

IBM Z Common Data Provider
2.1

User Guide



Figures

- 1. Flow of operational data among IBM Z Common Data Provider components to multiple analytics platforms..... 2
- 2. End-to-end flowchart of how to use Z Common Data Provider..... 6
- 3. Configure regex example 1..... 233
- 4. Configure regex example 2..... 234
- 5. Join data stream SMF_070_BCT and SMF_070_BPD_V2..... 269
- 6. Example of joining data stream SMF_070_BCT and SMF_070_BPD_V2.....270
- 7. Join data stream SMF_070_LCD and SMF_070_CPU_V2..... 270
- 8. Troubleshooting flowchart.....307

Tables

1. Required authorizations and associated information for each component.....	9
2. Working directories for IBM Z Common Data Provider components.....	10
3. Data gatherer configuration of Data Streamer port number.....	11
4. Target libraries for IBM Z Common Data Provider components.....	13
5. Protocol change rules.....	43
6. SMF record type 127 subtype 1000 layout.....	61
7. Fields generated by CICS with default dictionary.....	72
8. Fields collected from OMEGAMON XE for CICS OMEGBSC sections.....	90
9. Fields collected from OMEGAMON XE for CICS OMEGDB2 sections.....	91
10. Fields collected from OMEGAMON XE for CICS OMEGDLI sections.....	92
11. Fields collected from OMEGAMON XE for CICS OMEGMQ sections.....	92
12. Fields collected from OMEGAMON XE for CICS OMEGWLM sections.....	93
13. Fields collected from OMEGAMON XE for CICS OMEGCICS sections.....	93
14. Fields collected from OMEGAMON XE for CICS OMEGUEVNT sections.....	94
15. Common target destinations with the required streaming protocols and associated information.....	103
16. Mapping of the prefix that is used in a Logstash configuration file name to the content of the file.....	109
17. Mapping of the prefix that is used in a Logstash configuration file name to the content of the file.....	111
18. User exits for collecting z/OS SYSLOG data, with associated MVS installation exits and usage notes	128
19. Example System Data Engine interval values that are a factor of the total time in one day.....	136
20. Configuration reference information for managing policies.....	152
21. Subgroup and data streams of Starter Sets.....	158
22. Data stream names that IBM Z Common Data Provider uses to collect SMF data.....	159
23. Data stream names that IBM Z Common Data Provider uses to collect DCOLLECT data.....	176

24. Fields in the SMF_110_1_KPI data stream.....	177
25. Icons on each data stream node in a policy.....	180
26. Icons on each transform node in a policy.....	180
27. Icons on each subscriber node in a policy.....	181
28. Correlation between the sources from which the Log Forwarder gathers data and the data streams that can be defined for those sources.....	182
29. Support symbols.....	183
30. Support symbols.....	186
31. Support symbols.....	189
32. Correlation between the RMF III report type and data stream name.....	206
33. Parts in a CSV record.....	207
34. Logical operation NOT.....	244
35. Logical operations AND and OR.....	244
36. Field formats for the FIELDS clause of the DEFINE RECORD statement.....	254
37. z/OS console commands for starting, stopping, or viewing status or configuration information for individual Log Forwarder data streams.....	278
38. Headers for data that is sent by using the Data Transfer Protocol.....	283
39. Unsplit payload format.....	284
40. Split payload format.....	285
41. Metadata keywords and values.....	286
42. IBM Z Decision Support lookup table members to customize.....	294
43. Sample jobs for adding tables to IBM Db2 Analytics Accelerator for z/OS.....	295
44. Sample jobs for moving lookup table contents to IBM Db2 Analytics Accelerator for z/OS.....	295
45. IBM Z Common Data Provider lookup table members.....	296
46. Sample jobs for generating Db2 UNLOAD format.....	296
47. Sample jobs for loading data into IBM Db2 Analytics Accelerator.....	297
48. Sample jobs for enabling tables for acceleration in IBM Db2 Analytics Accelerator.....	297

49. Sample jobs that are provided by IBM Z Decision Support for removing tables from IBM Db2 Analytics Accelerator for z/OS.....	298
50. IBM Z Decision Support analytics components that can be loaded by the System Data Engine.....	299
51. Tables for Analytics - z/OS Performance component of IBM Z Decision Support, with corresponding base component tables.....	300
52. Tables for Analytics - Db2 component of IBM Z Decision Support, with corresponding base component tables.....	302
53. Tables for Analytics - KPM CICS component of IBM Z Decision Support, with corresponding base component tables.....	302
54. Tables for Analytics - KPM Db2 component of IBM Z Decision Support, with corresponding base component tables.....	303
55. Tables for Analytics - KPM z/OS component of IBM Z Decision Support, with corresponding base component tables.....	303
56. IBM Z Decision Support analytics component views that are based on multiple tables.....	304

Contents

- Figures..... iii**
- Tables..... v**
- Chapter 1. Overview..... 1**
 - Operational data..... 2
 - Analytics platforms..... 2
 - Components of Z Common Data Provider..... 3
 - End-to-end flowchart..... 5
- Chapter 2. Planning..... 7**
 - z/OS system requirements..... 7
 - Data Receiver system requirements..... 7
 - Configuration Tool browser requirements..... 8
 - Required authorizations for Common Data Provider components..... 9
 - Z hardware requirements for zIIP offload..... 10
 - Working directory definitions..... 10
 - Data Streamer port definition..... 10
- Chapter 3. Installing..... 13**
- Chapter 4. Upgrading..... 15**
 - Additional steps for upgrading IBM Z Common Data Provider from version 1.1.0 to 2.1.0..... 17
 - Upgrading user exit from IBM Z Common Data Provider version 1.1.0 to 2.1.0..... 17
 - Upgrading IMS user exit from IBM Z Common Data Provider version 1.1.0 to 2.1.0..... 18
 - UNIX System Services permission requirements change to some Log Forwarder scripts..... 19
 - Prefix change to the z/OS NetView message provider module and definitions..... 19
- Chapter 5. Configuring..... 21**
 - Getting started with the Configuration Tool..... 21
 - Getting started with the Configuration Tool on Liberty..... 22
 - Getting started with the Configuration Tool on z/OSMF..... 29
 - Output from the Configuration Tool..... 32
 - Managing policies..... 32
 - Purpose of transforming data in a policy..... 33
 - Subscribers to a data stream or transform..... 34
 - Creating a policy..... 36
 - Updating a policy..... 43
 - Migrating a policy..... 44
 - Adding a subscriber for a data stream or transform..... 44
 - Updating subscriptions of a subscriber..... 45
 - Exporting and importing subscribers..... 45
 - Managing custom data streams..... 46
 - Creating a System Data Engine data stream definition..... 46
 - Creating an application data stream definition..... 94
 - Deleting a custom data stream definition..... 96
 - Securing communications between the Data Streamer and its subscribers..... 96
 - Securing communications using self-signed certificate..... 97
 - Securing communications using non self-signed certificate..... 100

Preparing the IBM Z Common Data Provider and the target destinations to stream and receive data	103
Preparing to send data to Splunk.....	105
Preparing to send data to Elasticsearch.....	109
Preparing to send data to Humio.....	110
Configuring the Data Receiver.....	113
Configuring a Logstash receiver.....	117
Configuring Kafka.....	118
Configuring the Data Streamer.....	118
File buffer function in the Data Streamer.....	121
Metrics capture function in the Data Streamer.....	121
Configuring the data gatherer components.....	124
Configuring the Log Forwarder.....	124
Configuring the System Data Engine.....	133
Verifying the search order for the TCP/IP resolver configuration file.....	151
Configuration reference for managing policies.....	152
Global properties that you can define for all data streams in a policy.....	153
Groups of data streams in the Configuration Tool.....	157
SMF data stream reference.....	158
DCOLLECT Data stream reference.....	176
SMF_110_1_KPI data stream content.....	176
Icons on each node in a policy.....	180
Data stream configuration for data gathered by Log Forwarder.....	181
Data stream configuration for data gathered by System Data Engine.....	228
Transform configuration.....	229
Subscriber configuration.....	235
Language reference for System Data Engine.....	239
Language overview.....	239
DEFINE RECORD statement.....	250
DEFINE UPDATE statement.....	258
DEFINE TEMPLATE statement.....	263
DEFINE TABLE statement.....	264
Configuration scenarios.....	267
Binding the Data Streamer to a specific IP address.....	267
Join data streams for SMF record type 70 subtype 1.....	269
Chapter 6. Operating.....	271
Running the Data Receiver.....	271
Running the Data Receiver with scripts.....	272
Running the Data Receiver as systemd services on Linux.....	273
Running the Data Receiver as system services on Windows.....	274
Running the Data Streamer.....	276
Running the Log Forwarder.....	276
Running the System Data Engine.....	281
Chapter 7. Sending user application data to the Data Streamer.....	283
Data Transfer Protocol.....	283
Sending data by using the Java API.....	287
Sending data by using the REXX API.....	288
Chapter 8. Loading data to IBM Db2 Analytics Accelerator.....	291
Configuring IBM Z Decision Support for loading the data.....	291
Running the System Data Engine to write data in Db2 UNLOAD format.....	295
Loading data to IBM Db2 Analytics Accelerator.....	297
Removing tables from IBM Db2 Analytics Accelerator.....	298
IBM Z Decision Support analytics components that can be loaded by the System Data Engine.....	298
Analytics component tables.....	300
Analytics component views that are based on multiple tables.....	304

Chapter 9. Troubleshooting.....	305
Troubleshooting flowchart.....	306
Log Forwarder log files.....	308
Log Forwarder: enabling tracing.....	308
Enabling static tracing for the Log Forwarder.....	308
Enabling dynamic tracing for the Log Forwarder.....	309
System Data Engine log files.....	310
System Data Engine: enabling tracing and statistics data.....	310
Enabling tracing for the System Data Engine at startup.....	310
Enabling tracing for the System Data Engine after startup.....	311
Enabling statistics data for the System Data Engine after startup.....	312
Enabling dynamic tracing for the Data Streamer.....	312
Configuration Tool issues.....	314
Configuration Tool troubleshooting checklist.....	314
Subscribers are greyed out in the Subscribe to a data stream window.....	315
The Configuration Tool failed to load with the error message SRVE0295E.....	315
The Configuration Tool failed to load the policy with the error message HBO6502E regarding message queue size.....	315
Transform boxes are missing in the Configuration Tool after upgrading IBM Z Common Data Provider from Version 1.1 to Version 2.1.....	316
The text of the buttons or boxes doesn't appear correctly.....	316
No data streams available for creating policies after deploying the Configuration Tool on z/ OSMF.....	316
User ID of parameter AUTHORIZED_USER is not found.....	317
Failed to open the Configuration Tool on z/OSMF with the error message HBO6501W.....	317
The buttons in the Configuration Tool do not show the text but the variable name behind it.....	318
Troubleshooting Liberty issues.....	319
Log Forwarder issues.....	322
The Log Forwarder is not able to gather all data when a VSAM ESDS data set is deleted or redefined.....	322
The Log Forwarder procedure ended with a message saying out of memory	323
The Log Forwarder fails to warm start with the error message HBOB003E.....	323
Log Forwarder cannot gather job logs due to spool files purge.....	324
Log Forwarder user ID has insufficient authority.....	324
Log Forwarder cannot generate RACF PassTicket with return code and reason codes SafRc=8, racfRc=8 racfRsn=16.....	325
Log Forwarder failed to parse the data returned from the RMF Distributed Data Server with an HTTP response code 401.....	325
Log Forwarder gets message HBOD007E and EDC8128I at startup.....	326
Log Forwarder gets a HBOPROC CLASS error at startup.....	326
BPX messages precede HBO messages in the z/OS SYSLOG.....	327
Log Forwarder message states that PPI issued return code 24.....	328
NetView message provider HBONETV issues message HBOL004E with return code 15.....	328
NetView message provider HBONETV issues message HBOL006E.....	328
System Data Engine issues.....	329
System Data Engine does not start.....	329
The System Data Engine gets ABEND U006 at startup.....	329
The System Data Engine fails to start with SYSTEM COMPLETION CODE=DC4 REASON CODE=90041620.....	330
The System Data Engine has messages in its job log saying data fields being null when streaming SMF_030 and SMF_080.....	330
The message HBO0308I shows up frequently when collecting data with the System Data Engine.....	331
The SMF data packets that are sent to the target subscriber are very large.....	331
The System Data Engine ends with RC=8 when collecting SMF_110_2 related records.....	332
Data Streamer is not receiving data from System Data Engine.....	332

Data Streamer issues.....	333
Data Streamer does not start.....	333
Data Streamer gets message HBO6057E at startup.....	333
Data Streamer fails to start with the message JVMJ9VM015W.....	334
The Data Streamer issues a message about Java out of memory when a target subscriber remains unresponsive for a long time.....	335
Data Streamer is not receiving data from System Data Engine.....	335
The Data Receiver has a high CPU usage.....	336
Subscriber is not receiving data.....	337
syslogd message problems: inconsistencies in timestamp, or missing or misplaced messages.....	337
Logstash gets JSON parse error messages when receiving data from IBM Z Common Data Provider	338
Notices.....	339
Trademarks.....	340
Terms and conditions for product documentation.....	340

Chapter 1. Z Common Data Provider overview

IBM® Z Common Data Provider provides the infrastructure for accessing IT operational data from z/OS® systems and streaming it to the analytics platform in a consumable format. It is a single data provider for sources of both structured and unstructured data, and it can provide a near real-time data feed of z/OS operational data, like System Management Facilities (SMF) data and z/OS log data to your analytics platform.

IBM Z Common Data Provider automatically monitors SMF data and z/OS log data, it can collect SMF data and z/OS log data, and forwards it to the configured destination.

In each logical partition (LPAR) from which you want to analyze SMF data or z/OS log data, a unique instance of IBM Z Common Data Provider must be installed and configured to specify the type of data to be gathered and the destination for that data, which is called a *subscriber*.

IBM Z Common Data Provider includes a web-based configuration tool that is provided as an application for IBM WebSphere® Application Server for z/OS Liberty, or as a plug-in for IBM z/OS Management Facility (z/OSMF).

Flow of operational data to your analytics platform

As illustrated in [Figure 1 on page 2](#), operational data (such as SMF data or log data) is gathered by data gatherers, such as the System Data Engine or the Log Forwarder, and can be streamed to multiple subscribers.

The data gatherers send the data to the Data Streamer, which transforms the data before it sends the data to the subscribers.

The flow of data is controlled by a policy that you define in the IBM Z Common Data Provider Configuration Tool.

policy

In IBM Z Common Data Provider, a set of rules that define what operational data to collect and where to send that data. A policy is created in the Configuration Tool. See [“Managing policies” on page 32](#) for more information.

subscriber

In the IBM Z Common Data Provider configuration, the software that you define to receive operational data, like IBM Operations Analytics - Log Analysis, Elastic Stack, Splunk, Logstash, and Kafka. For more information about subscribers, see [“Subscribers to a data stream or transform” on page 34](#) and [“Subscriber configuration” on page 235](#).

Batch job

The System Data Engine can run as a batch job to create output data for the IBM Z® Decision Support, or for the Data Streamer as a part of the Data Gatherers. See the following tasks about running batch jobs.

- [“Creating the System Data Engine batch job for writing SMF data to data sets” on page 147](#)
- [“Creating the System Data Engine batch job for sending SMF data to the Data Streamer” on page 148](#)
- [“Creating the System Data Engine batch job for writing DCOLLECT data to data sets” on page 149](#)
- [“Creating the System Data Engine batch job for sending DCOLLECT data to the Data Streamer” on page 150](#)

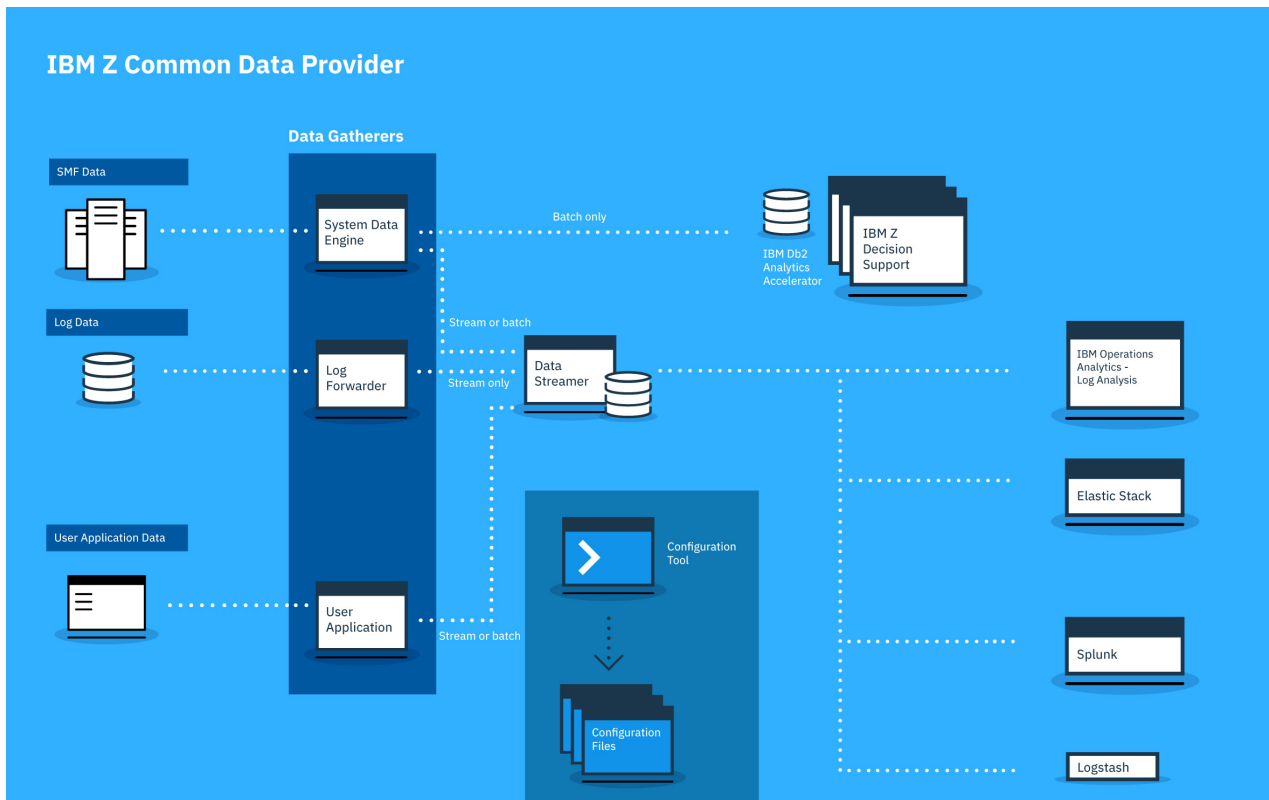


Figure 1. Flow of operational data among IBM Z Common Data Provider components to multiple analytics platforms

Operational data

Operational data is data that is generated by the z/OS system as it runs. This data describes the health of the system and the actions that are taking place on the system. The analysis of operational data by analytics platforms and cognitive agents can produce insights and recommended actions for making the system work more efficiently and for resolving, or preventing, problems.

IBM Z Common Data Provider can collect the following types of operational data:

- System Management Facilities (SMF) data
- z/OS log data from the following sources:
 - Job log, which is output that is written to a data definition (DD) by a running job
 - z/OS UNIX log file, including the UNIX System Services system log (syslogd)
 - Entry-sequenced Virtual Storage Access Method (VSAM) cluster
 - z/OS system log (SYSLOG)
 - IBM Tivoli® NetView® for z/OS messages
 - IBM WebSphere Application Server for z/OS High Performance Extensible Logging (HPEL) log
 - IBM Resource Measurement Facility (RMF) Monitor III reports
- User application data, which is operational data from your own applications

Analytics platforms

An analytics platform is a software program, or group of dedicated systems and software, that is configured to receive, store, and analyze large volumes of operational data.

The following analytics platforms are examples:

- IBM Db2® Analytics Accelerator for z/OS, a database application that provides query-based reporting
- IBM Z Operations Analytics, an on-premises product that can receive large volumes of operational data for analysis and can provide insights and recommended actions to the system owners, which are based on expert knowledge about z Systems® and applications
- Platforms such as Elasticsearch, Splunk, and Kafka that can receive and store operational data for analysis. These platforms do not include expert knowledge about z Systems and applications, but users can create or import their own analytics to run against the data.

Components of Z Common Data Provider

IBM Z Common Data Provider includes the following basic components: 1) a Configuration Tool for defining the sources from which you want to collect operational data, 2) the data gatherer components (System Data Engine and Log Forwarder) for gathering different types of operational data, and 3) a Data Streamer for streaming all data to its destination.

Other components include the Open Streaming API for gathering operational data from your own applications, a Data Receiver that acts as a target subscriber for operational data if the intended subscriber cannot directly ingest the data feed, the Buffered Splunk Ingestion App for data ingestion to Splunk via the Data Receiver, and the Elasticsearch ingestion kit for Logstash configuration for sending data to Elasticsearch.

The components are illustrated in [Figure 1 on page 2](#).

Basic components

Configuration Tool

The IBM Z Common Data Provider Configuration Tool is a web-based user interface that is provided as an application for IBM WebSphere Application Server for z/OS Liberty, or as a plug-in for IBM z/OS Management Facility (z/OSMF). In the tool, you specify the configuration information as part of creating a *policy* for streaming operational data to its destination.

In the policy definition, you must define a *data stream* for each source from which you want to collect operational data. A stream of data is a set of data that is sent from a common source in a standard format, is routed to, and transformed by, the Data Streamer in a predictable way, and is delivered to one or more subscribers.

You must specify the following information for each data stream in the policy:

- The source (such as SMF record type 30 or z/OS SYSLOG)
- The format to which to transform the operational data so that it is consumable by the analytics platform.
- The subscriber or subscribers for the operational data that is output by IBM Z Common Data Provider.

For example, subscribers include Logstash, the Data Receiver, the HTTP Event Collector (HEC) of Splunk, Kafka, and a generic HTTP receiver.

Data gatherer components

Each of the following components gathers a different type of data:

System Data Engine

The System Data Engine gathers System Management Facilities (SMF) data and IBM Information Management System (IMS) log data in near real time. It can also gather SMF data, IMS data, and DCOLLECT data in batch.

The System Data Engine can process all commonly used SMF record types from the following sources:

- SMF archive (which is processed only in batch)
- SMF in-memory resource (by using the SMF real-time interface)
- SMF user exit HBOSMFEX

- SMF log stream

It can also convert SMF records into a consumable format, such as a comma-separated values (CSV) file, or into Db2 UNLOAD format for loading in batch.

The System Data Engine can also be installed as a stand-alone utility to feed SMF data into IBM Db2 Analytics Accelerator for z/OS (IDAA) for use by IBM Z Decision Support.

To reduce general CPU usage and costs, you can run the System Data Engine on z Systems Integrated Information Processors (zIIPs). For more information, see [“Offloading the System Data Engine code to z Systems Integrated Information Processors”](#) on page 151.

Log Forwarder

The Log Forwarder gathers z/OS log data from the following sources:

- Job log, which is output that is written to a data definition (DD) by a running job
- z/OS UNIX log file, including the UNIX System Services system log (syslogd)
- Entry-sequenced Virtual Storage Access Method (VSAM) cluster
- z/OS system log (SYSLOG)
- IBM Tivoli NetView for z/OS messages
- IBM WebSphere Application Server for z/OS High Performance Extensible Logging (HPEL) log
- IBM Resource Measurement Facility (RMF) Monitor III reports

To reduce general CPU usage and costs, you can run the Log Forwarder on z Systems Integrated Information Processors (zIIPs).

User Application

The IBM Z Common Data Provider Open Streaming API provides an efficient way to gather operational data from your applications by enabling your applications to be data gatherers. You can use the API to send your application data to the Data Streamer and stream it to analytics platforms.

For more information about how to send user application data to the Data Streamer, see [Chapter 7, “Sending user application data to the Data Streamer,”](#) on page 283.

Data Streamer

The Data Streamer streams operational data to configured subscribers in the appropriate format. It receives the data from the data gatherers, alters the data to make it consumable for the subscriber, and sends the data to the subscriber. In altering the data to make it consumable, the Data Streamer can, for example, split the data into individual messages, or translate the data into a different encoding (such as from EBCDIC encoding to UTF-8 encoding).

The Data Streamer can stream data to both on-platform and off-platform subscribers. To reduce general CPU usage and costs, you can run the Data Streamer on z Systems Integrated Information Processors (zIIPs).

Other components

Depending on your environment, you might want also want to use one, or both, of the following components:

Open Streaming API

The Open Streaming API provides an efficient way to gather operational data from your own applications by enabling your applications to be data gatherers. You can use the API to send your application data to the Data Streamer and stream it to analytics platforms.

Data Receiver

The Data Receiver is required only if the intended subscriber of a data stream cannot directly ingest the data feed from IBM Z Common Data Provider. The Data Receiver writes any data that it receives to disk files, which can then be ingested into an analytics platform such as Splunk.

The Data Receiver typically runs on the same system as the analytics platform that processes the disk files. This system can be a distributed platform, or a z/OS system. For ingesting data to Splunk, install the Data Receiver on each Splunk forwarder that the Buffered Splunk Ingestion App is installed on.

Buffered Splunk Ingestion App

The IBM Z Common Data Provider Buffered Splunk Ingestion App must be installed in Splunk only if you are sending data to Splunk via the Data Receiver.

For more information about how to install the Buffered Splunk Ingestion App, see [“Preparing to send data to Splunk via the Data Receiver”](#) on page 106.

Elasticsearch ingestion kit

The IBM Z Common Data Provider Elasticsearch ingestion kit contains the Logstash configuration files that are provided by IBM Z Common Data Provider. Configure Logstash by using these configuration files before you can send data to Elasticsearch.

For more information about how to use the Elasticsearch ingestion kit, see [“Preparing to send data to Elasticsearch”](#) on page 109.

End-to-end flowchart

To use Z Common Data Provider, you must first ensure all the requirements are met. Then you can install, configure and run Z Common Data Provider, forward SMF data and other data types to analytics platforms, and view the data you streamed on your analytics platforms. This topic provides an end-to-end flowchart of how to use Z Common Data Provider. You can follow the step-by-step instructions to use Z Common Data Provider and click the blocks for more information.

