



# DPTOOL INSTALLATION AND USER GUIDE

V1.0

## Abstract

dpTool Utility is an easy to use DataPower configuration query tool that can be used by Administrators, Tech Leads and Developers in querying the configuration across devices

DEEPAK RAMABHAT  
deepaklr@in.ibm.com

## Contents

1.	About the Tool .....	2
2.	Installation Steps.....	4
3.	Configuration .....	9
4.	Use Cases .....	11
a)	Query Configuration .....	11
➤	Use Case-1: Compare System Settings of all DEV devices .....	12
➤	Use Case-2: List all MQ objects to find which domain connects to a specific Queue Manager .....	13
➤	Use Case-3: List all services to find a service with matching URI or Back-end IP: Port .....	13
➤	Use Case-4: List all Library Versions from device features .....	14
➤	Use Case-5: List all services where Debug Probe is enabled .....	15
b)	Validate Domain .....	16
c)	Search Domain.....	17
➤	Use Case-1: Search for all Unused Objects.....	17
➤	Use Case-2: Search for all Unused XSL files and Missing XSL files.....	18
➤	Use Case-3: Search for all User IDs used in a domain .....	18
➤	Use Case-4: Search for all WTX map files in a domain.....	19
➤	Use Case-5: Search for a file named 'log', used for logging.....	19
➤	Use Case-6: Search for 'uuid' function usage in all XSL files in a domain .....	19
➤	Use Case-7: Search for a field named ServiceTypeCd to see where all the field is mapped .....	20
d)	Query Certificate.....	21
e)	Host/IP Lookup .....	21
f)	Compare Configuration .....	22
g)	Compare Files .....	23
5.	Notes.....	24

## 1. About the Tool

dpTool is a service that is deployable to an existing DataPower device and can be configured to connect to multiple DataPower devices so users can access the device configurations using Browser and easy to view HTML tables. This will save Administrators from logging in to multiple Admin consoles and collate data from each device. This can also be used to search for files, search within files which otherwise could only be done by downloading the domain export.

dpTool displays data in a HTML table which can be copied and pasted to an excel sheet for easier comparisons between environments without logging in to each device individually. For example, comparing MQ parameters between multiple environments to see if connection parameters are correctly set would be much easier using this tool.

**Target users:** DataPower Administrators, Technical Leads, Architects and Developers

This Utility would be useful in following ways:

- Admins can extract and compare configurations from multiple devices
- Leads can extract configurations and check for standards and parameters
- Search for Services/Objects/Files or Code in a domain or on multiple domains
- Compare configurations for a domain on different devices without exporting the domain
- Extract list of services and URLs to create reference documentation for support
- Find a service with a specific URI/URL
- Find unused objects
- Search all XSL files for a specific code or a field
- View statistics of different services/domains

### Screenshot of the Tool

Extract Configuration for a domain or all domains in an environment  
For ex: List all MQ Queue Managers

Run predefined coding standard checks against a domain on a device

Search domains for a text or a file or name of a file or unused Objects  
For ex: find a field mapping by its name in all XSLs

List all digital certificates in any domain with its properties

Lookup IP or hostname. It searches against multiple hosts files / Aliases and DNS server ping

Compare configuration for a domain against different devices to know the code gap

Compare all files within a domain against different devices to know the code gap

**dpTool**

**Query Configuration**

Device\*:   
Type\*:   
Object\*:   
Domain\*:

**Validate Domain**

Device\*:   
Domain\*:

**Search Domain**

Device\*:   
Domain\*:   
Type\*:   
Search String:

**Query Certificate**

Device\*:   
Domain:

**Hostname Lookup**

Host or IP\*:

**Compare Config**

Device 1\*:   
Domain 1\*:   
Device 2\*:   
Domain 2\*:

**Compare Files**

Device 1\*:   
Domain 1\*:   
Device 2\*:   
Domain 2\*:

**Test Service**    **Interfaces**

**Request**

URL:

**Request Header Header Value**

SOAPAction:  Example: http://url/op  
Content-Type:  Example: text/xml; charset=UTF-8; action='http://url/op'  
MQMP:  Example: <MQMP> <Property name='usr.apphdr.xx'> </Property> </MQMP>  
SSLProfile:  Example: dpTool-trust-all-ssl  
Name:  Value:

