3.0 FAQ

Q: What does this tool cover?

A: The *Tool* tests your system's ability to discover organizationally-bound and user-bound certificates following the Certificate Discovery process required by Direct. It also tests that your certificates are discoverable following the Certificate Discovery process.

Q: Is there a User Guide?

A: Yes: The 3.0 Release User Guide is the latest. Previous user guides can be found in a particular release's wiki space.

Q: Is there a demo version of the *Tool* already deployed that I can use to test my Direct instance?

A: The demo version of the tool can be found here.

Q: Which browsers are supported by the *Tool*?

A: The *Tool* can be used on Chrome, Firefox, and Internet Explorer. Specifically, the Tool has been tested with the following versions: Chrome versions > 32, Firefox versions > 24. Internet Explorer may work for versions > 10.

Q: When I try to download the anchor file in Firefox, it asks me if I want to trust this and a RootCA for my local system. Should I do this?

A: This is not what you want to do. You need to right-click the link and choose "Save Link As...", save it somewhere on your system, and add it to your Direct instance as a trust anchor.

Q: My System doesn't trust the *Tool* even after uploading your anchor to my anchor store. Is there something wrong with your anchor?

A: We've noticed that some systems (including the Java Reference Implementation) take 5 minutes or longer to fully integrate any new anchors into their system. If you want to shorten this waiting period, restarting your James server should do the trick.

Q: I'm not receiving any response emails but my Direct messages are being sent to the demo site.

A: Check your junk mail or spam folders. Sometimes the messages are routed to these folders. Look out for messages coming from: results@dcdt30prod.sitenv.org.

Q: Which tests are required to demonstrate Meaningful Use Stage 2 (MU2) capabilities?

A: The Tool is divided into two types of tests, Hosting and Discovery. All Discovery tests are required for MU2 certification. However, for Hosting tests - the System Under Test (SUT) only has to take the tests that apply to their implementation - which could be as few as one test (e.g. Address-Bound DNS), 2 tests (Address-Bound DNS and Domain-Bound DNS), or all 4 tests (Address/Domain-Bound for both DNS and LDAP).

In other words, the SUT MUST be able to acquire certificates from any other conformant Direct implementation - regardless of the choices that system made; but for Hosting, the SUT only needs to prove at least one hosting method (systems should test for every hosting method they support, so if your product implements all optional methods - you MUST pass all 4 Hosting test cases).

Q: Why do some tests say that I failed because I didn't follow the correct SRV record priorities?

A: The specifications are written such that the priorities of the SRV Records should be taken into account by initiating Direct implementations. Here is a quote from the specifications regarding this notion:

> *"From the list of LDAP services the consumer should attempt to contact them based first on the priority value and, if there is more than one with the same priority value, they should then be ordered based on the weight value."*

Please note: this is a SHOULD requirement, but not a MUST. Our Tool highlights these discrepancies and warns the consumer when they ignore the priority values.

Q: What is meant by high priority LDAP instances vs. low priority LDAP instances?

A: If you fail a test for choosing the wrong priority valued LDAP server initially, we send a warning message stating that you chose the higher priority valued LDAP server (which is identified in an SRV Record) instead of the lower priority valued LDAP server. When this occurs, your system chooses an LDAP server with a higher valued priority (e.g. "2") instead of a lower valued priority (e.g. "1"). We send our diagnostic information in the human-readable format as opposed to the technical terminology for priority value. See RFC 2782 for more information about SRV records and their priorities.

Q: Where does the code live?

A: The code is in a GitHub repository and in a read-only Google Code repository, which is retired. The code can be built from source by following the 3.0 Source Build Guide.

Q: How do I submit a defect?

A: To submit a defect, please enter the issue using our JIRA issue tracker.

Q: If I choose to install the *Tool* locally, do I have to build it from source?

A: No, you can download the latest WAR file (here) and deploy it to your application server.

Q: How is the DCDT version 3.0 release different from previous releases?

A: The 3.0 release contains many improvements for the end user. Significant enhancements and improvements were made to: (1) Remove the dependency on the Direct Java RI, so that DCDT is now a stand-alone application with no dependency on the RI whatsoever; (2) Simplify the setup of DCDT: two fields -- domain name and IP address -- in the administration console are all that is needed to completely and automatically configure the tool and its certificates; and finally (3) Provide detailed feedback to the user at each step of every test case: detailed success and error messages provide specific, useful information to the user.

Q: How does the new DCDT release provide more detailed feedback?

A: Whenever a test is run, details of each step of every test case are displayed to the user. This makes it easy for the user to pinpoint the problem when a test fails. Whether a test passes or fails, all parameters and results are displayed to the user. Subsequent results are added to the web page as test cases are run, and previous results are minimized in accordion-like fashion.

Q: Is a full Java RI installation required for the new version 3 release?

A: No, DCDT is no longer dependent on the Direct Java RI. Previously, a full Java RI installation was required to provide underlying Direct functionality to DCDT, such as LDAP, DNS, email, and cryptography. DCDT now natively handles this functionality, so the installation process is much simpler.

Q: Why is DCDT trying to discover a sender certificate for Discovery testcases?

A: DCDT tries to discover a sender certificate for Discovery testcases by performing certificate discovery on the sender address of the Discovery testcase submission mapping and then verifies that it matches the certificate included in the mail signature. Although it is an optional requirement for sender/signer certificates to be discoverable, according to Section 1.4 of the Applicability Statement:

> *"The organization SHOULD publish the certificates for discovery by other implementations for the purposes of encryption and signature verification."*

this is the only way that DCDT can assert the authenticity of the sender and signer certificate since DCDT does not keep any trust chains for any sender and/or signer certificates.

In order to pass the Discovery testcases, sender certificates MUST be hosted properly (passes one of the Hosting testcases) so that it can be discovered by DCDT.