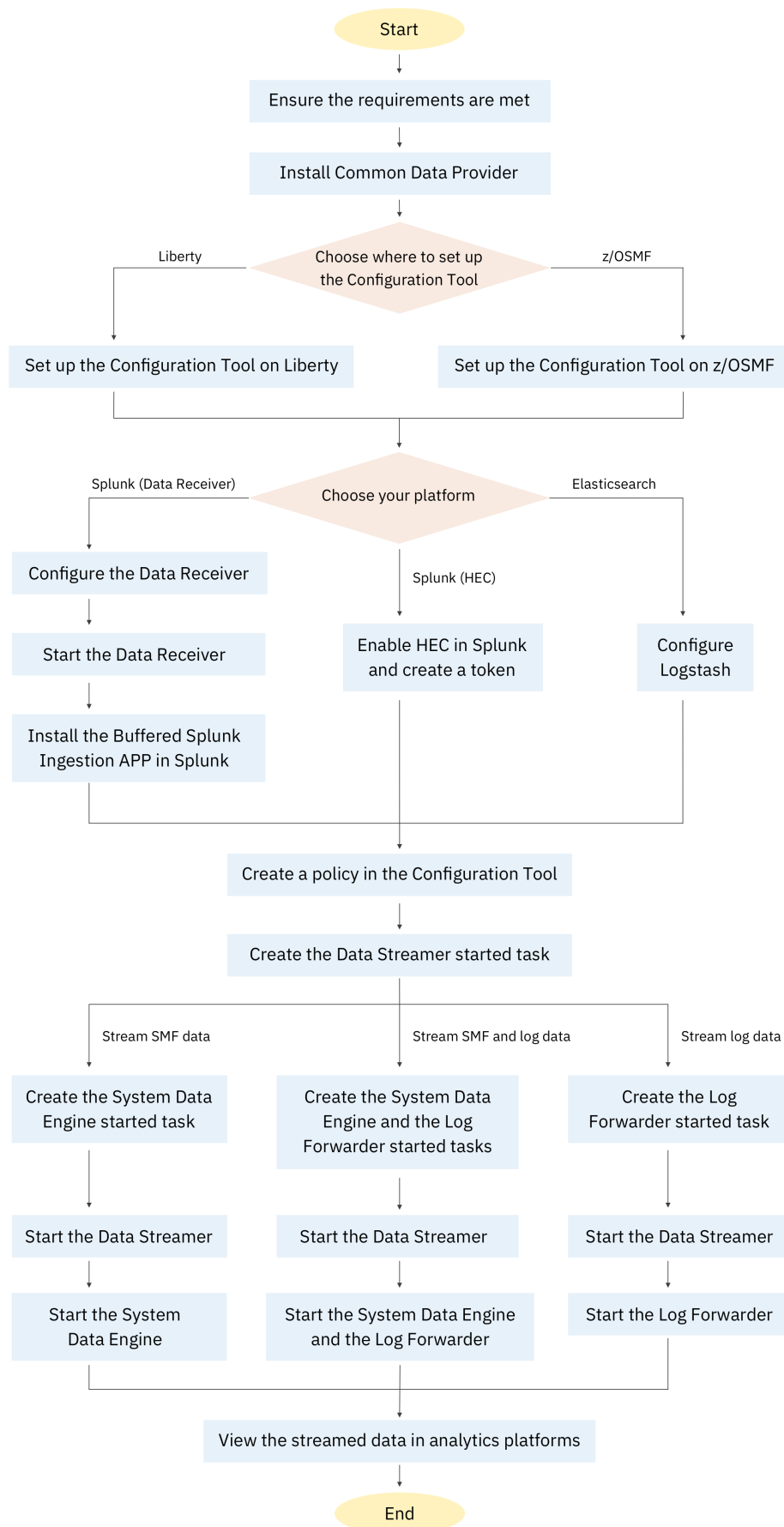


Figure 2. End-to-end flowchart of how to use Z Common Data Provider

Chapter 2. Planning to use Z Common Data Provider

Review the system and security requirements for using IBM Z Common Data Provider to provide z/OS operational data. Also, review the information about the Data Streamer port definition and about the working directories for IBM Z Common Data Provider components.

z/OS system requirements

Verify that your z/OS system meets the requirements for running IBM Z Common Data Provider. You must run the IBM Z Common Data Provider in each z/OS logical partition (LPAR) from which you want to gather z/OS operational data.

These requirements apply to the z/OS system where you are running the IBM Z Common Data Provider Data Streamer, Log Forwarder, and System Data Engine.

Basic requirements

IBM Z Common Data Provider must be run with the following software:

- IBM z/OS V2.2, or later (product number 5655-ZOS)

For z/OS V2.2 only: If you use IBM z/OS V2.2, the following software is also required on the system where configuration is done:

For z/OS V2.2

IBM z/OS Management Facility V2.2 (product number 5650-ZOS), with APAR/PTF PI52426/UI36315

- The following Java™ library:
 - IBM 64-bit SDK for z/OS Java Technology Edition V8 (product number 5655-DGH)

Important considerations:

- Use the latest available service release of the version of IBM SDK for z/OS, Java Technology Edition, that you choose, and apply fix packs as soon as possible after they are released. To find the latest service release or fix pack, see [IBM Java Standard Edition Products on z/OS](#).

Optional requirements

Depending on your environment, you might also want to run the following software with IBM Z Common Data Provider:

- On an IBM z/OS V2.2 system, to collect System Management Facilities (SMF) data from SMF in-memory resources, you must apply APAR OA49263.
- To load data to IBM Db2 Analytics Accelerator for z/OS, you must run IBM Db2 Analytics Accelerator Loader for z/OS V2.1 (product number 5639-OLE).

Data Receiver system requirements

If you plan to use the IBM Z Common Data Provider Data Receiver, verify that the system on which you plan to install the Data Receiver meets the requirements for running the Data Receiver.

The Data Receiver can be run on a Linux®, Windows system. It requires Java Runtime Environment (JRE) or Java Development Kit (JDK) version 1.8. If you plan to ingest data into Splunk via the Data Receiver, also see the [Splunk system requirements](#).

The disk usage, CPU usage and memory usage of the Data Receiver vary depending on the amount of data that is processed.

Disk usage

The disk usage of the Data Receiver varies depending on the amount of data that is processed per hour and the value of the cycle property that is defined in the `cdpdr.properties` file. The cycle property value determines the number of output data files generated by the Data Receiver. For more information about how the Data Receiver manages output files, see [“Data Receiver process for managing disk space” on page 115](#).

If you receive one data stream, the formula for calculating the disk usage of a single data stream is:

```
the amount of data streamed per hour by a data stream * the cycle property value
```

If you receive multiple data streams at the same time, you must calculate the disk usage for each data stream separately. The sum of the disk usage of each data stream is the total disk usage.

For example, if you have only one data stream streaming 20GB data per hour with its cycle property set to 3, the Data Receiver can take up to 60GB (20GB*3) of disk space. If you add one more data stream, which streams 10GB data per hour and its cycle value is 4, then the Data Receiver can take up to 100GB (20GB*3+10GB*4) of disk space. Determine the disk space required based on your environment.

To get a more robust disk space configuration, you can test the target data streams with the Data Receiver during peak hours, so that the resulting disk configuration can prevent your Data Receiver from running out of disk space during peak hours.

Tip:

- If you receive the same data stream through multiple LPARs, when calculating the disk space, you must multiply the disk space of the data stream for a single LPAR by the number of LPARs to get the total disk space. For example, if you receive one data stream through two LPARs, and this data stream receives 10GB per hour under one LPAR with the cycle value set to 3, then it can take up to 60GB (2*10GB*3) of disk space.
- You must re-evaluate your disk configuration when there are changes to your IBM Z Common Data Provider configurations and its operating environment. Some possible changes are:
 - Parameter value, such as the cycle property, change to the Data Receiver.
 - More LPARs are sending data to this Data Receiver.
 - New data streams are added to the policy through the Configuration Tool.

CPU usage

On our test environment, which has 8 CPU processors with 2.3 GHz, a Data Receiver that receives 600GB of data per day can take up to 15% of the system CPU utilization. Reserve the CPU capacity according to this reference.

Memory Usage

The minimum memory required is 2GB.

The recommended memory is 4GB.

Configuration Tool browser requirements

The Configuration Tool can be deployed as an application to IBM WebSphere Application Server for z/OS Liberty, or as a plug-in for IBM z/OS Management Facility (z/OSMF).

The following web browsers are supported by both platforms:

- Mozilla Firefox
- Microsoft Internet Explorer 11

Make sure that you always use the latest version of the browsers.

Required authorizations for Common Data Provider components

Various authorizations are required for installing and configuring IBM Z Common Data Provider components and for accessing component-related libraries and configuration files during run time.

Table 1 on page 9 references the information about the required authorizations for each IBM Z Common Data Provider component. The authorization requirements for installation of the components are described in the Program Directories.

Component	Information about required authorizations
Configuration Tool	<p>If your Configuration Tool is on Liberty:</p> <ul style="list-style-type: none"> • User IDs and group IDs for running the Configuration Tool: “Configuring user IDs, group IDs, and security product” on page 22 • User ID for running the setup script: “Setting up a Liberty server directory and a working directory for the Configuration Tool” on page 25 <p>If your Configuration Tool is on z/OSMF:</p> <ul style="list-style-type: none"> • User ID for running the setup script: “Setting up a working directory for the Configuration Tool” on page 29 • User ID for installing the tool: “Installing the Configuration Tool on z/OSMF” on page 30 • User ID for uninstalling the tool: “Uninstalling the Configuration Tool from z/OSMF” on page 31 • User ID for running the tool: “Running the Configuration Tool” on page 31
Data Streamer	<ul style="list-style-type: none"> • User ID that is associated with the Data Streamer started task: “Configuring the Data Streamer” on page 118
Log Forwarder	<ul style="list-style-type: none"> • User ID that is associated with the Log Forwarder started task: “Requirements for the Log Forwarder user ID” on page 126 • Security software updates to permit the Log Forwarder started task to run in your environment: “Creating the Log Forwarder started task” on page 125, step “4” on page 125
System Data Engine	<ul style="list-style-type: none"> • APF authorization: “Authorizing the System Data Engine with APF” on page 134 • User ID that is associated with the System Data Engine started task: “Requirements for the System Data Engine user ID” on page 139 • User ID for collecting IMS records: “Requirements for the System Data Engine user ID for collecting IMS records” on page 145

Z hardware requirements for zIIP offload

All IBM Z Common Data Provider runtime components can offload work to IBM System z Integrated Information Processors (zIIPs) when they are available. This operation frees the general-purpose processors (GCPs) for other work. It can also reduce software licensing costs.

The Log Forwarder and Data Streamer run in the Java runtime environment, which automatically make them eligible for offload to a zIIP engine. The System Data Engine component, however, must enable the zIIP offloading function by changing the value of a parameter in the System Data Engine started task procedure or batch job JCL. For more information about enabling the zIIP offload function for the System Data Engine, see [“Offloading the System Data Engine code to z Systems Integrated Information Processors” on page 151](#).

To offload work from GCPs to zIIPs, make sure one or more zIIPs are online on the LPAR before the IBM Z Common Data Provider components are started. Ensure that you have enough capacity on zIIPs. The Resource Measurement Facility (RMF) provides information on zIIP usage to help you identify when to add more zIIPs to the logical partition. Also, fields in SMF Type 30 records allow you to know how much time is spent on zIIPs, and how much time is spent on GCPs running zIIP eligible work.

A high CPU time consumed on GCPs by work that is eligible for a zIIP indicates high contention on the zIIP processors. In this case, you must add more zIIPs to the logical partition, or disable the zIIP offloading function of the System Data Engine.

If your system is consistently in shortage of zIIP resources, the additional CPU time to use zIIPs can overwhelm the CPU time that is offloaded to zIIPs.

Working directory definitions

When you configure some IBM Z Common Data Provider components, you must define a working directory for the component. To avoid possible conflicts, do not define the same directory as the working directory for multiple components.

Table 2 on page 10 indicates where you can find information about the working directories that must be defined.

Component	Information about working directory
Configuration Tool	“Setting up a working directory for the Configuration Tool” on page 29
Data Receiver	“Setting up a working directory and an output directory for the Data Receiver” on page 113
Data Streamer	“Configuring the Data Streamer” on page 118
Log Forwarder	“Log Forwarder properties configuration” on page 154

Data Streamer port definition

When you configure the IBM Z Common Data Provider Data Streamer, you define the port number on which the Data Streamer listens for data from the data gatherers. All data gatherers must send data to the Data Streamer through this port. If you update this port in the Data Streamer configuration, you must also update it in the configuration for all data gatherers.

For information about how to update this port for the Data Streamer, see [“Configuring the Data Streamer” on page 118](#).

For each data gatherer, [Table 3 on page 11](#) indicates where you can find information about the Data Streamer port number configuration.

Table 3. Data gatherer configuration of Data Streamer port number

Data gatherer	Information about Data Streamer port number configuration
Log Forwarder	“Log Forwarder properties configuration” on page 154
System Data Engine	“Creating the System Data Engine started task for streaming SMF data” on page 135
User Application	Chapter 7, “Sending user application data to the Data Streamer,” on page 283

Chapter 3. Installing Z Common Data Provider

Install IBM Z Common Data Provider by using SMP/E. The installation instructions are in the IBM Z Common Data Provider Program Directory.

About this task

The IBM Z Common Data Provider Program Directory is available at https://www.ibm.com/support/knowledgecenter/SSGE3R_2.1.0/pdf.html.

Optionally, if you plan to run the Configuration Tool on IBM WebSphere Application Server for z/OS Liberty, you must install the IBM Z Common Data Provider Embedded Liberty.

Table 4 on page 13 lists the target libraries.

Component	Target library
Configuration Tool	<ul style="list-style-type: none">• /usr/lpp/IBM/zcdp/v2r1m0/UI For the following libraries, customize the high-level qualifier (.hlq) according to site requirements. <ul style="list-style-type: none">• hlq.SHBODEFS• hlq.SHBOSAMP
Data Streamer	<ul style="list-style-type: none">• /usr/lpp/IBM/zcdp/v2r1m0/DS For the following library, customize the high-level qualifier (.hlq) according to site requirements. <ul style="list-style-type: none">• hlq.SHBOSAMP
Log Forwarder	<ul style="list-style-type: none">• /usr/lpp/IBM/zcdp/v2r1m0/LF For the following libraries, customize the high-level qualifier (.hlq) according to site requirements. <ul style="list-style-type: none">• hlq.SHBOCLST• hlq.SHBOLPA
System Data Engine	For the following libraries, customize the high-level qualifier (.hlq) according to site requirements. <ul style="list-style-type: none">• hlq.SHBOLLST• hlq.SHBOLOAD
Common dependencies	<ul style="list-style-type: none">• /usr/lpp/IBM/zcdp/v2r1m0/DEPS

Chapter 4. Upgrading Z Common Data Provider


To upgrade IBM Z Common Data Provider from an earlier version to a later version, you must first upgrade the components on the subscriber platforms. Secondly, copy existing policies to an independent later version Configuration Tool that is installed on a system with later version z/OS operational components installed, and migrate the policies. Lastly, upgrade the z/OS runtime components, Log Forwarder, System Data Engine (including definitions), and Data Streamer.

About this task

The term "version" in this topic generically refers to a level of IBM Z Common Data Provider or its components. It can refer to a version, release, or continuous delivery PTF.

The IBM Z Common Data Provider components (like the Buffered Splunk Ingestion App) in the subscriber environment are compatible with, or have coexisting versions that are compatible with the data streams produced by both the earlier and later versions of the components on z/OS. The upgrade of the components in the subscriber environment, or any other changes to the subscriber environment itself in preparation of receiving data from the upgraded z/OS components must be performed before you upgrade any z/OS components.

In the later version Configuration Tool, if the earlier version policies are not compatible, the **MIGRATE ALL**

POLICIES TO THE LATEST FORMAT button appears, and a migrate button  and a warning icon



appears on each box of the policy that needs migration. In this case, you must migrate the policies at the same time when you upgrade other components on z/OS.

The IBM Z Common Data Provider components on z/OS (including the Log Forwarder, the System Data Engine, and the Data Streamer) must be upgraded together. All your LPARs, however, do not have to be upgraded at the same time.

Important:

- You might have control procedures, other JCL, scripts and processes which use the names and paths of Version 1.1.0 components. In Version 2.1.0, the components are combined into a single z/OS File System (ZFS) path. As a result, the path names and part names of the Log Forwarder have changed. Starting from Version 2.1.0, the Log Forwarder uses the HBO prefix like other components of the product, and is installed in the same ZFS path in the LF subdirectory.
- If your z/OS is upgraded to V2R4, it is recommended that you upgrade your IBM Z Common Data Provider to version 2.1.0 with the latest PTF applied to support new RMF fields that are introduced by z/OS V2R4.

Procedure

Follow the steps to upgrade the Z Common Data Provider.

1. Install the later version by using SMP/E.
2. Deploy the later version components to the subscriber environment.
 - If you are streaming data to Splunk via the Data Receiver, follow the instructions in the Splunk documentation for upgrade when you install the Buffered Splunk Ingestion App. Back up any customized data types in the default directory and manually migrate them.
 - If you are streaming data to Elasticsearch, back up any custom data types before the upgrade. After the upgrade, you might need to manually merge these custom data types.
3. Copy the policies to the later version Configuration Tool, and if necessary, migrate your policies.
 - a) Create a separate Configuration Tool of the later version.

You must deploy the later version Configuration Tool on a separate Liberty server, or on a separate z/OSMF. For more information about creating a Configuration Tool, see [“Getting started with the Configuration Tool”](#) on page 21.

- b) Copy the existing policy files from the working directory of your earlier version Configuration Tool to that of the later version Configuration Tool.
Do not move the existing policy files.
 - c) If you see the **MIGRATE ALL POLICIES TO THE LATEST FORMAT** button in the later version Configuration Tool, click the button to migrate all earlier version policies.
4. Upgrade each of your z/OS LPARs in the order of your choice.
- All components in the same LPAR must be upgraded at the same time.
 - Upgrade the z/OS components by copying and configuring the new procedures of the later version.
 - Ensure that you use the same working directories for the new z/OS components.
 - Ensure that you stop the earlier version components before you start the later version components. For more information about the commands to start and stop the components, see [Chapter 6, “Operating Z Common Data Provider,”](#) on page 271.

Important: If you are upgrading IBM Z Common Data Provider from version 1.1.0 to version 2.1.0, you must inactivate the existing exit, delete all changed modules, load the new code, and then activate the new exit. For more information about upgrading the user exit, see [“Upgrading user exit from IBM Z Common Data Provider version 1.1.0 to 2.1.0”](#) on page 17.

What to do next

- The IBM Common Data Engine for z/OS changes the Access Control List (ACL) settings for the scripts and programs executable from the USS on z/OS. The ACL setting for such executables now is 750 with the following behaviors.
 - The ID which is used for your SMP/E updates has Read/Write (7) in order to do updates.
 - The everyone bit is set to 4 so the files are not executable by everyone who has access to the file system.
 - The group bit is set to 5 to enable reading and executing of the files by someone other than the installation ID which is likely to have root access. This setting allows an operations group to execute these commands and to be associated with started tasks which run the applications. Because it is unlikely that your operations group has the same Group ID (**GID**) as the installing ID, whenever you perform a SYSMOD update via the SMP/E APPLY or RESTORE of a PTF or ++APAR, the **GID** for these files must be changed to that of your operations team to give them the execute access. The sample job **HBOCHGP** is provided to handle this task. Make a copy of the sample job, update it to comply with your local conventions, and run it after every SMP/E update to the product.
- If you are upgrading from version 1.1.0 to 2.1.0, there are additional steps you must perform depending on the functions you use. For more information, see [“Additional steps for upgrading IBM Z Common Data Provider from version 1.1.0 to 2.1.0”](#) on page 17.

Additional steps for upgrading IBM Z Common Data Provider from version 1.1.0 to 2.1.0

If you are upgrading from IBM Z Common Data Provider version 1.1.0 to 2.1.0, besides the general upgrading steps, there are additional steps you must perform depending on the functions you use.

Upgrading user exit from IBM Z Common Data Provider version 1.1.0 to 2.1.0

If you are upgrading IBM Z Common Data Provider version 1.1.0 to version 2.1.0 with the Log Forwarder gathering z/OS SYSLOG data from a user exit, you must also upgrade the user exit modules.

About this task

The prefix of the modules in version 2.1.0 is changed from GLA to HBO.

Procedure

- If you want to update the user exit at system IPL, perform the following steps.
 - a) Update the LPA1STxx member with the LPA library updated by this PTF.
 - b) Verify that one of the following EXIT ADD statements is added to the PROGxx member so that the EXIT will be activated during IPL.

```
EXIT ADD EXITNAME(CNZ_MSGTOSYSLOG) MODNAME(HBOSYSG)
```

```
EXIT ADD EXITNAME(CNZ_WTOMDBEXIT) MODNAME(HBOMDBG)
```

- c) Perform system IPL.
- If you want to dynamically update the user exit without performing system IPL, perform the following steps.
 - a) Stop the Log Forwarder if it is running.
 - b) Complete one of the following actions to remove the old Program Call modules and user exit, and delete the data space.
 - If you have applied PTF UJ03590, run the batch utility HB0LFPCE in the *hlq.SHBOSAMP* data set. Customize the JCL according to the comments in the JCL and submit the JCL.
 - If you have not applied PTF UJ03590 yet, run the following shell command.

```
manageUserExit.sh -u environment_configuration_directory
```

The `manageUserExit.sh` file is in `zcdp_high_level/zcdp/v2r1m0/LF/samples`. The `environment_configuration_directory` is the complete path of the directory where your `zlf.conf` file is located.

Important: The user ID that runs this command must have the UID 0 attribute.

- c) If earlier version modules were dynamically added to the LPA library, delete the modules by running the following commands.

```
SETPROG LPA,DELETE,MODNAME=GLASYSYG,FORCE=YES  
SETPROG LPA,DELETE,MODNAME=GLAMDBG,FORCE=YES  
SETPROG LPA,DELETE,MODNAME=GLADSRW,FORCE=YES  
SETPROG LPA,DELETE,MODNAME=GLAGDSDL,FORCE=YES  
SETPROG LPA,DELETE,MODNAME=GLAGLMSG,FORCE=YES  
SETPROG LPA,DELETE,MODNAME=GLAUERQ,FORCE=YES
```

- d) Load the modules into the dynamic LPA by running the following commands, where `usre.lpalib` is the LPA library updated by this PTF.

```
SETPROG LPA,ADD,MODNAME=HBOSYSG,DSNAME=usre.lpalib  
SETPROG LPA,ADD,MODNAME=HBOMDBG,DSNAME=usre.lpalib  
SETPROG LPA,ADD,MODNAME=HBODSRW,DSNAME=usre.lpalib
```

```

SETPROG LPA,ADD,MODNAME=HBOGDSDL,DSNAME=usre.lpalib
SETPROG LPA,ADD,MODNAME=HBOGLMSG,DSNAME=usre.lpalib
SETPROG LPA,ADD,MODNAME=HBOUERQ,DSNAME=usre.lpalib

```

- e) Activate the user exit by running one of the following commands depending on the user exit you use:

```

SETPROG EXIT ADD,EXITNAME=CNZ_MSGTOSYSLOG,MODNAME=HBOSYSG

```

```

SETPROG EXIT ADD,EXITNAME=CNZ_WTOMDBEXIT,MODNAME=HBOMDBG

```

- f) Start the Log Forwarder.

What to do next

For more information about upgrading other components, see [Chapter 4, “Upgrading Z Common Data Provider,”](#) on page 15.

Upgrading IMS user exit from IBM Z Common Data Provider version 1.1.0 to 2.1.0

If you are upgrading IBM Z Common Data Provider version 1.1.0 to version 2.1.0 with the IMS LOGWRT user exit, you must update the user exit configuration with the new dataset. Make sure that the SHBOLLST library is APF-authorized.

Procedure

To update the user exit, complete the steps that apply for your installation option.

Installation option	Steps
IMS multi-user exit	<ol style="list-style-type: none"> Add the <i>hlq</i>.SHBOLLST data set to the STEPLIB concatenation of the IMS Control Region instead of the version 1.1.0 dataset. After the IMS Control Region JCL is updated, recycle the IMS system to activate the LOGWRT user exit.
IMS tools	<p>If IMS tools are implemented for the IMS environment, update the LOGWRT user exit. IMS Tools does not require the load library to be inserted into the IMS Control Region STEPLIB JCL.</p> <ol style="list-style-type: none"> Update the IMS tools user exit definition to the IMS PROCLIB member GLXEXIT0, as shown in the following example: <pre> EXITDEF(TYPE(LOGR) EXITNAME(HBOFLGX0) LOADLIB(<i>hlq</i>.SHBOLLST)) </pre> To activate the LOGWRT user exit, recycle the IMS system.
Stand-alone exit	<ol style="list-style-type: none"> Add the SHBOLLST library to the STEPLIB concatenation of the IMS Control Region instead of the version 1.1.0 dataset, and verify that the DFSFLGX0 module in this library is concatenated before any other module of the same name. After the IMS Control Region JCL is updated, recycle the IMS system to activate the LOGWRT user exit.

When the LOGWRT user exit initializes successfully, the following message is written to the z/OS console:

```

HB08101I CDP IMS LOGWRT EXIT ACTIVATED FOR IMSID=iii

```

UNIX System Services permission requirements change to some Log Forwarder scripts

The z/OS UNIX System Services permission requirements for some Log Forwarder scripts have changed in IBM Z Common Data Provider version 2.1.0 to provide better protection for your system.

The following scripts in the UNIX System Services directory `/usr/lpp/IBM/zcdp/v2R1M0/LF/samples` have ACLs of 755 so they can be started only by the owner of the file or the group with which the file is associated.

```
checkFilePattern.sh
encrypt.sh
manageUserExit.sh
startup.sh
```

Ensure that the UID of the started task user is the owner of the file or a member of the group.

Prefix change to the z/OS NetView message provider module and definitions

Starting from IBM Z Common Data Provider Version 2.1.0, the Log Forwarder component prefix is changed from GLA to HB0 to be consistent with other components. This change also applies to the z/OS NetView message provider module and its definitions in the CNMSTYLE member of NetView.