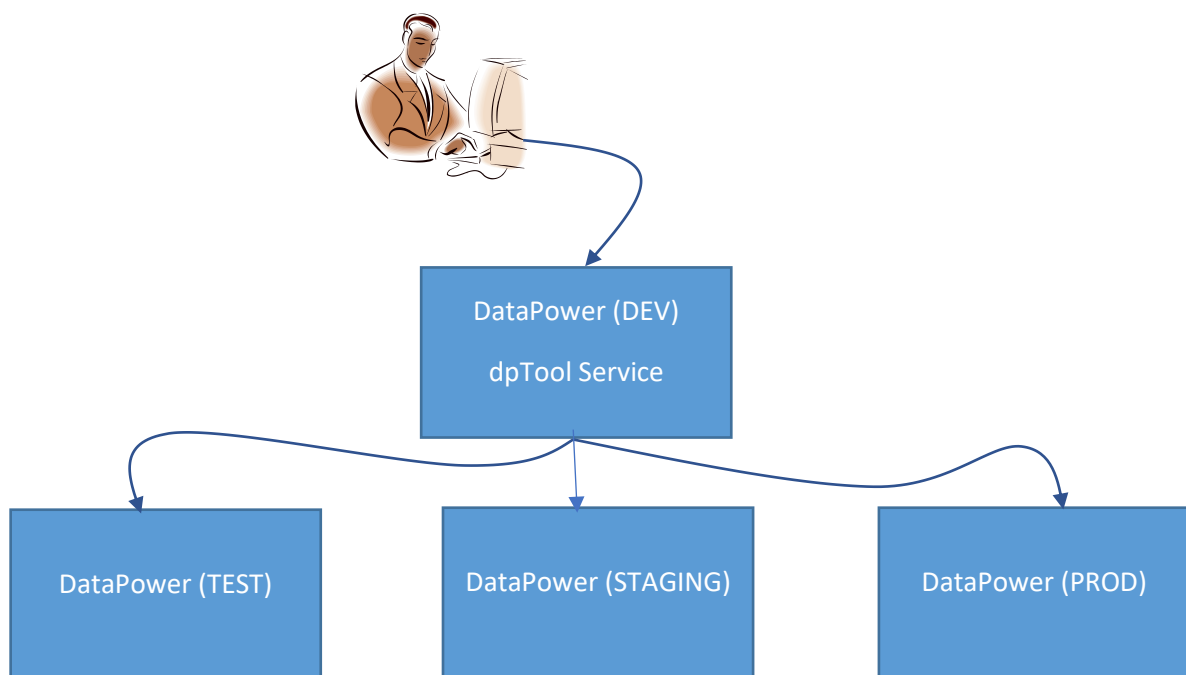
**Message**

**Response**

Use to test any service HTTP/MQ/FTP from within the DP environment. Helpful when services are not accessible outside DP.

Version: 1.0    Author: deepak1r@in.ibm.com    Tool Customized from IBM Repository

Simple topology diagram would be as below



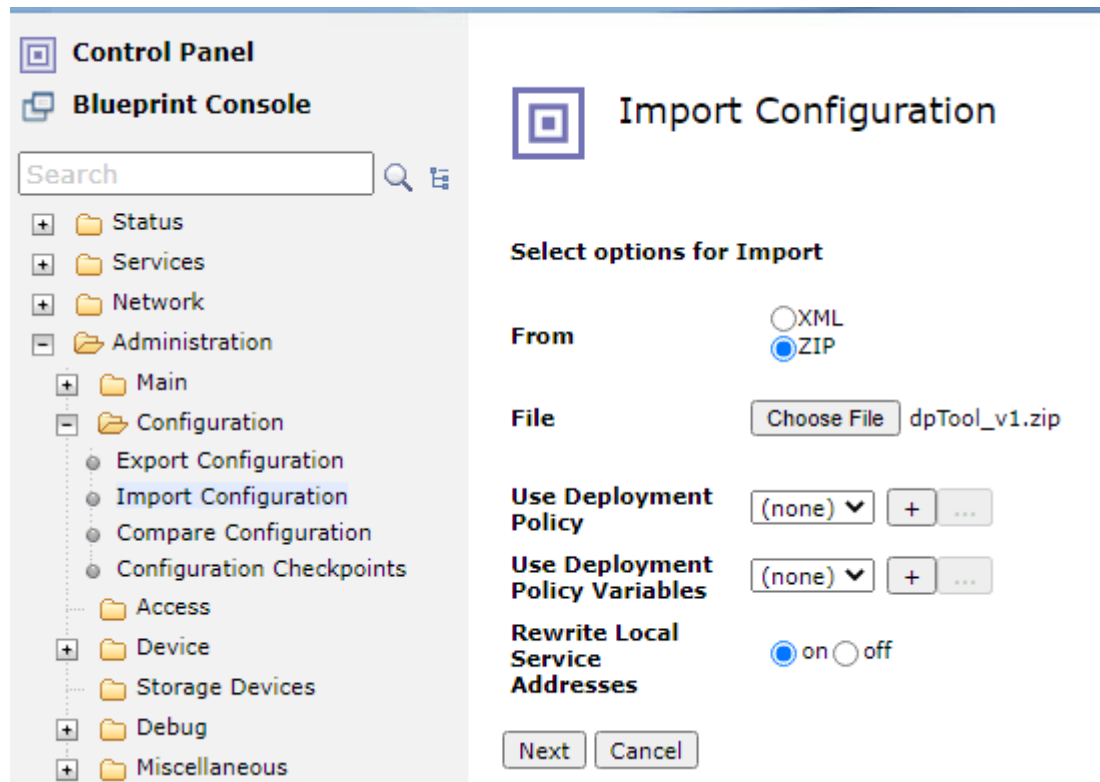
## 2. Installation Steps

Once you have downloaded the package follow below steps to install and configure the tool.

Make sure you have dpTool\_v1.zip file.

Login to the admin console of desired domain where you want to install the dpTool service, preferably DEV/TEST environment.

Select Import Configuration and choose dpTool\_v1.zip from local disk.



Proceed to Next Step and keep all default values and Click Import.



## Import Configuration

[Help](#)

### Object Import Results

- ✓ Crypto Key:nasesbdp
- ✓ Crypto Certificate:nasesbdp
- ✓ Crypto Identification Credentials:dpTool-ssl-id
- ✓ SSL Server Profile:dpTool-ssl-server
- ✓ HTTPS Handler:dpTool-https-fsh
- ✓ HTTPS Handler:dpTool-https-fsh2
- ✓ XML Manager:dpTool-xml-manager
- ✓ SSL Client Profile:dpTool-trust-all-ssl
- ✓ Matching Rule:dpTool-index
- ✓ Processing Action:dpTool-mpg-request-rule-index\_xform\_0

### File Import Results

- local:///dpTool/code/dpTool.xml: OK
- local:///dpTool/code/dptest-run.xml: OK
- local:///dpTool/code/collectInfo.xml: OK
- local:///dpTool/code/formatInfo.xml: OK
- local:///dpTool/code/validate.xml: OK
- local:///dpTool/code/searchDomain.xml: OK
- local:///dpTool/code/certInfo.xml: OK
- local:///dpTool/code/certToHTML.xml: OK
- local:///dpTool/code/tcplookup.xml: OK
- local:///dpTool/code/compareConfig.xml: OK
- local:///dpTool/code/compareFiles.xml: OK

Done

Make sure no errors while importing.

