How to diagnose connection issues with Transact Print Preview Interface

Ivana Dordevic - 2022-11-04 - Comments (0) - Temenos FAQs

Interface Overview

The Autoform Print Preview interface adds the ability to generate a PDF from raw Transact data 'on the fly'. Therefore, instead of printing out raw data for the customer, it is possible to use this functionality to instantly generate a properly formatted customer-facing document for local printing.

These instructions are only relevant to the older Transact installations.

How the connections work

The raw data is sent from Transact via an HTTP Post to Lasernet. Lasernet formats the data and then returns the PDF to Transact to be shown on the screen. The Interface is run via a browser tool that uses a specific Print Preview function via a Version or Enquiry.

The integration manual, supplied during the initial installation, contains the key details regarding the interface configuration and which files are necessary. The instructions below are based on a JBOSS Temenos installation, however, the functionality and any possible issues should be broadly similar for other platforms (though the file locations may vary).

Possible issues

Generally, most issues are caused by network interference, incorrect configuration or too much throughput for the current infrastructure. Basic troubleshooting steps for diagnosing common print preview issues are described below. However, due to the complexity of the interface, it is impossible to cover every scenario. If you experience any difficulty in troubleshooting the interface please contact Formpipe Support and we will be happy to discuss and assist with any issues that you may encounter.

How to test the network

Print Preview relies heavily on a stable network connection between the Temenos and

Lasernet servers. If there is any interference between the two, this could cause issues with the Print Preview functionality.

The following filter configuration can be found in the Web.XML in Transact (your XML may have some extra parameters):

```
<filter>
  <filter-name>PrintPdfFilter</filter-name>
  <filter-class>com.efstech.temenos.filters.PrintPdfFilter</filter-
class>
    <init-param>
      <param-name>target</param-name>
      <param-value>/servlet/LaserNetDataServlet</param-value>
    </init-param>
    <init-param>
        <param-name>lasernet</param-name>
        <param-value>
            http://LNSERVER:28080/webinputport/T24Input/?preview,
            http://LNSERVER:28081/webinputport/T24Input2/?preview
        </param-value>
    </init-param>
</filter>
```

POST data from Transact is received by Lasernet via two web ports. These are defined using the URLs listed under the param-name of “lasernet” (shown above). In order to function correctly, the Lasernet server needs to be reachable by the Transact server (across the network as well as having a resolvable domain name) and the port to be open in all relevant firewalls. All connections to the Lasernet server are made directly from the Transact server. No connections are needed between the teller machines and the Lasernet server. They can be tested using the following methods:

1. Ping the Lasernet server from the Transact server. This should resolve the Lasernet server name to an IP address and get a reply with an appropriate TTL time. If the IP address fails to resolve, the issue is likely to be with the server name or DNS. If the IP resolves but the server is unable to be found, this is likely to indicate a problem with the connection between Transact and the Lasernet server.

2. Telnet to the Lasernet Server Port from the Transact server. This should result in a blinking cursor on a blank screen. If this returns an error or times out, then the port on the Lasernet server is not reachable, most likely due to a firewall blocking connections to that port.

3. Access the URLs above using the Google Chrome web browser. A successful connection will return a blank page. In the Developer Tools Console (opened with F12), it will return a '405 Method Not Allowed' error. Successful connection attempts using Chrome will also be shown in the Lasernet Logs as "Ignoring request with non-allowed verb (GET)". If the connection is not established, an error will be shown in Chrome, whilst Developer Tools will show "ERR_CONNECTION_REFUSED".

Logging

Logging for Print Preview can be enabled in both Lasernet and Temenos. Lasernet logging is set to 'Debug' by default. Different logging can be filtered out by going to "Filter" in the Lasernet Monitor and turning the required logging on or off.

In Temenos, Print Preview logging can be enabled by turning on com.efstech logging in the log4j.properties file and setting it to DEBUG, this will be in a location similar to: `BrowserWeb.war\WEB-INF\classes\log4j.properties` (or alternatively jboss specific log config file).

The logging in JBOSS should show messages similar to this when it is set up and Print Preview is run:

```
INFO  [com.efstech.temenos.filters.PrintPdfFilter]
(http-0.0.0.0-8085-1) Forwarding to '/servlet/LaserNetDataServlet'

INFO  [com.efstech.temenos.servlets.LaserNetDataServlet]
(http-0.0.0.0-8085-1) Received request to LaserNetInputServlet
INFO  [com.efstech.temenos.servlets.LaserNetDataServlet]
(http-0.0.0.0-8085-1) Request processed successfully
```

This shows the Print Preview request was sent to Lasernet (via the LaserNetDataServlet servlet) and was returned and processed. Any errors that occur during the Print Preview from the Transact side will normally be shown in between these log entries.

Lasernet is where most of the processing takes place. An example extract of the logging from a successful print preview is shown below:

```
Lasernet Service;1;5652;56284;;POST request for
http://LNSERVER:28080/webinputport/T24Input?preview
10;10296;;;Passing job JOBNAME to FORM
Form Engine;10;10296;;;Applying module jobinfos to JOBNAME
Form Engine;12;10296;;;Set JobInfo FormType[0] = STATEMENT
Form Engine;12;10296;;;Set JobInfo RecognizedForm[0] = ACCOUNT
```

```
STATEMENT
Form Engine;10;10296;;;Number of pages generated from sheet: 1
Form Engine;10;10296;;;Printing page 1
Form Engine;10;10296;;;Form print done.
Form Engine;12;10296;;;Set JobInfo DataFormat[0] = EMF
Form Engine;10;10296;;;Passing job JOBNAME to PDF
PDF;1;8800;56303;;Processing page 1...
PDF;1;8800;56303;;Generating PDF...
PDF;10;8800;56303;;Passing job JOBNAME to Preview
Lasernet Service;11;5652;56314;;Committed job JOBNAME
```

When processing a job, the initial log message shows where the POST is sent to. In this example, LNSERVER, using Port 28080 and a Web Server input called T24Input. This should match the settings in the Web.XML.