In Z Common Data Provider Version 1.1.0, the NetView message provider module is GLANETV in the SHBOCLST data set. The GLANETV module is placed in the DSICLD data set that is defined in the NetView procedure. There are definitions in the CNMSTYLE member that have a prefix of GLA.

In Z Common Data Provider Version 2.1.0, the NetView message provider module is changed to HBONETV, and the prefix of the definitions in the CNMSTYLE member is changed to HB0.

After IBM Z Common Data Provider Version 2.1.0 is installed by using SMP/E, follow the instructions in [“Configuring the z/OS NetView message provider for collecting NetView messages” on page 130](#) to set up the NetView message provider.

Chapter 5. Configuring Z Common Data Provider

To configure IBM Z Common Data Provider, you must set up the Configuration Tool, use the Configuration Tool to create your policies for streaming data, prepare the target destinations to receive data from the Data Streamer, configure the Data Streamer, and configure the primary data gatherer components, which are the Log Forwarder and the System Data Engine.

Getting started with the Configuration Tool

The IBM Z Common Data Provider Configuration Tool is a web-based user interface that is provided as an application for IBM WebSphere Application Server for z/OS Liberty, or as a plug-in for IBM z/OS Management Facility (z/OSMF). You use the tool to specify what data you want to collect from your z/OS system and where you want that data to be sent. This configuration information is contained in a policy.

About this task

The Configuration Tool helps you create and manage policies for streaming operational data to its destination. The Log Forwarder and System Data Engine need this policy information to know what data to collect. Data Streamer needs this policy information to know what to do with the data that it receives from the data gatherers (such as the System Data Engine and the Log Forwarder).

Each policy definition is stored on the host and secured by the System Authorization Facility (SAF) product that is protecting the system.

There are two ways to run the Configuration Tool, you must determine on which platform you want to run the Configuration Tool and follow corresponding instructions to set up the Configuration Tool:

- Run the Configuration Tool on Liberty.

The Configuration Tool on Liberty can be deployed on IBM WebSphere Application Server for z/OS Liberty version 19.0.0.6 or later. IBM Z Common Data Provider provides a copy of IBM WebSphere Application Server for z/OS Liberty version 19.0.0.6 that is licensed for use with IBM WebSphere Application Server for z/OS.

If you do not have IBM z/OS Management Facility configured and operational for other applications, run the Configuration Tool on Liberty. For more information, see [“Getting started with the Configuration Tool on Liberty” on page 22](#).

- Run the Configuration Tool on IBM z/OS Management Facility.

z/OSMF is shipped with z/OS, and z/OSMF includes the following software:

- z/OSMF server.
- WebSphere Liberty profile, which provides an application server runtime environment for z/OSMF.
- Set of optional, system management functions or plug-ins, which you can enable when you configure z/OSMF.
- Technologies for serving the web browser interface, such as JavaScript, Dojo, and Angular.

If you have IBM z/OS Management Facility configured and operational for other applications, run the Configuration Tool on IBM z/OS Management Facility. For more information, see [“Getting started with the Configuration Tool on z/OSMF” on page 29](#).

Getting started with the Configuration Tool on Liberty

The IBM Z Common Data Provider Configuration Tool can be deployed as an application to IBM WebSphere Application Server for z/OS Liberty.

Configuring user IDs, group IDs, and security product

You must create user IDs and group IDs with necessary permissions to run the IBM Z Common Data Provider Configuration Tool.

About this task

A default properties file `/usr/lpp/IBM/zcdp/v2r1m0/UI/LIB/cdpui.properties` is provided with default user IDs and group IDs to run the Configuration Tool. You can run the `defracf.cmd` script to change the default values. The new values are saved in `/var/cdp-uiconfig/cdpui.properties` for the `savingpolicy.sh` script to use in the next task. If you are using RACF as your SAF product, you can allow the script to run necessary RACF commands to create the IDs and permissions. If you do not use RACF, you can exit the script after verifying or changing the values and continue with the configuration.

To run the `defracf.cmd` script, you must be logged in to the z/OS system with a user ID that has the RACF SPECIAL authority.

Important: If this is not the first time you run the script and you are trying to change the user ID and group ID for the started task, before you run the script, you must delete the certificate authority, the certificate, and the keyring that were created last time.

Procedure

1. Run the following script under UNIX System Services to start verifying the default values or changing the values. Only the default z/OS shell is supported.

```
/usr/lpp/IBM/zcdp/v2r1m0/UI/LIB/defracf.cmd
```

2. If necessary, change the user IDs and group IDs to meet your requirements.

All user IDs and group IDs that you specify must be unique. If **AUTOID** is set to OFF, the UIDs and GIDs that are specified must be unique.

STC_USRID

The user ID that is assigned to the Configuration Tool server started task procedure. The default value is HBOSTCID.

Tip: If you want to change this user ID after it is created by running the `defracf.cmd` script, you must first delete the profiles `HBOCFGA.*` and `HBOCFGT.*`, then you can rerun the `defracf.cmd` script to change values. Otherwise, you will see the following messages when you rerun the `defracf.cmd` script, and you will not be able to start the Liberty server:

```
ICH10102I HBOCFGA.* ALREADY DEFINED TO CLASS STARTED.  
ICH10102I HBOCFGT.* ALREADY DEFINED TO CLASS STARTED.
```

STC_GROUP

The group that contains **STC_USRID**. The default value is HBOSTCGP.

AUTHORIZED_GROUP

The group that is granted the permission of logging in and using the Configuration Tool. The default value is HB0USRGP.

GUEST_USER

The user ID that is used by Liberty for accessing the Configuration Tool login page. The default value is HBOGUEST.

GUEST_GROUP

The group that contains **GUEST_USER**. The default value is HBOUNGRP.

AUTHORIZED_USER

The user ID that is granted the permission of logging in and using the Configuration Tool. The default value is HB0USER. You must specify an existing user for this parameter. If you don't specify any value for this parameter, no user is able to access the Configuration Tool. To allow a user to use the Configuration Tool, you must connect the user to the **AUTHORIZED_GROUP** as instructed in [“Allowing users to use the Configuration Tool”](#) on page 28.

AUTOID

Determines whether the UID and GID are automatically assigned. The default value is OFF, and you must set values for the following parameters. Make sure that the UIDs and GIDs that you specify meet the requirements of your environment. If the UIDs and GIDs are not accepted by your security product, the Configuration Tool cannot be installed successfully.

STC_USRID_UID

The UID for **STC_USRID**.

STC_GROUP_GID

The GID for **STC_GROUP**.

AUTHORIZED_GROUP_GID

The GID for **AUTHORIZED_GROUP**.

GUEST_USER_UID

The UID for **GUEST_USER**.

GUEST_GROUP_GID

The GID for **GUEST_GROUP**.

If automatic assignment of UID and GID is enabled on your environment, you can change the value of this parameter to ON to have required UIDs and GIDs automatically assigned by the system. In this case, skip the UID and GID parameters that are listed previously.

HOSTNAME

The host name of the system. The default value is the output of the `hostname` command for your system. Usually the format of the host name is `XXXX.XXX.XXX.XXX`.

3. When you are prompted to choose exit or go, if you are using RACF as your SAF product and you want the script to run RACF commands to create the IDs and permissions, enter G0. Otherwise, enter EXIT to end the script.
 - If you enter G0, check the output from the RACF commands in the `/var/cdp-uiconfig/defracf.log` file and verify that all commands are successfully issued by the script.
 - There should be no RACF error messages from the UNIX System Services issued to the terminal after the script finishes running.
 - If you see the messages ICH10006I, ICH06011I, and IRRD175I indicating that RACLISTED PROFILES must be refreshed before they are effective, and a message "All related RACLIST CLASS are refreshed successfully" after the script finishes running, it means that the RACLISTED PROFILES are refreshed by the script and are effective.
 - Message ICH10102I that says BBG.AUTHMOD.BBGZSAFM, and BBG.AUTHMOD.BBGZSAFM.SAFCRED are already defined, can be safely ignored. These profiles are shared with other Liberty Angel Servers, and they might be defined by a Liberty Angel Server that was created before.
 - If you enter EXIT, you must configure your security product by using the information that is saved in `/var/cdp-uiconfig/cdpui.properties`. If you are using RACF as your SAF product, you can use the commands in [“Configuring the security product by running commands”](#) on page 24. If you are not using RACF, you can use these commands to compose equivalent commands for your SAF product.

Tip: If the user ID of **AUTHORIZED_USER** is not found after you run the script, see the troubleshooting topic [“User ID of parameter AUTHORIZED_USER is not found”](#) on page 317 for solution.

Important: If you run the script again to change the user ID and group ID for the started task, you must first delete the certificate authority, the certificate, and the keyring that are created this time.

Configuring the security product by running commands

If you are using RACF as your SAF product and you do not want to run the `defracf.cmd` script, you can run RACF commands to create the user IDs and group IDs, and grant them necessary permissions.

About this task

In this task, the following default values of the parameters in the `cdpui.properties` file are used in the code samples.

```
STC_USRID = HBOSTCID
STC_GROUP = HBOSTCGP
AUTHORIZED_GROUP = HBOUSRGP
GUEST_USER = HBOGUEST
GUEST_GROUP = HBOUNGRP
AUTHORIZED_USER = HBOUSER
```

If you are not using default values, make sure to change the values in the samples to the values that you use.

Procedure

1. If any one of classes `STARTED`, `APPL`, `FACILITY`, `SERVER`, `EJBRROLE`, `DIGTCERT`, and `DIGTRING` are not active, run one or more of the following RACF commands to activate them.

```
SETROPTS RACLIST(STARTED) CLASSACT(STARTED)
SETROPTS CLASSACT(APPL)
SETROPTS CLASSACT(FACILITY)
SETROPTS CLASSACT(SERVER)
SETROPTS CLASSACT(EJBRROLE)
SETROPTS CLASSACT(DIGTCERT)
SETROPTS CLASSACT(DIGTRING)
```

2. Run the following commands to define the groups and users that are specified in the `cdpui.properties` file.

```
ADDGROUP HBOSTCGP OMVS(GID(3701))
ADDGROUP HBOUSRGP OMVS(GID(3702))
ADDGROUP HBOUNGRP OMVS(GID(3703))
ADDUSER HBOSTCID DFLTGRP(HBOSTCGP) OMVS(UID(2701) HOME(/u/hbostcid)
PROGRAM(/bin/sh)) NAME('CDP UI Server Started Task USERID')
NOPASSWORD NOOIDCARD
ADDUSER HBOGUEST RESTRICTED DFLTGRP(HBOUNGRP) OMVS(UID(2702))
NAME('CDPz Unauthenticated USERID') NOPASSWORD NOOIDCARD
```

3. Run the following command to allow a user to use the Configuration Tool.

```
CONNECT HBOUSER GROUP(HBOUSRGP)
```

4. Run the following commands to define resource profiles and grant permission to these resource profiles to the user and group for the Configuration Tool server started task procedure.

```
RDEF STARTED HBOCFG.* UACC(NONE) STDATA(USER(HBOSTCID)
GROUP(HBOSTCGP) PRIVILEGED(NO) TRUSTED(NO) TRACE(YES))
RDEF STARTED HBOCFG.* UACC(NONE) STDATA(USER(HBOSTCID)
GROUP(HBOSTCGP) PRIVILEGED(NO) TRUSTED(NO) TRACE(YES))
RDEFINE SERVER BBG.ANGEL.HBOCFG UACC(NONE)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM UACC(NONE)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.SAFCRED UACC(NONE)
PERMIT BBG.ANGEL.HBOCFG CLASS(SERVER) ACCESS(READ) ID(HBOSTCID)
PERMIT BBG.AUTHMOD.BBGZSAFM CLASS(SERVER) ACCESS(READ) ID(HBOSTCID)
PERMIT BBG.AUTHMOD.BBGZSAFM.SAFCRED CLASS(SERVER) ACCESS(READ)
ID(HBOSTCID)
RDEFINE APPL HBOCFG UACC(NONE)
RDEFINE SERVER BBG.SECPF.HBOCFG UACC(NONE)
PERMIT BBG.SECPF.HBOCFG CLASS(SERVER) ACCESS(READ) ID(HBOSTCID)
RDEFINE FACILITY BBG.SYNC.HBOCFG UACC(NONE)
PERMIT BBG.SYNC.HBOCFG CLASS(FACILITY) ID(HBOSTCID)
ACCESS(CONTROL)
RDEFINE EJBRROLE HBOCFG.CDPUIserver.cdpUser UACC(NONE)
PERMIT HBOCFG CLASS(APPL) ID(HBOSTCID) ACCESS(READ)
PERMIT HBOCFG CLASS(APPL) ID(HBOGUEST) ACCESS(READ)
```

```

PERMIT HBOCFGT CLASS(APPL) ID(HBOUSRGP) ACCESS(READ)
PERMIT HBOCFGT.CDPUIserver.cdpUser CLASS(EJBR0LE) ID(HBOUSRGP)
ACCESS(READ)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(HBOSTCID)
ACCESS(READ)
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('CDPz CA Certification_HOSTNAME'))
WITHLABEL('HBOCA') TRUST NOTAFTER(DATE(2023/12/31))
RACDCERT ID (HBOSTCID) GENCERT SUBJECTSDN(CN('HOSTNAME'))
WITHLABEL('HB0DefaultCert') SIGNWITH(CERTAUTH LABEL('HBOCA'))
NOTAFTER(DATE(2023/12/31))
RACDCERT ADDRING(HB0.Keyring.DFLT) ID(HBOSTCID)
RACDCERT ID(HBOSTCID) CONNECT (LABEL('HB0DefaultCert')
RING(HB0.Keyring.DFLT) DEFAULT)
RACDCERT ID(HBOSTCID) CONNECT (LABEL('HBOCA')
RING(HB0.Keyring.DFLT) CERTAUTH)

```

Important: Change *HOSTNAME* in the CN field to the actual local host name. Usually the format of the host name is XXXX.XXX.XXX.XXX.

5. If the sharing of in-storage profile is active for any one of classes STARTED, APPL, FACILITY, SERVER, EJBR0LE, DIGTCERT, and DIGTRING, run one or more of the following RACF commands to refresh them so that the changes to these classes take effect.

```

SETROPTS RACLIST(STARTED) REFRESH
SETROPTS RACLIST(SERVER) REFRESH
SETROPTS RACLIST(FACILITY) REFRESH
SETROPTS RACLIST(EJBR0LE) REFRESH
SETROPTS RACLIST(APPL) REFRESH
SETROPTS RACLIST(DIGTCERT, DIGTRING) REFRESH

```

Setting up a Liberty server directory and a working directory for the Configuration Tool

You must set up a Liberty server directory to contain the configuration of the Configuration Tool server, and a working directory to store the policy definition files. A setup script (`savingpolicy.sh`) is provided to automate this process.

About this task

User ID criteria for running the setup script

To run the setup script, you must be logged in to the z/OS system with a user ID that has the UID 0 attribute.

Procedure

1. Enter the directory that contains the setup script.

```
cd /usr/lpp/IBM/zcdp/v2r1m0/UI/LIB/
```

2. Run the following command to start the setup script. Only the default z/OS shell is supported.

```
savingpolicy.sh
```

Important: Before you run the `savingpolicy.sh` script, make sure that the `/tmp` directory has at least 1 MB free space.

3. Follow the prompts of the script to provide necessary values.

To accept the default value that is shown in the parentheses, enter a blank value.

- a) When you are prompted to choose if you are deploying the Configuration Tool on z/OSMF or Liberty, enter 2 to select Liberty.
- b) When you are prompted to specify the full path of directory where the Configuration Tool server is installed, accept the default value, or specify the directory that you want to use.

- The directory must be readable by the **STC_USRID** and **STC_GROUP** that are specified in the `/var/cdp-uiconfig/cdpui.properties` file that is created in [“Configuring user IDs, group IDs, and security product”](#) on page 22.
 - To avoid possible conflicts, do not use a directory that is defined as the Data Streamer working directory (`CDP_HOME`) or the Log Forwarder working directory (`ZLF_WORK`).
- c) If you specify an existing directory, you are prompted for confirmation because the files in the existing directory might be replaced. Enter `y` to continue with the directory, or enter `n` to specify another directory.
 - d) When you are prompted to specify the Java home directory, accept the default value, or specify the Java home directory for your system.
 - e) When you are prompted to specify the Configuration Tool Source Script Directory, accept the default value, or specify a directory that is allowed on your system.

Results

The directory `config_tool_server_install_dir/servers/cdp_ui_server` is created as the Liberty server directory, and the directory `config_tool_server_install_dir/cdpConfig` is created as the working directory, where `config_tool_server_install_dir` is the full path of directory where the Configuration Tool server is installed that you specify in [“3.b”](#) on page 25.

What to do next

The Configuration Tool server uses 17977 as the default port number. If you want to change the port number, complete the following steps:

1. Open the file `server.xml` under the Liberty server directory of the Configuration Tool. By default, you can find this file under `/var/local/CDPServer/servers/cdp_ui_server/`.
2. Change the value of the variable `httpsPort` to the port number you want to use.

Creating the started tasks of the Configuration Tool server and its angel server

Before you can start the IBM Z Common Data Provider Configuration Tool server, you must create the started tasks for the Configuration Tool server and its angel server by copying the sample procedures into a user procedure library, and updating the copies.

Procedure

To create the started tasks, complete the following steps:

1. Copy the procedure `HBOCFGT` from `hlq.SHBOSAMP` to a user procedure library.
2. Update the procedure `HBOCFGT`.
 - Change the value of the variable `INSTDIR` to the path where the WebSphere Application Server for z/OS Liberty is installed.
 - Change the value of the variable `USERDIR` to the path where the Configuration Tool server is installed. This value is specified in [Step 2.b in Setting up a Liberty server directory and a working directory for the Configuration Tool](#)
3. Copy the procedure `HBOCFGGA` from `hlq.SHBOSAMP` to a user procedure library.
4. Change the value of the variable `WLPDIR` in `HBOCFGGA` to the path where the WebSphere Application Server for z/OS Liberty is installed.
5. You can set `HBOCFGT` and `HBOCFGGA` as low-priority WLM service class.

Specifying security ciphers for the Configuration Tool server connections

You can specify security ciphers for the Configuration Tool server connections. This step is optional, if you do not need to specify security ciphers, you can skip this step.

About this task

By default, the security ciphers that the Configuration Tool server supports depends on the underlying JRE that is used. You can check the JRE for valid ciphers. For more information, see [Cipher suites](#).

Note: It is a best practice to use the latest and securest TLS ciphers in Java, and avoid using deprecated TLS ciphers.

To specify security ciphers for the Configuration Tool server connections, indicate the ciphers in an editable file called the `server_override.xml` file. The override file is read at the initialization of the Configuration Tool server, and the values override default values for the Configuration Tool configuration properties.

Procedure

To specify security ciphers for the Configuration Tool server connections, complete the following steps:

1. Create an override file that is named `server_override.xml` in `config_tool_server_install_dir/servers/cdp_ui_server`.
2. Set the permissions to 755 for the `server_override.xml` in `config_tool_server_install_dir/servers/cdp_ui_server`. For example,

```
chmod 755 config_tool_server_install_dir/servers/cdp_ui_server/server_override.xml
```

3. Add the following statements to the `server_override.xml` file.

```
<server>  
  <ssl id="DefaultSSLSettings" enabledCiphers="<cipher_1> <cipher_2> ... <cipher_n>"/>  
</server>
```

4. Replace `<cipher_1> <cipher_2> ... <cipher_n>` with a list of ciphers, and separate each cipher with a space. In the following example, two ciphers are specified.

```
<server>  
  <ssl id="DefaultSSLSettings" enabledCiphers="TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256  
  TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384"/>  
</server>
```

5. Restart the Configuration Tool server.

As a result, you can only use the ciphers that you specify in the `server_override.xml` file for the Configuration Tool server connections.

Starting the Configuration Tool server

Start the angel server and the Configuration Tool server before you can access the Configuration Tool in a web browser.

Procedure

1. Start the angel server for the Configuration Tool server by running the following z/OS system console command:

```
START HBOCFGA
```

2. Verify that the angel server starts successfully.

You see the following message in the job log if the server starts successfully:

```
CWWKB0069I INITIALIZATION IS COMPLETE FOR THE HBOCFGA ANGEL PROCESS
```

3. Start the Configuration Tool server by running the following console command:

```
START HBOCFG
```

4. Verify that the Configuration Tool server starts successfully.

You see the following messages in the job log if the server starts successfully:

```
CWWKF0011I: The server cdp_ui_server is ready to run a smarter planet.  
CWWKT0016I: Web application available (default_host): https://hostname:port/cdp/
```

5. Verify that the Configuration Tool server and the angel server are connected successfully.

In the log file `/var/local/CDPServer/servers/cdp_ui_server/logs/messages.log`, you can see the following message:

```
CWWKB0103I: Authorized service group KERNEL is available.
```

If the SAF authorized user registry services and SAF authorization services (SAFCRED) are enabled, the following message is in the log file:

```
CWWKB0103I: Authorized service group SAFCRED is available
```

Tip: The user ID that can read the file `messages.log` must belong to the **STC_GROUP** user group that is specified in the `/var/cdp-uiconfig/cdpui.properties` file that is created in [“Configuring user IDs, group IDs, and security product”](#) on page 22.

Accessing the Configuration Tool

After the Configuration Tool server and the angel server are started, you can access the Configuration Tool in a web browser.

Procedure

1. Access the following URL in a web browser:

```
https://HostName:port/cdp
```

HostName is the host name of your server that runs the Configuration Tool server. *port* is the port number that is used by the Configuration Tool server. The default port number is 17977.

2. Log in as the user ID which is connected to the user group that is defined by the **AUTHORIZED_GROUP** parameter in the `/var/cdp-uiconfig/cdpui.properties` file that is created in [“Configuring user IDs, group IDs, and security product”](#) on page 22.

Results

The **"Common Data Provider"** tab opens. Predefined policies are listed.

Allowing users to use the Configuration Tool

To allow a user to use the Configuration Tool, you must connect the user to the user group that is specified in the `cdpui.properties` file. By default, the group ID is HBOUSRGP.

About this task

RACF is used as an example in the following instructions. For other SAF products, run corresponding commands to connect the user to the user group.

Procedure

1. Run the following RACF command on the TSO command line as a user ID with RACF SPECIAL authority.


```
CONNECT HBOUSER GROUP(HBOUSRGP)
```

Change *HBOUSER* to the user ID that you want to allow to access the Configuration Tool. If you don't use the default *HBOUSRGP* group ID, change it to the value of the **AUTHORIZED_GROUP** parameter that is specified in the `/var/cdp-uiconfig/cdpui.properties` file.

2. Verify that the user ID is connected to the group by running the following RACF command on the TSO command line:

```
LISTUSER HBOUSER
```

Stopping the Configuration Tool server

Run z/OS system console commands to stop the Configuration Tool server and the angel server when necessary.

Procedure

1. Run the following console command to stop the Configuration Tool server:

```
STOP HBOCFGT
```

2. Run the following console command to stop the angel server for the Configuration Tool server:

```
STOP HBOCFGGA
```

Getting started with the Configuration Tool on z/OSMF

The IBM Z Common Data Provider Configuration Tool can be deployed as a plug-in for IBM z/OS Management Facility (z/OSMF).

Before you begin

For information about how to set up z/OSMF so that you can use the IBM Z Common Data Provider Configuration Tool, see the [z/OSMF setup documentation for IBM Common Data Provider for z Systems](#).

Setting up a working directory for the Configuration Tool

Before you install the IBM Z Common Data Provider Configuration Tool, you must set up a working directory where the tool can store policy definition files. A setup script (`savingpolicy.sh`) is provided to automate this process.

About this task

Guidelines for the working directory

Use the following guidelines to help you decide which directory to use as the working directory:

- The directory must be readable and writable by the user ID that runs the Configuration Tool.
- To avoid possible conflicts, do not use a directory that is defined as the Data Streamer working directory (`CDP_HOME`) or the Log Forwarder working directory (`ZLF_WORK`).
- The setup script prompts you for input. To accept the default directory that is shown in the prompt, enter a blank value. If the name for the z/OSMF administrator group of your site is not `IZUADMIN`, ensure that you specify the correct name.

User ID criteria for running the setup script

To run the setup script, you must be logged in to the z/OS system with a user ID that meets the following criteria:

- Because IBM z/OS Management Facility (z/OSMF) administrators need to write updates to the policy definition files, the user ID must be in z/OSMF administrator group 1, which is the UNIX group to which z/OSMF administrators are added. By default, z/OSMF administrator group 1 is IZUADMIN.
- The user ID must be a TSO ID that has the UID 0 attribute.

Procedure

To set up the working directory, run the following command with a user ID that meets the criteria that is specified in [About this task](#):

```
sh /usr/lpp/IBM/zcdp/v2r1m0/UI/LIB/savingpolicy.sh
```

Important: Before you run the `savingpolicy.sh` script, make sure that the `/tmp` directory has at least 1 MB free space.

The script creates the following path and file:

`/u/userid/cdpConfig/HBOCDEUI/v2r1m0/LIB` path

This path is a symbolic link to target libraries. The `cdpConfig.properties` file is in this path, and it must be imported into z/OSMF when you install the Configuration Tool. For more information about the installation steps, see [“Installing the Configuration Tool on z/OSMF” on page 30](#).

`/u/userid/cdpConfig/HBOCDEUI/v2r1m0/LIB/cdpConfig.json` file

This file includes the variable `configPath` that defines the working directory for the Configuration Tool.

Installing the Configuration Tool on z/OSMF

To install the IBM Z Common Data Provider Configuration Tool, you must log in to the IBM z/OS Management Facility (z/OSMF), and import the `cdpConfig.properties` file.

Before you begin

Before you install the Configuration Tool, the following tasks must be complete:

1. IBM Z Common Data Provider is installed, and all SMP/E tasks are complete.
2. z/OSMF is installed, configured, and running.

Tip: z/OSMF is running if the started tasks CFZCIM, IZUANG1, and IZUSVR1 are active.

3. The working directory for the Configuration Tool must be set up. For more information, see [“Setting up a working directory for the Configuration Tool” on page 29](#).