Once service is imported, go to Multi-Protocol Gateway list and Edit “dpTool” service.

Edit the HTTPS Handler in the list – dpTool-https-fsh



## Configure Multi-Protocol Gateway

General Advanced Subscriptions Policy SLA Policy Details Stylesheet Params Headers Monitors WS-Addr

Apply Cancel Delete

[Export](#) | [View Log](#) | [View Status](#) | [Show Probe](#) | [Validate Conformance](#) | [Help](#)

Multi-Protocol Gateway status: [down - Failed to install on port]

## General Configuration

## Multi-Protocol Gateway Name

dpTool \*

## Summary

Tool to query datapower configurati

## Type

☒ dynamic-backends  
☐ static-backend \*

## XML Manager

dpTool-xml-manager + ... \*

## Multi-Protocol Gateway Policy

dpTool-mpg-policy + ... \*

## URL Rewrite Policy

(none) + ...

## Back side settings

With a dynamic proxy back end type, the back end server address and port are determined by a stylesheet in a policy action.

## Front side settings

## Front Side Protocol

dpTool-https-fsh (HTTPS Handler) [down]	✕
dpTool-https-fsh2 (HTTPS Handler) [down]	✕
<input type="text"/>	+ ... Add

\*

Select appropriate local interface where you want to expose the service. Change the port if required.

You can also choose to create plain HTTP Handler instead of HTTPS – even though not preferred.

HTTPS Handler: dpTool-https-fsh [down - Required referenced object not up]

Apply

Cancel

Undo

Administrative state

☒ enabled ☐ disabled

Comments

Local IP address

ExternalDEV

Select Alias \*

Port

8008 \*

Allowed methods and versions

- ☒ HTTP 1.0
- ☒ HTTP 1.1
- ☐ HTTP/2
- ☒ POST method
- ☒ GET method
- ☐ PUT method
- ☐ PATCH method
- ☐ HEAD method
- ☐ OPTIONS
- ☐ TRACE method
- ☐ DELETE method
- ☐ Custom methods
- ☒ URL with ?
- ☐ URL with #
- ☐ URL with ..
- ☐ URL with cmd.exe

Allow WebSocket upgrade

☐ on ☒ off

SSL server type

Server Profile ▼

SSL server profile

dpTool-ssl-server ▼

[down]

+

...

\*

Edit the SSL Server Profile named – dpTool-ssl-server.

On the next page, Edit the Identification Credentials – dpTool-ssl-id

## Credentials

Identification credentials

dpTool-ssl-id ▼

[down]

+

...



On next page, Edit the Crypto Key and select a local private key available. If not available use Crypto Tools to create a self-signed key-cert pair on the local device and select the newly created key.

Similarly create the local public certificate for the above matching key for the certificate.

Administrative state	<input checked="" type="radio"/> enabled <input type="radio"/> disabled
Crypto Key	<div>nasesbdp ▾ [down] + ... *</div>
Certificate	<div>nasesbdp ▾ + ... *</div>
Intermediate CA Certificate	<div>(empty)</div> <div><div>▾</div> add + ...</div>

Save all changes on all the pages.

Save configuration changes to the domain.

Verify that dpTool service has no errors and in UP status.

### 3. Configuration

Tool requires two configuration files to be updated

1. local://dptool/config/dpTool-config.xml [REQUIRED]
2. local://dptool/config/hosts.xml [OPTIONAL]

Edit local://dptool/config/dpTool-config.xml file and add all the devices in your environment.

```
<!--
    Source: dpTool from IBM code repository, Copyrights may apply

    Contains configuration for dpTool
    Add devices in each environment along with ENCRYPTED AUTH information and XMA/XMI URLs for each device.
    If AUTH information is empty then users will be asked to enter their own credentials to login to datapower
    Use supplied powershell script dpTool.auth.ps1 to create the encrypted string from the plain text user id and password

    Change History:

-->
<dpTool>
    <!-- Add devices in each environment, it can be just one or more. Each device needs to have XMI and XMA URL defined as attribute
        If users have different user id and password then you will need to define one user id and password that can be used to query the devices,
        as users can not enter multiple ID/Password for each device.
        auth takes base-64 encoded string similar to HTTPS BASIC Authentication.
        For example: auth="Basic Sk9LRVI6Sk9LRVI=" -->
    <environment name="DEV">
        <device name="ESB" auth="" XMI="https://9.12.125.253:5550/service/mgmt/current" XMA="https://9.12.125.253:5550/service/mgmt/amp/1.0" />
        <device name="GW" auth="" XMI="https://9.12.125.239:5550/service/mgmt/current" XMA="https://9.12.125.239:5550/service/mgmt/amp/1.0" />
    </environment>

    <environment name="INT">
        <device name="ESB" auth="" XMI="https://9.12.125.254:5550/service/mgmt/current" XMA="https://9.12.125.254:5550/service/mgmt/amp/1.0" />
        <device name="GW" auth="" XMI="https://9.12.125.238:5550/service/mgmt/current" XMA="https://9.12.125.238:5550/service/mgmt/amp/1.0" />
    </environment>

    <environment name="PROD">
        <device name="ESB1" auth="" XMI="https://9.12.126.253:5550/service/mgmt/current" XMA="https://9.12.126.253:5550/service/mgmt/amp/1.0" />
        <device name="GW1" auth="" XMI="https://9.12.126.252:5550/service/mgmt/current" XMA="https://9.12.126.252:5550/service/mgmt/amp/1.0" />
        <device name="ESB2" auth="" XMI="https://9.12.126.254:5550/service/mgmt/current" XMA="https://9.12.126.254:5550/service/mgmt/amp/1.0" />
        <device name="GW2" auth="" XMI="https://9.12.126.251:5550/service/mgmt/current" XMA="https://9.12.126.251:5550/service/mgmt/amp/1.0" />
    </environment>

    <!-- static domains list for drop down selection : following sample domains are pre-listed -->
    <domains>
        <option value="Claim"/>
        <option value="Member"/>
        <option value="Provider"/>
        <option value="Utility"/>
        <option value="default"/>
    </domains>
</dpTool>
```

