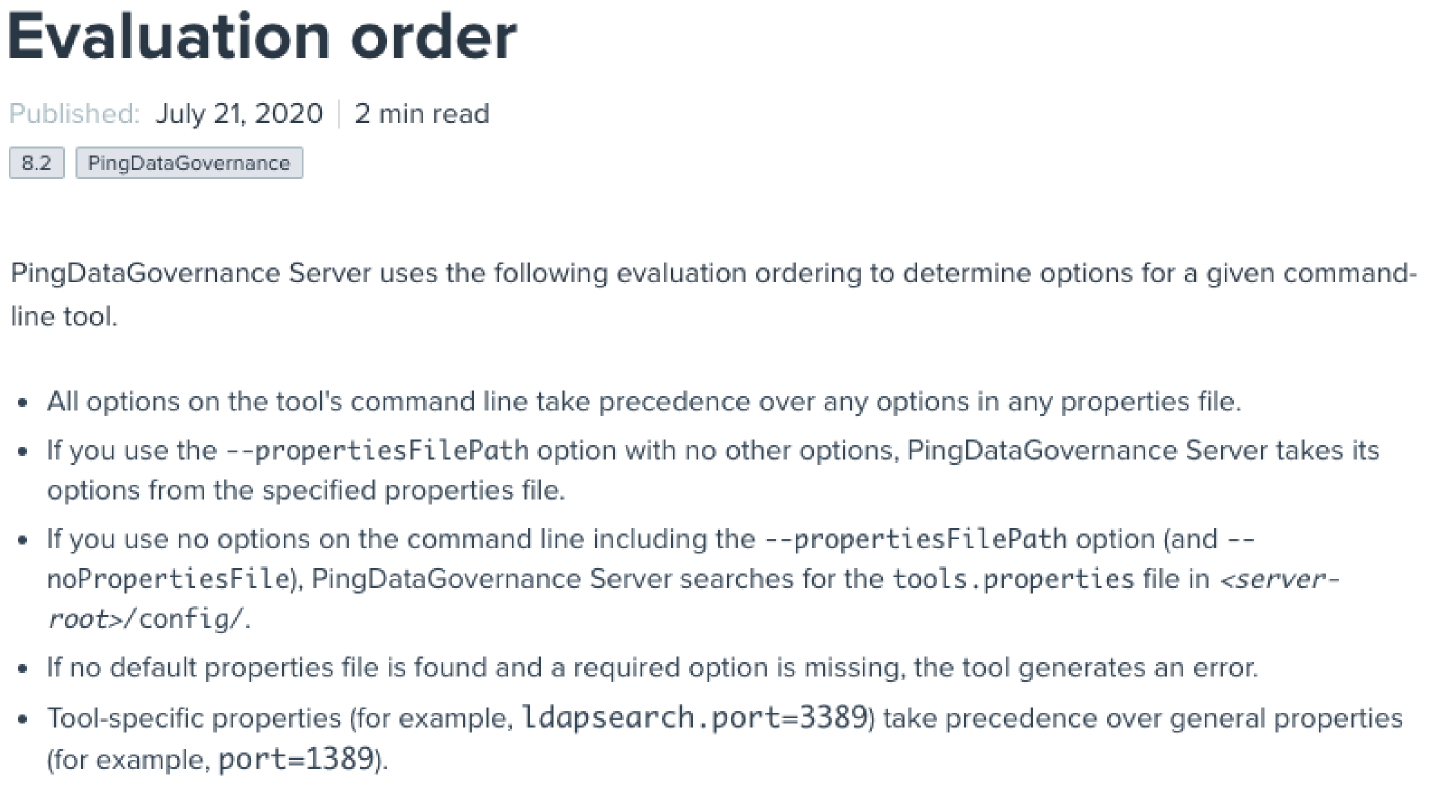
This document provides:

- A “before” version of a topic I edited
- Comments about the changes I made
- An “after” version that reflects the comments

Before version

The original text spans two pages.



Evaluation order

Published: July 21, 2020 | 2 min read

8.2 PingDataGovernance

PingDataGovernance Server uses the following evaluation ordering to determine options for a given command-line tool.

- All options on the tool's command line take precedence over any options in any properties file.
- If you use the `--propertiesFilePath` option with no other options, PingDataGovernance Server takes its options from the specified properties file.
- If you use no options on the command line including the `--propertiesFilePath` option (and `--noPropertiesFile`), PingDataGovernance Server searches for the `tools.properties` file in *<server-root>*`/config/`.
- If no default properties file is found and a required option is missing, the tool generates an error.
- Tool-specific properties (for example, `ldapsearch.port=3389`) take precedence over general properties (for example, `port=1389`).

Consider this example properties file that is saved as *<server-root>*`/bin/tools.properties`:

```
hostname=server1.example.com
port=1389
bindDN=cn=Directory\ Manager
bindPassword=secret
```

PingDataGovernance Server locates a command-line option in a specific priority order.

1. All options presented with the tool on the command line take precedence over any options in any properties file. In the following example, the client request is run with the options specified on the command line (`--port` and `--baseDN`). The command uses the `bindDN` and `bindPassword` values specified in the properties file.

