Information Engineering Technology

Install Guide - pathvIEw

Release 8.7.3

Table Of Contents

pathvIEw Architecture

pathvIEw is a code coverage testing tool that provides logging and reporting information on the code you have executed in your application. This information allows you to quickly review what code has been executed, and therefor see what level of testing has been done, at the statement level.

In order to achieve this, a number of simple components and processes are required:

(1) Standard CA Gen source code is generated (from the toolset or encyclopaedia). Prior to compilation, it is modified by the pathvIEw *Source Code Post-Processor* (PVIEWSPP), which adds in calls to the pathvIEw runtime to log statement executions. Immediately after the Source Code Post-Processor is run, details of that code is stored in the pathvIEw database, including the PAD Listing, generation timestamp etc.

(2) The modified source code is then installed (compile, link, etc.) and made available for execution in your testing region.

(3) On execution of the application, statement coverage data is gathered by the pathvIEw runtime, and then sent (using TCP/IP sockets) to the pathvIEw *Listener*, which stores the execution results in the pathvIEw database.

(4) The code coverage data can be viewed at any time using the pathvIEw *Client*.
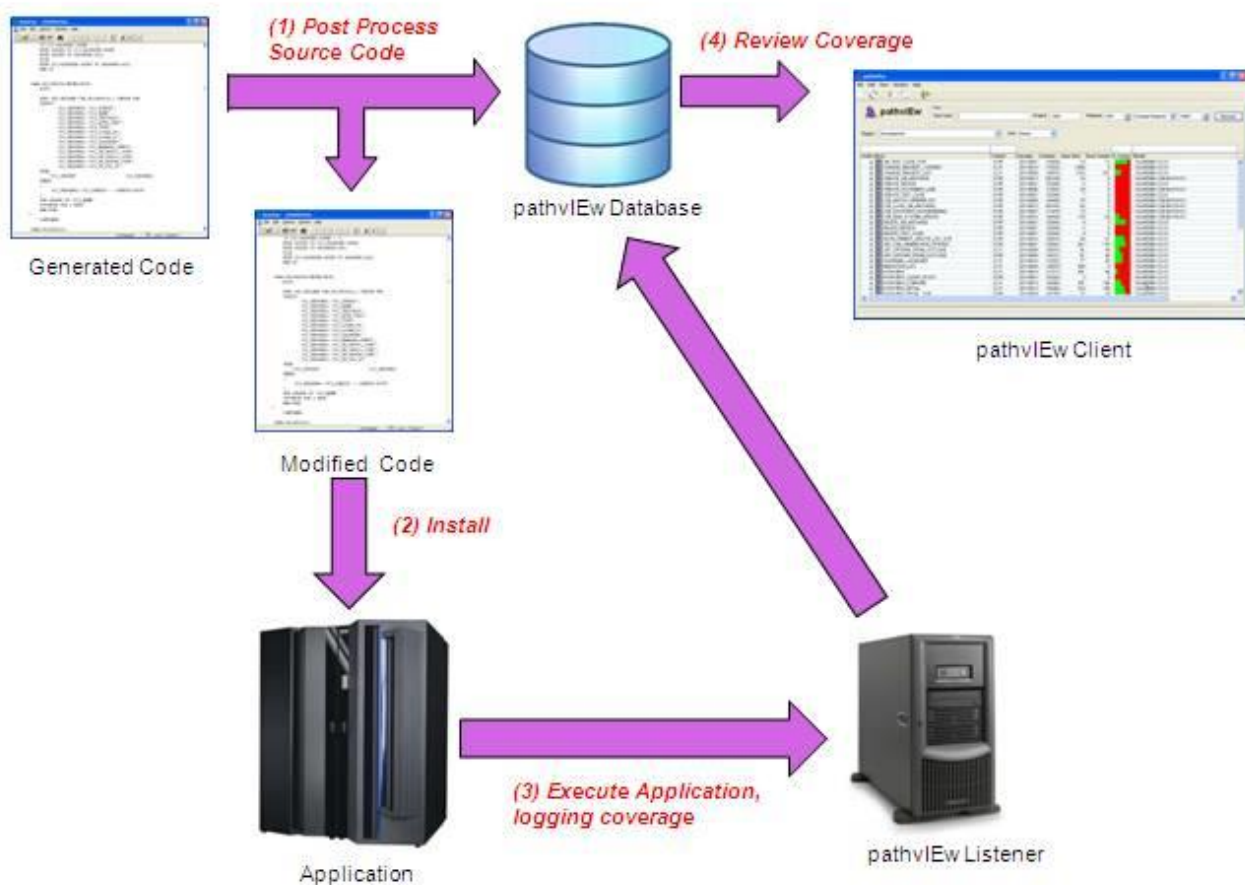
Installation Overview

The installation and configuration of pathvIEw comprises the following steps:

1. Configure the pathvIEw Listener
2. Configure the application build process so that the generated applications have statement logging enabled
3. Configure the application runtime environment so that pathvIEw will log statement executions
4. Configure the system parameters used by pathvIEw
5. Setup housekeeping routines

Each of these steps is described in the chapters below.

Pre-Requisites

You should ensure that the following prerequisites are already in place before attempting the install of pathvIEw:

- GuardIEn 8.7 Server and Client components – the pathvIEw Listener software is provided within the standard GuardIEn installation

Windows pathvIEw Listener

- Microsoft Visual C++ 2015 runtime – will already be available if you have a GuardIEn 8.7 Server installed

Toolset Generation

- ODBC driver for CSE/HE database if using Toolset generation

pathvIEw Runtimes Environments Supported

pathvIEw supports a variety of runtime environments enabling Gen applications for numerous targets to have their executions logged. Different runtimes can communication with a single pathvIEw Listener, including:
- z/OS – CICS, IMS, IEFAE and Batch
- UNIX – IBM AIX, Sun Solaris, HP IA64 Itanium and Linux – all types of packaging e.g. Co-Operative Servers, Online, Non-Screen
- Windows – all types of packaging e.g. Windows packaged, Co-Operative Clients & Servers, Online, Non-Screen
- JVM – all types of packaging e.g. Rapide, Web Generation, Web View, EJB etc.

Compatibility With Different Releases of Gen

pathvIEw supports the following combinations of Gen version and platform, providing the necessary facilities in the Source code Post Processor, as well as any supporting pathvIEw runtime required:

	C on Windows	C on UNIX	COBOL	Java
Gen 6.5	YES			
Gen 7.0				
Gen 7.5				
Gen 7.6	YES	YES	YES	YES
Gen 8.0	YES	YES	YES	YES
Gen 8.5	YES	YES	YES	YES
Gen 8.6	YES	YES	YES	YES

Download Files

All IET software is available for download from the IET support centre: https://support.iet.co.uk. Software is secured on the web site, so you will need register and then request access.

