
Article

[Eduard Lebedyuk](#) · Jun 7, 2017 10m read

Debugging Web - Part 2

In the [first article](#) I discussed testing and debugging Caché web applications with external tools. Second part would cover Caché tools.


They are:

- CSP Gateway and Webapp configuration
- CSP Gateway logging
- CSP Gateway tracing
- ISCLOG
- Custom logging
- Session events
- Output to device

CSP Gateway and Webapp configuration

First of all, if you're debugging and especially developing front-end application you don't need caching. It's useful on a live system, but not during development. To disable logging for a web application you need to go to: SMP Menu → Manage Web Applications → <Your Web App> and set `Serve Files Timeout` setting equal to 0. Then press "Save" button.

Next, you need to purge existing web-application cache. To do that go to: SMP → System Administration Configuration → CSP Gateway Management → System Status. In there find "Cached Forms" table, last row is a Total line, press clear button (with the dot) there to clear web application cache:



[About CSP Gateway](#)

Management

[System Status](#)

[Test Server Connection](#)

[View Event Log](#)

[View HTTP Trace](#)

Configuration

[Default Parameters](#)

[Server Access](#)

[Application Access](#)

Caché Management

[Back to Management Portal](#)

[Logout](#)

Caché Server Pages

Web Gateway Management

System Status *(Gateway Initialized: Fri Jun 2 11:35:05 2017)*

Connection Number	Gateway PID	Server Name	Caché PID	Status	Idle time / Timeout	Activity	Interrupt	Close
0	6471	LOCAL	6617	Free	0/0	3934		•
1	6469	LOCAL	6620	Server	0/0	7066		•
2	6471	LOCAL	6621	Server	0/0	0		•
3	6469	LOCAL	6622	Free	0/0	3002		•
4	6469	LOCAL	6623	Free	0/0	1008		•
5	6471	LOCAL	6639	Free	0/0	866		•
6	6471	LOCAL	6662	Free	0/0	69		•
7	6469	LOCAL	6776	Free	0/0	37		•

Server Number	Server Name	Mirror Member	Mirror Status	Total Connections	Connections In-Use	Private Connections	Total Activity	Queued Requests	Close
0	LOCAL			8	0	0	8919	0	•
Total	•	•	•	8	0	0	8919	0	•

Path Number	Path	Server Number	Server Name	Activity	Status	Action
0	/	0	LOCAL	380	Enabled	
1	/csp	0	LOCAL	8539	Enabled	

Cached Forms	Cached Data (Bytes)	Cache Blocks In Use	Cache File	Cached Form Activity	Clear
/csp/wiseweb/deepsee/ds2_dashboard_22.png	1948	1	c6471_796.dat	0	•
/csp/wiseweb/deepsee/ds2_x_44.png	1405	1	c6469_797.dat	0	•
Total: 2	3353	2/251	•	0 8	•

CSP Gateway logging


While we are on a topic of CSP Gateway, it supports logging of incoming requests ([documentation](#)). In Default Parameters tab specify desired log level (for example v9a) and save changes. v9a (check other options are in docs) records all HTTP requests to http.log in the Gateway home directory. Here's a sample captured request:

```
GET /forms/form/info/Form.Test.Person HTTP/1.1
Host: localhost:57772
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:57772/formsui/index.html
Cookie: CSPSESSIONID=SP-57772-UP-forms-=001000000000yxioCLLb8bbc9SVXQJC5WMU831n2sENf40GeGa; CSPWSERVERID=144zwBTP
Dnt: 1
Connection: keep-alive
Cache-Control: max-age=0
<>
```

There are also options to record performance. Results can be written to file or viewed from View Event Log tab.

CSP Gateway tracing

And, finally, you can trace requests and responses in the View HTTP Trace tab of CSP Gateway. Enable tracing and requests would start getting captured immediately. Don't forget to switch it off after debugging. Here's a sample debugging session:


Caché Server Pages
 Web Gateway HTTP Trace Facility

Trace OFF Trace ON

[Jun 7 10:15:56 fe6](#)

[Jun 7 10:16:00 fe7](#)

[Jun 7 10:16:00 fe8](#)

[Jun 7 10:16:00 fe9](#)

[Jun 7 10:16:00 fea](#)

[Jun 7 10:16:00 feb](#)

[Jun 7 10:16:00 1322](#)

