysis.Analyzer.getXMLVars(Analyzer.java:908)
```

Solution:

The following rule is coded wrong. There is no space after variable `$CURRENT_BSM_DATAPART_SIZE.`

```
<SEC_BSM_NEXT_DATASPACE_TOO_SMALL>
  <rule>
    <green>
      ($NEXT_BSM_DATASPACE_SIZE &gt;= $CURRENT_BSM_DATAPART_SIZE);
    </green>
    ...
```

The correct way of coding this rule is:

```
<SEC_BSM_NEXT_DATASPACE_TOO_SMALL>
  <rule>
    <green>
      ($NEXT_BSM_DATASPACE_SIZE &gt;= $CURRENT_BSM_DATAPART_SIZE );
    </green>
    ...
```

10.2 MLOG considerations

MLOG changes the output of the MAP nn and GETVIS nn console commands, so the Health Checker is not able to parse them correctly.

Example of a standard VSE output

```
map r1
AR 0015  PARTITION: R1        SPACE-GETVIS......:     128K  ADDR: 500000
AR 0015  CLASS....: R         ALLOC (VIRTUAL)...:    8064K  ADDR: 520000
AR 0015  STATUS...: VIRTUAL     SIZE............:    1024K
AR 0015  POWER-JOB: STARTVCS     EXEC-SIZE......:    1024K
AR 0015  JOBNUMBER: 22352       GETVIS..........:    7040K
AR 0015  JOBNAME..: STARTVCS     EXEC-GETVIS....:    7040K  ADDR: 620000
AR 0015  PHASE....: IESVCSRV
AR 0015                       PFIX(BELOW)-LIMIT :       0K
AR 0015                                  -ACTUAL:       0K
AR 0015                       PFIX(ABOVE)-LIMIT :       0K
AR 0015                                  -ACTUAL:       0K
AR 0015 1I40I  READY
```

Example of the corresponding MLOG output

```
map r1
17:05:24 24/08/2005 SYSA
AR 0015 ID  M JOB      EXEC     PB  P  STATUS   WORKLOAD  JOBTIME  JOBSTART
17:05:24 24/08/2005 SYSA
AR 0015 R1  V STARTVCS IESVCSRV 83 11  RUNNING      *     00.06.43 16.58.41
17:05:24 24/08/2005 SYSA
```

The problem can be solved by running following VSE/POWER job.

```
* $$ JOB JNM=MLOGMAPS,CLASS=0,DISP=D
// JOB MLOGMAPS
/*
/*  See MLOG Product Description, Chapter 17
/*  ARMAP {YES/NO}
/*  Specify whether you want the MAP command to be
/*  intercepted by the Attention Routine.
/*  The default is YES.
/*
// LIBDEF PHASE,SEARCH=lib.sublib
// EXEC M4USRMAP
ARMAP NO
/*
/&
* $$ EOJ
```

10.3 More typical errors

For more typical errors, open the Health Checker help index and click on link “Syntax errors in rules file”.

11 More information

You can find more information in these books and web sites:

Download connector components from the VSE homepage
http://www.ibm.com/servers/eserver/zseries/zvse/downloads/

“Health Checking for z/VSE “, article in zJournal issue June/July 2006
http://www.zjournal.com/index.cfm?section=article&aid=399

z/VSE e-business Connectors User’s Guide
http://www.ibm.com/servers/eserver/zseries/zvse/documentation/#conn

VSESystem class library as a separate package
http://www.ibm.com/systems/z/os/zvse/downloads/index.html

Presentation: z/VSE Health Checker
ftp://public.dhe.ibm.com/eserver/zseries/zos/vse/pdf3/techconf2007/munich/E76.pdf