Download pviewxxx.exe (where xxx is the version required) into a directory where you want to temporarily store the pathvIEw software.

Extract the files from the self-extracting archive pviewxxx.exe to the chosen directory.

Configure the Listener

The pathvIEw Listener receives TCP/IP messages from the pathvIEw runtime and logs the statement execution data in the pathvIEw database.

It is recommended that the Listener is installed on the same server as the GuardIEn database for best performance, but this need not be the case if the database access calls performed by the Listener are transparent, i.e. using Oracle SQL*Net or ODBC. For DB2, it is recommended that the Listener is installed on MVS since DB2 ODBC performance is often very poor.

The Listener does not need to be on the same server as your application. For a client/server application, there might be logging from multiple clients and multiple servers, all communicating with a single pathvIEw Listener. The Listener may run on Windows, UNIX (not Solaris or Linux) or MVS.

MVS Listener

The Listener software is installed with the GuardIEn 8.7 host server software.

For MVS two types of Listener are provided:
- Started Task
- CICS

The recommended Listener on MVS is the Started Task, but a CICS Listener is also available for sites that prefer not to implement Started Tasks.

Configure Started Task Listener

Review the parameters for the started task JCL:

Parameter	Description
DSN	DB2 sub-system
PLAN	DB2 Plan
PORT	The port that the listener will listen on
WAIT	Controls error handling for thread creation errors: 0: terminate on thread creation 1: ignore a thread creation >1: wait for n milliseconds before a second attempt at a thread creation. If this fails, terminate

Start Listener

Submit JCLRUN(PVSTART) to test that the Listener starts as a normal batch job

Ping Listener

Edit JCLRUN(PVPING) and enter your IP Address and Port and then submit to test that you can communicate with the Listener

Stop Listener

Edit JCLRUN(PVSTOP) and enter your IP Address and Port and then submit to stop the Listener

Note that you can also use an MVS operator command to stop the Listener: STOP <job/task name>

Create Started Task

Copy the PROCLIB(PVLSNR) member to your started task dataset so that the Listener can be executed as a started task.

Configure CICS Listener

Check IBM Listener Installed

The pathvIEw CICS Listener is started by the standard IBM Listener program EZACIC02. The default transaction code for this is CSKL.

Check that you have started the default Listener using the CICS command EZAC,DISplay,LISTENER and note the port defined for the Listener since this is the port that you will have to use for your pathvIEw-enabled applications.

Copy Executable

Copy SVRLOAD(PVCICLSN) to a CICS program library if the GuardIEn SVRLOAD dataset is not included in your CICS DFHRPL dataset list.

Check CICS Resources

Ensure that the following resources have been defined in CICS when GuardIEn 8.6 was installed:

TRANS(PVCL)
PROG(PVCICLSN)
DB2TRAN(PVD2T)
DB2ENTRY(PVD2E)

If they are not present then use the GuardIEn installation JCL (INS.JCL) member SPVCSD to create them. Follow the instructions contained within the JCL to configure the job.

Ping Listener

Edit JCLRUN(PVPING) and enter your IP Address and CICS Listener Port and then submit to test that you can communicate with the Listener.

Windows Listener

- The Windows Listener is installed with the GuardIEn 8.7 CSE server software.

- Register the Listener Service by executing PVLSNRRegSvc.cmd. By default, the service is registered to run using the Local System account credentials, but if you want to execute the service using a specific account, either amend the cmd file to specify the account user/password or amend the service Log On properties.

- Start the Listener Service by executing StartPVLSNRService.bat, or by using standard Windows Services – to test that the Listener works ok for the first time, you can start it as a normal application using script StartPVLSNR.bat instead.

- Test that the Listener Service is active by executing PingPVLSNR.bat (by default this script communicates with 'localhost' on port 50877, so you may need to modify it if you chose different hostname or port number).

- Stop the Listener using StopPVLSNR.bat, or StopPVLSNRService.bat (similar parameters to Ping above).

- You can now start the Listener Service using Windows Services and set the service to automatically start. Note that the database to which the Listener writes to must be available permanently while Listener is active, as it maintains a single database connection for its life.

UNIX Listener

- On HP IA64 Itanium or AIX, Start the Listener Service by executing startPVLSNR

- Test that the Listener Service is active by executing pingPVLSNR

- Stop the Listener using stop PVLSNR

- Note that the database to which the Listener writes to must be available permanently while Listener is active, as it maintains a single database connection for its life.

Configure the Build Process

To enable an application to communicate with pathvIEw, the generated source code needs to be modified using the pathvIEw Source Code Post-Processor (PVIEWSPP). The output from the post-processor is stored in the pathvIEw database and the build process (compile & link) is modified so that the pathvIEw runtime modules are correctly referenced by the application.

The configuration of the build process varies by platform and additionally, can be greatly simplified by using an automated build such as GuardIEn.

Invoking PVIEWSPP at Generation Time

PVIEWSPP for Toolset Generation

If you are generating code using the Gen Toolset, then you can either execute PVIEWSPP.exe after generating the code and before the build, or as part of the Gen build script:

- Copy the <pathvIEw>\Windows\Build\<ency> folder to your workstation where <ency> is "CSE" if the pathvIEw database is installed on a CSE and "HE" if it is installed on an HE.

- Edit pvSPP1.bat and amend the following two lines to reference the drive and path where the files have been installed.
    o   set GDDRV=C:
    o   set GDHOME=\??????\

- If you want to automatically invoke PVIEWSPP when performing a windows build of C code using the Build Tool, amend the build_lm_c.scr file in Gen\bt\scripts and insert the line highlighted below. The source code will be processed if an environment variable PVIEWSPP is set to YES. Gen 7.6 and Gen 6.5 modifications will differ slightly from this.

```
@instmsgj {execunit.MEMBER} "Compiling {source.MEMBER}"
@echo.
…
if "%PVIEWSPP%" == "YES" call pvspp1.bat {source.NONDIAL}{source.MEMBER}.{LEXT} {LOC.CODE_SRC}
cl $(CFLAGS) -Fo"{LOC.CODE_OBJ}{source.NONDIAL}{source.MEMBER}.{OBJEXT}" …
```