About this task

To install the Configuration Tool, you must be logged in to z/OSMF with a TSO user ID that is in z/OSMF administrator group 1, which is the UNIX group to which z/OSMF administrators are added. By default, z/OSMF administrator group 1 is IZUADMIN.

Procedure

To install the Configuration Tool, complete the following steps:

1. Open z/OSMF in a web browser, and log in with your TSO ID.

Tips:

- The web address, such as `https://host/zosmf/`, is dependent on the configuration of your z/OSMF installation.
- If you cannot log in to z/OSMF, verify that the z/OSMF WebSphere Liberty Profile server is started. The default procedure is IZUSVR1.

2. In the left navigation pane, expand **z/OSMF Administration**, and click **Import Manager**.

3. Type the path and name of the `cdpConfig.properties` file and click **Import**. The `cdpConfig.properties` file is in the working directory which is created after running the `savingpolicy.sh` script. By default, the path is `/u/userid/cdpConfig/HBOCDEUI/v2r1m0/LIB`.

The import can take several seconds. When it is complete, the following message is shown:

```
Plug-in "Common Data Provider" with tasks
"Common Data Provider" was added to z/OSMF.
To control access, define SAF resource profiles in the
"ZMFAPLA" class for the following SAF resources:
"IZUDFLT.ZOSMF.IBM_CDP.CONFIG.CDPConfiguration".
```

Tip: If you click this resulting message, you are directed to documentation that your systems programmer might need to grant permissions for accessing the plug-in, as indicated in the next step.

4. Have your systems programmer run a SAF command to grant permissions for the z/OSMF administrator group 1 (default is IZUADMIN group) to access the plug-in. For example, if you are using RACF, run the following command:

```
RDEFINE ZMFAPLA IZUDFLT.ZOSMF.IBM_CDP.CONFIG.CDPConfiguration) UACC(NONE)
PERMIT IZUDFLT.ZOSMF.IBM_CDP.CONFIG.CDPConfiguration CLASS(ZMFAPLA) ID(IZUADMIN)
ACCESS(CONTROL)
PERMIT IZUDFLT.ZOSMF.IBM_CDP.CONFIG.CDPConfiguration CLASS(ZMFAPLA) ID(IZUUSER) ACCESS(READ)
SETROPTS RACLIST(ZMFAPLA) REFRESH
```

Running the Configuration Tool

To run the IBM Z Common Data Provider Configuration Tool, you must log in to the IBM z/OS Management Facility (z/OSMF).

Before you begin

Install the Configuration Tool, as described in [“Installing the Configuration Tool on z/OSMF”](#) on page 30.

About this task

To run the Configuration Tool, you must be logged in to z/OSMF with a TSO user ID that is in z/OSMF administrator group 1, which is the UNIX group to which z/OSMF administrators are added. By default, z/OSMF administrator group 1 is IZUADMIN.

Procedure

1. Open z/OSMF in a web browser, and log in with your TSO ID.
2. In the left navigation pane, expand **Configuration**, and click **Common Data Provider**.

The Common Data Provider tab opens, and you can manage your policies for streaming z/OS operational data to subscribers.

Uninstalling the Configuration Tool from z/OSMF

To uninstall the IBM Z Common Data Provider Configuration Tool, you must log in to the IBM z/OS Management Facility (z/OSMF), and import the `cdpConfig.remove.properties` file.

About this task

To uninstall the Configuration Tool, you must be logged in to z/OSMF with a TSO user ID that is in z/OSMF administrator group 1, which is the UNIX group to which z/OSMF administrators are added. By default, z/OSMF administrator group 1 is IZUADMIN.

Procedure

To uninstall the plug-in, complete the following steps:

1. Open z/OSMF in a web browser, and log in with your TSO ID.
2. In the left navigation pane, expand **z/OSMF Administration**, and click **Import Manager**.
3. Type the path and name of the `cdpConfig.remove.properties` file, which is in the path `/u/userid/cdpConfig/HBOCDEUI/v2r1m0/LIB`, and click **Import**.

Results

The Configuration Tool is removed after you import the `cdpConfig.remove.properties` file. There is no confirmation message for the removal.

Output from the Configuration Tool

For each policy that you save, the IBM Z Common Data Provider Configuration Tool creates several files in its working directory.

For example, if you create and save a policy that is named `Sample1`, the following files are created in the Configuration Tool working directory:

- `Sample1.policy` if one or more SMF data streams are selected
- `Sample1.layout`
- `Sample1.sde`
- `Sample1.zlf.conf`
- `Sample1.config.properties` if logs (like SYSLOG and OPERLOG) that are gathered by the Log Forwarder are selected
- `cdpkey`

The `cdpkey` file generated only once for each Configuration Tool instance, not for each policy.

Important: Do not edit the files that the Configuration Tool creates.

The following descriptions explain the purpose of each output file, based on the file name extension:

.policy file

Contains configuration information for the Data Streamer.

.layout file

Contains information about how a policy definition is visually presented in the Configuration Tool.

.sde file

Contains configuration information for the System Data Engine.

.zlf.conf file

Contains environment variables for the Log Forwarder.

.config.properties file

Contains configuration information for the Log Forwarder.

cdpkey file

Contains the key for encrypting the HEC token if used in a policy. This file must be in the same directory as the policy files, which means if you copy the policy files to other directories, you must copy the `cdpkey` file together.

Managing policies

From the IBM Z Common Data Provider Configuration Tool, you can manage (for example, create, update, and search for) your policies for streaming z/OS operational data to subscribers.

Before you begin

For information about how to run the Configuration Tool, see [“Running the Configuration Tool” on page 31](#).

About this task

A policy includes the following components, which are shown in the Configuration Tool as interconnected nodes in a graph so that you can more easily see how data flows through your system:

Data streams

You must define a *data stream* for each source (such as SMF record type 30 or z/OS SYSLOG) from which you want to collect operational data.

Transforms

In a transform, you specify how to alter the operational data in the stream so that the data is consumable at its target destination. For example, you can use the FixedLength Splitter transform to split data records that have a fixed record length into multiple messages, based on configuration values that you provide.

Subscribers

You must define the subscriber or subscribers for each data stream. A subscriber can be subscribed to multiple data streams. A data stream can be the source of multiple subscribers.

Tip: Only one policy can be active in each logical partition (LPAR).

Purpose of transforming data in a policy

In a transform, you specify how to alter the operational data in the stream so that the data is consumable at its target destination. The two categories of transform are splitter transforms and filter transforms.

Split and unsplit data

The Data Streamer accepts data in two formats, split and unsplit.

Split data

Split data is divided into records and is sent as an ordered list of individual text strings. Each individual text string represents an individual record from the data source.

Unsplit data

Unsplit data is sent as a single text string and is not divided into records.

Overview of transform categories

The following information provides a brief overview of each transform category and some tips about when to use a transform from that category. Each category includes one or more transforms, which are listed and described in [“Transform configuration”](#) on page 229.

Splitter transforms

Based on specified criteria, a splitter transform splits data that is received as one message into multiple messages. Splitter transforms are available only for Generic z/OS Job Output data stream, Generic ZFS File data stream, Generic VSAM Cluster data stream, and, if you have IBM Z Operations Analytics, zSecure Access Monitor,

Usage tips

- For splitting a single message into multiple messages, based on occurrences of carriage return (CR) or line feed (LF) characters, use the [“CRLF Splitter transform”](#) on page 229.
- For splitting data records that have a fixed record length into multiple messages, based on configuration values that you provide, use the [“FixedLength Splitter transform”](#) on page 230.

Filter transforms

Based on specified criteria, a filter transform discards messages from the data stream.

Usage tips

- For filtering messages in the data stream according to a regular expression (regex) pattern, which you can define, use the [“Regex Filter transform”](#) on page 231.

- For filtering messages in the data stream according to a specified schedule, which you can define, use the [“Time Filter transform”](#) on page 234.

For filtering SMF or IMS data: For information about how to filter System Management Facilities (SMF) or IBM Information Management System (IMS) data, see [“Filtering a System Data Engine data stream by using a data stream definition”](#) on page 54.

Subscribers to a data stream or transform

A subscriber defines a destination that IBM Z Common Data Provider sends operational data to. A subscriber can be an analytics platform like Splunk, or intermediary software (such as Logstash and the Data Receiver) that is configured to send data to its target destination, which can include analytics platforms such as IBM Z Operations Analytics, Splunk, Elasticsearch, and others.

Streaming protocols for sending data from the Data Streamer to its subscribers

If the subscriber is Logstash or the Data Receiver, the Data Streamer uses a persistent TCP socket connection to send data to the subscriber.

If the subscriber is the Splunk HTTP Event Collector or a generic HTTP or HTTPS subscriber, the Data Streamer uses HTTP or HTTPS to send data to the subscriber.

When you configure a subscriber in a policy, the streaming protocol that you select in the configuration (in the **Protocol** field) defines the following characteristics for sending the data:

- Whether the data will be used by IBM Z Operations Analytics
- The destination software (such as Splunk, Elasticsearch, IBM Operations Analytics - Log Analysis, and so on)
- The intermediary software that the data will pass through (such as Logstash, Data Receiver, and so on)
- The communications protocol, such as a persistent TCP socket connection, HTTP, or HTTPS
- Whether communications are secured by using Transport Layer Security (TLS)

For more information about the streaming protocols, see [“Subscriber configuration”](#) on page 235.

IBM Operations Analytics - Log Analysis

IBM Operations Analytics - Log Analysis provides a semi-structured data analytics solution. It can be used to reduce problem diagnosis and resolution time, helping you to manage your infrastructure and applications more effectively. It can also help you to identify problems and propose solutions.

Splunk

Splunk Enterprise and Splunk Cloud enable you to search, analyze, and visualize the data gathered from the components of your IT infrastructure or business.

Humio

Humio is powerful and extremely useful for system administrators. It provides a fast, but flexible platform for logs and server metric.

Elasticsearch

Elasticsearch is a distributed, open source search and analytics engine for all types of data, including textual, numerical, geospatial, structured, and unstructured.

Kafka

Kafka is an open-source distributed event streaming platform.

Logstash

Logstash is an open source data collection engine that in near real-time, can dynamically unify data from disparate sources and normalize the data into the destinations of your choice for analysis and visualization.

Because IBM Z Common Data Provider does not support forwarding data directly to IBM Z Operations Analytics, for example, the z/OS log data and SMF data that is gathered by IBM Z Common Data Provider must be forwarded to a Logstash instance. The Logstash instance then forwards the data to IBM Z Operations Analytics. Through Logstash, you can also forward data to other destinations, such as Elasticsearch, or route data through a message broker, such as Apache Kafka.

For more information about Logstash, see the [Logstash documentation](#).

Data Receiver

The IBM Z Common Data Provider Data Receiver is software that runs on any platform supporting Java and acts as a target subscriber when the intended subscriber of a data stream cannot directly ingest the data feed from IBM Z Common Data Provider. The Data Receiver writes any data that it receives to disk into files that are sorted by the data source type. These files can then be ingested by an analytics platform.

The Data Receiver protocol is typically used to send data to Splunk. In this case, Data Receiver should run on a platform that can be accessed using conventional Splunk capabilities.

Splunk HTTP Event Collector (HEC)

HEC is an HTTP API endpoint that allows you to send data directly to Splunk over HTTP or HTTPS. For more information about HEC, see [Splunk documentation](#).

If this feature is enabled on Splunk, IBM Z Common Data Provider is able to send data directly to Splunk via HEC instead of sending data through the Data Receiver.

Enabling HEC on Splunk generates a token value for communicating with Splunk. You must provide this value in the subscriber settings when you create a policy.

Generic HTTP subscriber

A generic HTTP subscriber is an application that can receive data that is sent by using an HTTP POST request. An example of a generic HTTP subscriber is a Logstash event processing pipeline that is configured with an HTTP input that uses the JavaScript Object Notation (JSON) codec.

The body of the HTTP POST request is a JSON object with the following members:

host

The TCP/IP host name of the system from which the data originates.

path

The virtual path of the data, which can be a real file path, such as `/home/etc/cdpConfig.log` or a virtual file path, such as `SMF/SMF_030`.

sourceType

The type of the data.

sourceName

A name that is associated with the data stream.

timezone

The base time zone offset for the timestamp in the data.

systemName

The name of the system from which the data originates, as it is defined to the z/OS system.

sysplexName

The name of the sysplex from which the data originates, as it is defined to the z/OS system.

message

The data itself. Depending on whether the data was split, this value might be one string or an array of strings.

Creating a policy

From the Common Data Provider tab in the IBM Z Common Data Provider Configuration Tool, you can create a policy.

About this task

In a policy, you must define at least one data stream with at least one subscriber. For each data stream, you can also define one or more transforms and multiple subscribers. A subscriber can also be subscribed to multiple data streams and transforms.

[“Icons on each node in a policy” on page 180](#) describes the icons that are shown on each data stream, transform, and subscriber node that you define in a policy.

Procedure

Watch this interactive video [How to create a policy in the Configuration Tool](#) to understand the general procedures of creating a policy.

You can also see the following instructions:

1. Click the **Create a new policy** box.
2. In the resulting **Policy Profile Edit** window, type or update the required policy name and, optionally, a policy description.
3. Define any global properties, which are properties that apply to all data streams in the policy.
For information about what you can define, see [“Global properties that you can define for all data streams in a policy” on page 153](#).

4. Click the **Add Data Stream** icon .

The **"Select data stream"** window is shown with a list of categorized data streams. You can expand the categories to view the possible data streams that you can define for this policy.





The data stream group **Starter Sets** includes commonly used data streams. These data streams are categorized into various data stream units based on some z/OS components and subsystems instead of based on data types. For more information about **Starter Sets** and other data stream groups, see [“Groups of data streams in the Configuration Tool” on page 157](#).

Tip: If you create any custom data stream definitions, the associated data streams are listed under the groups that you specified in the stream definitions. For more information about these definitions, see the following information:


- [“Creating a System Data Engine data stream definition” on page 46](#)
- [“Creating an application data stream definition” on page 94](#)

5. Select one or more data streams and click **Select**.

After you click **Select**, each data stream that you choose is shown as a node in the graph. Each node includes the icons that are described in [“Icons on each node in a policy” on page 180](#).

6. Depending on what you want to define in the policy, use the **Configure** , **Transform** , and **Subscribe**  icons on each node to complete the policy definition. If you want to add more data streams to the policy, use the **Add Data Stream** icon .

Tips:

- “Icons on each node in a policy” on page 180 indicates where you can find more information about configuring data streams, transforms, and subscribers, including information about the configuration values.
 - To remove a data stream, transform, or subscriber node from a policy definition, click the **Remove** icon (X mark) on the node. When you remove a data stream node or a transform node, any connected transform nodes are also removed.
 - If you select Generic z/OS Job Output, Generic ZFS File, Generic VSAM Cluster, or, if you have IBM Z Operations Analytics, zSecure Access Monitor data streams, you must click the **Transform**  icon on a data stream node, and configure **CRLF Splitter** transform or **FixedLength Splitter** transform. For more information about transforms, see [Transform configuration](#).
7. To save the policy, click **Save**.
- The box for the new policy is then shown on the page.

Creating a policy to stream SMF data





From IBM Z Common Data Provider Configuration Tool, you can create a policy to stream SMF data to various subscribers.

About this task

In the policy, select one or more SMF data streams, and add at least one subscriber.

For detailed steps about creating a policy to stream SMF data, refer to the following procedure.

Procedure

1. From the Configuration Tool, click the **Create a new policy** box.
2. In the resulting **Policy Profile Edit** window, type the required policy name and, optionally, a policy description.
3. Click the **Add Data Stream** icon  **DATA STREAM**.
The "**Select data stream**" window is shown with a list of categorized data streams. You can expand the categories to view the possible data streams that you can define for this policy.
4. Select one or more SMF data streams and click **Select**. Each data stream that you choose is shown as a node in the graph.
5. If you want to alter the operational data in the stream, click the **Transform**  icon on a data stream node.
For more information about transforms, see [Transform configuration](#).
6. Click the **Subscribe**  icon on a data stream node, the **Policy Profile Edit** window opens where you can select a previously defined subscriber, or define a new subscriber by completing the following steps.
 - a) In the "**Subscribe to a data stream**" or "**Subscribe to a transform**" window, click the **Add Subscriber** icon  **ADD SUBSCRIBER**.
 - b) In the resulting "**Add subscriber**" window, update the associated configuration values, and click **OK** to save the subscriber.

You can update the following values in the "**Add subscriber**" window:

Name

The name of the subscriber.

Description

An optional description for the subscriber.

Protocol

The streaming protocol that the Data Streamer uses to send data to the subscriber. For example, you can choose **CDP Elasticsearch via Logstash** if you want to use Elastic Stack as a subscriber, or you can choose **CDP Splunk via Data Receiver** if you want to use Splunk as a subscriber. Make sure you choose the protocol that meets your requirements. For more information about protocols, see [“Subscriber configuration”](#) on page 235.

Host

The host name or IP address of the subscriber.

Port

The port on which the subscriber listens for data from the Data Streamer.

Auto-Qualify

A specification of whether to prepend system names and sysplex names to data source names in the data streams that are sent to the subscriber. The data source name is the value of the **dataSourceName** field in the data stream configuration.

If you use the same policy file for multiple systems within one sysplex, the data source names must be unique across all systems in that sysplex. If you use the same policy file for multiple sysplexes, the data source names must be unique across all systems in all sysplexes. You can use this field to fully qualify these data source names.

You can choose any of the following values. The default value is None.

None

Indicates that the data source name from the **dataSourceName** field in the data stream configuration is used.

System

Specifies that the system name and the data source name are used in the following format:

```
systemName-dataSourceName
```

systemName represents the name of the system on which the IBM Z Common Data Provider runs.

If you use the same policy file for multiple systems within one sysplex, you might want to use the System value.

Sysplex

Specifies that the sysplex name, system name, and data source name are used in the following format:

```
sysplexName-systemName-dataSourceName
```

systemName represents the name of the system on which the IBM Z Common Data Provider runs. *sysplexName* represents the name of the sysplex in which the IBM Z Common Data Provider runs.

If you use the same policy file for multiple sysplexes, you might want to use the Sysplex value.

For more information about the **dataSourceName** field in the data stream configuration, see the following topics:

- [“Data stream configuration for data gathered by Log Forwarder”](#) on page 181
- [“Data stream configuration for data gathered by System Data Engine”](#) on page 228

7. In the **"Subscribe to a data stream"** or **"Subscribe to a transform"** window, select one or more subscribers, and click **Update Subscriptions**.

The subscribers that you choose are then shown on the graph.

8. Click the **SYSTEM DATA ENGINE** button to set the configuration values for your System Data Engine environment.

a) In the Global Properties section of the **Policy Profile Edit** window, click **SYSTEM DATA ENGINE**.

- b) In the "**Configure System Data Engine properties**" window, update the following configuration values for your environment, and click **OK**.

CDP Concatenation

This value must be the name of the SHBODEFS data set that is installed with IBM Z Common Data Provider in your environment. This data set is also referenced in the `concats.json` file, which is in the working directory for the IBM Z Common Data Provider Configuration Tool.

9. To save the policy, click **Save**.





Creating a policy to stream log data

From IBM Z Common Data Provider Configuration Tool, you can create a policy to stream log data to various subscribers.

About this task

In the policy, select one or more log data, and add at least one subscriber. For example, you can complete the following steps to create a policy to stream SYSLOG or job log data.

Procedure

1. From the Configuration Tool, click the **Create a new policy** box.
2. In the resulting **Policy Profile Edit** window, type the required policy name and, optionally, a policy description.
3. Click the **Add Data Stream** icon  **DATA STREAM** .
The "**Select data stream**" window is shown with a list of categorized data streams. You can expand the categories to view the possible data streams that you can define for this policy.
4. Select one or more SYSLOG or job log data streams and click **Select**. Each data stream that you choose is shown as a node in the graph.
5. If you want to alter the operational data in the stream, click the **Transform**  icon on a data stream node.
For more information about transforms, refer to [Transform configuration](#).
6. Click the **Subscribe**  icon on a data stream node, the **Policy Profile Edit** window opens where you can select a previously defined subscriber, or define a new subscriber to include in the selection list.
To define a new subscriber for a data stream or transform node, complete the following steps:
 - a) In the "**Subscribe to a data stream**" or "**Subscribe to a transform**" window, click the **Add Subscriber** icon  **ADD SUBSCRIBER** .
 - b) In the resulting "**Add subscriber**" window, update the associated configuration values, and click **OK** to save the subscriber.

You can update the following values in the "**Add subscriber**" window:

Name

The name of the subscriber.

Description

An optional description for the subscriber.

Protocol

The streaming protocol that the Data Streamer uses to send data to the subscriber. For example, you can choose **CDP Elasticsearch via Logstash** if you want to use Elastic Stack as a subscriber, or you can choose **CDP Splunk via Data Receiver** if you want to use Splunk as a subscriber. Make sure you choose the protocol that meets your requirements. For more information about protocols, see "[Subscriber configuration](#)" on page 235.

Host

The host name or IP address of the subscriber.

Port

The port on which the subscriber listens for data from the Data Streamer.

Auto-Qualify

A specification of whether to prepend system names and sysplex names to data source names in the data streams that are sent to the subscriber. The data source name is the value of the **dataSourceName** field in the data stream configuration.

If you use the same policy file for multiple systems within one sysplex, the data source names must be unique across all systems in that sysplex. If you use the same policy file for multiple sysplexes, the data source names must be unique across all systems in all sysplexes. You can use this field to fully qualify these data source names.

You can choose any of the following values. The default value is None.

None

Indicates that the data source name from the **dataSourceName** field in the data stream configuration is used.

System

Specifies that the system name and the data source name are used in the following format:

```
systemName-dataSourceName
```

systemName represents the name of the system on which the IBM Z Common Data Provider runs.

If you use the same policy file for multiple systems within one sysplex, you might want to use the System value.

Sysplex

Specifies that the sysplex name, system name, and data source name are used in the following format:

```
sysplexName-systemName-dataSourceName
```

systemName represents the name of the system on which the IBM Z Common Data Provider runs. *sysplexName* represents the name of the sysplex in which the IBM Z Common Data Provider runs.

If you use the same policy file for multiple sysplexes, you might want to use the Sysplex value.

For more information about the **dataSourceName** field in the data stream configuration, see the following topics:

- [“Data stream configuration for data gathered by Log Forwarder” on page 181](#)
- [“Data stream configuration for data gathered by System Data Engine” on page 228](#)

7. In the "**Subscribe to a data stream**" or "**Subscribe to a transform**" window, select one or more subscribers, and click **Update Subscriptions**.

The subscribers that you choose are then shown on the graph.

8. Click the **LOG FORWARDER** button to set the configuration values for your Log Forwarder environment.

For detailed steps to set the configuration values, see [“Log Forwarder properties configuration” on page 154](#).

9. To save the policy, click **Save**.





Creating a policy to stream custom System Data Engine data streams

From IBM Z Common Data Provider Configuration Tool, you can create a policy to stream one or more custom System Data Engine data streams to various subscribers.

About this task

In the policy, select one or more custom System Data Engine data streams, and add at least one subscriber. The steps are similar to creating a policy for streaming SMF data.

Procedure

1. From the Configuration Tool, click the **Create a new policy** box.
2. In the resulting **Policy Profile Edit** window, type the required policy name and, optionally, a policy description.
3. Click the **Add Data Stream** icon  **DATA STREAM**.
The "**Select data stream**" window is shown with a list of categorized data streams. You can find the custom data streams under the **Custom Data Streams** group.
4. Select one or more custom data streams and click **Select**. Each data stream that you choose is shown as a node in the graph.
5. If you want to alter the operational data in the stream, click the **Transform**  icon on a data stream node.
For more information about transforms, see [Transform configuration](#).
6. Click the **Subscribe**  icon on a data stream node, the **Policy Profile Edit** window opens where you can select a previously defined subscriber, or define a new subscriber by completing the following steps.
 - a) In the "**Subscribe to a data stream**" or "**Subscribe to a transform**" window, click the **Add Subscriber** icon  **ADD SUBSCRIBER**.
 - b) In the resulting "**Add subscriber**" window, update the associated configuration values, and click **OK** to save the subscriber.

You can update the following values in the "**Add subscriber**" window:

Name

The name of the subscriber.

Description

An optional description for the subscriber.

Protocol

The streaming protocol that the Data Streamer uses to send data to the subscriber. For example, you can choose **CDP Elasticsearch via Logstash** if you want to use Elastic Stack as a subscriber, or you can choose **CDP Splunk via Data Receiver** if you want to use Splunk as a subscriber. Make sure you choose the protocol that meets your requirements. For more information about protocols, see "[Subscriber configuration](#)" on page 235.

Host

The host name or IP address of the subscriber.

Port

The port on which the subscriber listens for data from the Data Streamer.

Auto-Qualify

A specification of whether to prepend system names and sysplex names to data source names in the data streams that are sent to the subscriber. The data source name is the value of the **dataSourceName** field in the data stream configuration.

If you use the same policy file for multiple systems within one sysplex, the data source names must be unique across all systems in that sysplex. If you use the same policy file for multiple sysplexes, the data source names must be unique across all systems in all sysplexes. You can use this field to fully qualify these data source names.

You can choose any of the following values. The default value is None.

None

Indicates that the data source name from the **dataSourceName** field in the data stream configuration is used.

System

Specifies that the system name and the data source name are used in the following format:

```
systemName-dataSourceName
```

systemName represents the name of the system on which the IBM Z Common Data Provider runs.

If you use the same policy file for multiple systems within one sysplex, you might want to use the System value.

Sysplex

Specifies that the sysplex name, system name, and data source name are used in the following format:

```
sysplexName-systemName-dataSourceName
```

systemName represents the name of the system on which the IBM Z Common Data Provider runs. *sysplexName* represents the name of the sysplex in which the IBM Z Common Data Provider runs.

If you use the same policy file for multiple sysplexes, you might want to use the Sysplex value.

For more information about the **dataSourceName** field in the data stream configuration, see the following topics:

- [“Data stream configuration for data gathered by Log Forwarder” on page 181](#)
- [“Data stream configuration for data gathered by System Data Engine” on page 228](#)

7. In the "**Subscribe to a data stream**" or "**Subscribe to a transform**" window, select one or more subscribers, and click **Update Subscriptions**.

The subscribers that you choose are then shown on the graph.

