Use Discovery Analyzer to Debug Content

The Discovery Analyzer tool is intended for debugging purposes when developing packages, scripts, or any other content. The tool runs a job against a remote destination and returns logs containing information, warning and error details, and results of discovered CIs.

This section includes the following topics:

- "Tasks and Records" on page 604
- "Logs" on page 604

Tasks and Records

A task file contains data regarding a task to be executed. The task consists of information such as the job's name and required parameters that define the trigger CI, for example, the remote destination address.

A record file contains task information as well as the results of a specific execution. An execution is the detailed communication (including a response) between the Probe or Discovery Analyzer (whichever module executed the task) and the remote destination.

A task that is defined by a task file can be executed against a remote destination, whereas a task that is defined by a record file (that contains extra data regarding a specific execution) can be executed and can also be played back (that is, can reproduce the same execution documented in the record file).

Logs

Logs provide information about the latest run, as follows:

- **General Log**. This log includes all information data, errors, and warnings that occurred during the run.
- **Communication Log**. This log contains the detailed communication between the Discovery Analyzer and the remote destination (including its response). After the execution, the log can be saved as a record file.

➤ **Results Log**. Displays a list of discovered CIs. The appearance time of each CI depends on the design of the patterns and scripts.

You can save all logs together or each log separately. When you save all the logs, they are saved together under one name.

If you replay a record file, the same data is displayed in the communication log, the only difference being the time of execution.

The following section includes the following topics:

➤ "Working with Discovery Analyzer" on page 605

Working with Discovery Analyzer

The following procedure explains how to work with Discovery Analyzer.

This section includes the following topics:

➤ "Prerequisites" on page 606

➤ "Access Discovery Analyzer" on page 606

➤ "Define a Task" on page 607

➤ "Edit a Task" on page 609

➤ "Save the Task and Logs" on page 609

➤ "Run the Task" on page 610

➤ "Send a Task Result to the Server" on page 610

➤ "Import Settings" on page 611

➤ "Breakpoints" on page 611

➤ "Troubleshooting and Limitations" on page 611

1 Prerequisites

- The Probe must be installed. (The Discovery Analyzer is installed as part of the Probe installation process and shares resources with it.)
- The Probe does not need to be running while you are working with Discovery Analyzer.

  However, if the Probe has already run against a UCMDB server, all the required resources are already downloaded to the file system. If the Probe has not run, you can upload resources needed by Discovery Analyzer through the Settings menu. For details, see "Import Settings" on page 611.
- The UCMDB server does not need to be installed.

2 Access Discovery Analyzer

You access Discovery Analyzer either:

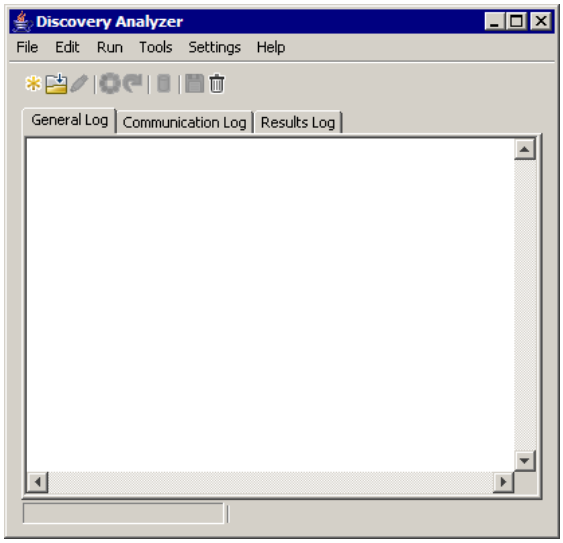
- **When working with Eclipse**. The recomended plugin is **pydev**. Install PyDev and PyDev Extensions, available at: **http://pydev.sourceforge.net/** (PyDev) and **http://www.fabioz.com/pydev/** (PyDev Extensions).

Note: Ensure that you configure the pydev plug-in correctly, including the Jython interpreter. The recommended way is to run the Jython interpreter that is included with the DDM Probe installation. In Eclipse, choose **Preferences**. The workspace includes a .pydevproject file that defines the class path of pydev. It is selected automatically if pydev is correctly installed.

The Probe installation comes with a default Eclipse workspace located at **C:\hp\DDM\DiscoveryProbe\discoveryAnalyzerWorkspace**. This workspace includes a Jython script to start DiscoveryAnalyzer (**startDiscoveryAnalyzerScript.py**) as well as a link to all DDM scripts. If you start the tool in this way, you can locate breakpoints within the Jython scripts for debugging purposes.

➤ **Directly, by double-clicking the file** in the following folder: **C:\hp\DDM\DiscoveryProbe\root\lib\collectors\discoveryAnalyzer.cmd**. For details, see the following section.

The Discovery Analyzer window opens:



3 Define a Task

You define a task using one of the following methods:

➤ By defining a new task. For details, see "Define a New Task" on page 607.

➤ By importing a task from a record file. For details, see "Retrieve a Record" on page 609.