   ```
   $ bin/ldapsearch --port 2389 --baseDN ou=People,dc=example,dc=com \
     --propertiesFilePath bin/tools.properties “(objectclass=*)”
   ```

2. Next, if you specify the properties file using the `--propertiesFilePath` option and no other command-line options, PingDataGovernance Server uses the specified properties file as follows:

   ```
   $ bin/ldapsearch --propertiesFilePath bin/tools.properties \
     “(objectclass=*)”
   ```

3. If no options are presented with the tool on the command line and the `--noPropertiesFile` option is not present, PingDataGovernance Server attempts to locate the default `tools.properties` file in the following location:

   ```
   <server-root>/config/tools.properties
   ```

   If you move your `tools.properties` file from *<server-root>*`/bin` to the *<server-root>*`/config` directory, you can then run your tools as follows:

   ```
   $ bin/ldapsearch "(objectclass=*)"
   ```

   You can configure PingDataGovernance Server so that it does not search for a properties file by using the `--noPropertiesFile` option. This option tells PingDataGovernance Server to use only those options specified on the command line. You cannot use the `--propertiesFilePath` and `--noPropertiesFile` options together.

4. If no default `tools.properties` file is found and no options are specified with the command-line tool, the tool generates an error for any missing arguments.

Comments

The following list summarizes the updates I made to the text and why.

- The “Evaluation order” title works okay when seen in context. However, the new title, “Evaluation of command-line options and file options,” immediately tells the reader what we are evaluating.

- The section focused on the order of evaluation. However, the content is actually about different scenarios where order of evaluation does not matter. To make those scenarios and their results easier to scan, I created a table with the scenarios in the first column and the product responses to the scenarios in the second column.

- In addition to easier scanning, the table removes filler text and discusses the scenarios in the same way, reducing the cognitive load from the original text.

- The original text has this line:

  If no default properties file is found and a required option is missing, the tool generates an error.

  Given that the tool does produce an error in this case, this line seems unnecessary. I removed it from the rewrite.

- The original text has this line:

  Tool-specific properties (for example, ldapsearch.port=3389) take precedence over general properties (for example, port=1389).

  Describing how the properties file works does not seem relevant in this section. Also, the section about creating the properties file already mentions this behavior. I removed this line from the rewrite.

- I like having an example to illustrate the behavior. To help the reader see the example, I created an “Example” subsection for this content.

- The example in the original text mentioned order and even has a numbered list to reflect the order. However, as mentioned, this content covers scenarios where order does not matter. I changed the numbered list to a bulleted list.

- If I had kept the numbered list though, I would have needed to change step 2 to remove “Next.” The word is not needed because the numbered list already indicates an order.

- In the rewrite, step 1 becomes the first bullet, but I use multiple paragraphs in the bullet to make the text less dense. Also, I changed the “client request” text to “command” to match the wording used later in the paragraph. I also added the “With the port value both on the command line and in …” sentence to clarify how that relationship works.

- Step 3 becomes the third bullet in the rewrite. The step has this text:

  If no options are presented with the tool on the command line and the --noPropertiesFile option is not present,

  I reduced that text to the following text to reduce the cognitive load:

  If you don’t specify any command-line options,

- The rewrite also uses some contractions.