8. Click the **SYSTEM DATA ENGINE** button to set the configuration values for your System Data Engine environment.

- a) In the Global Properties section of the **Policy Profile Edit** window, click **SYSTEM DATA ENGINE**.
- b) In the "**Configure System Data Engine properties**" window, update the following configuration values for your environment, and click **OK**.

USER Concatenation

This field appears only if you are using custom System Data Engine data streams. It is required as part of enabling the Configuration Tool to support custom System Data Engine data streams. For more information, [“Creating a System Data Engine data stream definition” on page 46](#).

The value must be the name of the USERDEFS data set that contains the custom System Data Engine definitions. This data set is also referenced in the `concat.s.json` file, which is in the working directory for the IBM Z Common Data Provider Configuration Tool.

CDP Concatenation

This value must be the name of the SHB0DEFS data set that is installed with IBM Z Common Data Provider in your environment. This data set is also referenced in the `concat.s.json` file, which is in the working directory for the IBM Z Common Data Provider Configuration Tool.

- 9. To save the policy, click **Save**.

Updating a policy

From the Common Data Provider tab in the IBM Z Common Data Provider Configuration Tool, you can update a policy, which can include editing, renaming, duplicating, or deleting the policy.

Procedure

Complete one or more of the following steps, depending on what you want to do:

- To edit a policy, click the box that has the policy name, make changes in the resulting **Policy Profile Edit** window, and click **Save** to save your changes.

Important: Starting from 2Q 2019 PTF for IBM Z Common Data Provider version 1.1.0, when you edit policies that were created before you apply this PTF, consider the following changes:

- Only the following transforms are available.

- CRLF Splitter transform
- FixedLength Splitter transform
- Regex Filter transform
- Time Filter transform




The transcribe transform nodes and their related links will be removed from the UI. The subscribers and other transform nodes that used to link to the transcribe transform will be re-connected to the data stream node directly. Other existing transforms are cleared after you save the changes to the policy.

- The protocols in the subscribers are automatically changed according to the following rules:

Old protocol	New protocol
CDP Logstash with the Send as value <code>unsplit</code>	IZOA on IOA-LA via Logstash
CDP Logstash with the Send as value <code>split</code>	IZOA on Elasticsearch via Logstash
CDP Data Receiver	IZOA on Splunk via Data Receiver

Tip: If you create any custom data stream definitions, the associated data streams are listed under the groups that you specified in the stream definitions. For more information about these definitions, see the following information:

- [“Creating a System Data Engine data stream definition” on page 46](#)
- [“Creating an application data stream definition” on page 94](#)

- To rename a policy, click the **Rename** icon  on the box for the policy.
- To duplicate a policy, click the **Duplicate** icon  on the box for the policy.
- To delete a policy, click the **Delete** icon  on the box for the policy.



Tip: [“Output from the Configuration Tool” on page 32](#) describes the files that the Configuration Tool creates for each policy that you save. For recovery purposes, when you delete a policy, the file extension `.hidden` is appended as a suffix to the standard file extension of all the associated policy files. To recover a deleted policy, rename each of the associated policy files to remove `.hidden` from the file extension.

Migrating a policy

If maintenance or a new release of IBM Z Common Data Provider requires additional or different information in the policy files, you must migrate the policies by using the IBM Z Common Data Provider Configuration Tool before you restart IBM Z Common Data Provider. Each policy should only be migrated when the maintenance or new release is applied to the LPAR (or LPARs).


About this task

After you install the PTF and start the Configuration Tool, the **MIGRATE ALL POLICIES TO THE LATEST**

FORMAT button appears, and a migrate button  and a warning icon  appears on each box of the policy that needs migration.

Procedure


Depending on how you want to migrate the policies, complete one of the following steps:

- To migrate all policies, click the **MIGRATE ALL POLICIES TO THE LATEST FORMAT** button.
- To migrate a specific policy, click the migrate button  in the lower-left of each policy box.
- For more information about migrating policies in the process of upgrading the product to a later version, see [Chapter 4, “Upgrading Z Common Data Provider,” on page 15.](#)

Results

If a policy is migrated, the warning icon  and the migrate button  disappear from its policy box. If all policies are migrated, the **MIGRATE ALL POLICIES TO THE LATEST FORMAT** button disappears.

Adding a subscriber for a data stream or transform


When you click the **Subscribe** icon  on a data stream or transform node, a window opens where you can select a previously defined subscriber, or define a new subscriber to include in the selection list. This procedure focuses on how to define a new subscriber.

Before you begin

For information about the types of subscribers that you can choose, see [“Subscribers to a data stream or transform” on page 34.](#)


Procedure

To define a new subscriber for a data stream or transform node, complete the following steps:

1. In the "**Subscribe to a data stream**" or "**Subscribe to a transform**" window, click the **Add Subscriber** icon  **ADD SUBSCRIBER**.
2. In the resulting "**Add subscriber**" window, update the associated configuration values, and click **OK** to save the subscriber.
For more information about the configuration values, see [“Subscriber configuration” on page 235.](#)
3. In the "**Subscribe to a data stream**" or "**Subscribe to a transform**" window, select one or more subscribers, and click **Update Subscriptions**.

The subscribers that you chose are then shown on the graph.

Updating subscriptions of a subscriber


When you click the **Subscribe** icon  on a subscriber node, a window opens where you can update the subscriptions.

Before you begin

For information about how to define a subscriber, see [“Adding a subscriber for a data stream or transform”](#) on page 44.

Procedure

To update subscriptions of a subscriber, complete the following steps:

1. In the "**Policy Profile Edit**" window, click the **Subscribe** icon  on a subscriber node.
2. In the resulting "**Select streams for subscription**" window, select or deselect data streams from the list.
The data streams that are invalid for the subscriber can not be selected.
3. Click **UPDATE SUBSCRIPTIONS** for the changes to take effect.

Results

The connection between data streams and the subscriber is shown in the "**Policy Profile Edit**" window.

Exporting and importing subscribers


You can export a subscriber with all transforms and data streams to which it is subscribed. The resulting subscriber file can then be imported into another policy.

About this task

Each subscriber node includes the **Export** icon  that you can use to export the subscriber and its associated data stream and transform nodes.

Procedure

To export a subscriber and import it into another policy, complete the following steps:

1. On the subscriber node, click the **Export** icon .

In the resulting **Export** window, the following check boxes are shown:

Omit layout

By default, an exported subscriber file includes information about how a policy definition is visually presented in the Configuration Tool. This layout information is used to reproduce the positioning of the subscriber node and its associated data stream and transform nodes.

If you do not want to save the layout information, select this check box to omit it. A new layout is then generated when the subscriber is imported.

Export as template

By default, an exported subscriber file includes configuration information that was previously provided (for example, ports, IP addresses, and user names).

If you do not want to save this configuration information, select this check box to export the subscriber file as a template in which all configurable information is reset to the default values.

2. After you optionally select one or both check boxes, click **Export** to download the policy information, which is in a file with the extension `.subscriber`.

The exported policy information in the `.subscriber` file can then be imported into another policy.

3. To import the preconfigured subscriber into a policy, complete either of the following actions:

- Drag and drop the `.subscriber` file onto the policy graph.
- Click the **Import** button at the top of the **Policy Profile Edit** window to browse for the `.subscriber` file to import.

The existing and imported graphs are then merged.

Managing custom data streams

From the IBM Z Common Data Provider Configuration Tool, you can manage custom data stream definitions. For example, you might want to create custom SMF data stream definitions, or application data stream definitions for your own applications, so that you can add these data streams to your policy.


Before you begin

For information about how to run the Configuration Tool, see [“Running the Configuration Tool” on page 31](#).

About this task

You must create a stream definition for each custom data type (analogous to types such as SMF record type 30 or z/OS SYSLOG) from which you want to collect operational data.

These stream definitions are used to populate the Configuration Tool with your data streams, which you can then use in your policy. For example, in the **Policy Profile Edit** window of the Configuration Tool, when

you click the **Add Data Stream** icon  **DATA STREAM** to add a data stream to your policy, the **"Select data stream"** window is shown with a list of categorized data streams. You can expand the categories to view the possible data streams that you can define for the policy. After you create your custom data stream definitions, your custom data streams are included in this categorized list.

You can create the following two categories of custom data stream definition:

System Data Engine data stream definition

Specifies a data stream for a type of data that is gathered by the System Data Engine.

Application data stream definition

Specifies a data stream for a type of user application data.

After you create an application data stream definition, you use the Open Streaming API to send your application data to the Data Streamer and stream it to analytics platforms. For more information, see [Chapter 7, “Sending user application data to the Data Streamer,” on page 283](#).

Creating a System Data Engine data stream definition

From the Common Data Provider tab in the IBM Z Common Data Provider Configuration Tool, you can create a System Data Engine data stream definition to use in a policy.

Procedure

1. Before you create a System Data Engine data stream definition, complete the following steps:
 - a) On the z/OS system where the Configuration Tool runs, create a partitioned data set (PDS) that can be used as the concatenation library for storing the data set members that contain the record, update, and, optionally, template and table definitions for this data stream. Only one concatenation library is required for each Configuration Tool. If you already create a concatenation library, skip this step.

The data set must be defined with the attributes RECFM=VB and LRECL=255, which are the same as those for the SMP/E target data set *hlq.SHBODEFS*. The following example shows the PDS file *USERID.LOCAL.DEFS*, where *USERID* represents your user ID:

```

Data Set Name . . . . : USERID.LOCAL.DEFS

General Data                               Current Allocation
Management class . . : **None**           Allocated cylinders : 10
Storage class . . . . : CLASS2             Allocated extents . : 1
Volume serial . . . . : T10062            Maximum dir. blocks : 20
Device type . . . . . : 3390
Data class . . . . . : **None**
Organization . . . . : PO                 Current Utilization
Record format . . . . : VB                 Used cylinders . . . : 1
Record length . . . . : 255               Used extents . . . . : 1
Block size . . . . . : 27998             Used dir. blocks . . : 1
1st extent cylinders: 10                  Number of members . : 0
Secondary cylinders : 5

```

Tip: For each new concatenation library (USER concatenation), you must later set it in the System Data Engine properties, as described in [What to do next](#).

- b) Use the System Data Engine language to create the custom record, update, and optionally template and table definitions, as described in [“Language reference for System Data Engine”](#) on page 239.
2. In the Configuration Tool, click the **MANAGE CUSTOM DATA STREAM DEFINITIONS** button.
3. In the resulting **Manage Custom Data Stream Definitions** window, click the **Create System Data Engine data stream definition** box.
4. In the resulting **Define System Data Engine Data Stream** window, provide values for the following fields:

Name

Specifies the name of the data stream. The name must be the same as the name of the update definition. For example, if the update is defined by the statement `DEFINE UPDATE SMF_CUST_030`, the data stream name must be `SMF_CUST_030`.

This name is converted to uppercase characters when it is saved.

The **Name** value must contain only alphanumeric characters and underscores. The maximum length is 243 characters.

Group

In the Configuration Tool, the data stream is included in the list of categorized data streams under the main category **Customer Data Streams**. For the hierarchy under **Customer Data Streams**, you must specify the group and subgroup under which you want to include the data stream. The value of **Group** specifies the group. The following example illustrates the hierarchy, and indicates that MYGROUP is specified as the **Group** value:

- **Customer Data Streams**
 - **MYGROUP**

The **Group** value is case-insensitive. For example, if you specify MyGroup as the value, and a group that is named MYGROUP exists under the main category **Customer Data Streams**, the Configuration Tool includes the data stream under MYGROUP, and does not create the category MyGroup.

The **Group** value must contain only alphanumeric characters and underscores. The maximum length is 243 characters.

Subgroup

In the Configuration Tool, the data stream is included in the list of categorized data streams under the main category **Customer Data Streams**. For the hierarchy under **Customer Data Streams**, you must specify the group and subgroup under which you want to include the data stream. The value of **Subgroup** specifies the subgroup. The following example illustrates the hierarchy, and indicates that MYSUBGROUP is specified as the **Subgroup** value:

- **Customer Data Streams**

- **MYGROUP**
- **MYSUBGROUP**

The **Subgroup** value is case-insensitive. For example, if you specify MySubGroup as the value, and a subgroup that is named MYSUBGROUP exists under the group, the Configuration Tool includes the data stream under MYSUBGROUP, and does not create the category MySubGroup.

The **Subgroup** value must contain only alphanumeric characters and underscores. The maximum length is 243 characters.

SHBODEFS data set members

Specifies the names of the data set members that contain the record, update, and, optionally, template definitions for this data stream.

The value of **SHBODEFS data set members** is case-insensitive and is converted to uppercase characters when it is saved.

You must list one member per line, and the data stream definitions must be listed in the following order:

- a. Record definition
- b. Update definition
- c. Template definition

Therefore, if the data stream definitions are in separate data set members, the members must be listed in the following order:

- a. The member that contains the record definition
- b. The member that contains the update definition
- c. The member that contains the template definition

For example, assume that you want to specify the following data set members:

- HBORSZ30 for record definition
- HBOUSZ30 for update definition
- HBOTSZ30 for template definition

In this case, you must list the members as shown in the following example:

```
HBORSZ30  
HBOUSZ30  
HBOTSZ30
```

Important:

- When a *custom* data stream is added to a policy, the Configuration Tool searches the concatenation libraries in the following order to find the specified data set members:
 - a. USER concatenation library
 - b. CDP concatenation library

The concatenation libraries are defined in the System Data Engine properties, as described in [“SYSTEM DATA ENGINE properties: Defining your System Data Engine environment” on page 155.](#)

5. Click **OK**.

The data stream is created and available to be included in policies.

What to do next

Add the custom data stream to your policy. For information about creating or updating a policy, see the following information:

- [“Creating a policy” on page 36](#)
- [“Updating a policy” on page 43](#)

Remember: Before you save a policy that includes a custom System Data Engine data stream, you must set the USER concatenation library in the System Data Engine properties, as described in [“SYSTEM DATA ENGINE properties: Defining your System Data Engine environment” on page 155](#).

Streaming IMS user log records to your analytics platform by using a data stream definition

When the IMS Log Write (LOGWRT) user exit is enabled, IMS log records, including IMS user log records, are written to System Management Facilities (SMF) for processing by the System Data Engine. However, to have the System Data Engine process the IMS user log records, you must also use System Data Engine language statements to define custom record and update definitions, and, optionally, to define template definitions for the advanced data filtering of IMS user log records.

About this task

After you create custom definitions for System Data Engine data streams, create one or more data streams for the IMS user log records and then update your analytics platform so that it can process the data streams. After that, create or update the policy in the Configuration Tool to include the data streams.

Procedure

1. If you do not already have one, create a partitioned data set (PDS) that is used as the user concatenation library for the custom record and update definitions.

For more information about how to create the data set, see step 1a in [“Creating a System Data Engine data stream definition” on page 46](#).

2. Copy the sample record definition IMS_USR_F801 from the member HBOURIMS of the SMP/E target data set *hlq.SHBODEFS* to a new member of the user concatenation library and edit the definition based on your requirements.

The following example shows how to define a custom record definition for IMS user log record type x 'F801' based on the sample record definition.

```

/*****/
/*
/* IMS Ilog Record Type x'F801'
/*
/*
/*****/
DEFINE RECORD IMS_USR_F801
  VERSION 'CDP.210'
  IN LOG SMF
  BUILT BY HBOSDIMS
  IDENTIFIED BY USRTYPE = 248
                AND USRSUBT = 1
  FIELDS
  (
-----
--- 1. IMS Record Prefix
-----
    USRLL    LENGTH 2 BINARY,      -- Length of log record
    USRZZ    LENGTH 2 HEX,         -- QSAM reserved bits
    USRTYPE  LENGTH 1 BINARY,      -- Record type
    USRTYPEH OFFSET 4 LENGTH 1 HEX, -- Record type in HEX
    USRSUBT  LENGTH 1 BINARY,      -- Record subtype
    USRSUBTH OFFSET 5 LENGTH 1 HEX, -- Record subtype in HEX
-----
--- 2. IMS Record Data
-----
    fld1 ,
    fld2 ,
    .....
    fldn
  )
-----

```

```

--- 3. IMS Record Suffix
-----
SECTION SUFFIX
  OFFSET USRLL - 16
  LENGTH 16
  NUMBER 1
  FIELDS (
    USRSTCK LENGTH 8 TIMESTAMP(TOD), -- timestamp
    USRLSN  LENGTH 8 HEX             -- log sequence number
  );

```

DEFINE RECORD

The following naming convention is used for the record name:

- `IMS_USR_XX` is used if the IMS user log record has no subtype. `xx` is the HEX value of the IMS user log record type.
- `IMS_USR_XXYY` is used if the IMS user log record has subtypes. `xx` is the HEX value of the IMS user log record type, and `yy` is the subtype.

```
DEFINE RECORD IMS_USR_F801
```

USRTYPE

USRTYPE is the binary value of the IMS user log record type. In the example, 248 is the equivalent binary value for HEX `x'F8'`.

```
IDENTIFIED BY USRTYPE = 248
```

USRSUBT

USRSUBT is the binary value of the IMS user log record subtype. In the example, 1 is the equivalent binary value for HEX `x'01'`. If the IMS user log record does not have any subtype, remove this line.

```
AND USRSUBT = 1
```

USRSUBT LENGTH and USRSUBTH OFFSET

If the IMS user log record does not have any subtype, remove these lines.

```

USRSUBT LENGTH 1 BINARY, -- Record subtype
USRSUBTH OFFSET 5 LENGTH 1 HEX, -- Record subtype in HEX

```

fld1, fld2, fldn

This section defines the fields in the IMS user log record. Fields are separated by commas.

```

fld1 ,
fld2 ,
.....
fldn

```

You can define multiple record definitions in the same member if you have multiple IMS user log record types. For the language reference of the `DEFINE RECORD` statement, see [“DEFINE RECORD statement”](#) on page 250.

3. Copy the sample update definition `IMS_USR_F801` from the member `HB0UUIMS` of the SMP/E target data set `h1q.SHB0DEFS` to a new member of the user concatenation library and edit the definition based on your requirements.

The following example shows how to define an update definition for IMS user log record type `x'F801'` based on the sample update definition.

```

SET IBM_FILE = 'IMSF801';

DEFINE UPDATE IMS_USR_F801
  VERSION 'CDP.210'
  FROM IMS_USR_F801
  TO &IBM_UPDATE_TARGET
  &IBM_CORRELATION
  AS &IBM_FILE_FORMAT SET(ALL);

```

SET

The SET statement is needed only when the target of the update definition is a file, which means the variable `IBM_UPDATE_TARGET` is set to `FILE &IBM_FILE`.

The following naming convention for IMS log record definitions is used in this example:

- `IMSxx` is used if the IMS user log record has no subtype. `xx` is the HEX value of the IMS user log record type.
- `IMSxxyy` is used if the IMS user log record has subtypes. `xx` is the HEX value of the IMS user log record type, and `yy` is the subtype.

```
SET IBM_FILE = 'IMSF801';
```

DEFINE UPDATE

The custom update definition name must be unique among update definitions. In this example, the custom update definition name is the same as the name of the custom record definition that the update definition is associated with.

```
DEFINE UPDATE IMS_USR_F801
```

FROM

The FROM clause identifies the source of the update definition, which is the name of the custom record definition.

```
FROM IMS_USR_F801
```

You can define multiple custom update definitions in the same member if you have multiple IMS user log record types. Also you can use the WHERE clause to select certain records for collection. For the language reference of the DEFINE UPDATE statement, see [“DEFINE UPDATE statement” on page 258](#).

4. Optional: If you want to use advanced data filters to filter the fields to be collected from the user log records, add a DEFINE TEMPLATE statement for the update definition in the same data set member of that update definition.

In the template definition, you must include the field `USRSTCK`, which is required for timestamp resolution when you ingest data to your analytics platform. For the language reference of the DEFINE TEMPLATE statement, see [“DEFINE TEMPLATE statement” on page 263](#).

5. Validate the syntax of the custom record, update, and, optionally, template definitions.

Use the following example job to verify the members for the custom record and update definitions.

```
//HBOJBCOL JOB (), 'DUMMY',MSGCLASS=X,MSGLEVEL=(,0),
//          CLASS=A,NOTIFY=&SYSUID
//*
//HBOSMFCB EXEC PGM=HBOPDE,REGION=0M,PARM='SHOWINPUT=YES'
//STEPLIB DD DISP=SHR,DSN=hlq.SHB0LOAD
//HBOOUT DD SYSOUT=*
//HBODUMP DD SYSOUT=*
//HBOIN DD DISP=SHR,DSN=hlq.SHB0DEFS(HBOCCSV)
// DD DISP=SHR,DSN=hlq.SHB0DEFS(HBOCCORY)
// DD DISP=SHR,DSN=hlq.SHB0DEFS(HBOLLSMF)
// DD DISP=SHR,DSN=hlq.SHB0DEFS(HBORSIMS)
// DD DISP=SHR,DSN=USERID.LOCAL.DEFS(USRRSIMS)
// DD DISP=SHR,DSN=USERID.LOCAL.DEFS(USRUSIMS)
// DD *
COLLECT SMF
WITH STATISTICS
BUFFER SIZE 1 M;
//*
//HBOLOG DD
DUMMY
```

hlq

Change `hlq` to the high-level qualifier for the IBM Z Common Data Provider SMP/E target data set.

```
//STEPLIB DD DISP=SHR,DSN=hlq.SHB0LOAD
```


sourceType

The value of `sourceType` must match the data source type of the data stream. The naming convention is `zOS-data_stream_name`.

```
if [sourceType] == "zOS-IMS_USR_F801"
```

csv{ columns =>

If you have a custom template definition, change the column list to match the fields and order in the template definition. If the first column is `Correlator`, leave it as the first column in the list.

USRSUBT and USRSUBTH

If the IMS user log record does not have any subtype, remove `USRSUBT` and `USRSUBTH`.

f1d1, f1d2, and f1dn

Replace `f1d1`, `f1d2`, and `f1dn` with field names in your custom record definition.

Timestamp resolution configuration file

The file is named `N_data_stream_name.lsh`, for example, `N_IMS_USR_F801.lsh`. Here is an example of the file:

```
# CDPz ELK Ingestion
#
# Timestamp Extraction for stream zOS-IMS_USR_F801
#

filter {
  if [sourceType] == "zOS-IMS_USR_F801" {
    mutate{ add_field => {
      "[@metadata][timestamp]" => "%{USRSTCK}"
    }}

    date{ match => [
      "[@metadata][timestamp]", "yyyy-MM-dd HH:mm:ss.SSSSS"
    ]}
  }
}
```

sourceType

The value of `sourceType` must match the data source type of the data stream. The naming convention is `zOS-data_stream_name`.

```
if [sourceType] == "zOS-IMS_USR_F801"
```

Restart Logstash after you create the files for all data streams. Refer to Logstash documentation for more information about the configuration files.

- If you are ingesting data to Splunk, define the layout of the data streams to the Splunk server by creating the `props.conf` file in the `Splunk_Home/etc/apps/ibm_cdpz_buffer/local` directory with the following content. If the `props.conf` already exists, append the following lines to that file.

```
#
# IMS_USR_F801
#

[zOS-IMS_USR_F801]
TIMESTAMP_FIELDS = USRSTCK, timezone
TIME_FORMAT= %F %H:%M:%S.%6Q %z
FIELD_NAMES = "sysplex", "system", "hostname", "", "", "sourcename", "timezone",
"Correlator", "USRLI", "USRZZ", "USRTYPE", "USRTYPEH", "USRSUBT", "USRSUBTH",
"f1d1", "f1d2", "f1dn", "USRSTCK", "USRLSN"
INDEXED_EXTRACTIONS = csv
KV_MODE = none
NO_BINARY_CHECK = true
SHOULD_LINEMERGE = false
category = Structured
disabled = false
pulldown_type = true
```

[zOS-IMS_USR_F801]

You must specify the data source name of the data stream. The naming convention is zOS-*data_stream_name*.

FIELD_NAMES

If you have a custom template definition, change the column list to match the fields and order in the template definition. If the column `Correlator` exists, do not remove it.

USRSUBT and USRSUBTH

If the IMS user log record does not have any subtype, remove USRSUBT and USRSUBTH.

f1d1, f1d2, and f1dn


Replace f1d1, f1d2, and f1dn with field names in your record definition.

Tip: The example includes only one data stream. Duplicate the sample code for each new data stream.

In the Splunk user interface, you must also configure the file to data source type mapping for the new data stream. The file that the Data Receiver saves is named zOS-*data_stream_name*-*.cdp. For example, the data stream IMS_USR_F801 has the file named CDP-zOS-IMS_USR_F801-*.cdp.

Restart the Splunk server after you make the changes.

Refer to Splunk documentation for more information.

8. Create or update the policy to add the new System Data Engine data streams so that the IMS user log records can be processed and streamed by the IBM Z Common Data Provider.
 - a) In the Configuration Tool primary window, select the policy that you want to update .
 - b) Click the **Add Data Stream** icon  **DATA STREAM** in the **Policy Profile Edit** window.
 - c) Find and select the new data stream from the list in the **select data stream** window. Select more data streams if you have multiple IMS user log record types.
 - d) Assign a subscriber for each new data stream.
 - e) In the **Policy Profile Edit** window, click **SYSTEM DATA ENGINE** to ensure that values are provided for **USER Concatenation** and **CDP Concatenation** fields, and click **OK**. Following previous examples, `USERID . LOCAL . DEFS` should be specified for the **USER Concatenation** field. Fill in this field with the name of your user concatenation library.
 - f) Click **Save** to save the policy.

Important: Each time that the associated record definition or update definition is changed, you must edit and save the policy in the Configuration Tool so that the changes are reflected in the policy.

For more information on how to update a policy, see [“Updating a policy”](#) on page 43.

9. Restart the Data Streamer and the System Data Engine.

Results

The IMS user log records are streamed to your analytics platform.

Filtering a System Data Engine data stream by using a data stream definition

For a System Data Engine data stream, you can use the WHERE clause in the custom update definition to filter the records to be processed, and use the custom template definition that is associated with the update definition to filter the fields to be streamed by IBM Z Common Data Provider.

About this task

After you create custom definitions for System Data Engine data streams, create a custom System Data Engine data stream and then update your analytics platform so that it can process the new data stream. After that, create or update the policy in the Configuration Tool to include the new data stream. If you are adding filters to an existing data stream, you can create the custom definitions and data streams, and update your analytics platform based on the settings for that data stream.