[Jun 7 10:16:00 1320](#)

[Jun 7 10:16:00 1321](#)

[Jun 7 10:16:00 fec](#)

[Jun 7 10:16:00 fed](#)

[Jun 7 10:16:00 fee](#)

[Jun 7 10:16:00 1323](#)

[Jun 7 10:16:00 1324](#)

[Jun 7 10:16:00 1325](#)

[Jun 7 10:16:00 fef](#)

[Jun 7 10:16:00 ff0](#)

[Jun 7 10:16:00 ff1](#)

[Jun 7 10:16:00 1326](#)

[Jun 7 10:16:00 1327](#)

[Jun 7 10:16:00 1328](#)

[Jun 7 10:16:01 ff2](#)

[Jun 7 10:16:01 ff3](#)

[Refresh](#) [Clear](#)

Date: Jun 7 10:16:01; Request ID: ff3; Session ID: BIQFY33Bue; Remote-Addr:

[Show Response](#)

```
GET /workflow/rest/tasks HTTP/1.1
Host:
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
Cookie: CSPSESSIONID-SP-57772-UP-workflow-rest=003000000000BIQFY33Bueyepvuk90MOig6Wf2gkKVMsxsMrZn; CSPSESSIONID-SP-57772-UP-workflow=003000000000RJZ57LWtcPgD5uyvHg5oQ4GLjgu3yvQg0AFsIP; FA3D69260D25=DEEPSEE%3A; CacheBrowserId=VWaneTnwK7KA95PVgpcMKg--; CSPSERVERID=hzZqGKSh; CSPSESSIONID-SP-57772-UP=000000010000Cj9RarukjbWjTm_uvldAapwjHUKeESeLSdu85m; CSPSMTOKEN=101658808521071974907911572786
Dnt: 1
Connection: keep-alive
```

[Show Request](#)

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Set-Cookie: CSPSESSIONID-SP-57772-UP-workflow-rest=003000000000BIQFY33Bueyepvuk90MOig6Wf2gkKVMsxsMrZn; path=/workflow/rest/; httpOnly;
Access-Control-Allow-Headers: Content-Type, Authorization, Accept-Language
Access-Control-Allow-Methods: GET, PUT, POST, DELETE, OPTIONS
Access-Control-Allow-Origin: *
Access-Control-Max-Age: 10000
Cache-Control: no-cache
Date: Wed, 07 Jun 2017 10:16:01 GMT
Expires: Thu, 29 Oct 1998 17:04:19 GMT
Pragma: no-cache
Set-Cookie: CSPSERVERID=hzZqGKSh; path=/; httpOnly;
Content-Length: 23

{
  "children": [
  ]
}
```

Note: use tracing if you're can know what's wrong and can easily reproduce the problem. Use logging for statistics gathering, performance profiling, etc.

Additionally most web servers provide logging and performance tracking tools.

ISCLOG

CSP Gateway is useful in determining network problems and tracking performance, but to log what's going on inside Caché other tools are required. One of the more versatile of such tools is ISCLOG. [Documentation](#).

It's a global which can store information about current request processing. To start logging execute:

```
set ^%ISCLOG = 2
```

And to end logging execute:

```
set ^%ISCLOG = 0
```

Here's what a sample request looks like:

```
^%ISCLOG=0
^%ISCLOG("Data")=24
^%ISCLOG("Data",1)=$lb(2,"CSPServer","Header from CSP Size:3744 CMD:h IdSource:3","4664","FORMS","2017-06-07 10:49:21.341","%SYS.cspServer2","","")
^%ISCLOG("Data",1,0)=""^h30000 "$c(14,0,0)"_A"
^%ISCLOG("Data",2)=$lb(2,"CSPServer",["UpdateURL] Looking up: //localhost/forms/form/info path found: //localhost/forms/ Appl= "$c(2,1,3,4)"_@"_$(3,4,1,2,1,9,1)"/forms/"_$(2,1,3,4,1,2,1,2,1,2,4,3,4,1,2,1,9,1,7,1)_" :%All"_$c(8,1)"/forms"_$c(7,1)"FORMS"_$c(2,1,2,1,3,4,1,2,1,2,1,3,4,1,2,1,4,4,132,3,3,4,2,3,4,2,2,1,4,4,16,14,2,4,3,4,1,3,4,1,2,1,3,4,1,2,1,16,1)"_Form.REST.Main"_$c(2,4,2,4),"4664","FORMS","2017-06-07 10:4
```