Whenever user tries to use dpTool, they will be asked for their user id and password to login to the devices. If a single environment has multiple devices, then tool expects single user id and password will work on all the device. If this condition is not true in your environment, then you can use a hard-coded authentication for each of the device.

For each device there is an attribute named *auth*, set this value with a pre-defined userid/password in an encoded format. For this use the supplied dpTool-auth.ps1 file to convert plain text user id, password to required base-64 encoded format.

Edit PowerShell script dpTool-auth.ps1 file and add your ID and Password in the first two fields.

Save and Click Run – which will print the Encoded value and use this to define the attribute as below:

*auth="Basic Sk9LRVI6Sk9LRVI="*

```
dpTool-auth.ps1 X
1 #Enter you user id and password
2 $userid="JOKER"
3 $password="JOKER"
4
5 $bytes = [System.Text.Encoding]::ASCII.GetBytes($userid + ":" + $password)
6 $base64 = [System.Convert]::ToBase64String($bytes)
7 $basicAuthValue = "Basic $base64"
8 Write-Output $basicAuthValue
9
PS C:\Users\DEEPAKRABHAT\Desktop> C:\Users\DEEPAKRABHAT\Desktop\dpTool-auth.ps1
Basic Sk9LRVI6Sk9LRVI=
```

Similarly add the individual authentication information for each device.

*Note: If you want the user id and password not be copied by others on the domain then restrict the domain access policy so others can not read this configuration file.*

Once all environments are added, then add an initial set of Domains in the same file. This will be used for the dropdowns, but user will still be able to key-in any new domains defined. This list is only to have a pre-populated domain names in the dropdowns.

---

Edit local://dptool/config/hosts.xml file if you have lot of the servers in the environment and user often requires translating IP to host or host to IP. You can even add load balancer IPs to assist developers. Add each host and respective IP like hosts file on windows but in the XML format.

```
<!-- add hosts and IPs similar to hosts file -->
<hosts>
  <host ip="127.0.0.1" name="localhost">Local server IP</host>
  <host ip="xx.xxx.xxx.xxx" name="xxx.net">#NAT: xx.xxx.xxx.xxx</host>
</hosts>
```

---

Now you can test the tool by accessing the tool from your browser.

Type in the address of the service like <https://yourIP:8008/dpTool/>

This should bring up the dpTool page if all is well.

## 4. Use Cases

dpTool contains multiple functions. Summary of each function and some of the use cases are given below.

### a) Query Configuration

This function is useful in querying one or more devices configured in an environment. You can either query on the XML configuration objects or the device status like webservice metrics, connection status etc.

The image shows a web form titled "Query Configuration". It has four input fields: "Device\*" (a text box), "Type\*" (a dropdown menu), "Object\*" (a dropdown menu), and "Domain" (a text box). Below these fields is a "Query" button.

Multiple devices can be present in a given environment and device drop down groups' devices by environment.

The image shows the "Query Configuration" form with the "Device\*" dropdown menu open. The dropdown list contains the following options: DEV-ALL, DEV-ESB, DEV-GW, INT-ALL, INT-ESB, and INT-GW. The "Type\*" dropdown is also visible but not open.

Type can be Configuration, Status or Debug.

Select configuration to query the XML configuration objects.

Select Status to see the device/domain/service/object status.

Select debug to list all debug probes enabled.

The image shows the "Query Configuration" form with the "Type\*" dropdown menu open. The dropdown list contains the following options: Configuration, Status, and debug. The "Object\*" dropdown is also visible but not open.

Object will list the type of Objects that can be queried. This list depends on the Type selected above.

The image shows two side-by-side screenshots of the 'Query Configuration' window. Both windows have a header bar with the title 'Query Configuration'.  
 The left window has the following fields: 'Device\*' (empty), 'Type\*' set to 'Configuration', 'Object\*' set to 'AccessControlList', and 'Domain' (empty). Below these fields is a scrollable list of objects: AccessControlList, AppSecurityPolicy, CompactFlash, CompileOptionsPolicy, ConfigDeploymentPolicy, ConformancePolicy, CertMonitor, CRLFetch, CryptoCertificate, CryptoFWCred, CryptoIdentCred, CryptoKerberosKDC, CryptoKerberosKeytab, CryptoKey, CryptoProfile, CryptoSSKey, CryptoValCred, OAuthSupportedClient, OAuthSupportedClientGroup, and SSHClientProfile. There are also sections for 'Validate Domain' and 'Search Domain' with their respective fields.  
 The right window has the following fields: 'Device\*' (empty), 'Type\*' set to 'Status', 'Object\*' set to 'ActiveUsers', and 'Domain' (empty). Below these fields is a scrollable list of objects: ActiveUsers, ARPStatus, Battery, ChangeGroupRetryQueue, ChangeGroups, ConnectionsAccepted, CPUUsage, CryptoEngineStatus, CryptoHwDisableStatus, CryptoModeStatus, CurrentSensors, DateTimeStatus, DNSCacheHostStatus, DNSCacheHostStatus2, DNSNameServerStatus, DNSNameServerStatus2, DNSSearchDomainStatus, DNSStaticHostStatus, DocumentCachingSummary, and DocumentStatus. Similar to the left window, it has 'Validate Domain' and 'Search Domain' sections.