The rest of the logging may be slightly different, depending on the engines used. However, the normal path is likely to take the job through the Form Engine and the PDF Converter before being passed to the Preview destination that returns it to Transact to be displayed on the front end.

The logging can be improved slightly by adding another parameter to the custom.js functions. The Print Preview functions (normally called “efs_enquiry_launchPdfRequest” and “efs_version_launchPdfRequest”) can take custom parameters and pass them to Lasernet as Jobinfos when the parameter name has “efs_” prefixing it.

We suggest the addition of the following line to the two functions, after the variable “enqselForm” gets set:

```
  _efs_addOrUpdateInputWithValue(enqselForm, "efs_correlationId",
_efs_createCorrelationId());
```

This will create a variable and jobinfo called “efs_correlationId” that is set to a unique ID for that job. This shows in Print Preview Debugging logging as:

```
2016-03-01 11:22:13,346 INFO
[com.efstech.temenos.servlets.LaserNetDataServlet]
(http-0.0.0.0-8085-4) Got custom parameter [efs_correlationId] value
[493227_1456831333260]
```

You can output this in Lasernet by running a script on a modifier and logging out the jobinfo using the following code (or variation thereof):

```
var corrID = job.getJobInfo("efs_correlationId")

logger.logEvent(123,"Input efs_correlationId is " + corrID);
```

This logs as:
`Input efs_correlationId is 493227_1456831333260`

A print preview request can then be tracked through the Transact logs, into Lasernet and back to Transact, to assist in diagnosing any issues.

When reporting issues to Formpipe Support, best practice is to have Debug Transact and Lasernet logs for the time of the error, to assist with diagnosing the issue.

Improving throughput
One of the most common issues with Print Preview is having higher throughput than can be delivered through the current Lasernet server. Although each Lasernet service can handle multiple Web Server Input ports (by default Lasernet starts with 2 (from Interfaces v4 and above)), the more inputs that are used, the higher the required resources are. As such, the process will eventually succumb to the law of diminishing returns, as the load on the rest of the system will eventually cause bottlenecks.

Whilst there are no set figures for the volume that can be handled by Print Preview, throughput issues should be relatively easy to identify by the number of failed jobs, combined with increased resource usage and higher loads, during peak times.

Therefore, it is important that the system is load-tested *before* going live with Print Preview. We recommend it is tested to at least 1.5x to 2x the expected peak throughput.

To try and mitigate the effects of high loads across multiple ports, Print Preview v4 (and above) handles ports in a 'round-robin' fashion, skipping one when it is inaccessible and moving on to the next. These can be listed in the Web.XML file.

An alternative solution to speed up throughput and even out load is to run multiple Lasernet servers, each using an optimum number of ports (to be determined through testing & resource monitoring). This will deliver higher overall throughput than running a single server with too many ports. However, this solution does require additional Lasernet licenses.

As well as improving the throughput using an optimised number of inputs, throughput can also be improving by speeding up how quickly a job is processed. New features in Lasernet v7 and above make it easier to build forms without needing to use overlays. Using the

Shapes, Tables and Regions tools you can create a form that will result in a faster and smaller output without any loss of style. In our tests, files that used these tools were one third of the file size and were three to four times faster to process.

To take advantage of the new features found in Lasernet, we provide Familiarisation Training – which includes designing "overlays" forms as mentioned above. To book a training session, please contact the Formpipe Sales Team. Training can be carried out on-site or remotely according to your schedule.

Timeouts and retries

There are a number of additional configuration options available in the Web.XML (under the list of Lasernet servers), in Print Preview Interface v4 (and above). These can be configured to tweak the performance of the Print Preview interface:

```
<init-param>
    <param-name>timeout</param-name>
>    <param-value>30</param-value>
</init-param>
<init-param>
    <param-name>retries</param-name>
    <param-value>3</param-value>
</init-param>
<init-param>
    <param-name>wait-time</param-name>
    <param-value>3</param-value>
</init-param>
```

- The Timeout parameter is the length of time Print Preview will try to connect to the Lasernet port before timing out.
- The Retries parameter is the number of times Print Preview will retry before failing.
- The Wait-Time parameter is the amount of time it waits between each retry.

Catch-All/Form Matching

Even if the connections between Transact and Lasernet are correctly established, there could still be issues with the data getting returned. Lasernet uses parts of the input data (configured in the form) to decide what form to process. If none of these match, then a generic "Catch-All" form is matched and returned to Transact (note: the behaviour and content of the "Catch-All" can be configured differently for each install).

If this generic PDF is returned, then the issue is within the Lasernet build. To solve this problem you first need to locate the example input file. This can be done by collecting the

autoformlasernet.txt file generated by Lasernet, or by turning on Grab mode (either via the Developer and uploading the build or directly through the Monitor) and reprocessing the Print Preview.

The input file can be then be opened in its matching Form and the criteria can be checked to determine why the match failed.