```

9:21.342", "%CSP.Request.1", "124", "L3DfNILTaE")
^%ISCLOG("Data",3)=$lb(2,"CSPServer","[UpdateURL] Found cls: Form.REST.Main nocharset
convert: charset=UTF-8 convert charset=UTF8", "4664", "FORMS", "2017-06-07 10:49:21.342
", "%CSP.Request.1", "124", "L3DfNILTaE")
^%ISCLOG("Data",4)=$lb(2,"CSPServer","[HTML] Determined request type", "4664", "FORMS",
"2017-06-07 10:49:21.342", "%SYS.cspServer2", "124", "L3DfNILTaE")
^%ISCLOG("Data",4,0)=$lb("usesession",1,"i%Class", "Form.REST.Main", "i%Service", "REST"
, "NOLOCKITEM", "", "i%GatewayError", "")
^%ISCLOG("Data",5)=$lb(2,"CSPSession","[%LoadData] Loading CSP session, nosave=0", "46
64", "FORMS", "2017-06-07 10:49:21.342", "%CSP.Session.1", "", "L3DfNILTaE")
^%ISCLOG("Data",5,0)=$lb(900,,0,5567742244,$c(149)_"Û"_$c(3)_"ó»à"_$c(127)_"½"_$c(14
9,10)_"\"_$c(18)_"v"_$c(128,135)_"3Vô"_$c(11)_"*"_$c(154)_"PİĞ¥"_$c(140,157,145,10,13
1)_"*",2,"FORMS", "001000000000L3DfNILTaElcDBJNjyQdyLwKq4wCXP82ld8gic", ,0,"ru", "L3DfNI
LTaE",2,1,"/forms/", $lb("UnknownUser", "%All,%Developer", "%All,%Developer", 64,-5590387
37), "", "", "", "2017-06-07 10:48:51", "2017-06-07 10:49:04", "Basic ZGV2OjEyMw==", "Mozill
a/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0", "", "", 0, "/forms/
", "", "", "", 4, "", "", "", "", "http://localhost:57772/formsui/index.html")
^%ISCLOG("Data",6)=$lb(2,"CSPServer","[CSPDispatch]Requested GET /forms/form/info", "4
664", "FORMS", "2017-06-07 10:49:21.342", "%SYS.cspServer", "124", "L3DfNILTaE")
^%ISCLOG("Data",7)=$lb(2,"CSPServer","[CSPDispatch] ** Start processing request newSe
s=0", "4664", "FORMS", "2017-06-07 10:49:21.342", "%SYS.cspServer", "124", "L3DfNILTaE")
^%ISCLOG("Data",7,0)="/forms/form/info"
^%ISCLOG("Data",8)=$lb(2,"CSPServer","[CSPDispatch] Service type is REST has-soapacti
on=0 nosave=0", "4664", "FORMS", "2017-06-07 10:49:21.342", "%SYS.cspServer", "124", "L3DfN
ILTaE")
^%ISCLOG("Data",9)=$lb(2,"CSPServer","[CSPDispatch]About to run page: Form.REST.Main"
, "4664", "FORMS", "2017-06-07 10:49:21.342", "%SYS.cspServer", "124", "L3DfNILTaE")
^%ISCLOG("Data",9,0)=$lb("UnknownUser", "%All,%Developer", "%All,%Developer", 64,-559038
737)
^%ISCLOG("Data",10)=$lb(2,"CSPServer","[callPage] url=/forms/form/info ; Appl: /forms
/ newsession=0", "4664", "FORMS", "2017-06-07 10:49:21.342", "%SYS.cspServer", "124", "L3Df
NILTaE")
^%ISCLOG("Data",11)=$lb(2,"CSPServer","[callPage]Imported security context ; User: Un
knownUser ; Roles: %All,%Developer", "4664", "FORMS", "2017-06-07 10:49:21.342", "%SYS.cs
pServer", "124", "L3DfNILTaE")
^%ISCLOG("Data",12)=$lb(2,"CSPServer","[OutputCSPGatewayData]: chd=1;", "4664", "FORMS"
, "2017-06-07 10:49:21.431", "%CSP.Response.1", "", "L3DfNILTaE")
^%ISCLOG("Data",13)=$lb(2,"CSPResponse","[WriteHTTPHeaderCookies] Session cookie: CSP
SESSIONID-SP-57772-UP-forms-=001000000000L3DfNILTaElcDBJNjyQdyLwKq4wCXP82ld8gic; path
=/forms/; httpOnly;", "4664", "FORMS", "2017-06-07 10:49:21.431", "%CSP.Response.1", "124
", "L3DfNILTaE")
^%ISCLOG("Data",14)=$lb(2,"CSPServer","[callPage] Return Status", "4664", "FORMS", "2017
-06-07 10:49:21.431", "%SYS.cspServer", "124", "L3DfNILTaE")
^%ISCLOG("Data",14,0)=1
^%ISCLOG("Data",15)=$lb(2,"CSPServer","[OutputCSPGatewayData]: chd=1;", "4664", "FORMS"
, "2017-06-07 10:49:21.431", "%CSP.Response.1", "", "L3DfNILTaE")
^%ISCLOG("Data",16)=$lb(2,"CSPServer","[Cleanup]Page EndSession=0; needToGetALicense=
-1; nosave=0; loginredirect=0; sessionContext=1", "4664", "FORMS", "2017-06-07 10:49:21.
431", "%SYS.cspServer", "124", "L3DfNILTaE")
^%ISCLOG("Data",17)=$lb(2,"CSPSession","[Cleanup] EndSession=0 nosave=0", "4664", "FORM
S", "2017-06-07 10:49:21.431", "%SYS.cspServer", "124", "L3DfNILTaE")
^%ISCLOG("Data",18)=$lb(2,"CSPSession","[%SaveData] Saved: ", "4664", "FORMS", "2017-06-
07 10:49:21.431", "%CSP.Session.1", "", "L3DfNILTaE")
^%ISCLOG("Data",18,0)=$lb(900,,0,5567742261,$c(149)_"Û"_$c(3)_"ó»à"_$c(127)_"½"_$c(1
49,10)_"\"_$c(18)_"v"_$c(128,135)_"3Vô"_$c(11)_"*"_$c(154)_"PİĞ¥"_$c(140,157,145,10,1
31)_"*",2,"FORMS", "001000000000L3DfNILTaElcDBJNjyQdyLwKq4wCXP82ld8gic", ,0,"ru", "L3DfN
ILTaE",2,1,"/forms/", $lb("UnknownUser", "%All,%Developer", "%All,%Developer", 64,-559038
737), "", "", "", "2017-06-07 10:48:51", "2017-06-07 10:49:21", "Basic ZGV2OjEyMw==", "Mozil
la/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0", "", "", 0, "/forms

```