Domain is an optional field, Type in or select a single domain if you want to query only a single domain. If not selected, tool will collect data from all domains in one shot.

Example use cases to begin with the tool.

➤ Use Case-1: Compare System Settings of all DEV devices

This screenshot shows the 'Query Configuration' window with the following settings: 'Device\*' is 'DEV-ALL', 'Type\*' is 'Configuration', 'Object\*' is 'TimeSettings', and 'Domain' is 'default'. A 'Query' button is located at the bottom of the window.

Select DEV-ALL to query multiple devices.

Select domain as default for any device level info.

This screenshot shows a 'Sign in' dialog box. It contains the URL 'https://10.201.195.253:8008', a 'Username' input field, a 'Password' input field, and two buttons: 'Sign in' and 'Cancel'.

On the authentication prompt, type in your DataPower credentials.

Result will display in a Table where values can be compared easily.

Device	Domain	Name	mAdminState	UserSummary	ProductOID	Description	SerialNumber	EntitlementNumber	ProductID	Cc
ESB	default	System-Settings	enabled	NUESBDEV01 (VMWare)	1.3.6.1.4.1.14685.1.8	IBM DataPower Gateway	0000000	0000000	5725 [Rev None]	IG
Device	Domain	Name	mAdminState	UserSummary	ProductOID	Description	SerialNumber	EntitlementNumber	ProductID	Cc
GW	default	System-Settings	enabled	NUGWDEV01 (VMWare)	1.3.6.1.4.1.14685.1.8	IBM DataPower Gateway	0000000	0000000	5725 [Rev None]	IG

- Use Case-2: List all MQ objects to find which domain connects to a specific Queue Manager

**Query Configuration**

Device\* DEV-ALL

Type\* Configuration ▼

Object\* MQQM ▼

Domain

Query

All MQ Objects with properties will be displayed in a table and can be easily searched. You can also export it to excel for all future reference or to be shared with support team.

Device	Domain	Name	Connection	QMGR Name	Channel	Max Msg Size	AutomaticBackout	BackoutQueueName	Total Connections Limit	InitialConnections	State
GW	WP	REG_GW_BKND_QM1	mqdev01(1421)	MQDEV1D08F	PSVRCONN	20971520	off		4	2	enabled
GW	WP	REG_GW_BKND_QM2	mqdev01(1435)	MQDEV1D22F	PSVRCONN	20971520	off		4	2	enabled
GW	WP	REG_GW_FSH_QM1	mqdev01(1421)	MQDEV1D08F	PSVRCONN	20971520	off		4	2	enabled
GW	WP	REG_GW_FSH_QM2	mqdev01(1435)	MQDEV1D22F	PSVRCONN	20971520	off		4	1	enabled
GW	C	P7_BKND_QM1	mqdev01(1426)	MQDEV1D13F	SVRCONN	1048576	on	MS.REQUEST.Q001.BACKQ	4	2	enabled
GW	C	P7_BKND_QM2	mqdev01(1427)	MQDEV1D14F	SVRCONN	1048576	on	MS.REQUEST.Q001.BACKQ	4	2	enabled
GW	C	P7_FSH_QM1	mqdev01(1426)	MQDEV1D13F	SVRCONN	1048576	on	MS.REQUEST.Q001.BACKQ	4	2	enabled
GW	C	P7_FSH_QM2	mqdev01(1427)	MQDEV1D14F	SVRCONN	1048576	on	MS.REQUEST.Q001.BACKQ	4	2	enabled

- Use Case-3: List all services to find a service with matching URI or Back-end IP: Port

**Query Configuration**

Device\* DEV-GW

Type\* Status ▼

Object\* WSOperationsStatus ▼

Domain

Query

This will list all Webservice proxies with front-end and back-end URLs and one can export/search the list to see which domain contains specific service and the name of the service proxy.

## ➤ Use Case-4: List all Library Versions from device features

Query Configuration	
Device*	DEV-ALL
Type*	Status ▼
Object*	LibraryVersion ▼
Domain	default
Query	

Device	Domain	Name	Library	Version
ESB	default		Access Manager	all
ESB	default		Coproc	1.2
ESB	default		DCO	5.14
ESB	default		DCO Oracle	1.0
ESB	default		JMS	2.0.3
ESB	default		MQ	8.0.0.0
ESB	default		ODBC	7.16
ESB	default		ODR	ODRLIBX.ODRLIB_a1929.01
Device	Domain	Name	Library	Version
GW	default		Access Manager	all
GW	default		Coproc	1.2
GW	default		DCO	5.14
GW	default		DCO Oracle	1.0
GW	default		JMS	2.0.3
GW	default		MQ	8.0.0.0
GW	default		ODBC	7.16
GW	default		ODR	ODRLIBX.ODRLIB_a1929.01

- Use Case-5: List all services where Debug Probe is enabled

Query Configuration	
Device*	DEV-ALL
Type*	debug ▼
Object*	Debug Probes ▼
Domain	
Query	