- Similarly, if you want to automatically invoke PVIEWSPP when performing a windows build of Java code using the Build Tool, amend the build_lm_java.scr file in Gen\bt\scripts and insert the line highlighted below. The source code will be processed if an environment variable PVIEWSPP is set to YES. Replace the explicit path to the pathvIEw SPP directory with the path that you installed it into on your build machine:

```
<exec executable="instmsgj">
  <arg value="{ICM_FILENAME}"/>
  <arg value="Compiling Action Block Classes"/>
</exec>

<!-- Modified by IET for pathvIEw SPP -->
  <echo message=""/>
  <echo message="...pathvIEw SPP Source Code Post-Processor for Action Blocks"/>
  {[FOREACH]} ACBLK AS acblk
    {[IF]} NOT_EQUAL "{acblk.ABTYPE}" "CASCADE"
      <echo message="SPP AB: {acblk.PACKAGE}\{acblk.MEMBER}.{LEXT}"/>
      <exec executable="C:\Gdn\pview81\Windows\Build\CSE\pvspp1.bat">
        <arg value="{acblk.PACKAGE}\{acblk.MEMBER}.{LEXT}"/>
        <arg value="{LOC.CODE_SRC}"/>
      </exec>
    {[ENDIF]}
  {[ENDFOR]}

<echo message=""/>
<echo message="...Compiling Action Block Classes"/>
```

and also:

```
<exec executable="instmsgj">
  <arg value="{ICM_FILENAME}"/>
  <arg value="Compiling Procedure Step Classes"/>
</exec>
```

```
<!-- Modified by IET for pathvIEw SPP -->
  <echo message=""/>
  <echo message="...pathvIEw SPP Source Code Post-Processor for Procedure Steps"/>
  {[FOREACH]} PSTEP AS pstep
    {[IF]} NOT_EQUAL "{pstep.ABTYPE}" "CASCADE"
      <echo message="SPP PSTEP: {pstep.PACKAGE}\{pstep.MEMBER}.{LEXT}"/>
      <exec executable="C:\Gdn\pview81\Windows\Build\CSE\pvspp1.bat">
        <arg value="{pstep.PACKAGE}\{pstep.MEMBER}.{LEXT}"/>
        <arg value="{LOC.CODE_SRC}"/>
      </exec>
    {[ENDIF]}
  {[ENDFOR]}

  <echo message=""/>
  <echo message="...Compiling Procedure Step Classes"/>
```

PVIEWSPP for GuardIEn CSE Generation

- The source code generated using GuardIEn 8.5 and above can be automatically processed by setting the Target Property PV<n>SPP to YES where <n> is the code gen variant, for example PV1SPP for code gen variant 1.

PVIEWSPP for GuardIEn HE Generation

- The source code generated on the host encyclopaedia is post-processed using either a separate step that should be scheduled to run after the source code generation and before the compile, or if you are using the GuardIEn compile procedures, as an alternative compile procedure.
- To implement the separate step, define a new step type that uses the PVSPP action. The step type parameter specifies whether the source code dataset name is specified using the model business system libraries or GuardIEn target properties.
- To use the combined spp and compile procedure, GuardIEn 8.6 provides two example procedures GDCMPLP and GDCMPLPN. You should amend your project definition to use these procedures instead of the default procedures.
- If you want to integrate the post-processing into your own custom build process, you can use the sample PVSPP JCL in the PROCLIB dataset.

Including pathvIEw at Build Time

Configure MVS Build

- The modified source code contains a call to PVLOG and therefore PVLOG needs to be available in the link-edit SYSLIB, either by adding the pathvIEw NCAL dataset into the SYSLIB datasets or copying PVLOG from NCAL to a dataset that is in the SYSLIB concatenation.
- For Gen 7.6 and earlier, PVLOG will be resolved using AUTOCALL.
- For Gen 8.0 and above, AUTOCALL is not used at link-time and therefore an INCLUDE SYSLIB(PVLOG) statement needs to be added to the link-edit control cards. This is best accomplished by editing the CEHBSKL0 members listed below and adding in the card.

  TICLBTCH, TICLDLI, TICLDSN, TICLAED, TICLCICD, TICLIMSD, TI$LIMSD
  TI$LBTCH, TI$LDLI, TI$LDSN, TI$LAED, TI$LCICD, (if using Compatibility)

Configure Windows Build for Building C

- The <pathvIEw>\Windows\Build\Include folder contains the C header file (PVIEW001.h) used by the post-processed code. You should either add this folder to the INCLUDE environment variable or copy the header file to a folder that is already contained in the INCLUDE path.
- The <pathvIEw>\Windows\Build\Lib folder contains the lib file, pathvIEw.lib, to resolve the calls to the pathvIEw runtime. You should either add this folder to the LIB environment variable or copy the contents to a folder that is contained in the LIB path.
- Amend the Build Tool profile to ensure that pathvIEw.lib is referenced as an external library.

Configure Windows Build for Building Java

- The <pathvIEw>\JVM\<Gen version>\pathvIEw<rel> folder (where <Gen version> should match the release of your Gen Build Tool e.g. Gen86, and <rel> is the pathvIEw Release e.g. 87) contains the Classes and Exit Source files used by the post-processed code. You will need to add these Classes to your CLASSPATH for Gen builds. For example, in a Toolset/Build Tool startup script you could add:

  SET CLASSPATH=C:\PV\JVM\Gen80\pathvIEw87\pathvIEw87.jar;%CLASSPATH%.