```

/","","","5","","","","","http://localhost:57772/formsui/index.html")
^%ISCLOG("Data",19)=$lb(2,"CSPServer","[Cleanup] Restoring roles before running destr
uctor","4664","FORMS","2017-06-07 10:49:21.431","%SYS.cspServer","","L3DfNILTaE")
^%ISCLOG("Data",19,0)=$lb("UnknownUser","%All,%Developer","%All,%Developer",64,-55903
8737)
^%ISCLOG("Data",20)=$lb(2,"CSPServer","[Cleanup] End","4664","FORMS","2017-06-07 10:4
9:21.431","%SYS.cspServer","","L3DfNILTaE")
^%ISCLOG("Data",20,0)="-Finish processing request->"
^%ISCLOG("Data",21)=$lb(2,"GatewayRequest","[CSPGWClientRequest] GWID: ed-pc:57772; R
equest: sys_get_system_metricsTimeout: 5","11112","%SYS","2017-06-07 10:49:23.141","%
SYS.cspServer3","","")
^%ISCLOG("Data",22)=$lb(2,"GatewayRequest","[CSPGWClientRequest] GWID: 127.0.0.1:5777
2; Request: sys_get_system_metricsTimeout: 5","11112","%SYS","2017-06-07 10:49:23.141
","%SYS.cspServer3","","")
^%ISCLOG("Data",23)=$lb(2,"GatewayRequest","[SendSimpleCmd:Server:Failed] WebServer:
127.0.0.1:57772; Gateway Server Request Failed","11112","%SYS","2017-06-07 10:49:23.1
41","%CSP.Mgr.GatewayMgrImpl.1","","")
^%ISCLOG("Data",23,0)=0
^%ISCLOG("Data",24)=$lb(2,"GatewayRequest","[GetMetrics]","11112","%SYS","2017-06-07
10:49:23.141","%CSP.Mgr.GatewayMgrImpl.1","","")
^%ISCLOG("Data",24,0)="-End Request Client->"

```

Also here's a quick script to output global to file:

```

set p="c:\temp\isclog.txt"
open p:"NW"
use p zw ^%ISCLOG
close p

```

Custom logging

While default logging tools are quite good, but they have several problems:

- They are generic and don't know about your application
- More verbose options affect performance
- They are unstructured, so extracting information can be difficult

Therefore, more specific cases can be covered by writing your own custom logging systems. Here's a sample of a persistent class that records part of %request object:

```

/// Incoming request
Class Log.Request Extends %Persistent
{
/// A string indicating HTTP method used for this request.
Property method As %String;
/// A string containing the URL up to and including the page name
/// and extension, but not including the query string.
Property url As %String(MAXLEN = "");
/// A string indicating the type of browser from which the request
/// originated, as determined from the HTTP_USER_AGENT header.
Property userAgent As %String(MAXLEN = "");
/// A string indicating the MIME Content-Type of the request.
Property contentType As %String(MAXLEN = "");

```

```

/// Character set this request was send in, if not specified in the HTTP headers
/// it defaults to the character set of the page it is being submitted to.
Property charSet As %String(MAXLEN = "");
/// A <class>%CSP.Stream</class> containing the content submitted
/// with this request.
Property content As %Stream.GlobalBinary;
/// True if the communication between the browser and the web server was using
/// the secure https protocol. False for a normal http connection.
Property secure As %Boolean;
Property cgiEnvs As array Of %String(MAXLEN = "", SQLPROJECTION = "table/column");
Property data As array Of %String(MAXLEN = "", SQLPROJECTION = "table/column");
ClassMethod add() As %Status
{
    set request = ..%New()
    quit request.%Save()
}
Method %OnNew() As %Status [ Private, ServerOnly = 1 ]
{
    #dim %request As %CSP.Request
    #dim sc As %Status = $$$OK
    quit: '$isObject($g(%request)) $$$ERROR($$$$GeneralError, "Not a web context")
    set ..charSet = %request.CharSet
    if $isObject(%request.Content) {
        do ..content.CopyFromAndSave(%request.Content)
    } else {
        set ..content = ""
    }
    set ..contentType = %request.ContentType
    set ..method = %request.Method
    set ..secure = %request.Secure
    set ..url = %request.URL
    set ..userAgent = %request.UserAgent
    set cgi = ""
    for {
        set cgi=$order(%request.CgiEnvs(cgi))
        quit:cgi=""
        do ..cgiEnvs.SetAt(%request.CgiEnvs(cgi), cgi)
    }
    // Only gets first data if more than one data with the same name is present
    set data = ""
    for {
        set data=$order(%request.Data(data))
        quit:data=""
        do ..data.SetAt(%request.Get(data), data)
    }
    quit sc
}
}

```

To add a new record to Log.Request table, add a call in your code:

```
do ##class(Log.Request).add()
```

This is a very basic sample, it can and should be extended to record comments, variables or anything else you might need. The main advantage of this approach is an ability to execute SQL queries over recorded data. For more info on building your own logging system please refer to [this article](#).

Session events

Event class is class which defines the interfaces that are called during the lifetime of a [%CSP.Session](#) object. In order to use this you should subclass [%CSP.SessionEvents](#) and implement the method code you wish to be executed. Then within the CSP application configuration set the event class to the class you have created.

The following callbacks are available:

- OnApplicationChange
- OnEndRequest
- OnEndSession
- OnLogin
- OnLogout
- OnStartRequest
- OnStartSession
- OnTimeout

For example custom logging discussed above can be called from these methods.

Output to device

One of the most simple options - a CSP utility method which outputs all objects as a response. Just add this to any part of your code:

```
set %response.ContentType = "html"  
do ##class(%CSP.Utils).DisplayAllObjects()  
return $$$OK
```

Conclusion

There are many tools you can use to debug your web application. Choose the best one for the task at hand.

What are your tips and tricks to debugging web applications from Caché?

[#Best Practices](#) [#Debugging](#) [#Frontend](#) [#Caché](#)

Source URL: <https://community.intersystems.com/post/debugging-web-part-2>