Device	Domain	Name	type	probe
ESB	esce	esce_Inbound_Event_Processor_v1	MultiProtocolGateway	on
ESB	esce	esce_Outbound_Event_Processor_v1	MultiProtocolGateway	on
ESB	esce	esce_Outbound_Event_Router_v1	MultiProtocolGateway	on
ESB	esce	esceMembers_v1	MultiProtocolGateway	on
ESB	erServicing-API	MemberAdvisories_v1	MultiProtocolGateway	on
ESB	erServicing-HMHS	API_CSGroupInquiry_v1	MultiProtocolGateway	on
ESB	erServicing-HMHS	SOAP_CSCustomerInteractionInquiry_v3	MultiProtocolGateway	on
ESB	erServicing-HMHS	SOAP_CSMemberInquiry_v4	MultiProtocolGateway	on
ESB	External	ing_Rest_MPG	MultiProtocolGateway	on
ESB	MLE	aimAdjudication_WSP	WSGateway	on
ESB	MLE	aimPreparation_WSP	WSGateway	on
Device	Domain	Name	type	probe
GW	ExternalGateway	External_Outbound_API_GW	MultiProtocolGateway	on
GW	ExternalGateway	External_Outbound_API_GW_OLD	MultiProtocolGateway	on



## b) Validate Domain

This function can be used to validate a domain against few coding style/standards. This can be run every time a change goes in, to make sure properties are correctly set. One can add more checks in the XSL code as per the needs.

Validate Domain	
Device*	DEV-ESB
Domain*	Generic
Validate	

Generic	DEV-ESB:https://[redacted]:5550/service/mgmt/current
WSP	2
[redacted]MediationInt	<ul style="list-style-type: none"> <li>✓ XML Manager [Generic_WSP_XML_MNGR]</li> <li>✓ FrontTimeout [15]</li> <li>✓ BackTimeout [14]</li> <li>✓ backend URL [WAS_F5_VIP]</li> <li>✗ DO NOT validate Request [on], validation is turned off</li> <li>✗ DO NOT validate Response [on], validation is turned off</li> </ul>
[redacted]efitInformationService	<ul style="list-style-type: none"> <li>✓ XML Manager [Generic_WSP_XML_MNGR]</li> <li>✓ FrontTimeout [13]</li> <li>✓ BackTimeout [13]</li> <li>✓ backend URL [NAS1]</li> <li>✓ Request Validation disable [off]</li> <li>✗ DO NOT validate Response [on], validation is turned off</li> </ul>
MPG	2
AuditLogging_[redacted]MPG	<ul style="list-style-type: none"> <li>✓ XML Manager [Generic_MPG_XML_MNGR]</li> <li>✓ FrontTimeout [8]</li> <li>✓ BackTimeout [8]</li> <li>① Transaction Timeout [0] is defaulted to zero</li> </ul>
[redacted]AuditLog_MPG	<ul style="list-style-type: none"> <li>✓ XML Manager [Generic_MPG_XML_MNGR]</li> <li>① FrontTimeout [120] may be too large</li> <li>✓ BackTimeout [5]</li> <li>① Transaction Timeout [0] is defaulted to zero</li> </ul>
LBG	0
Log Target	4
default-log GenericServicesFileLog Latency [redacted]-Off-Box-LogTarget	
MQQM	18
[redacted]AuditLog_MPG_QM1	✓ Hostname [redacted]mqdev01(1422)]
[redacted]AuditLog_MPG_QM10	✓ Hostname [redacted]mqdev01(1422)]
[redacted]AuditLog_MPG_QM11	✓ Hostname [redacted]mqdev01(1423)]
[redacted]AuditLog_MPG_QM12	✓ Hostname [redacted]mqdev01(1422)]

## c) Search Domain

This function can be used to search domains for files matching a name pattern or file content for a string or unused objects that needs to be cleaned up.

Search Domain	
Device*	DEV-ESB
Domain*	Generic
Type*	Unused Objects ▼
Search Pattern	Unused Objects
	Unused XSLs
	Credentials
Query Certific	Search Text
Device*	List Files

Type can be – Unused Objects that do not have a reference in the configuration, Unused XSLs that no Processing Rule refers, credentials used on any object, Search text within all local files or List all files matching a pattern.

Domain is mandatory.

Search Pattern is applicable only for Type – Search Text & List Files.

## ➤ Use Case-1: Search for all Unused Objects

Search Domain	
Device*	DEV-ESB
Domain*	Sandbox-IBM
Type*	Unused Objects ▼
Search Pattern	
Search	

<b>Sandbox-IBM</b>	DEV-ESB:https://[REDACTED]:53:5550/service/mgmt/current
Orphan Objects	
PasswordAlias	WSRRPass_v8
SSLProxyProfile	wsrr_int
DocumentCryptoMap	EOBSIgn2
DocumentCryptoMap	Q1
DocumentCryptoMap	Q2
DocumentCryptoMap	Q3
StylePolicyAction	EOBSIgn_request_sign_0
StylePolicy	JsonJsonx_PP
XMLManager	default-wsrr
XMLManager	SchedulerXMLMgr
HTTPSSourceProtocolHandler	dpTool-https-fsh2
PolicyAttachments	JsonJsonx
TCPProxyService	test1
Note:These objects do not have reference within the config but can be used elsewhere, confirm before deleting	

- Use Case-2: Search for all Unused XSL files and Missing XSL files

Search Domain	
Device*	DEV-ESB
Domain*	Sandbox-IBM
Type*	Unused XSLs ▼
Search Pattern	
Search	

This will list both Unused XSL files and missing XSL files that a Processing Action is referring.