- The <pathvIEw>\JVM\<Gen version>\pathvIEw<rel> folder also contains the jar file, pathvIEw87.jar, which you will need to specify for the Assemble of your Java application in the *Additional Files* section, so that the pathvIEw runtime is distributed with your application:
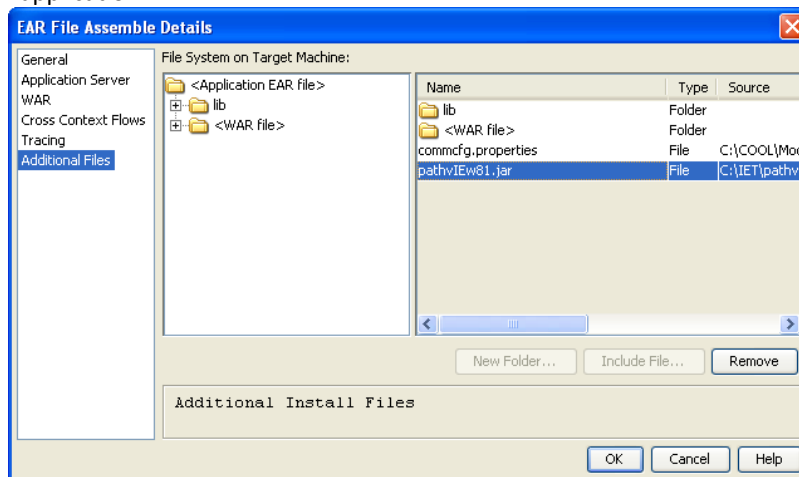
- Before Assembling your application you will need to tailor (edit the source code) the pathvIEw Exit to specify how communications with the pathvIEw Listener should be performed, as well as specifying how USERID, REGION and TEST CASE are determined at runtime. The source is provided in the same directory as the Classes, and is called pathvIEw_Exit.java. Once you have edited the Java source for this Exit, you need to rebuild the pathvIEw runtime .jar using the provided DOS command script in the directory above the source, called Make-pathvIEw81ExitAndRuntimeJar.cmd. You may need to tailor this script if your environment does not contain everything necessary to build Java code by default.
- A sample Java program to *Ping* the pathvIEw Listener is also provided here, which can be tested with batch script Test-pathvIEw87JavaPING.cmd. This may be useful in validating that you have modified the Exit source correctly – particularly in terms of specifying the Host and Port for the Listener.

Configure UNIX Build

- The <pathvIEw>\UNIX\Include folder contains the C header file (PVIEW001.h) used by the post-processed code. You should copy the contents to a central directory so that this header file can be found at compile time during the build process. e.g. $IEFH/include
- The <pathvIEw>\UNIX\HP|AIX|Solaris|Linux<-32/64> folder contains the PVLSPING utility and the shared library file (libpathvIEw.s*) to resolve the calls to the pathvIEw runtime. The 32/64 suffix relates to the 32bit/64bit architecture of the binary files. The appropriate library for your application environment should be placed in a central directory that is already assigned to the SHLIB_PATH/LIBPATH/LD_LIBRARY_PATH environment variable. e.g. $IEFH/lib
- Amend the Build Tool profile to ensure that this pathvIEw shared library is references as an external library.

Configure the Runtime Environment

Install MVS Runtime Environment

This section is only required if you are executing a Gen application on MVS and do not use GuardIEn for the host encyclopaedia.

- Edit the MVS\pvFTP.cmd command file and enter your MVS userid, IP address and dataset high level qualifier in the SET statements
- FTP the XMT files to the host using the MVS\pvFTP.cmd command file. This should be executed from a command prompt specifying your MVS password as a parameter, for example pvFTP.cmd password
- Check that the datasets have been transferred by reviewing the pvftp.out file
- Logon to TSO and edit <HLQ>.CNTL(PVINST)
- Review the job card and submit the job
- Review the job output. This should have created the <PVHLQ>.NCAL, LOADB and LOADC datasets.
- Review the exit code in CNTL members PVENVB and PVENVC and check your IP address, region code, etc.
- Submit jobs CNTL(MKENVB) to compile the batch PVENV exit and CNTL(MKENVC) to compile the CICS PVENV exit if you use CICS as a runtime environment.

Configure MVS Runtime Environment

- Customise the PVENVB and PVENVC Exits with your Listener IP Address and Port and submit the following jobs to compile the Exits. If you are using GuardIEn for the host encyclopaedia, the source code is provided in the EXITSRC dataset.

Job	Description	Notes
PVENVB	Compile PVENV batch exit	
PVENVC	Compile PVENV CICS Exit	Only required if using CICS for the application runtime

- Ensure that the PVENV and PVSTOR load modules are available at runtime.
- If you are using GuardIEn for the host encyclopaedia, these pathvIEw runtime modules are provided with the GuardIEn host software. Separate Batch and CICS versions are located in the GuardIEn BATLOAD and SVRLOAD datasets. For your batch applications, either include the GuardIEn BATLOAD in your batch STEPLIB or copy these load modules to an library in the application STEPLIB. See the separate instructions below on installing the pathvIEw runtime on the host.
- If you are using GuardIEn for the CSE, then the runtime modules are provided in the pathvIEw installation suite and will require installation on the host.
- For CICS you should define the PVENV and PVSTOR programs if you do not use autoinstall.
- The pathvIEw region code is defined using the PVENV exit, so if you want to use pathvIEw in multiple test regions, you should create separate copies of the PVENV exit. The Region allows you to log which of your test environments the execution was run within.

Configure Windows Runtime Environment

- Ensure that the <pathvIEw>\Windows\Runtime\[GenVersion]\pathvIEw.dll is available in the PATH by distributing this with your application. Note that different versions of the DLL are provided for each release of Gen, so choose the appropriate version to match the version of Gen that you use to generate your application.
- The pathvIEw runtime parameters are defined using the following environment variables that should be set when you start your application:

SET PATHVIEW_REGION=<Region Code>
SET PATHVIEW_IP=<IP Address for Listener?
SET PATHVIEW_PORT=<Port for Listener>
SET PATHVIEW_DEBUG=N (Y to enable debug from the application side)

Configure UNIX Runtime Environment

- Ensure that the appropriate pathvIEw shared library is available by distributing this with your application. Versions for AIX 7.1, Solaris and Linux are currently available – please contact IET if you require a pathvIEw runtime for an older release of UNIX, for example to use with Gen 7.6.
- The pathvIEw runtime parameters are defined using the following environment variables that should be set when you start your application:

```
PATHVIEW_REGION=<Region code>
PATHVIEW_IP=<IP Address for Listener>
PATHVIEW_PORT=<Port for Listener>
PATHVIEW_DEBUG=<Y/N> (Y to enable debug from the application side)
export PATHVIEW_REGION PATHVIEW_IP PATHVIEW_PORT PATHVIEW_DEBUG
```

Configure Java Runtime Environment

There is no special configuration to perform for pathvIEw in a JVM environment – all tailoring is done in the pathvIEw_Exit.java exit instead, and distributed in pathvIEw87.jar along with your application.

Configure System Parameters

Administrator Authority

A GuardIEn user must be defined as a pathvIEw administrator to be able to edit Regions and Test Cases.

By default, a user is defined as a pathvIEw Administrator if they are defined as a GuardIEn System Administrator.

You can define additional users as pathvIEw administrators by creating a group id (for example PVADMIN) and assigning the users to this group. You then define the group id as a pathvIEw admin group by creating a system parameter with a code of PVADMIN and a text value specifying the group id, in this example, also PVADMIN.

Housekeeping

The housekeeping job GD70 uses the PVHKEEP system parameter numeric value to specify how many previous instances to keep.