Procedure

1. If you do not already have one, create a partitioned data set (PDS) that is used as the user concatenation library for the custom definitions.

For more information about how to create the data set, see step 1a in [“Creating a System Data Engine data stream definition”](#) on page 46.

2. Create a custom update definition in a new or an existing data set member of the user concatenation library. If you want to filter the records to be processed, add a WHERE clause to the definition. The name of the member for the custom update definition cannot be the same as any existing member in the SHBODEFS data set.
 - You can create a new custom update definition by using the DEFINE UPDATE statement. For the language reference of the DEFINE UPDATE statement, see [“DEFINE UPDATE statement”](#) on page 258.
 - To filter an existing data stream, copy the update definition that is used by the data stream to a new or an existing data set member of the user concatenation library and update the definition based on your requirements.
 - a. Locate the update definition for the existing data stream by using one of the following methods. The data set members for update definitions are named HBOUxxxx.
 - Because the IBM Z Common Data Provider names the data stream with the name of the associated update definition, in your z/OS environment, use ISPF option 3.14 or SRCHF0R ISPF command to search the data stream name.
 - Review the data stream definition in the sde.streams.json file or the ims.streams.json file in the Configuration Tool directory /usr/lpp/IBM/zcdp/v2r1m0/UI/LIB/. Check the hboin parameter that specifies all data set members for required definitions. Usually the last one is for the update definition.
 - b. Copy the update definition to a new or an existing data set member of the user concatenation library.

The following code sample shows the update definition SMF_101_1_PACKAGE in the member HBOUS101 of the data set SHBODEFS.

```
SET IBM_FILE = 'SMF1011K';

DEFINE UPDATE SMF_101_1_PACKAGE
  VERSION 'CDP.210'
  FROM SMF_101_1 SECTION PACKAGE
  TO &IBM_UPDATE_TARGET
  &IBM_CORRELATION
  AS &IBM_FILE_FORMAT SET(ALL);
```

Copy the update definition SMF_101_1_PACKAGE to the data set member USRUS101 in the user concatenation library USERID.LOCAL.DEFS with the following changes.

SET

The SET statement is needed only when the target of the update definition is a file, which means the variable *IBM_UPDATE_TARGET* is set to FILE &IBM_FILE. You can change it to USR1011K.

```
SET IBM_FILE = 'USR1011K';
```

DEFINE UPDATE

The data streams must have unique names, so you must rename the update definition to avoid conflict with the existing data stream. You can change it to USR_101_1_PACKAGE.

The updated USERID.LOCAL.DEFS(USRUS101) member has the following content:

```
SET IBM_FILE = 'USR1011K';

DEFINE UPDATE USR_101_1_PACKAGE
  VERSION 'CDP.210'
  FROM SMF_101_1 SECTION PACKAGE
```

```
TO &IBM_UPDATE_TARGET
&IBM_CORRELATION
AS &IBM_FILE_FORMAT SET(ALL);
```

- c. If you want to filter the records to be processed, add a WHERE clause to the custom update definition. For example, if you want to collect only Db2 package accounting records whose transaction name starts with MG, or the authorization ID is U@MUPJ2, add the following WHERE clause:

```
WHERE (SUBSTR(QWHCEUTX,1,2) = 'MG')
OR (QWHCAID = 'U@MUPJ2')
```

The updated *USERID*.LOCAL.DEFS(USRUS101) member has the following content:

```
SET IBM_FILE = 'USR1101K';

DEFINE UPDATE USR_101_1_PACKAGE
VERSION 'CDP.210'
FROM SMF_101_1 SECTION PACKAGE
WHERE (SUBSTR(QWHCEUTX,1,2) = 'MG')
OR (QWHCAID = 'U@MUPJ2 ')
TO &IBM_UPDATE_TARGET
&IBM_CORRELATION
AS &IBM_FILE_FORMAT SET(ALL);
```

For more information about the WHERE clause, see [“WHERE” on page 259](#).

3. If you want to filter the fields to be streamed, add a DEFINE TEMPLATE statement for the update definition in the same data set member of that update definition.

Verify that the template definition is placed after the update definition. The following example shows a template definition in the member *USERID*.LOCAL.DEFS(USRUS101) for the update definition USR_101_1_PACKAGE to stream only a few fields in the PACKAGE section of record SMF_101_1 record.

```
SET IBM_FILE = 'USR1101K';

DEFINE UPDATE USR_101_1_PACKAGE
VERSION 'CDP.210'
FROM SMF_101_1 SECTION PACKAGE
WHERE (SUBSTR(QWHCEUTX,1,2) = 'MG')
OR (QWHCAID = 'U@MUPJ2 ')
TO &IBM_UPDATE_TARGET
&IBM_CORRELATION
AS &IBM_FILE_FORMAT SET(ALL);

DEFINE TEMPLATE USR_101_1_PACKAGE FOR USR_101_1_PACKAGE
ORDER
(SM101TME,
SM101DTE,

QPACLOCN,
QPACCOLN,
QPACPKID,
QPACSQLC,
QPACSCB,
QPACSCS,
QPACBJST,
QPACJST)
AS &IBM_FILE_FORMAT;
```

DEFINE TEMPLATE

The template definition name must be the same as the update definition name to replace the default template definition that streams all fields for the update definition. In the template definition, you must include the date and time fields from the SMF record header for an SMF record, or the timestamp field in the record suffix for an IMS log record. These fields are required for timestamp resolution when you ingest data to your analytics platform. In this example, the fields are SM101DTE and SM101TME.

For the language reference of the DEFINE TEMPLATE statement, see [“DEFINE TEMPLATE statement” on page 263](#).

4. Validate the syntax of the custom update and template definitions.

Use the following example job to verify the members for the custom update and template definitions.

```
//HBOBCOL JOB ( ), 'DUMMY',MSGCLASS=X,MSGLEVEL=(,0),
// CLASS=A,NOTIFY=&SYSUID
//*
//HBOSMFCB EXEC PGM=HBOPDE,REGION=0M,PARM='SHOWINPUT=YES'
//STEPLIB DD DISP=SHR,DSN=hlq.SHBOLoad
//HBOOUT DD SYSOUT=*
//HBODUMP DD SYSOUT=*
//HBOIN DD DISP=SHR,DSN=hlq.SHBODEFS(HBOCCSV)
// DD DISP=SHR,DSN=hlq.SHBODEFS(HBOCCORY)
// DD DISP=SHR,DSN=hlq.SHBODEFS(HBOLLSMF)
// DD DISP=SHR,DSN=hlq.SHBODEFS(HBORS101)
// DD DISP=SHR,DSN=USERID.LOCAL.DEFS(USRUS101)
// DD *
COLLECT SMF
WITH STATISTICS
BUFFER SIZE 1 M;
//*
//HBOLOG DD DUMMY
```

hlq

Change *hlq* to the high-level qualifier for the IBM Z Common Data Provider SMP/E target data set.

```
//STEPLIB DD DISP=SHR,DSN=hlq.SHBOLoad
//HBOOUT DD SYSOUT=*
//HBODUMP DD SYSOUT=*
//HBOIN DD DISP=SHR,DSN=hlq.SHBODEFS(HBOCCSV)
// DD DISP=SHR,DSN=hlq.SHBODEFS(HBOCCORY)
// DD DISP=SHR,DSN=hlq.SHBODEFS(HBOLLSMF)
// DD DISP=SHR,DSN=hlq.SHBODEFS(HBORS101)
```

HBORS101

HBORS101 contains the record definition SMF_101_1. This member must be included before the member that contains your custom update and template definitions.

```
// DD DISP=SHR,DSN=hlq.SHBODEFS(HBORS101)
```

// DD DISP=SHR,DSN=USERID.LOCAL.DEFS(USRUS101)

Specifies the data set member for the custom update and template definition.

Important: Verify that the definitions are error-free by running the validation job before you create the custom data stream.

If there is no syntax error, you see the following messages.

```
HB00201I Update USR_101_1_PACKAGE was successfully defined.
HB00500I Template USR_101_1_PACKAGE was successfully defined.
```

If there are syntax errors, correct the errors according to the messages in the output file that is defined by HBOOUT.

5. Create a custom System Data Engine data stream in the Configuration Tool based on the update definition and template definition that are created in previous steps.

For more information, see [“Creating a System Data Engine data stream definition”](#) on page 46. Verify that the data stream name, the custom update definition name, and the custom template definition name are the same, and that you specify the member for the record definition before the member for the custom update and template definitions in the **SHBODEFS data set members** field.

6. Update your analytics platform so that it can process the new data stream.

- If you are ingesting data to the Elastic Stack, for each data stream, create a field name annotation configuration file, and a timestamp resolution configuration file in the Logstash configuration directory.

If your new data stream is created based on an existing one, you can create the two files by copying and editing the files for the old data stream. In previous examples, the new data stream

USR_101_1_PACKAGE is created based on the existing data stream SMF_101_1_PACKAGE, and the two configuration files are H_SMF_101_1_PACKAGE.lsh and N_SMF_101_1_PACKAGE.lsh in the Logstash configuration directory. Copy these two files and change the file names to H_USR_101_1_PACKAGE.lsh and N_USR_101_1_PACKAGE.lsh, then edit the files according to the following instructions.

Field name annotation configuration file

The file is named `H_data_stream_name.lsh`, for example, `H_USR_101_1_PACKAGE.lsh`. See the following example of the file:

```
# CDPz ELK Ingestion
#
# Field Annotation for stream zOS-USR_101_1_PACKAGE
#

filter {
  if [sourceType] == "zOS-USR_101_1_PACKAGE" {

    csv{ columns => [ "Correlator", "SM101TME",
"SM101DTE", "QPACLOCN", "QPACCOLN", "QPACPKID",
"QPACSQLC", "QPACSCB", "QPACSCCE", "QPACBJST",
"QPACEJST"]
      separator => "," }

  }
}
```

sourceType

The value of `sourceType` must match the data source type of the data stream. The naming convention is `zOS-data_stream_name`.

```
if [sourceType] == "zOS-USR_101_1_PACKAGE"
```

csv{ columns => []

If you have a custom template definition, change the column list to match the fields and order in the template definition.

Timestamp resolution configuration file

The file is named `N_data_stream_name.lsh`, for example, `N_USR_101_1_PACKAGE.lsh`. See the following example of the file:

```
# CDPz ELK Ingestion
#
# Timestamp Extraction for stream zOS-USR_101_1_PACKAGE
#

filter {
  if [sourceType] == "zOS-USR_101_1_PACKAGE" {
    mutate{ add_field => {
      "[@metadata][timestamp]" => "%{SM101DTE} %{SM101TME}"
    }}

    date{ match => [
      "[@metadata][timestamp]", "yyyy-MM-dd HH:mm:ss:SS"
    ]}

  }
}
```

sourceType

The value of `sourceType` must match the data source type of the data stream. The naming convention is `zOS-data_stream_name`.

```
if [sourceType] == "zOS-USR_101_1_PACKAGE"
```

add_field =>

For an SMF record, you must specify the date and time fields in the SMF record header. In this example, the fields are SM101DTE and SM101TME.

```
"[@metadata][timestamp]" => "%{SM101DTE} %{SM101TME}"
```

For an IMS log record, you must specify the timestamp field in the record suffix. For example, the timestamp field in the IMS_07 record suffix is DLRSTCK.

```
"[@metadata][timestamp]" => "%{DLRSTCK}"
```

match =>

For an SMF record, use the following time format.

```
"[@metadata][timestamp]", "yyyy-MM-dd HH:mm:ss:SS"
```

For an IMS log record, use the following time format.

```
"[@metadata][timestamp]", "yyyy-MM-dd HH:mm:ss.SSSSSS"
```

Restart Logstash after you create the files for the new data stream. Refer to Logstash documentation for more information about the configuration files.

- If you are ingesting data to Splunk, define the layout of the data stream to the Splunk server by creating the `props.conf` file in the `Splunk_Home/etc/apps/ibm_cdpz_buffer/local` directory on the Splunk server.

If your new data stream is created based on an existing one, you can create the file by copying and editing the content for the old data stream. Based on previous examples, open the `props.conf` file in the `Splunk_Home/etc/apps/ibm_cdpz_buffer/default` directory and copy the section for `SMF_101_1_PACKAGE`. Paste the content to the `props.conf` file in `Splunk_Home/etc/apps/ibm_cdpz_buffer/local` and edit it according to the following example. If the `props.conf` file exists, append the content to the file.

```
#
# USR_101_1_PACKAGE (zOS-USR_101_1_PACKAGE)
#

[zOS-USR_101_1_PACKAGE]
TIMESTAMP_FIELDS = SM101DTE, SM101TME, timezone
TIME_FORMAT = %F %H:%M:%S:%2Q %z
FIELD_NAMES = "sysplex","system","hostname","","","sourcename",
"timezone", "Correlator", "SM101TME", "SM101DTE", "QPACLOCN",
"QPACCOLN", "QPACPKID", "QPACSQLC", "QPACSCB", "QPACSCCE",
"QPACBJST", "QPACEJST"
INDEXED_EXTRACTIONS = csv
KV_MODE = none
NO_BINARY_CHECK = true
SHOULD_LINEMERGE = false
category = Structured
disabled = false
pulldown_type = true
TRUNCATE = 20000
```

[zOS-USR_101_1_PACKAGE]

You must specify the data source name of the data stream. The naming convention is `zOS-data_stream_name`.

TIMESTAMP_FIELDS

For an SMF record, you must specify the date and time fields in the SMF record header. In this example, the fields are `SM101DTE` and `SM101TME`.

```
TIMESTAMP_FIELDS = SM101DTE, SM101TME, timezone
```

For an IMS log record, you must specify the timestamp field in the record suffix. For example, the timestamp field in the `IMS_07` record suffix is `DLRSTCK`.

```
TIMESTAMP_FIELDS = DLRSTCK, timezone
```

TIME_FORMAT

For an SMF record, use the following time format.

```
TIME_FORMAT = %F %H:%M:%S:%2Q %z
```

For an IMS log record, use the following time format.

```
TIME_FORMAT = %F %H:%M:%S.%6Q %z
```

FIELD_NAMES


If you have a custom template definition, change the column list to match the fields and order in the template definition. If the column `Correlator` exists, do not remove it.

In the Splunk user interface, you must also configure the file to data source type mapping for the new data stream. The file that the Data Receiver saves is named `zOS-data_stream_name-*.cdp`. For example, the data stream `USR_101_1_PACKAGE` has the file that is named `CDP-zOS-USR_101_1_PACKAGE-*.cdp`.

Restart the Splunk server after you make the changes.

Refer to Splunk documentation for more information.

7. Create or update the policy to add the new System Data Engine data stream.

- a) In the Configuration Tool primary window, select the policy that you want to update.
- b) Click the **Add Data Stream** icon  **DATA STREAM** in the **Policy Profile Edit** window.
- c) Find and select the new data stream from the list in the **select data stream** window.
- d) Assign a subscriber for each new data stream.
- e) In the **Policy Profile Edit** window, click **SYSTEM DATA ENGINE** to ensure that values are provided for **USER Concatenation** and **CDP Concatenation** fields, and click **OK**. Complete the field **USER Concatenation** with the data set name of your user concatenation library. Based on previous examples, `USERID . LOCAL . DEFS` should be specified for the field.
- f) Click **Save** to save the policy.

Important: Each time that the associated update definition or template definition is changed, you must edit and save the policy in the Configuration Tool so that the changes are reflected in the policy.

For more information on how to update a policy, see [“Updating a policy” on page 43](#).

8. Restart the Data Streamer and the System Data Engine.

Results

The records and fields are filtered according to your configuration.

Streaming structured data of your application to analytics platforms by using a data stream definition

In addition to the SMF records that are produced by IBM products and some third-party products, you can use IBM Z Common Data Provider to stream your own SMF records and application data. By using this function, you can stream structured data of your application to analytics platforms.

About this task

If the structured data of your application is not SMF data, you must first write the data to an SMF record. Create custom record definitions, update definitions, and optionally template definitions for this special SMF record by using IBM Z Common Data Provider System Data Engine language to support the structured data of your application. In the Configuration Tool, create data streams for your definitions, and then create or update policies to include the data streams. On the analytics platforms, update the configuration files to support the new data source types.

Procedure

1. If the structured data of your application is available in SMF records, skip this step. If the structured data of your application is not SMF data, write the data to an SMF record as the payload by using the `SMFWTM` or `SMFEWTM` macro, or other methods.

You can use SMF record type 128 through 255, which are for user-written records, to include your structured data.

Because the SMF user exit that is provided by IBM Z Common Data Provider suppresses the recording of the SMF record type 127 subtype 1000, if your SMF is in data set recording mode and you want the records to be processed by IBM Z Common Data Provider only and not recorded to VSAM data sets, you can write your structured data to SMF record type 127 subtype 1000. If you use this record type, you must define the record layout according to the following table. Because this record type is used in IBM Z Common Data Provider by multiple data providers, two additional fields SM127SRC and SM127SRS are defined.

Offset	Data Field	Length	Note
0 (x00)	SM127LEN	2	Record length (maximum size of 32,756) Must be the logical record length including the RDW.
2 (x02)	SM127SEG	2	Segment descriptor Initialize the field with zeros.
4 (x04)	SM127FLG	1	System indicator Turn on bit 1 (x'40') indicating record with subtypes.
5 (x05)	SM127RTY	1	Record type The value must be 127.
6 (x06)	SM127TME	4	Time of record written No need to supply this field if you use SMFWTM or SMFEWTM macro.
10(x0A)	SM127DTE	4	Date of record written No need to supply this field if you use SMFWTM or SMFEWTM macro.
14(x0E)	SM127SID	4	System ID No need to supply this field if you use SMFWTM or SMFEWTM macro.
18(x12)	SM127SSI	4	Subsystem ID The value must be CDP.
22(x16)	SM127STY	2	Record subtype This value must be 1000.
24(x18)	SM127SRC	2	User payload type Specify a number between 128 and 255.
26(x1A)	SM127SRS	2	User payload subtype Use this field to further identify your payload application structured data.
28(x1C)			Start of user payload data.

Refer to the MVS™ System Management Facilities (SMF) manual for more details on writing SMF records.

- If you do not already have one, create a partitioned data set (PDS) that is used as the user concatenation library for the custom definitions.

For more information about how to create the data set, see step 1a in [“Creating a System Data Engine data stream definition”](#) on page 46.

- In a data set member of the user concatenation library, create a custom record definition for the SMF records that contain the structured data of your application.

The custom record definition must match the layout of the SMF records. The following code sample creates a member `USERID.LOCAL.DEFS(USRRS127)` to define the record definition for SMF record type 127 subtype 1000 that contains the structured data.

```

/*****/
/*
/* SMF Record Type 127 SubType 1000 for User Data          */
/*
/*
/*****/
SET SMF_ABC_RECTYPE = '127' ;
SET SMF_ABC_RECSTYP = '1000' ;
SET SMF_ABC_RECSID  = 'CDP' ;
SET SMF_ABC_SRCID   = '128' ;

DEFINE RECORD ABC_01
  VERSION 'CDP.210'
  IN LOG SMF
  IDENTIFIED BY SM127RTY = &SMF_ABC_RECTYPE
                AND SM127STY = &SMF_ABC_RECSTYP
                AND SM127SID = &SMF_ABC_RECSID
                AND SM127SRC = &SMF_ABC_SRCID
                AND SM127SRS = 1

  FIELDS (
-----
--- Standard SMF record header
-----
SM127LEN  LENGTH 2  BINARY,      -- Record length
SM127SEG  LENGTH 2  BINARY,      -- Segment descriptor
SM127FLG  LENGTH 1  HEX,         -- System indicator
SM127RTY  LENGTH 1  BINARY,      -- Record Type
SM127TME  LENGTH 4  TIME(1/100S), -- Time
SM127DTE  LENGTH 4  DATE(0CYDDDF), -- Date
SM127SID  LENGTH 4  CHAR,        -- System ID
SM127SSI  LENGTH 4  CHAR,        -- Subsystem ID
SM127STY  LENGTH 2  BINARY,      -- Record subtype
-----
--- CDP fields for payload type/subtype
-----
SM127SRC  LENGTH 2  BINARY,      -- Payload Type
SM127SRS  LENGTH 2  BINARY,      -- Payload Subtype
-----
--- SMF User Data
-----
fld1 ,
fld2 ,
.....
fldn
);

```

SMF_ABC_SRCID

Identifies the payload type of your data. Specify a number between 128 and 225.

DEFINE RECORD

Specifies the name for the custom record definition. The name must be different from other record definitions.

SM127SRC and SM127SRS

Specifies the user data payload type and subtype. Use these fields to further identify the SMF record. SMF record type 127 subtype 1000 uses these two additional fields following the standard SMF record header. If you are using a different SMF record type, ensure that you create the record definition with the correct SMF header.

fld1, fld2, and fldn

Specify the fields for the user data in the SMF record. Fields are separated by commas.

You can define multiple record definitions in the same data set member. Use a different SM127SRS value for each record.

4. In a data set member of the user concatenation library, create a custom update definition to collect the SMF user records.

The following code sample creates a member `USERID.LOCAL.DEFS(USRUS127)` for the update definition.

```
SET IBM_FILE = 'ABC01';

DEFINE UPDATE ABC_01
  VERSION 'CDP.210'
  FROM ABC_01
  TO &IBM_UPDATE_TARGET
  &IBM_CORRELATION
  AS &IBM_FILE_FORMAT SET(ALL);
```

IBM_FILE

Set a different value of `IBM_FILE` for each update definition. This value must be a valid DD name.

DEFINE UPDATE

Specifies the update definitions name. The name can be the same as the custom record definition, but it must be different from other update definitions.

FROM

Identifies the source of the update definition, which is the name of the record definition.

You can define multiple update definitions in the same data set member. Also, you can use the `WHERE` clause to select records to collect. For more information about the `WHERE` clause, see [“WHERE” on page 259](#).

5. If you want to filter the fields to be streamed, add a `DEFINE TEMPLATE` statement for the update definition in the same data set member of that update definition.

Verify that the template definition is placed after the update definition and that the template definition name is the same as the update definition name.

In the template definition you must include the `SM127TME` and `SM127DTE` fields in the SMF record header for timestamp resolution when you ingest data to analytics platforms.

For the language reference of the `DEFINE TEMPLATE` statement, see [“DEFINE TEMPLATE statement” on page 263](#).

6. Validate the syntax of the custom record, update, and, optionally, template definitions.

Use the following example job to verify the members for the custom definitions.

```
//HBOJBCOL JOB ( ), 'DUMMY',MSGCLASS=X,MSGLEVEL=(,0),
//          CLASS=A,NOTIFY=&SYSUID
//*
//HBOSMFCB EXEC PGM=HBOPDE,REGION=0M,PARM='SHOWINPUT=YES'
//STEPLIB DD DISP=SHR,DSN=hlq.SHB0LOAD
//HBOOUT DD SYSOUT=*
//HBODUMP DD SYSOUT=*
//HBOIN DD DISP=SHR,DSN=hlq.SHB0DEFS(HBOCCSV)
// DD DISP=SHR,DSN=hlq.SHB0DEFS(HBOCCORY)
// DD DISP=SHR,DSN=hlq.SHB0DEFS(HBOLLSMF)
// DD DISP=SHR,DSN=USERID.LOCAL.DEFS(USRRS127)
// DD DISP=SHR,DSN=USERID.LOCAL.DEFS(USRUS127)
// DD *
COLLECT SMF
WITH STATISTICS
BUFFER SIZE 1 M;
//*
//HBOLOG DD DUMMY
```

hlq

Change `hlq` to the high-level qualifier for the IBM Z Common Data Provider SMP/E target data set.

```
// DD DISP=SHR,DSN=USERID.LOCAL.DEFS(USRRS127)
```

// DD DISP=SHR, DSN=USERID.LOCAL.DEFS (USRUS127)

Specify the data set members for the custom definitions. *USERID.LOCAL.DEFS* is the user concatenation library. *USRRS127* is the member that contains the record definitions. *USRUS127* is the member that contains the update and template definitions. Replace the values based on your configuration. Verify that the record definition member is included before the update definition member.

Important: Ensure that the definitions are error-free by running the validation job before you create the custom data stream.

Messages are in the output file that is defined by *HBOOUT*. If there is no syntax error, you see the following messages.

```
HB00125I ABC_01 was successfully defined.
HB00201I Update ABC_01 was successfully defined.
```

If there are syntax errors, correct the errors according to the messages in the output file.

7. Validate the data collection with the custom record, update, and, optionally, template definitions. Collect data from an SMF data set that contains your SMF records by using a batch System Data Engine job, and validate the data by reviewing the output data set.

Use the following example job to verify the data that is collected with the custom definitions.

```
//HBOJBCOL JOB ( ), ' DUMMY ', MSGCLASS=X, MSGLEVEL=( , 0 ),
//          CLASS=A, NOTIFY=&SYSUID
// *
//HBOSMFCB EXEC PGM=HBOPDE, REGION=OM, PARM=' ALLHDRS=YES '
//STEPLIB DD DISP=SHR, DSN=hlq.SHBOLoad
//HBOOUT DD SYSOUT=*
//HBODUMP DD SYSOUT=*
//HBOIN DD DISP=SHR, DSN=hlq.SHBODEFS (HBOCCSV)
// DD DISP=SHR, DSN=hlq.SHBODEFS (HBOCCORY)
// DD DISP=SHR, DSN=hlq.SHBODEFS (HBOLLSMF)
// DD DISP=SHR, DSN=USERID.LOCAL.DEFS (USRRS127)
// DD DISP=SHR, DSN=USERID.LOCAL.DEFS (USRUS127)
// DD *
COLLECT SMF
WITH STATISTICS
BUFFER SIZE 1 M;
/*
//HBOLOG DD DISP=SHR, DSN=HLQ.LOCAL.SMFLOGS
// *
//ABC01 DD DSN=USERID.ABC01.CSV,
// DISP=(NEW, CATLG, DELETE), SPACE=(CYL, (10, 10)),
// DCB=(RECFM=V, LRECL=32756)
```

hlq

Change *hlq* to the high-level qualifier for the IBM Z Common Data Provider SMP/E target data set.