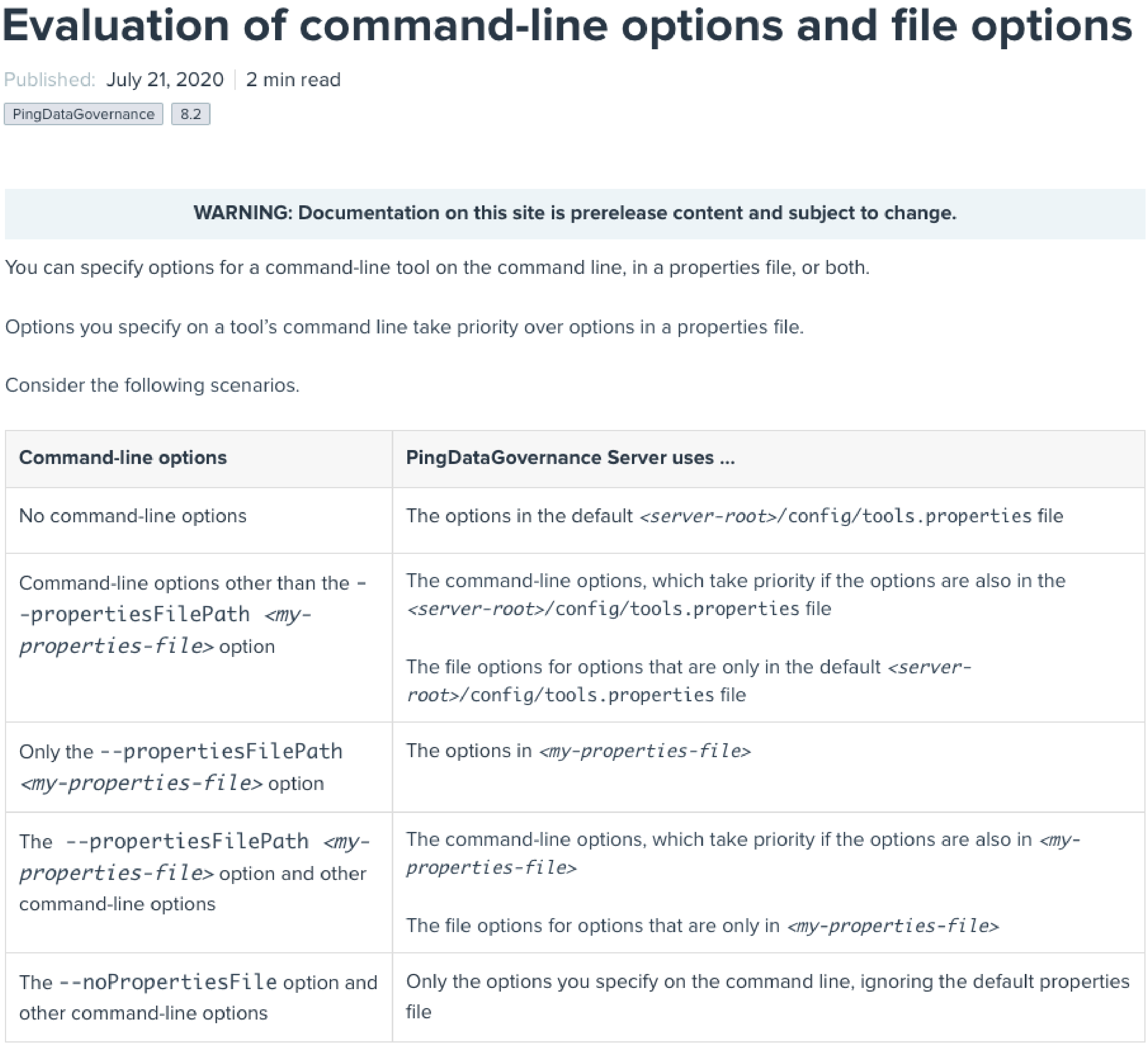
- Step 3 has a para that starts “You can configure PingDataGovernance Server so that it does not search …” Most of that para is a scenario, so I moved it to the table with the other scenarios.

  I removed the last line from that para, which starts with “You cannot use.” Presumably, the tool will produce an error in the specified case that indicates proper usage.

- Step 4 is another example of text that mentions when the tool produces an error. I left out this text in the rewrite.

After version

The rewrite reflects my updates.



Evaluation of command-line options and file options

Published: July 21, 2020 | 2 min read

PingDataGovernance 8.2

WARNING: Documentation on this site is prerelease content and subject to change.

You can specify options for a command-line tool on the command line, in a properties file, or both.

Options you specify on a tool's command line take priority over options in a properties file.

Consider the following scenarios.

Command-line options	PingDataGovernance Server uses ...
No command-line options	The options in the default *<server-root>*`/config/tools.properties` file
Command-line options other than the `--propertiesFilePath` *<my-properties-file>* option	The command-line options, which take priority if the options are also in the *<server-root>*`/config/tools.properties` file The file options for options that are only in the default *<server-root>*`/config/tools.properties` file
Only the `--propertiesFilePath` *<my-properties-file>* option	The options in *<my-properties-file>*
The `--propertiesFilePath` *<my-properties-file>* option and other command-line options	The command-line options, which take priority if the options are also in *<my-properties-file>* The file options for options that are only in *<my-properties-file>*
The `--noPropertiesFile` option and other command-line options	Only the options you specify on the command line, ignoring the default properties file

Example

Consider this example properties file that is saved as *<server-root>*`/bin/tools.properties`:

```
hostname=server1.example.com
port=1389
bindDN=cn=Directory\ Manager
bindPassword=secret
```

PingDataGovernance Server checks command-line options and file options to determine the options to use, as explained below.

- All options presented with the tool on the command line take precedence over any options in a properties file.

  In the following example, the command runs with the options specified on the command line (`--port` and `--baseDN`). With the `port` value both on the command line and in the properties file, the command-line value takes priority. The command uses the `bindDN` and `bindPassword` values specified in the properties file.

  ```
  $ bin/ldapsearch --port 2389 --baseDN ou=People,dc=example,dc=com \
    --propertiesFilePath bin/tools.properties “(objectclass=*)”
  ```

- If you specify the properties file using the `--propertiesFilePath` option and no other command-line options, PingDataGovernance Server uses only the options in the specified properties file:

  ```
  $ bin/ldapsearch --propertiesFilePath bin/tools.properties \
    “(objectclass=*)”
  ```

- If do not specify any command-line options, PingDataGovernance Server attempts to locate the default properties file in the following location:

  ```
  <server-root>/config/tools.properties
  ```

  By moving your `tools.properties` file from *<server-root>*`/bin` to *<server-root>*`/config`, you do not have to specify the `--propertiesFilePath` option. That change shortens the previous command to the following command.

  ```
  $ bin/ldapsearch "(objectclass=*)"
  ```