See the Background Jobs manual for further details.

CICS Listener LogLevel

The CICS Listener obtains its loglevel from a system parameter with a code of PVLOGLVL and a text value of 0, 1, 2 or 3.

Configuring your ODBC Data Source

You will need to define an ODBC data source if you are executing pathvIEw on a workstation for Toolset generation.

- Open the Windows ODBC Data Sources application, Control Panel->Administrative Tools->Data Sources (ODBC).
- Change to the 'System DSN' tab.
- Click 'Add' to add a new System ODBC Data Source.
- Follow the steps below for the type of Data Source that you need to connect to your encyclopaedia database with…
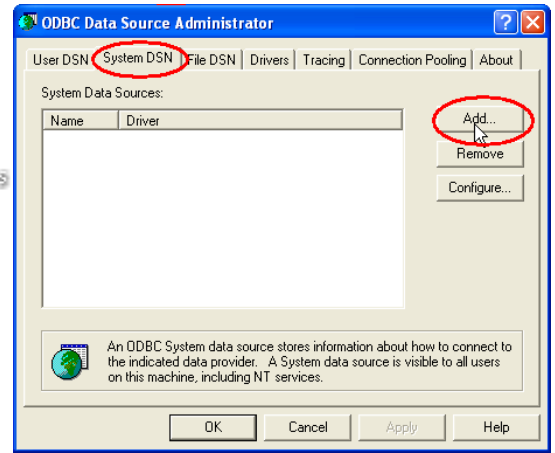
Oracle CSE (Windows & UNIX)

- Use Oracle's ODBC driver. Do not use the Microsoft Oracle ODBC Driver (there are known problems with this using GuardIEn).
- Press Finish to create the new Data Source.
- The Oracle-specific connection parameters dialog will appear next. Enter a name for your new ODBC Data Source and the TNS Service Name (e.g. DBIE) that you use to connect to the CSE database from your workstation. Note that if you are using a centralised home directory, then you should use the same name for the Data Source as specified in the central gdologon file in that home directory.
- For the User Id, if you have a full GuardIEn database on the server, then enter the 'gdn' Oracle userid. Otherwise, enter the 'ency' Oracle userid.
- If you know the password for that Oracle userid, then use the *Test Connection* button to make sure it works ok.
- For Oracle 11g, you either need to give your ODBC Data Source the same name as the server's hostname, or ensure that the Oracle XML Database component is installed for that Oracle Instance.
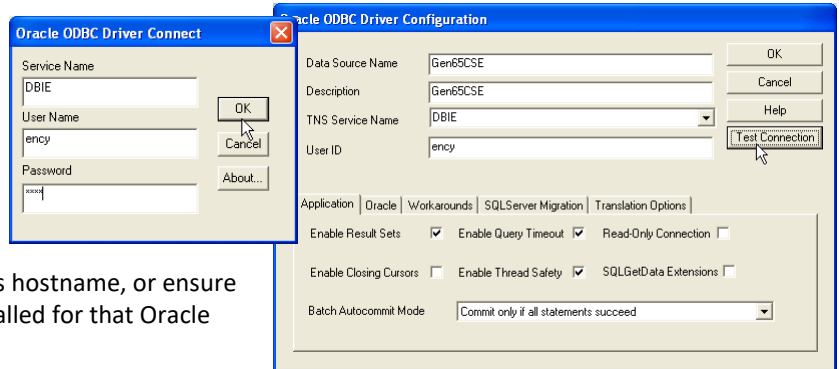
SQLServer CSE

- Choose the SQL Server ODBC Driver.
- Press Finish to create the new Data Source.
- The SQLServer-specific connection parameters dialog will appear next. Enter a name for your new ODBC Data Source and the Server name for the SQLServer that holds you CSE. Note that if you are using a centralised home directory, then you should use the same name for the Data Source as specified in the central gdologon file in that home directory.
- Press *Next* to move onto the Authentication panel.

- It is most common to use *SQL Server authentication* when connecting to SQL Server i.e. userid *sa*. Check with your CSE administrator what the connection details should be here.
- If you do not know the correct userid/password, then the test will fail, but this does not matter – as long as the Data Source itself is created.
- If you get a connection failure when pressing *Next* then simply hit *OK*, then *Back*, then *Finish* (back on the original dialog). The Data Source will be created anyway.
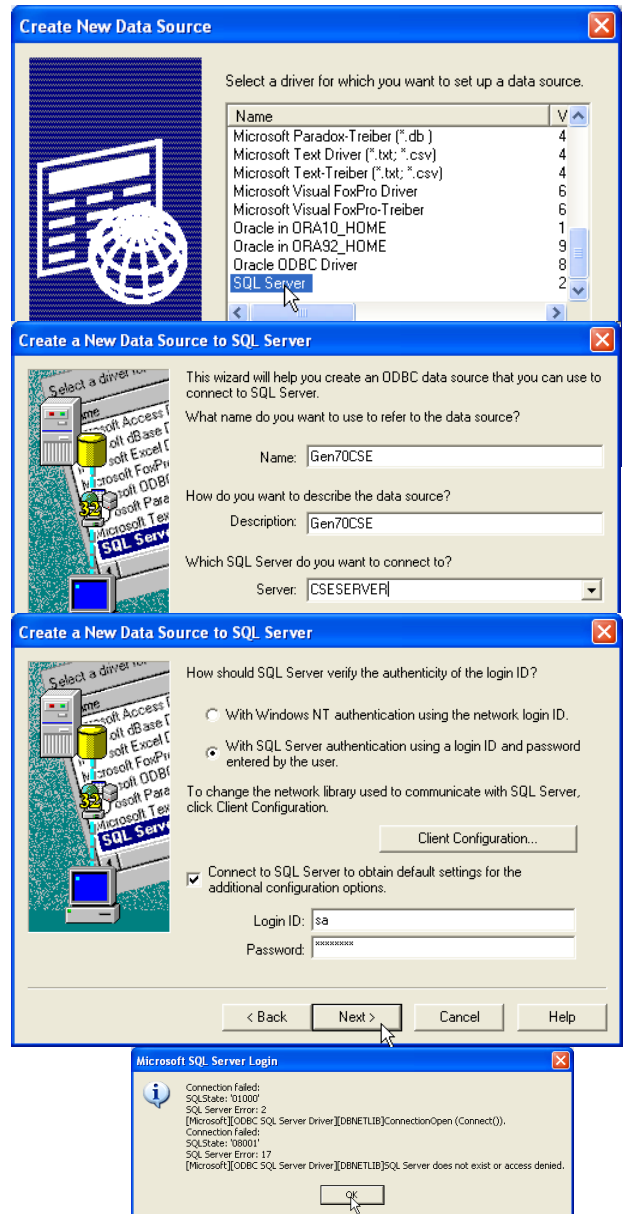
MVS DB2 Host Ency

The IBM DB2 ODBC Driver for Windows is provided with IBM DB2 Connect.