// DD DISP=SHR, DSN=USERID.LOCAL.DEFS (USRRS127)**// DD DISP=SHR, DSN=USERID.LOCAL.DEFS (USRUS127)**

Specify the data set members for the custom definitions. *USERID.LOCAL.DEFS* is the user concatenation library. *USRRS127* is the member that contains the record definitions. *USRUS127* is the member that contains the update and template definitions. Replace the values based on your configuration. Ensure that the record definition member is included before the update definition member.

//HBOLOG DD DSN=

Specifies the SMF data set that contains your SMF records.

//ABC01 DD DSN=

Specifies the data set that stores the output data. Ensure that this value is the same as the value of the statement `SET IBM_FILE=` in the corresponding update definition. The output data set is a CSV file which you can download and open with spreadsheet applications for validation.

8. Create a custom System Data Engine data stream in the Configuration Tool based on the update definition and template definition that are created in previous steps.

Create a data stream for each update definition.

Verify that the data stream name, the custom update definition name, and the custom template definition name are the same, and that in the **SHBODEFS data set members** field you specify the member for the custom record definition before the member for the custom update and template definitions. For more information, see [“Creating a System Data Engine data stream definition”](#) on page 46.

9. Update your analytics platform so that it can process the new data stream.

- If you are ingesting data to the Elastic Stack, for each data stream, create a field name annotation configuration file, and a timestamp resolution configuration file in the Logstash configuration directory.

Field name annotation configuration file

The file is named `H_data_stream_name.lsh`, for example, `H_ABC_01.lsh`. See the following example of the file:

```
# CDPz ELK Ingestion
#
# Field Annotation for stream zOS-ABC_01
#

filter {
  if [sourceType] == "zOS-ABC_01" {

    csv{ columns => [ "Correlator", "SM127LEN", "SM127SEG",
"SM127FLG", "SM127RTY", "SM127TME", "SM127DTE", "SM127SID",
"SM127SSI", "SM127STY", "SM127SRC", "SM127SRS", "fld1", "fld2",
"fldn" ]
      separator => "," }

  }
}
```

sourceType

The value of `sourceType` must match the data source type of the data stream. The naming convention is `zOS-data_stream_name`.

```
if [sourceType] == "zOS-ABC_01"
```

fld1, fld2, and fldn

Replace these values with the fields in your custom record definition. If you have a custom template definition, change the column list to match the fields and order in the template definition. Keep `Correlator` as the first column in the list.

Timestamp resolution configuration file

The file is named `N_data_stream_name.lsh`, for example, `N_ABC_01.lsh`. See the following example of the file:

```
# CDPz ELK Ingestion
#
# Timestamp Extraction for stream zOS-ABC_01
#

filter {
  if [sourceType] == "zOS-ABC_01" {
    mutate{ add_field => {
      "[@metadata][timestamp]" => "%{SM127DTE} %{SM127TME}"
    }}

    date{ match => [
      "[@metadata][timestamp]", "yyyy-MM-dd HH:mm:ss:SS"
    ]}

  }
}
```

sourceType

The value of `sourceType` must match the data source type of the data stream. The naming convention is `zOS-data_stream_name`.

```
if [sourceType] == "zOS-ABC_01"
```

add_field =>

For an SMF record, you must specify the date and time fields in the SMF record header. In this example, the fields are SM127DTE and SM127TME.

Restart Logstash after you create the files for the new data stream. Refer to Logstash documentation for more information about the configuration files.

- If you are ingesting data to Splunk, define the layout of the data stream to the Splunk server by creating the `props.conf` file in the `Splunk_Home/etc/apps/ibm_cdpz_buffer/local` directory on the Splunk server. If the `props.conf` file exists, append the following content to the file.

```
#
# ABC_01
#

[zOS-ABC_01]
TIMESTAMP_FIELDS = SM127DTE, SM127TME, timezone
TIME_FORMAT= %F %H:%M:%S:%Q %z
FIELD_NAMES = "sysplex","system","hostname","","","sourcename","timezone",
"Correlator", "SM127LEN", "SM127SEG", "SM127FLG", "SM127RTY", "SM127TME",
"SM127DTE", "SM127SID", "SM127SSI", "SM127STY", "SM127SRC", "SM127SRS",
"f1d1", "f1d2", "f1dn"
INDEXED_EXTRACTIONS = csv
KV_MODE = none
NO_BINARY_CHECK = true
SHOULD_LINEMERGE = false
category = Structured
disabled = false
pulldown_type = true
```

[zOS-ABC_01]

You must specify the data source name of the data stream. The naming convention is `zOS-data_stream_name`.

TIMESTAMP_FIELDS

For an SMF record, you must specify the date and time fields in the SMF record header. In this example, the fields are SM127DTE and SM127TME.

FIELD_NAMES


Replace `f1d1`, `f1d2`, and `f1dn` with the fields in your custom record definition. If you have a custom template definition, change the column list to match the fields and order in the template definition. If the column `Correlator` exists, do not remove it.

In the Splunk user interface, you must also configure the file to data source type mapping for the new data stream. The file that the Data Receiver saves is named `zOS-data_stream_name-*.cdp`. For example, the data stream `ABC_01` has the file that is named `CDP-zOS-ABC_01-*.cdp`.

Restart the Splunk server after you make the changes.

Refer to Splunk documentation for more information.

10. Create or update the policy to add the new System Data Engine data stream.

- a) In the Configuration Tool primary window, create a new policy or select the policy that you want to update.
- b) Click the **Add Data Stream** icon  **DATA STREAM** in the **Policy Profile Edit** window.
- c) Find and select the new data stream from the list in the **select data stream** window.
- d) Assign a subscriber for each new data stream.
- e) In the **Policy Profile Edit** window, click **SYSTEM DATA ENGINE** to ensure that values are provided for **USER Concatenation** and **CDP Concatenation** fields, and click **OK**. Complete the field **USER Concatenation** with the data set name of your user concatenation library. Based on previous examples, `USERID.LOCAL.DEFS` should be specified for the field.
- f) Click **Save** to save the policy.

Important: Each time that the associated record definition or update definition is changed, you must edit and save the policy in the Configuration Tool so that the changes are reflected in the policy.

For more information on how to update a policy, see [“Updating a policy” on page 43](#).

11. Restart the Data Streamer and the System Data Engine.

Streaming key performance indicators for CICS Transaction Server for z/OS monitoring

IBM Z Common Data Provider uses SMF_110_1_KPI to collect key performance indicators for CICS® Transaction Server for z/OS monitoring. In addition to the fields in SMF_110_1_KPI, you can use DEFINE TEMPLATE to stream more data fields.

Before you begin

For more information about the content of SMF_110_1_KPI data stream, see [“SMF_110_1_KPI data stream content” on page 176](#). For more information about the fields in SMF_110_1_KPI, see [Table 24 on page 177](#).

About this task

You can create DEFINE TEMPLATE statement to filter more fields of SMF_110_1_KPI records and customize the data streams to stream the fields. After that, create and update the policy in the Configuration Tool to include the data stream.

Procedure

1. If you do not already have one, create a partitioned data set (PDS) that is used as the user concatenation library for the custom template definition.
For more information about how to create the partitioned data set, see step [“1.a” on page 46](#) in [“Creating a System Data Engine data stream definition” on page 46](#).
2. Copy the sample update and template definitions from the member HBOUUKPI of the SMP/E target data set *hlq*.SHBODEFS to the user concatenation library, and edit the definitions based on your requirements.

The following example shows how to define an update and a template definitions for filtering more fields of SMF_110_1_KPI records based on the sample template definition.

```
SET IBM_FILE = 'SMF110xx';

DEFINE UPDATE SMF_110_1_CUST
  VERSION 'CDP.210'
  FROM SMF_CICS_T
  TO &IBM_UPDATE_TARGET
  AS &IBM_FILE_FORMAT SET(ALL);

DEFINE TEMPLATE SMF_110_1_CUST FOR SMF_110_1_CUST
  ORDER
  (SMFMNTME,
   SMFMNDTE,
   fld1,
   fld2,
   .....
   fldn)
  AS &IBM_FILE_FORMAT;
```

SET

The SET statement is needed only when the target of the data stream is a file, which means the variable *IBM_UPDATE_TARGET* is set to FILE &IBM_FILE.

DEFINE UPDATE

The custom update definition name must be unique among update definitions. For the language reference of the DEFINE UPDATE statement, see [“DEFINE UPDATE statement”](#) on page 258.

```
DEFINE UPDATE SMF_110_1_CUST
```

You can change the value of *CUST* in *SMF_110_1_CUST* according to your needs.

DEFINE TEMPLATE

For filtering more fields of *SMF_110_1_KPI* records, add a DEFINE TEMPLATE statement for the update definition in the same data set member of that update definition. The template definition name must be the same as the update definition name to replace the default template definition that streams all fields for the update definition.

For versions before 4Q2019 PTF, in the template definition, you must include the date *SMFMNDTE* and time *SMFMNTME* fields from the SMF record header of *SMF_CICS_T*. These fields are required for timestamp resolution when you ingest data to your analytics platform.

fld1, fld2, fldn

This section defines the fields in *SMF_110_1_KPI* record. These fields are separated by commas. You can select the fields that are listed in [“Fields for SMF_110_1_CUST data stream”](#) on page 71.

For the language reference of the DEFINE TEMPLATE statement, see [“DEFINE TEMPLATE statement”](#) on page 263.

3. Validate the syntax of the custom update and template definitions.

Use the following example job to verify the member of the custom definitions.

```
//HBOJBCOL JOB (), 'DUMMY',MSGCLASS=X,MSGLEVEL=(,0),
//          CLASS=A,NOTIFY=&SYSUID
//*
//HBOSMFCB EXEC PGM=HBOPDE,REGION=0M,PARM='SHOWINPUT=YES'
//STEPLIB DD DISP=SHR,DSN=h1q.SHBOLoad
//HBOOUT DD SYSOUT=*
//HBODUMP DD SYSOUT=*
//HBOIN DD DISP=SHR,DSN=h1q.SHBODEFS(HBOCCSV)
// DD DISP=SHR,DSN=h1q.SHBODEFS(HBOCCORY)
// DD DISP=SHR,DSN=h1q.SHBODEFS(HBOLLSMF)
// DD DISP=SHR,DSN=h1q.SHBODEFS(HBOTCIFI)
// DD DISP=SHR,DSN=h1q.SHBODEFS(HBORS110)
// DD DISP=SHR,DSN=USERID.LOCAL.DEFS(HBOUUKPI)
// DD *
COLLECT SMF
WITH STATISTICS
BUFFER SIZE 1 M;
//*
//HBOLOG DD DUMMY
```

hlq

Change the *hlq* to the high-level qualifier for the IBM Z Common Data Provider SMP/E target data set.

// DD DISP=SHR,DSN=USERID.LOCAL.DEFS(HBOUUKPI)

Specifies the data set member for the custom definitions. *USERID.LOCAL.DEFS* is the user concatenation library. *HBOUUKPI* is the member that contains the update and template definitions. Replace the values based on your configuration.

Important: Ensure that the definitions are error-free by running the validation job before you create the custom data stream.

Messages are in the output file that is defined by *HBOOUT*.

If there is no syntax error, you see the following messages.

```
HB00201I Update SMF_110_1_CUST was successfully defined.
HB00500I Template SMF_110_1_CUST was successfully defined.
```

If there are syntax errors, correct the errors according to the messages in the output file.

4. Validate the data collection with the custom update and template definitions.

Collect data from an SMF data set that contains SMF type 110 subtype 1 records by using a batch System Data Engine job, and validate the data by reviewing the output data set.

Use the following example job to verify the data that is collected with the custom definitions.

```
//HBOJBCOL JOB ( ), 'DUMMY',MSGCLASS=X,MSGLEVEL=(,0),
//          CLASS=A,NOTIFY=&SYSUID
//*
//HBOSMFCB EXEC PGM=HBOPDE,REGION=0M,PARM=' ALLHDRS=YES'
//STEPLIB DD DISP=SHR,DSN=hlq.SHBLOAD
//HBOOUT DD SYSOUT=*
//HBOUMP DD SYSOUT=*
//HBOIN DD DISP=SHR,DSN=hlq.SHBODEFS(HBOCCSV)
// DD DISP=SHR,DSN=hlq.SHBODEFS(HBOCCORY)
// DD DISP=SHR,DSN=hlq.SHBODEFS(HBOLLSMF)
// DD DISP=SHR,DSN=hlq.SHBODEFS(HBOTCIFI)
// DD DISP=SHR,DSN=hlq.SHBODEFS(HBORS110)
// DD DISP=SHR,DSN=USERID.LOCAL.DEFS(HBOUUKPI)
// DD *
COLLECT SMF
WITH STATISTICS
BUFFER SIZE 1 M;
/*
//HBOLOG DD DISP=SHR,DSN=HLQ.LOCAL.SMFLOGS
//*
//SMF110xx DD DSN=USERID.SMF110xx.CSV,
//          DISP=(NEW,CATLG,DELETE),SPACE=(CYL,(10,10)),
//          DCB=(RECFM=V,LRECL=32756)
```

hlq

Change *hlq* to the high-level qualifier for the IBM Z Common Data Provider SMP/E target data set.

// DD DISP=SHR,DSN= USERID.LOCAL.DEFS(HBOUUKPI)

Specifies the data set member for the custom definitions. *USERID.LOCAL.DEFS* is the user concatenation library. *HBOUUKPI* is the member that contains the update and template definitions. Replace the values based on your configuration. Ensure that the record definition member is included before the update definition member.

//HBOLOG DD DSN=

Specifies the SMF data set that contains your SMF records.

//SMF110xx DD DSN=

Specifies the data set that stores the output data. Ensure that this value is the same as the value of the statement `SET IBM_FILE=` in the corresponding update definition. The output data set is a CSV file which you can download and open with spreadsheet applications for validation.

5. Create a custom System Data Engine data stream named `SMF_110_1_CUST` in the Configuration tool.

For more information about how to create the custom System Data Engine data stream, see [“Creating a System Data Engine data stream definition”](#) on page 46.

Verify that the data stream name, the custom update definition name, and the custom template definition name are the same.

Fill in the **SHBODEFS data set members** field as:

```
HBOLLSMF
HBORS110
HBOTCIFI
HBOUUKPI
```

6. Update your analytics platform so that it can process the new data stream.

- If you are ingesting `SMF_110_1_CUST` data to the Elastic Stack, for each data stream, create a field name annotation configuration file, and a timestamp resolution configuration file in the Logstash configuration directory.

Field name annotation configuration file

The file is named `H_SMF_110_1_CUST.lsh`. Here is an example of the file:

```
# CDPz ELK Ingestion
#
# Field Annotation for stream zOS-SMF_110_1_CUST
```

```
#
filter {
  if [sourceType] == "zOS-SMF_110_1_CUST" {
    csv{ columns => [ "Correlator", " SMFMNTME", "SMFMNDTE", "fld1", "fld2",
"fldn" ]
      separator => "," }
  }
}
```

Make sure the value of *CUST* in *SMF_110_1_CUST* is the same as the value that is specified for the update definition name.

sourceType

The value of *sourceType* must match the data source type of the data stream. The naming convention is *zOS-SMF_110_1_CUST*.

```
if [sourceType] == "zOS-SMF_110_1_CUST"
```

fld1, fld2, and fldn

Replace *fld1*, *fld2*, and *fldn* with the fields and order in your custom define template definition. Keep *Correlator* as the first column in the list.

Timestamp resolution configuration file

The file is named *N_SMF_110_1_CUST.lsh*. Here is an example of the file:

```
# CDPz ELK Ingestion
#
# Timestamp Extraction for stream zOS-SMF_110_1_CUST
#
filter {
  if [sourceType] == "zOS-SMF_110_1_CUST" {
    mutate{ add_field => {
      "[@metadata][timestamp]" => "%{SMFMNDTE} %{SMFMNTME}"
    }}
    date{ match => [
      "[@metadata][timestamp]", "yyyy-MM-dd HH:mm:ss:SS"
    ]}
  }
}
```

Make sure the value of *CUST* in *SMF_110_1_CUST* is the same as the value that is specified for the update definition name.

sourceType

The value of *sourceType* must match the data source type of the data stream. The naming convention is *zOS-SMF_110_1_CUST*.

```
if [sourceType] == "zOS-SMF_110_1_CUST"
```

Restart Logstash after you create the files for the new data stream. Refer to Logstash documentation for more information about the configuration files.

- If you are ingesting *SMF_110_1_CUST* data to Splunk, define the layout of the data stream to the Splunk server by creating the *props.conf* file in the *Splunk_Home/etc/apps/ibm_cdpz_buffer/local* directory on the Splunk server. If the *props.conf* file exists, append the following content to the file.

```
#
# SMF_110_1_CUST
#
[zOS-SMF_110_1_CUST]
TIMESTAMP_FIELDS = SMFMNDTE, SMFMNTME, timezone
TIME_FORMAT= %F %H:%M:%S:%2Q %z
FIELD_NAMES =
"sysplex","system","hostname","","","sourcename","timezone","Correlator","SMFMNTME","SMFMN
DTE","fld1","fld2","fldn"
INDEXED_EXTRactions = csv
KV_MODE = none
```

```
NO_BINARY_CHECK = true
SHOULD_LINEMERGE = false
category = Structured
disabled = false
pulldown_type = true
```

Make sure the value of *CUST* in *SMF_110_1_CUST* is the same as the value that is specified for the update definition name.

[zOS-SMF_110_1_CUST]


You must specify the data source name of the data stream. The naming convention is zOS-SMF_110_1_CUST.

FIELD_NAMES

Replace *f1d1*, *f1d2*, and *f1dn* with the fields and order in your custom template definition. If the column *Correlator* exists, do not remove it.

In the Splunk user interface, you must also configure the file to data source type mapping for the new data stream. The file that the Data Receiver saves is named *zOS-data_stream_name-*.cdp*. For example, the data stream *SMF_110_1_CUST* has the file that is named *CDP-zOS-SMF_110_1_CUST-*.cdp*.

Restart the Splunk server after you make the changes. Refer to Splunk documentation for more information.

7. Create or update the policy to add the new System Data Engine data stream **SMF_110_1_CUST**.
 - a) In the Configuration Tool primary window, create a new policy or select the policy that you want to update.
 - b) Click the **Add Data Stream** icon  **DATA STREAM** in the **Policy Profile Edit** window.
 - c) Find and select the new data stream from the list in the **select data stream** window.
 - d) Assign a subscriber for each new data stream.
 - e) In the **Policy Profile Edit** window, click **SYSTEM DATA ENGINE** to ensure that values are provided for **USER Concatenation** and **CDP Concatenation** fields, and click **OK**. Fill in the field **USER Concatenation** with the data set name of your user concatenation library.
 - f) Click **Save** to save the policy.

Important: Each time that the associated update definition or template definition is changed, you must edit and save the policy in the Configuration Tool so that the changes are reflected in the policy.

For more information on how to update a policy, see [“Updating a policy” on page 43](#).

8. Restart the Data streamer and the System Data Engine.

Fields for SMF_110_1_CUST data stream

For *SMF_110_1_CUST* data stream, this reference lists the fields that you can stream and includes a brief description of the fields. You can use `DEFINE TEMPLATE` statement to customize data streams to stream more data fields.

The following tables provide fields that you can select for *SMF_110_1_CUST* data stream:

Table 7. Fields generated by CICS with default dictionary

Field name	Description	Dictionary entry ID
SMFMNLEN	Record length. This field and the next field (total of four bytes) form the RDW (record descriptor word).	SMF header
SMFMNSEG	Segment descriptor	
SMFMNFLG	System indicator. Turn on bit 1 (x'40') indicating record with subtypes.	
SMFMNRTY	Record type	
SMFMNTME	Time since midnight, in hundredths of a second, that the record was moved into the SMF buffer.	
SMFMNDTE	Date when the record was moved into the SMF buffer, in the form 0cyydddf.	
SMFMNSID	System identification (from the SID parameter)	
SMFMNSSI	Subsystem identification	
SMFMNSTY	Record subtype	
SMFMNTRN	Number Of triplets	
SMFMNAPS	Offset to product section	
SMFMNLPS	Length of product section	
SMFMNPNPS	Number of product sections	
SMFMNASS	Offset to data section	
SMFMNASL	Length of data section	
SMFMNASN	Number of data sections	
SMFMNRVN	Record version(CICS)	
SMFMNRVN_CHAR	Record version(CICS)	
SMFMNPRN	Product name (Generic APPLID)	
SMFMNSPN	Product name (Specific APPLID)	
SMFMNMFL	Record maintenance indicator	
SMFMNCL	Class of data	
SMFMNDCA	Offset to CICS field connectors	
SMFMNDCL	Length of each CICS field connector	
SMFMNDCN	Number OF CICS field connector	
SMFMNDRA	Offset to first CICS data record	
SMFMNDRL	Length of each CICS data record	
SMFMNDRN	Number of CICS data records	
SMFMNTAD	Local TOD clock adjustment value	
SMFMNLISO	Leap second offset TOD format	
SMFMNDTO	Local TIME/DATE offset	
SMFMNJBN	JOB name	
SMFMNRSD	JOB date	
SMFMSRST	JOB time	
SMFMNUIF	User identification	
SMFMNPDN	Operating system product level	
START	The start time of the transaction. The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.	DFHCICS T005
STOP	The stop time of the transaction. The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.	DFHCICS T006
TRAN	CICS Transaction Server for z/OS transaction ID. The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.	DFHTASK C001
TERM	Terminal identification	DFHTERM C002
USERID	Current CICS Transaction Server for z/OS user ID. The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.	DFHCICS C089
T_CHAR	Transaction type in CHAR format	DFHTASK C004
T_HEX	Transaction type in HEX format	

Table 7. Fields generated by CICS with default dictionary (continued)		
Field name	Description	Dictionary entry ID
TRANNUM	CICS Transaction Server for z/OS transaction number. The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.	DFHTASK P031
TRANNUM_CHAR	CICS Transaction Server for z/OS transaction number. The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.	
TRANPRI	Transaction priority	DFHTASK C109
LUNAME	The z/OS Communications Server SNA logical unit name (if available) of the terminal that is associated with this transaction. If the task is executing in an application-owning or file-owning region, the LUNAME is the generic applid of the originating connection for MRO, LUTYPE6.1, and LUTYPE6.2 (APPC). The LUNAME is blank if the originating connection is an external CICS interface (EXCI).	DFHTERM C111
PGMNAME	CICS Transaction Server for z/OS program name. The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.	DFHPROG C071
NETNAME	Network Unit-of-Work Netname	DFHTASK C097
UOWID1	Network Unit-of-Work Instance/Seqno: First 6 bytes in HEX	DFHTASK C098
UOWID2	Network Unit-of-Work Instance/Seqno: Last 2 bytes in binary	
TASKFLAG	Transaction error flags	DFHTASK A064
ABCODEO	Original CICS Transaction Server for z/OS abend code. The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.	DFHPROG C113
ABCODEC	Current CICS Transaction Server for z/OS abend code. The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.	DFHPROG C114
RTYPE	Record type	DFHCICS C112
TCMSGIN1	Primary TC messages - in	DFHTERM A034
TCCHRIN1	Primary TC characters - in	DFHTERM A083
TCMSGOU1	Primary TC messages - out	DFHTERM A035
TCCHROU1	TCCHROU1 - Primary TC characters - out	DFHTERM A084
TCMSGIN2	Secondary TC messages - in	DFHTERM A067
TCCHRIN2	Secondary TC characters - in	DFHTERM A085
TCMSGOU2	Secondary TC messages - out	DFHTERM A068
TCCHROU2	Secondary TC characters - out	DFHTERM A086
TCALLOCT	No. TCTTE allocate requests	DFHTERM A069
TCSTG	Terminal storage allocated to the terminal for one transaction	DFHSTOR A104
SCUGETCT	Number of user-storage GETMAIN requests issued by the user task for storage below the 16 MB line, in the UDSA.	DFHSTOR A054
SCUGETCTE	Number of user-storage GETMAIN requests issued by the user task for storage above the 16 MB line, in the extended user dynamic storage area (EUDSA).	DFHSTOR A105
SCCGETCT	Number of user-storage GETMAIN requests issued by the user task for storage below the 16 MB line, in the CDSA.	DFHSTOR A117
SCCGETCTE	Number of user-storage GETMAIN requests issued by the user task for storage above the 16 MB line, in the ECDSA.	DFHSTOR A120
SCUSRHWM	Maximum amount (high-water mark) of user storage allocated to the user task below the 16 MB line, in the user dynamic storage area (UDSA).	DFHSTOR A033
SCUSRHWME	Maximum amount (high-water mark) of user storage allocated to the user task above the 16 MB line, in the EUDSA.	DFHSTOR A106
SC24CHWM	Maximum amount (high-water mark) of user storage allocated to the user task below the 16 MB line, in the CICS dynamic storage area (CDSA).	DFHSTOR A116
SC31CHWM	Maximum amount (high-water mark) of user storage allocated to the user task above the 16 MB line, in the extended CICS dynamic storage area (ECDSA).	DFHSTOR A119
SCUSRSTG	Storage occupancy of the user task below the 16 MB line, in the UDSA. This measures the area under the curve of storage in use against elapsed time. For more information about storage occupancy, see Storage occupancy counts.	DFHSTOR A095
SCUSRSTGE	Storage occupancy of the user task above the 16 MB line, in the EUDSA. This measures the area under the curve of storage in use against elapsed time. For more information, see Storage occupancy counts.	DFHSTOR A107
SC24COCC	Storage occupancy of the user task below the 16 MB line, in the CDSA. This measures the area under the curve of storage in use against elapsed time. For more information, see Storage occupancy counts.	DFHSTOR A118
SC31COCC	Storage occupancy of the user task above the 16 MB line, in the ECDSA. This measures the area under the curve of storage in use against elapsed time. For more information, see Storage occupancy counts.	DFHSTOR A121