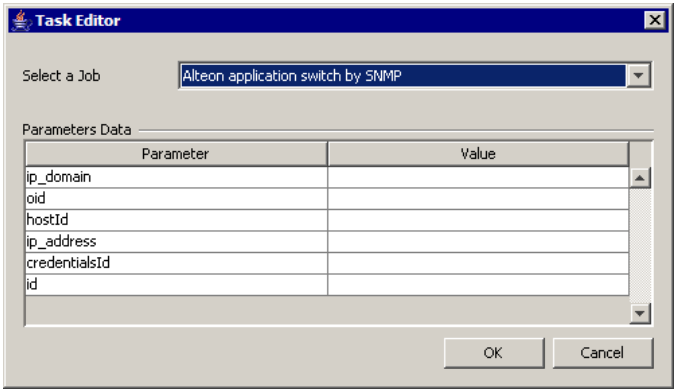
➤ By importing a saved task from a task file. For details, see "Open a Task File" on page 609.

➤ By retrieving a job from the Probe's internal database. For details, see "Import a Job from the Database" on page 609.

Define a New Task

a Display the Task Editor: click the **New Task** button.

The Task Editor displays a list of jobs that currently exist in the file system. This list is updated each time the Probe receives tasks from the server, or packages are deployed manually from the Settings menu.



b Select a job.

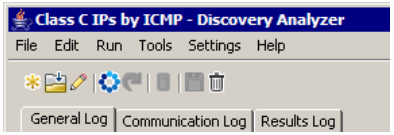
c Enter values for all parameters.

The parameters displayed here are DDM pattern parameters. They can be viewed in the Discovery Pattern Parameters pane in the Pattern Signature tab. For details, see "Discovery Pattern Parameters Pane" on page 97.

All fields are mandatory (unless a job's script demands that the field be empty).

For parameters that require an ID or credentials ID input value, you can use randomly created IDs: right-click the value box and select **Generate random CMDB ID** or **Credential Chooser**.

The task is now active and the name of the open task is displayed in the title bar:



d Continue with the procedure for defining a task. For details, see "Save the Task and Logs" on page 609.

Retrieve a Record

You can define a task by opening a record file containing data regarding a specific execution. If a task is defined in this way, you can reproduce the specific execution by selecting the playback option. (If a task is replayed, responses are received from the data stored in the record file and not from the remote destination.)

Select **File** > **Open Record**. Browse to the folder where you saved the record. The record is now active and the name of the task is displayed in the title bar.

Open a Task File

You can define a task from a task file: Select **File** > **Open Task**.

Import a Job from the Database

You can retrieve a job from the Probe database on condition that the Probe has already run and has active tasks in its internal database. You can use the parameter values to define the task.

a Select **File** > **Import Task from Probe Database**.

b In the dialog box that opens, select the job to run and click **OK**.

c Continue with the procedure for defining a task. For details, see "Save the Task and Logs" on page 609.

4 Edit a Task

After a task is defined, the name of the task (or the file) is displayed in the title bar. Now the file can be edited.

a Select **Edit** > **Edit Task**.

b Make any changes to the task and click **OK**.

5 Save the Task and Logs

You can save task parameters: Select **File** > **Save Task**.

The following options are available only after a task is executed.

- Save a record of the task. You can save the task parameters and the results of the task run: Select **File** > **Save Record**.
- Save a log of the task: Select **File** > **Save General Log**.
- Save results: Select **File** > **Save Results**.

6 Run the Task

The next step in the procedure is to run the task you created.

a To execute the task only against a remote destination, click the **Run Task** button.

Discovery Analyzer executes the job and displays information in the three log files: **General**, **Communication**, and **Results**.

b You can save the log files, either together or separately: Select **File** > **Save General Log**, **Save Record**, **Save Results**, or **Save All Logs**. For details on the log files, see "Logs" on page 604.

c If a task is retrieved from a record file, the execution that is documented in this file can be reproduced by clicking the **Playback** button. The same Communication log is displayed, but the execution time is updated.

7 Send a Task Result to the Server

If a task's execution ends with results (that is, the Results Log tab displays a list of discovered CIs), you can send the results to the UCMDB server. This is useful if, for example, you were previously testing a script when the server was down.

Note: You can send results only to a UCMDB server that receives tasks from the Probe that is installed on the same machine as Discovery Analyzer.

---

8 Import Settings

If the Probe has not yet run, you can import files needed by Discovery Analyzer. Access the Settings menu and import **domainScopeDocument.xml** or **domainScopeDocument.bin**. If you import the .bin file, you must import **key.bin** too. You can deploy any necessary DDM packages (including scripts, patterns, and so on) by selecting: **Settings** > **Import packages**.

9 Breakpoints

If you run Discovery Analyzer from the Python script, you can add breakpoints to your script.

10 Troubleshooting and Limitations

To prevent an error being thrown during the tool run, before using the DDM Analyzer, verify that the following line exists in the DiscoveryProbe.properties file:

```
# Fix used to run DiscoveryAnalyzer from Eclipse (should be true if running from eclipse else should be false)
appilog.agent.local.discoveryAnalyzerFromEclipse = true
```