- Install the DB2 Connect software first…
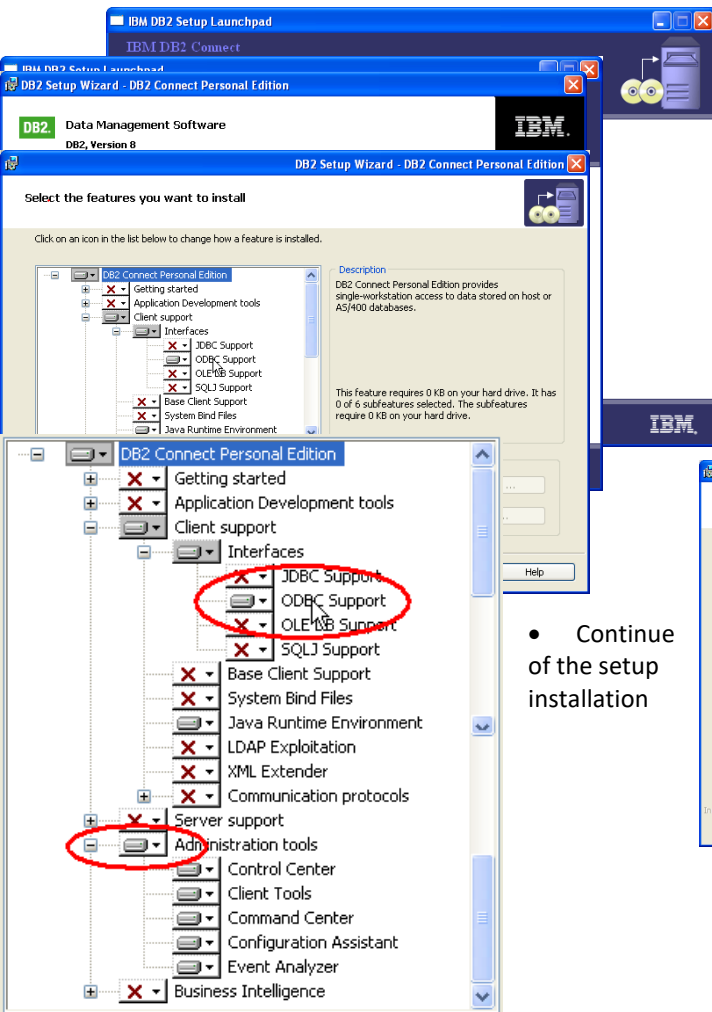
- Continue of the setup installation
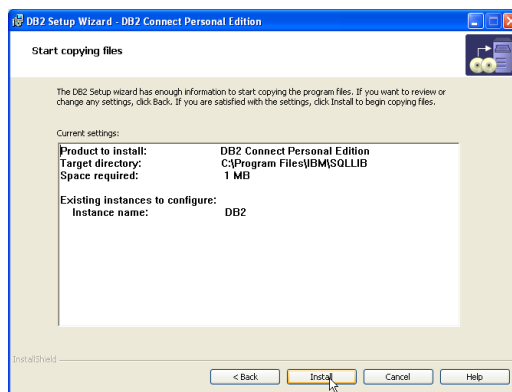
- Ensure that you have the options selected for *ODBC Support*, and *Administration Tools*. through the remainder and complete the of this software.

- Next you need to configure DB2 Connect…
- Launch the *Configuration Assistant*, which should have been installed above.
- On the *Selected* menu, choose *Add Database Using Wizard*…

- Choose *Manually Configure* and press *Next*.

  Manually configure a connection to a database

  Select if you know all the information necessary to connect to the desired database. This includes the protocols supported by the rem... name of the database.

  Add Database Wizard
  1. Source
  2. Protocol
  3. TCP/IP
  4. Datab...
  5. Data ...
  6. Node ...
  7. Syste...
  8. Securi...

  Select a communications protocol
  Select the communications protocol that you want to use to connect protocol. Indicate if your database is located on a host or OS/400 sy database is a gateway connection, then select the type of connection
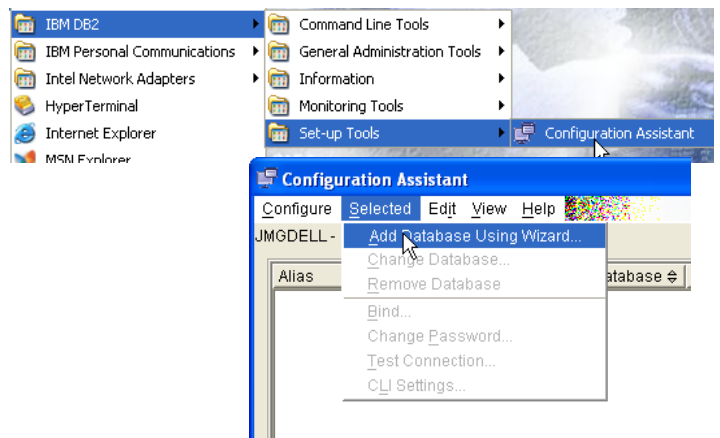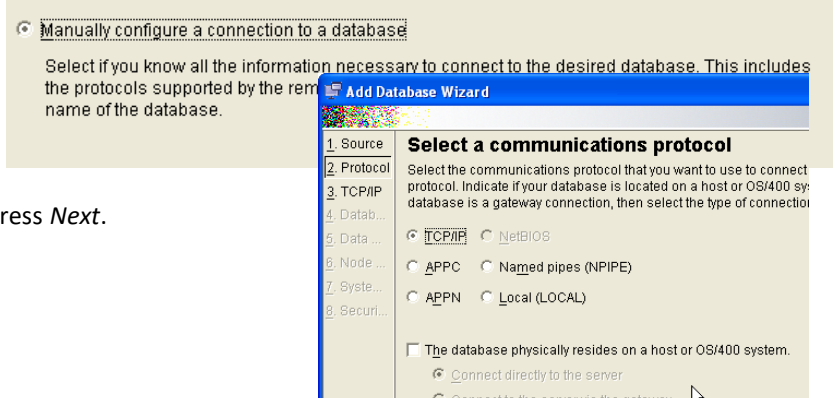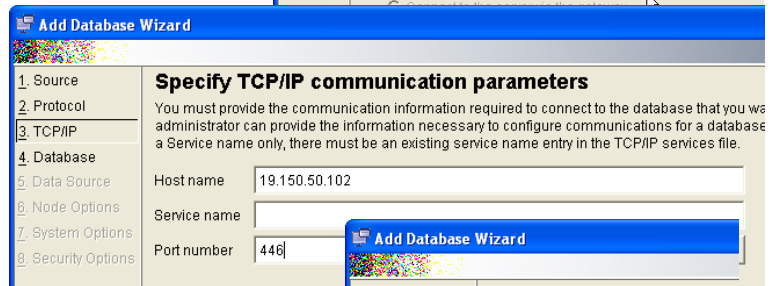
  ⦿ TCP/IP ○ NetBIOS
  ○ APPC ○ Named pipes (NPIPE)
  ○ APPN ○ Local (LOCAL)