<b>Sandbox-IBM</b>	DEV-ESB:https://10.10.10.253:5550/service/mgmt/current
Unused XSLs	
local:/Service/CSCOBInquiry_v2/Common/XSL/CommonUtilities.xsl	Unused
local:/Service/CSMemberInquiry_v4/Common/XSL/CommonUtilities.xsl	Unused
local:/Service/CSCOBInquiry_v2/setCache_xform:local:/Service/CSCOBInquiry_v2/Common/XSL/cache-set.xsl	Transform Action XSLT is pointing to invalid file
local:/Service/Sign_request_get_plaintext:local:/Service/SignStatement.xsl	Transform Action XSLT is pointing to invalid file
JsonJsonx_PP_rule_2_xform_1:local:/scheduler.xsl	Transform Action XSLT is pointing to invalid file

- Use Case-3: Search for all User IDs used in a domain

Search Domain	
Device*	DEV-ESB
Domain*	Generic
Type*	Credentials ▼
Search Pattern	
Search	

<b>Generic</b>	DEV-ESB:https://10.10.10.253:5550/service/mgmt/current
Login IDs	
LogTarget/dp:config	perfre
LogTarget/dp:config	mwops
MQQM/dp:config	dpclient
MQQM/dp:config	dpclient
MQQM/dp:config	dpclient

- Use Case-4: Search for all WTX map files in a domain

Search Domain	
Device*	DEV-ESB
Domain*	Generic
Type*	List Files ▼
Search Pattern	dpa
Search	

Generic
Search For: dpa
local:/usr/share/AuditLog/wtx/WTX_TRNS_HRRK_EXTRACT_XML.dpa
local:/usr/share/AuditLog/wtx/WTX_TRNS_HIQK_EXTRACT_XML.dpa

Similarly, you can search for any file/extension.

- Use Case-5: Search for a file named 'log', used for logging

Search Domain	
Device*	DEV-ESB
Domain*	Generic
Type*	List Files ▼
Search Pattern	log
Search	

Generic
Search For: log
local:/usr/share/AuditLog/xsl/logAuditEvent.xsl
local:/usr/share/AuditLog/xsl/logErrorMessage.xsl
local:/usr/share/AuditLog/xsl/logAuditLog.xsl
local:/usr/share/Service/logTransaction.xsl
local:/usr/share/Service/xsl/logTransaction.xsl

- Use Case-6: Search for 'uuid' function usage in all XSL files in a domain

Search Domain	
Device*	DEV-ESB
Domain*	Generic
Type*	Search Text ▼
Search Pattern	uuid
Search	

Generic	DEV-ESB:https://[redacted]:253:5550/service/mgmt/current	
Search For: uuid		
local:/XSL/setSystem.xsl		
local:/[redacted]AuditLog/xml/Environment.xml		
local:/[redacted]2AuditLog/xsl/logHIQKAuditEvent.xsl	[30]	<xsl:variable name="transactionID" select="dp:generate-uuid()" />
local:/[redacted]AuditLog/xsl/logErrorMessage.xsl		
local:/[redacted]AuditLog/xsl/setResponse.xsl		
local:/[redacted]AuditLog/xsl/logCLS2AuditLog.xsl	[30]	<xsl:variable name="transactionID" select="dp:generate-uuid()" />
local:/[redacted]Service/buildError.xsl	[53]	<dp:set-http-response-header name="'uuid'" value="dp:variable('var://context/log/uuid')"/>
local:/[redacted]Service/logTransaction.xsl	[53]	<xsl:variable name="uuid-in" select="dp:http-request-header('uuid')" />
	[54]	<xsl:variable name="uuid-saved" select="dp:variable('var://context/log/uuid')"/>
	[58]	<xsl:when test="string-length(\$uuid-in) > 0">
	[59]	<xsl:value-of select="\$uuid-in" />
	[60]	<dp:set-variable name="'var://context/log/uuid'" value="\$uuid-in" />
	[62]	<xsl:when test="string-length(\$uuid-saved) > 0">
	[63]	<xsl:value-of select="\$uuid-saved" />
	[64]	<dp:set-http-response-header name="'uuid'" value="\$uuid-saved" />
	[67]	<xsl:variable name="uuid" select="dp:generate-uuid()" />
	[68]	<xsl:value-of select="\$uuid" />
	[69]	<dp:set-variable name="'var://context/log/uuid'" value="\$uuid" />
	[70]	<dp:set-http-request-header name="'uuid'" value="\$uuid" />
	[327]	<xsl:variable name="uuid-in" select="dp:http-request-header('uuid')"/>
	[328]	<xsl:variable name="uuid-saved" select="dp:variable('var://context/log/uuid')"/>
	[332]	<xsl:when test="string-length(\$uuid-in) > 0">
	[333]	<xsl:value-of select="\$uuid-in" />
	[334]	<dp:set-variable name="'var://context/log/uuid'" value="\$uuid-in" />
	[336]	<xsl:when test="string-length(\$uuid-saved) > 0">
	[337]	<xsl:value-of select="\$uuid-saved" />
	[338]	<dp:set-http-response-header name="'uuid'" value="\$uuid-saved" />
	[341]	<xsl:variable name="uuid" select="dp:generate-uuid()" />
	[342]	<xsl:value-of select="\$uuid" />
	[343]	<dp:set-variable name="'var://context/log/uuid'" value="\$uuid" />
	[344]	<dp:set-http-request-header name="'uuid'" value="\$uuid" />

Search returns all the lines that matches the given string, along with line numbers.

Similarly, you can search for any field or a variable or a property within all the local files.

➤ Use Case-7: Search for a field named ServiceTypeCd to see where all the field is mapped

Search Domain		
Device*	DEV-ESB	
Domain*	Accums	
Type*	Search Text	▼
Search Pattern	ServiceTypeCd	
	Search	
local:/Service/[redacted]tion/xsl/Benefit_Suppression.xsl	[24]	<xsl:when test="(ServiceTypeCd='[redacted]' and (BenefitTypeCode='[redacted]'
local:/Service/[redacted]tion/xsl/Check_Eligibility_v2.xsl		
local:/Service/[redacted]tion/xsl/ErrorResponseMapping.xsl		
local:/Service/[redacted]tion/xsl/CallClaimLineDetails.xsl		
local:/Service/[redacted]tion/xsl/ToHTML.xsl	[222]	<td><xsl:value-of select="ServiceTypeCd"/></td>
	[223]	<td><xsl:value-of select="ServiceTypeCdDesc"/></td>

You can also search for Comments Section for a specific Change Number or a date or a Change History etc.