  ☐ The database physically resides on a host or OS/400 system.
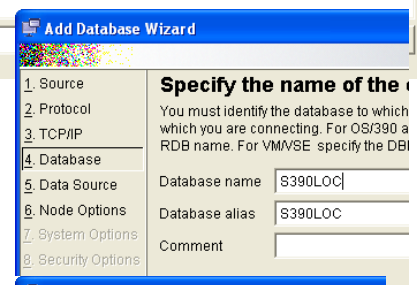 ⦿ Connect directly to the server

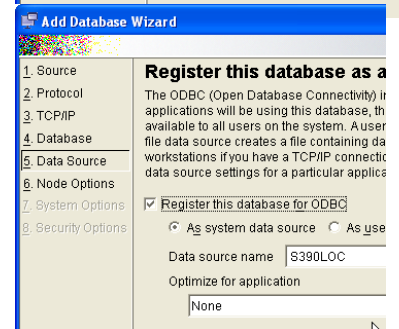- For step 2 (protocol), choose TCP/IP only and press *Next*.

- For step 3 (TCP/IP), enter the IP address or DNS Host Name for your OS/390 or z/OS server.
- Also enter the Port number that will service DB2 Connect communications. Default is 446, but check with your DB2 administrator if they have used a different value.

  Add Database Wizard
  1. Source
  2. Protocol
  3. TCP/IP
  4. Database
  5. Data Source
  6. Node Options
  7. System Options
  8. Security Options

  Specify TCP/IP communication parameters
  You must provide the communication information required to connect to the database that you wa administrator can provide the information necessary to configure communications for a database a Service name only, there must be an existing service name entry in the TCP/IP services file.

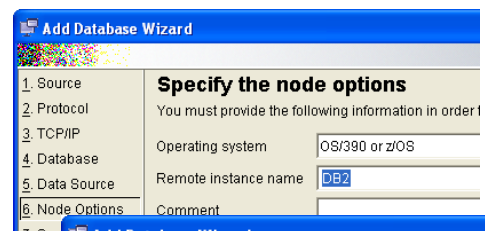 Host name   19.150.50.102
  Service name
  Port number   446

- For step 4 (Database), enter the Database name (alias is normally the same). Note that if you are using a centralised home directory, then you should use the same name for the Data Source as specified in the central gdologon file in that home directory.

  Add Database Wizard
  1. Source
  2. Protocol
  3. TCP/IP
  4. Database
  5. Data Source
  6. Node Options
  7. System Options
  8. Security Options

  Specify the name of the
  You must identify the database to which which you are connecting. For OS/390 a RDB name. For VM/VSE specify the DB

  Database name   S390LOC
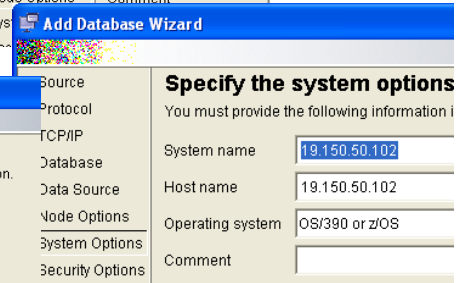 Database alias   S390LOC
  Comment

- For step 5 (Data Source), tick the box for *Register this database for ODBC* and make sure that *As system data source* is selected. The Data source name should be the same as the Database name specified on the previous dialog.

  Add Database Wizard
  1. Source
  2. Protocol
  3. TCP/IP
  4. Database
  5. Data Source
  6. Node Options
  7. System Options
  8. Security Options

  Register this database as a
  The ODBC (Open Database Connectivity) i applications will be using this database, th available to all users on the system. A user file data source creates a file containing da workstations if you have a TCP/IP connectio data source settings for a particular applica

  ☑ Register this database for ODBC
  ⦿ As system data source ○ As use
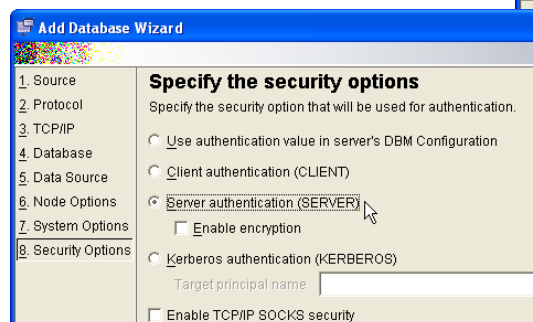 Data source name   S390LOC
  Optimize for application
  None

- For step 6 (Node Options), choose "OS/390 or z/OS" from the drop-down for *Operating System*.
- For *Remote instance name* type in "DB2", although check with your DB2 administrator that this is the correct instance name to use.

  Add Database Wizard
  1. Source
  2. Protocol
  3. TCP/IP
  4. Database
  5. Data Source
  6. Node Options
  7. Sys...

  Specify the node options
  You must provide the following information in order

  Operating system   OS/390 or z/OS
  Remote instance name   DB2
  Comment

- For step 7 (System Options), the defaults suggested should be correct (determined from previous dialogs in the wizard).

  Add Database Wizard
  Source
  Protocol
  TCP/IP
  Database
  Data Source
  Node Options
  System Options
  Security Options

  Specify the system options
  You must provide the following information i

  System name   19.150.50.102
  Host name   19.150.50.102
  Operating system   OS/390 or z/OS
  Comment

- For step 8 (Security Options), simply choose *Server authentication* and press the *Finish* button to complete the Wizard.

  Add Database Wizard
  1. Source
  2. Protocol
  3. TCP/IP
  4. Database
  5. Data Source
  6. Node Options
  7. System Options
  8. Security Options

  Specify the security options
  Specify the security option that will be used for authentication.
  ○ Use authentication value in server's DBM Configuration
  ○ Client authentication (CLIENT)
  ⦿ Server authentication (SERVER)
  ☐ Enable encryption
  ○ Kerberos authentication (KERBEROS)
  Target principal name
  ☐ Enable TCP/IP SOCKS security

- After completion of the Wizard, you will be given the opportunity to test the connection. If you have a suitable MVS logon that can do this, then perform this test here.
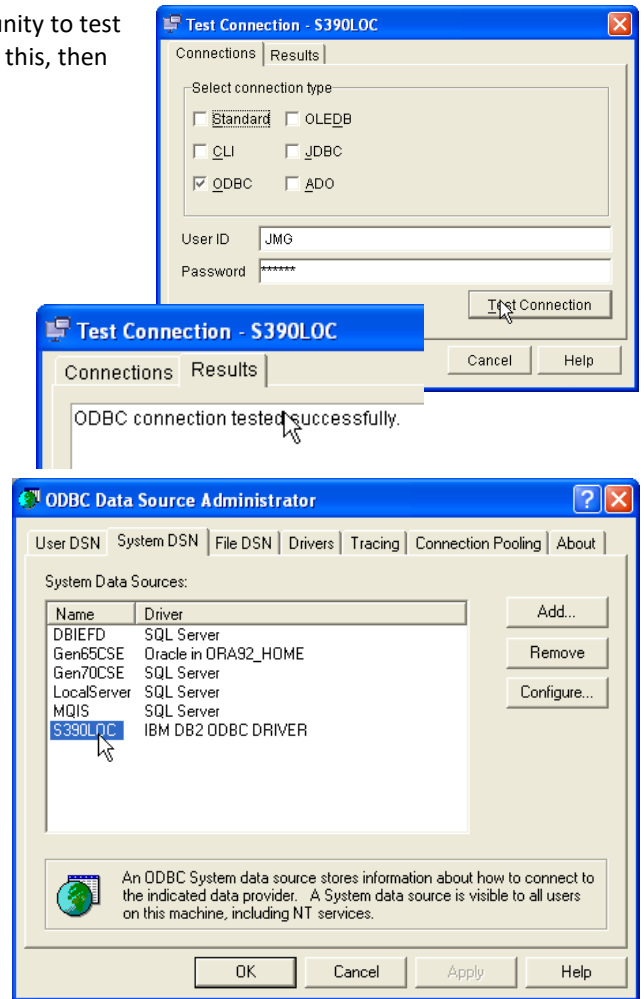
In your Windows ODBC Data Sources, you should find that you have a new Data Source defined for your DB2 connection.

Configuring a 32-bit ODBC Data Source on 64-bit Windows

pathvIEw is built to use a 32-bit ODBC Data Source. Therefore if you are installing on a 64-bit Windows Operating System, then you will need to be careful that the ODBC Data Source you define and use is a 32-bit Data Source, and not the default 64-bit Data Source.

Both 32-bit and 64-bit Data Sources are defined in the same manner using the Windows *ODBC Data Source Manager*. The version of this which is launched via the standard Start Menu shortcut and Control Panel is the 64-bit version. To launch the 32-bit version you must navigate to the appropriate directory and run the executable directly:

…windows\system\odbcad32.exe (the 64-bit ODBC Data Source Manager)
…windows\sysWOW64\odbcad32.exe (the 32-bit ODBC Data Source Manager)

Only one of these applications can be active at once, so if you already have the 64-bit DSM running, you will need to close it before running the 32-bit version.

Database Logon File

If your administrator has not provided a centralised gdologon file, then you will need to create this yourself. This file is used to provide the database connection credentials to the Data Source. The file is encrypted, so the connection details are not readable.

- To create the gdologon file, run the \Windows\Build\Logon\GDODLOGN.exe program.
- Enter the ODBC Data Source that you just created above.
- Enter the userid and password required for the connection to the GuardIEn database
- Use the *Test* button to test the connection.
- If successful, press the *Create* button and the gdologon file will be created.
- Copy this file to the HOME folder specified in PVSPP1.bat

Troubleshooting

Issue	Resolution
Generated code instance not visible in pathvIEw client	1) Browse the generated source code and verify that it has been post-processed by pathvIEw. Look in the top block of comments for the string "(modified for pathvIEw)". 2) Check that the GD69 job was able to store the PAD instance in the pathvIEw database.
Statement execution not logged in pathvIEw database	1) Check that you are displaying execution data for the same region that is specified in the runtime environment. 2) Check that the pathvIEw listener is started 3) Use the pathvIEw ping utility to test that you can communicate with the listener 4) Check that the listener ip address and port are correctly defined in the pathvIEw runtime environment 5) Check for error messages in the listener logs 6) Set the listener loglevel to 3 and see if there are any messages in the listener to indicate that it is receiving messages from the application 7) Enable debug mode in the application runtime and check for any error messages issued by pathvIEw in the pathvIEw log. See below for notes on enabling logging and debug.

Listener LogLevel

The logging output for the Listener is defined using either a startup parameter or a system parameter for the CICS listener.

The loglevel values are:
- 0 – no logging (recommended)
- 1 – Information
- 2 – Trace
- 3 - Verbose

For the CICS Listener, the loglevel is defined using the PVLOGLVL system parameter. The loglevel should be defined as a text parameter, for example '0'.

For Windows or UNIX, a startup parameter of LOGLEVEL=n may be specified where n is 0, 1, 2 or 3. Logging on Windows/UNIX may go to stdout (default) or to a file defined by another startup parameter LOGLOCATION=<filename>. Note that if the listener fails to write to the specified file, it will default back to logging only to stdout.

Runtime Debug

The pathvIEw application runtime can be configured to output debug information that can be useful in troubleshooting.

C Code
- Set the environment variable PATHVIEW_DEBUG to YES to enable debug.
- Debug output is written to a file called pathvIEw-AppLog-XXXXnnnn.log, where XXXX is the executable name or trancode and nnnn is the process id of the executable.

Java Code
- Amend the exit pathvIEw_Exit.java and enable the debug flag.
- Debug output is written to the application server console.

COBOL Code
- Amend the PVENVB or PVENVC exit and set the debug flag to Y. Re-compile and deploy the amended PVENV runtime module.
- Debug output is written to SYSOUT for batch programs and the CICS logs for CICS.