Note: Search is case sensitive in case of Host Alias or DNS server look up.

## f) Compare Configuration

This function can be used to compare configuration of a specific domain on different devices. This will extract the property/value differences that exists within the DataPower Configuration. Use this extensively to see what code has not been promoted to next environment.

One can use PowerShell script effectively to use this HTTP GET URL to scan for all domains and dump the differences to an excel file.

Compare Config		
Device 1*	DEV-GW	
Domain 1*	CommonServices	
Device 2*	MO-GW1	
Domain 2*	CommonServices	
		Compare

Domain	Object	Difference
CommonServices	CryptoCertificate [Client_cert]	Not found
CommonServices	CryptoCertificate [ibm_test_client]	Not found
CommonServices	CryptoidentCred [GatewayESBIDCred]	Key[ri...wdp...bluesnet.net != services-mo...bluesnet.net] Certificate[nase...
CommonServices	CryptoValCred [GatewayCryptoValidationCred]	Not found
CommonServices	SSLServerProfile [GatewayESBSSLServerProfile]	Not found
CommonServices	SSLServerProfile [GatewayTwoWayESBServerProfile]	Not found
CommonServices	LogTarget [CommonServices-Log]	RemoteAddress[...utlsg01 != ...utlmo01] RemoteLogin[wdprcdev != wdprcmod] R...
CommonServices	LogTarget [default-log]	Class[all != ] Priority[error != ] Class[mgmt != ] Priority[notice != ]
CommonServices	LogTarget [...Off-Box-LogTarget]	LocalFile[logstore:///CommonServices-NUGWDEV01.log != logstore:///CommonServ...
CommonServices	XMLManager [CommonServices_XML_Manager]	Not found
CommonServices	HTTPSourceProtocolHandler [..._OnewaySSL_HTTPS_FSH]	SSLServer[GatewayESBSSLServerProfile != GatewayESBSSLProxyProfile]
CommonServices	HTTPSourceProtocolHandler [..._TwowaySSL_HTTPS_FSH]	GET[off != on] SSLServer[GatewayTwoWayESBServerProfile != GatewayTwowayProxi...
CommonServices	MultiProtocolGateway [..._OnewaySSL_RoutingProxy_MPGW]	XMLManager[CommonServices_XML_Manager != CommonService_XML_Manager] F...
CommonServices	MultiProtocolGateway [..._TwowaySSL_RoutingProxy_MEGW]	XMLManager[CommonServices_XML_Manager != CommonService_XML_Manager]

Differences are listed as [Property A != Property B] for each Object and its child elements.

**Note:** This is not a perfect comparison and has limitations. Suppose order of the property is differing but all values match, even then this would list it as a difference as order is not considered. Use this only as a reference or a starting point.

## g) Compare Files

This function compares all files within a domain from one device to another to find out the gap in the code between the environments. Compare Configuration and Compare Files can be used in combination to find the overall gap between the environments.

Compare Files		
Device 1*	DEV-GW	
Domain 1*	CommonServices	
Device 2*	MO-GW1	
Domain 2*	CommonServices	
		Compare

File	Difference
local:/xml/URLList.xml	#3 <!-- URL for [REDACTED] Service Start --> <!-- URL for [REDACTED] C EOB Service Start -->
	#10 <!-- URL for [REDACTED] Service End --> <!-- URL for [REDACTED] OB Service End -->
	#12 <!-- Contract and Member service--> <!--ContractAndMemberService-->
	#16 <!-- [REDACTED] PMBR - Internal --> <!-- [REDACTED] PMBR Service -->
	#176 <!-- [REDACTED] PMNA - Internal --> <!-- [REDACTED] PMNA Service -->
	#180 <!-- [REDACTED] PBIL - Internal --> <!-- [REDACTED] PBIL Service -->
	#220 <!-- [REDACTED] PMST - Internal --> <!-- [REDACTED] PMST Service -->
	#278 <!-- [REDACTED] PPER - Internal --> <!-- [REDACTED] PPER Service -->
	#288 <!-- [REDACTED] PREF - Internal --> <!-- [REDACTED] PREF Service -->
	#301 <!-- URL For [REDACTED] Service End --> <!-- URL for [REDACTED] Service End -->
	#310 <!-- URL for [REDACTED] etBenefits Service --> <URI id=[REDACTED]enefits-core-webservicesBxBenefits">
	Another 226 differences found
local:/xslt/common/BuildRoutingURL.xsl	Same Size
local:/CommonHC/CommonHealthCheck.xsl	Same Size

**Note:** This would list only first 10 differences and then a count of total differences to know that file differs. If file sizes are exactly matching, then files are not compared.



## 5. Notes

- ❖ All functions respond to HTTP GET URL and hence can be called by a PowerShell Script or a Batch Script or an Excel Web Query and hence can be used effectively to automate many functions outside the browser
- ❖ Feel free to modify the code and add your own functions and checks
- ❖ Restrict the access to the service so that unauthorised users do not extract information
- ❖ Some of the functions like file comparison can be stressful on the DataPower system usage as it must extract the contents of all files within the domain for comparison. Educate developers/users to use these functions off hours to avoid any performance impacts to production or live transactions

I would be delighted to know if this tool has saved lot of hours for you/your team, drop me an email 😊