Table 7. Fields generated by CICS with default dictionary (continued)		
Field name	Description	Dictionary entry ID
PCSTGHWM	Maximum amount (high-water mark) of program storage in use by the user task both above and below the 16 MB line	DFHSTOR A087
PC31AHWM	Maximum amount (high-water mark) of program storage in use by the user task above the 16 MB line. This field is a subset of PCSTGHWM (field id 087) that resides above the 16 MB line.	DFHSTOR A139
PC24BHWM	Maximum amount (high-water mark) of program storage in use by the user task below the 16 MB line. This field is a subset of PCSTGHWM (field id 087) that resides below the 16 MB line.	DFHSTOR A108
PC31UHWM	Program storage above the 16MB line in the extended user dynamic storage area (EUDSA), in bytes.	DFHSTOR A140
PC24UHWM	Program storage below the 16MB line in the user dynamic storage area (UDSA), in bytes.	DFHSTOR A141
PC31CHWM	Maximum amount (high-water mark) of program storage in use by the user task above the 16 MB line, in the extended CICS dynamic storage area (ECDSA). This field is a subset of PC31AHWM (139) that resides in the ECDSA.	DFHSTOR A142
PC24CHWM	Maximum amount (high-water mark) of program storage in use by the user task below the 16 MB line, in the CICS dynamic storage area (CDSA). This field is a subset of PC24BHWM (108) that resides in the CDSA.	DFHSTOR A143
PC31RHWM	Maximum amount (high-water mark) of program storage in use by the user task above the 16 MB line, in the extended read-only dynamic storage area (ERDSA). This field is a subset of PC31AHWM (field id 139) that resides in the ERDSA.	DFHSTOR A122
FCGETCT	Number of file GET requests issued by the user task	DFHFILE A036
FCPUTCT	Number of file PUT requests issued by the user task	DFHFILE A037
FCBRWCT	Number of file browse requests issued by the user task. This number excludes the START and END browse requests.	DFHFILE A038
FCADDCT	Number of file ADD requests issued by the user task	DFHFILE A039
FCDELCT	Number of file DELETE requests issued by the user task	DFHFILE A040
FCTOTCT	Total number of file control requests issued by the user task. This number excludes any request for OPEN, CLOSE, ENABLE, or DISABLE of a file.	DFHFILE A093
FCAMCT	Number of times the user task invoked file access-method interfaces. This number excludes requests for OPEN and CLOSE.	DFHFILE A070
TDGETCT	Number of transient data GET requests issued by the user task	DFHDEST A041
TDPUTCT	Number of transient data PUT requests issued by the user task	DFHDEST A042
TDPURCT	Number of transient data PURGE requests issued by the user task	DFHDEST A043
TDTOTCT	Total number of transient data requests issued by the user task. This field is the sum of TDGETCT, TDPUTCT, and TDPURCT.	DFHDEST A091
TSGETCT	Number of temporary storage GET requests to auxiliary or main temporary storage issued by the user task	DFHTEMP A044
TSPUTACT	Number of PUT requests to auxiliary temporary storage issued by the user task	DFHTEMP A046
TSPUTMCT	Number of PUT requests to main temporary storage issued by the user task	DFHTEMP A047
TSTOTCT	Total number of temporary storage requests issued by the user task. This field is the sum of the temporary storage READQ (TSGETCT), READQ shared (TSGETSCT), WRITEQ AUX (TSPUTACT), WRITEQ MAIN (TSPUTMCT), WRITEQ shared (TSPUTSCT), and DELETEQ requests issued by the user task.	DFHTEMP A092
BMSMAPCT	Number of BMS MAP requests issued by the user task. This field corresponds to the number of RECEIVE MAP requests that did not incur a terminal I/O, and the number of RECEIVE MAP FROM requests.	DFHMAPP A050
BMSINCT	Number of BMS IN requests issued by the user task. This field corresponds to the number of RECEIVE MAP requests that incurred a terminal I/O.	DFHMAPP A051
BMSOUTCT	Number of BMS OUT requests issued by the user task. This field corresponds to the number of SEND MAP requests.	DFHMAPP A052
BMSTOTCT	Total number of BMS requests issued by the user task. This field is the sum of BMS RECEIVE MAP, RECEIVE MAP FROM, SEND MAP, SEND TEXT, and SEND CONTROL requests issued by the user task.	DFHMAPP A090
PCLINKCT	Number of program LINK and INVOKE APPLICATION requests issued by the user task, including the link to the first program of the user task. This field does not include program LINK URM (user-replaceable module) requests.	DFHPROG A055
PCXCTLCT	Number of program XCTL requests issued by the user task	DFHPROG A056
PCLOADCT	Number of program LOAD requests issued by the user task	DFHPROG A057
JCPUWRCT	Number of journal write requests issued by the user task	DFHJOUR A058
ICPUINCT	Number of interval control START or INITIATE requests during the user task	DFHTASK A059
SPSYNCT	The total number of syncpoint requests that are issued by the user task. The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.	DFHSYNC A060
TCLNAME	Transaction class name. This field is null if the transaction is not in a TRANCLASS.	DFHTASK C166

Table 7. Fields generated by CICS with default dictionary (continued)		
Field name	Description	Dictionary entry ID
RSYSID	The name (SYSID) of the remote system to which this transaction was routed either statically or dynamically	DFHCICS C130
TCM62IN2	Number of messages received from the alternate facility by the user task for LUTYPE6.2 (APPC) sessions	DFHTERM A135
TCM62OU2	Number of messages sent to the alternate facility by the user task for LUTYPE6.2 (APPC) sessions	DFHTERM A136
TCC62IN2	Number of characters received from the alternate facility by the user task for LUTYPE6.2 (APPC) sessions	DFHTERM A137
TCC62OU2	Number of characters sent to the alternate facility by the user task for LUTYPE6.2 (APPC) sessions	DFHTERM A138
PC24SHWM	Maximum amount (high-water mark) of program storage in use by the user task below the 16 MB line, in the shared dynamic storage area (SDSA). This field is a subset of PC24BHWM (108) that resides in the SDSA.	DFHSTOR A160
PC31SHWM	Maximum amount (high-water mark) of program storage in use by the user task above the 16 MB line, in the extended shared dynamic storage area (ESDSA). This field is a subset of PC31AHWM (139) that resides in the ESDSA.	DFHSTOR A161
PC24RHWM	Maximum amount (high-water mark) of program storage in use by the user task below the 16 MB line, in the read-only dynamic storage area (RDSA). This field is a subset of PC24BHWM (108) that resides in the RDSA.	DFHSTOR A162
SZALLOCT	Number of conversations allocated by the user task. This number is incremented for each FEPI ALLOCATE POOL or FEPI CONVERSE POOL.	DFHFEPI A150
SZRCVCT	Number of FEPI RECEIVE requests made by the user task. This number is also incremented for each FEPI CONVERSE request.	DFHFEPI A151
SZSENDCT	Number of FEPI SEND requests made by the user task. This number is also incremented for each FEPI CONVERSE request.	DFHFEPI A152
SZTRTCT	Number of FEPI START requests made by the user task	DFHFEPI A153
SZCHROUT	Number of characters sent through FEPI by the user task	DFHFEPI A154
SZCHRIN	Number of characters received through FEPI by the user task	DFHFEPI A155
SZALLCTO	Number of times the user task timed out while waiting to allocate a conversation	DFHFEPI A157
SZRCVTO	Number of times the user task timed out while waiting to receive data	DFHFEPI A158
SZTOTCT	Total number of all FEPI API and SPI requests made by the user task	DFHFEPI A159
PERRECNT	The number of performance class records written by the CICS Monitoring Facility (CMF) for the user task	DFHCICS A131
SRVCLASS	The z/OS Workload Manager (WLM) service class for this transaction. This field is null if no transaction classification rules are defined for CICS subsystems in the active z/OS Workload Manager (WLM) service policy, or if the transaction was WLM-classified in another CICS region.	DFHCICS C167
RPTCLASS	The z/OS Workload Manager (WLM) report class for this transaction. This field is null if no transaction classification rules are defined for CICS subsystems in the active z/OS Workload Manager (WLM) service policy, or if the transaction was WLM-classified in another CICS region.	DFHCICS C168
LOGWRTCT	Number of CICS log stream write requests issued by the user task	DFHJOUR A172
SC24SGCT	Number of storage GETMAIN requests issued by the user task for shared storage below the 16 MB line, in the CDSA or SDSA.	DFHSTOR A144
SC24GSHR	Number of bytes of shared storage obtained by the user task by using a GETMAIN request below the 16 MB line, in the CDSA or SDSA.	DFHSTOR A145
SC24FSHR	Number of bytes of shared storage released by the user task by using a FREEMAIN request below the 16 MB line, in the CDSA or SDSA.	DFHSTOR A146
SC31SGCT	Number of storage GETMAIN requests issued by the user task for shared storage above the 16 MB line, in the ECDSA or ESDSA.	DFHSTOR A147
SC31GSHR	Number of bytes of shared storage obtained by the user task by using a GETMAIN request above the 16 MB line, in the ECDSA or ESDSA.	DFHSTOR A148
SC31FSHR	Number of bytes of shared storage released by the user task by using a FREEMAIN request above the 16 MB line, in the ECDSA or ESDSA.	DFHSTOR A149
RMUOWID	The identifier of the unit of work (unit of recovery) for this task. Unit of recovery values are used to synchronize recovery operations among CICS and other resource managers, such as IMS and Db2.	DFHTASK T132
FCTYNAME	Transaction facility name. This field is null if the transaction is not associated with a facility. The transaction facility type (if any) can be identified using byte 0 of the transaction flags, TRANFLAG, (164) field.	DFHTASK C163
TRANFLAG	Transaction flags, a string of 64 bits used for signaling transaction definition and status information.	DFHTASK A164
TERMINFO	Terminal or session information for the principal facility of this task, as identified in the 'TERM' field id 002. This field is null if the task is not associated with a terminal or session facility.	DFHTERM A165
TERMCNNM	Terminal session connection name. If the terminal facility associated with this transaction is a session, this field is the name of the owning connection (sysid). A terminal facility can be identified as a session by using byte 0 of the terminal information, TERMINFO (165), field. If the value is x'02', the terminal facility is a session.	DFHTERM C169

Table 7. Fields generated by CICS with default dictionary (continued)

Field name	Description	Dictionary entry ID
PCLURMCT	Number of program LINK URM (user-replaceable module) requests issued by, or on behalf of, the user task.	DFHPROG A072
ICTOTCT	Total number of Interval Control Start, Cancel, Delay, and Retrieve requests issued by the user task.	DFHTASK A066
BRDGTRAN	Bridge listener transaction identifier. For CICS 3270 Bridge transactions, this field is the name of the Bridge listener transaction that attached the user task.	DFHTASK C124
PRCSNAME	The name of the CICS business transaction service (BTS) process of which the user task formed part	DFHCBTS C200
PRCSTYPE	The process-type of the CICS BTS process of which the user task formed part	DFHCBTS C201
PRCSID	The CICS-assigned identifier of the CICS BTS root activity that the user task implemented	DFHCBTS C202
ACTVTYID	The CICS-assigned identifier of the CICS BTS activity that the user task implemented	DFHCBTS C203
ACTVTYNM	The name of the CICS BTS activity that the user task implemented	DFHCBTS C204
BARSYNCT	The number of CICS BTS run process, or run activity, requests that the user task made in order to execute a process or activity synchronously.	DFHCBTS A205
BARASYCT	The number of CICS BTS run process, or run activity, requests that the user task made in order to execute a process or activity asynchronously.	DFHCBTS A206
BALKPACT	The number of CICS BTS link process, or link activity, requests that the user task issued.	DFHCBTS A207
BADPROCT	The number of CICS BTS define process requests issued by the user task	DFHCBTS A208
BADACTCT	The number of CICS BTS define activity requests issued by the user task	DFHCBTS A209
BARSPACT	The number of CICS BTS reset process and reset activity requests issued by the user task	DFHCBTS A210
BASUPACT	The number of CICS BTS suspend process, or suspend activity, requests issued by the user task.	DFHCBTS A211
BARMPACT	The number of CICS BTS resume process, or resume activity, requests issued by the user task.	DFHCBTS A212
BADCPCACT	The number of CICS BTS delete activity, cancel process, or cancel activity, requests issued by the user task.	DFHCBTS A213
BAACQPCT	The number of CICS BTS acquire process, or acquire activity, requests issued by the user task.	DFHCBTS A214
BATOTPCT	Total number of CICS BTS process and activity requests issued by the user task	DFHCBTS A215
BAPRDCT	The number of CICS BTS delete, get, move, or put, container requests for process data containers issued by the user task.	DFHCBTS A216
BAACDCT	The number of CICS BTS delete, get, move, or put, container requests for current activity data containers issued by the user task.	DFHCBTS A217
BATOCCT	Total number of CICS BTS delete, get, move, or put, process container and activity container requests issued by the user task.	DFHCBTS A218
BARATECT	The number of CICS BTS retrieve-reattach event requests issued by the user task	DFHCBTS A219
BADFIECT	The number of CICS BTS define-input event requests issued by the user task	DFHCBTS A220
BATIAECT	The number of CICS BTS DEFINE TIMER EVENT, CHECK TIMER EVENT, DELETE TIMER EVENT, and FORCE TIMER EVENT requests issued by the user task.	DFHCBTS A221
BATOTECT	Total number of CICS BTS event-related requests issued by the user task	DFHCBTS A222
CFCAPICT	Number of CICS OO foundation class requests, including the Java API for CICS (JCICS) classes, issued by the user task.	DFHCICS A025
IMSREQCT	The number of IMS (DBCTL) requests issued by the user task	DFHDATA A179
DB2REQCT	The total number of Db2 EXEC SQL and Instrumentation Facility Interface (IFI) requests issued by the user task	DFHDATA A180
DHCRECT	The number of document handler CREATE requests issued by the user task	DFHDOCH A226
DHINSCT	The number of document handler INSERT requests issued by the user task	DFHDOCH A227
DHSETCT	The number of document handler SET requests issued by the user task	DFHDOCH A228
DHRETCT	The number of document handler RETRIEVE requests issued by the user task	DFHDOCH A229
DHTOTCT	The total number of document handler requests issued by the user task	DFHDOCH A230
DHTOTDCL	The total length of all documents created by the user task	DFHDOCH A240
PCDPLCT	Number of distributed program link (DPL) requests issued by the user task. For a breakdown by program name and system identifier (sysid) of the individual distributed program link (DPL) requests, you can request transaction resource monitoring. For more details, see Transaction resource class data: Listing of data fields.	DFHPROG A073
SOBYENCT	The number of bytes decrypted by the secure sockets layer for the user task	DFH SOCK A242
SOBYDECT	The number of bytes decrypted by the secure sockets layer for the user task	DFH SOCK A243
CLIPADDR	Client IP Address	DFH SOCK C244

<i>Table 7. Fields generated by CICS with default dictionary (continued)</i>		
Field name	Description	Dictionary entry ID
TRNGRPID	The transaction group ID is assigned at transaction attach time, and can be used to correlate the transactions that CICS runs for the same incoming work request.	DFHTASK C082
RRMSURID	RRMS/MVS unit-of-recovery ID (URID)	DFHTASK C190
TCBATTCT	The number of CICS TCBS attached by or on behalf of the user task	DFHTASK A251
WBRCVCT	The number of CICS web support RECEIVE requests issued by the user task	DFHWEBB A231
WBCHRIN	The number of bytes received by the CICS web support RECEIVE requests issued by the user task	DFHWEBB A232
WBSENDCT	The number of CICS web support SEND requests issued by the user task	DFHWEBB A233
WBCHROUT	The number of bytes sent by the CICS web support SEND requests issued by the user task	DFHWEBB A234
WBTOTCT	The total number of CICS web support requests issued by the user task	DFHWEBB A235
WBREPRCT	The number of reads from the repository in temporary storage issued by the user task	DFHWEBB A236
WBREPWCT	The number of writes to the repository in temporary storage issued by the user task	DFHWEBB A237
NETID	NETID if a network qualified name has been received from the Communications Server. If it is a resource and the network qualified name has not yet been received, NETID is 8 blanks. In all other cases, it is nulls.	DFHTERM C197
RLUNAME	Real network name if a network qualified name has been received from the Communications Server. In all other cases, this field is the same as LUNAME (field ID 111). For non-Communications Server resources, it is nulls.	DFHTERM C198
TCPSRVCE	The TCP/IP service name that attached the user task	DFH SOCK C245
PORTNUM	The TCP/IP port number of the TCP/IP service that attached the user task	DFH SOCK C246
OTSTID	This field is the first 128 bytes of the Object Transaction Service (OTS) Transaction ID (TID)	DFHTASK C194
WBEXTRCT	The number of CICS web support EXTRACT requests issued by the user task	DFHWEBB A238
WBBRWCT	The number of CICS web support browsing requests for HTTPHEADER, FORMFIELD, and QUERYPARM (STARTBROWSE, READNEXT, and ENDBROWSE) issued by the user task.	DFHWEBB A239
WBREADCT	The number of CICS web support READ HTTPHEADER, READ FORMFIELD, and READ QUERYPARM requests issued by the user task.	DFHWEBB A224
WBWRITCT	The number of CICS web support WRITE HTTPHEADER requests issued by the user task	DFH SOCK A225
SOEXTRCT	The number of EXTRACT TCPIP and EXTRACT CERTIFICATE requests issued by the user task	DFH SOCK A289
SOCNPST	The total number of requests made by the user task to create a nonpersistent outbound socket	DFH SOCK A290
SOCPSCT	The total number of requests made by the user task to create a persistent outbound socket	DFH SOCK A291
SONPSHWM	The peak number of nonpersistent outbound sockets owned by the user task	DFH SOCK A292
SOPSHWM	The peak number of persistent outbound sockets owned by the user task	DFH SOCK A293
SORCVCT	The total number of receive requests issued for outbound sockets (persistent and nonpersistent) by the user task	DFH SOCK A294
SOCHRIN	The total number of bytes received on outbound sockets by the user task	DFH SOCK A295
SOSENDCT	The total number of send requests issued for outbound sockets (persistent and nonpersistent) by the user task	DFH SOCK A296
SOCHROUT	The total number of bytes sent on outbound sockets by the user task	DFH SOCK A297
SOTOTCT	The total number of socket requests issued by the user task	DFH SOCK A298
SOMSGIN1	The number of inbound socket receive requests issued by the user task	DFH SOCK A301
SOCHRIN1	The number of characters received by inbound socket receive requests issued by the user task	DFH SOCK A302
SOMSGOU1	The number of inbound socket send requests issued by the user task	DFH SOCK A303
SOCHROU1	The number of characters sent by inbound socket send requests issued by the user task	DFH SOCK A304
DSTCBHWM	The peak number of CICS open TCBS (in TCB modes L8, L9, S8, T8, X8, and X9) that have been concurrently allocated to the user task	DFHTASK A252
CBSRVNRM	The CorbaServer for which this request processor instance is handling requests. Request processor transactions can be identified using byte 4 of the transaction flags, TRANFLAG (164), field.	DFHEJBS C311
EJBSACCT	The number of bean activations that have occurred in this request processor	DFHEJBS A312
EJBSPACT	The number of bean passivations that have occurred in this request processor	DFHEJBS A313
EJBRECT	The number of bean creation calls that have occurred in this request processor	DFHEJBS A314
EJBREMT	The number of bean removal calls that have occurred in this request processor	DFHEJBS A315
EJBMTHCT	The number of bean method calls executed in this request processor	DFHEJBS A316
EJBTOTCT	The total for this request processor of fields 312–316	DFHEJBS A317

<i>Table 7. Fields generated by CICS with default dictionary (continued)</i>		
Field name	Description	Dictionary entry ID
WBREDOCT	The number of CICS web support READ HTTPHEADER requests issued by the user task when CICS is an HTTP client	DFHWEBB A331
WBWRTOCT	The number of CICS web support WRITE HTTPHEADER requests issued by the user task when CICS is an HTTP client	DFHWEBB A332
WBRCVIN1	The number of CICS web support RECEIVE and CONVERSE requests issued by the user task when CICS is an HTTP client	DFHWEBB A333
WBCHRIN1	The number of bytes received by the CICS web support RECEIVE and CONVERSE requests issued by the user task when CICS is an HTTP client. This number includes the HTTP headers for the response.	DFHWEBB A334
WBSNDOU1	The number of CICS web support SEND and CONVERSE requests issued by the user task when CICS is an HTTP client	DFHWEBB A335
WBCHROU1	The number of bytes sent by the CICS web support SEND and CONVERSE requests issued by the user task when CICS is an HTTP client. This number includes the HTTP headers for the request.	DFHWEBB A336
WBPARSCT	The number of CICS web support PARSE URL requests issued by the user task	DFHWEBB A337
WBBRWOCCT	The number of CICS web support BROWSE HTTPHEADER requests (STARTBROWSE, READNEXT, and ENDBROWSE) issued by the user task when CICS is an HTTP client	DFHWEBB A338
WBIWBSCT	The number of EXEC CICS INVOKE SERVICE and EXEC CICS INVOKE WEBSERVICE requests issued by the user task	DFHWEBB A340
WBREPRDL	The total length, in bytes, of the data read from the repository in temporary storage by the user task.	DFHWEBB A341
WBREPWDL	The total length, in bytes, of the data written to the repository in temporary storage by the user task.	DFHWEBB A342
ICSTACCT	Total number of local interval control START requests, with the CHANNEL option, issued by the user task.	DFHTASK A065
ICSTACDL	Total length, in bytes, of the data in the containers of all the locally executed START CHANNEL requests issued by the user task. This total includes the length of any headers to the data.	DFHTASK A345
ICSTRCCT	Total number of interval control START CHANNEL requests, to be run on remote systems, issued by the user task.	DFHTASK A346
ICSTRCDL	Total length, in bytes, of the data in the containers of all the remotely executed START CHANNEL requests issued by the user task. This total includes the length of any headers to the data.	DFHTASK A347
PCDLCSDL	The total length, in bytes, of the data in the containers of all the distributed program link (DPL) requests issued with the CHANNEL option by the user task. This total includes the length of any headers to the data.	DFHPROG A286
PCDLCRDL	The total length, in bytes, of the data in the containers of all DPL RETURN CHANNEL commands issued by the user task. This total includes the length of any headers to the data.	DFHPROG A287
PCLNKCCT	Number of local program LINK and INVOKE APPLICATION requests, with the CHANNEL option, issued by the user task. This field is a subset of the program LINK and INVOKE APPLICATION requests field, PCLINKCT (055).	DFHPROG A306
PCXCLCCT	Number of program XCTL requests issued with the CHANNEL option by the user task. This field is a subset of the program XCTL requests field, PCXCTLCT (056).	DFHPROG A307
PCDPLCCT	Number of program distributed program link (DPL) requests issued with the CHANNEL option by the user task. This field is a subset of the distributed program link requests field, PCDPLCT (073).	DFHPROG A308
PCRTNCCT	Number of remote pseudoconversational RETURN requests, with the CHANNEL option, issued by the user task.	DFHPROG A309
PCRTNCDL	The total length, in bytes, of the data in the containers of all the remote pseudoconversational RETURN CHANNEL commands issued by the user task. This total includes the length of any headers to the data.	DFHPROG A310
PGTOTCCT	The number of CICS requests for channel containers issued by the user task	DFHCHNL A321
PGBRWCCT	The number of CICS browse requests for channel containers issued by the user task	DFHCHNL A322
PGGETCCT	The number of GET CONTAINER and GET64 CONTAINER requests for channel containers issued by the user task	DFHCHNL A323
PGPUTCCT	The number of PUT CONTAINER and PUT64 CONTAINER requests for channel containers issued by the user task	DFHCHNL A324
PGMOVCCT	The number of MOVE CONTAINER requests for channel containers issued by the user task	DFHCHNL A325
PGGETCDL	The total length, in bytes, of the data in the containers of all the GET CONTAINER CHANNEL and GET64 CONTAINER CHANNEL commands issued by the user task.	DFHCHNL A326
PGPUTGDL	The total length, in bytes, of the data in the containers of all the PUT CONTAINER CHANNEL and PUT64 CONTAINER CHANNEL commands issued by the user task.	DFHCHNL A327
PGCRECCT	The number of containers created by MOVE, PUT CONTAINER, and PUT64 CONTAINER requests for channel containers issued by the user task.	DFHCHNL A328
OAPPLID	The APPLID of the CICS region in which this work request (transaction) originated, for example, the region in which the CWXN task ran.	DFHCICS C360
OSTART	The time at which the originating task, for example, the CWXN task, was started.	DFHCICS T361
OTRANNUM	The number of the originating task, for example, the CWXN task.	DFHCICS P362

Table 7. Fields generated by CICS with default dictionary (continued)		
Field name	Description	Dictionary entry ID
OTRAN	The transaction ID (TRANSID) of the originating task, for example, the CWXN task.	DFHCICS C363
OUSERID	The originating Userid-2 or Userid-1, for example, from CWBA, depending on the originating task.	DFHCICS C364
OUSERCOR	The originating user correlator	DFHCICS C365
OTCPSVCE	The name of the originating TCPIP SERVICE	DFHCICS C366
OPORTNUM	The port number used by the originating TCPIP SERVICE	DFHCICS A367
OCLIPORT	The TCP/IP port number of the originating client or Telnet client	DFHCICS C369
OTRANFLG	Originating transaction flags, a string of 64 bits used for signaling transaction definition and status information.	DFHCICS A370
OFCTYNME	The facility name of the originating transaction. If the originating transaction is not associated with a facility, this field is null. The transaction facility type, if any, can be identified using byte 0 of the originating transaction flags, OTRANFLG (370), field.	DFHCICS C371
DHDELCT	The number of document handler DELETE requests issued by the user task	DFHDOCH A223
ISALLOCT	The number of allocate session requests issued by the user task for sessions using IPIC	DFH SOCK A288
ISIWTT_TOD	The elapsed time for which a user task waited for control at this end of an IPIC connection	DFH SOCK S300
ISIWTT_BT		
ISIWTT_CT		
ISIPICNM	The name of the IPIC connection for the TCP/IP service that attached the user task	DFH SOCK C305
CLIPPORT	The port number of the client or Telnet client	DFH SOCK A330
USRDISPT_TOD	Total elapsed time during which the user task was dispatched on each CICS TCB under which the task ran. The TCB modes managed by the CICS dispatcher are: QR, RO, CO, FO, SZ, RP, SL, SP, SO, EP, L8, L9, S8, TP, T8, X8, X9, and D2. Be aware that, for each CICS release, new TCB modes might be added to this list, or obsolete TCB modes might be removed. For more information about dispatch time and CPU time, see Transaction dispatch time and CPU time.	DFHTASK S007
USRDISPT_BT		
USRDISPT_CT		
USRCPUT_TOD	Processor time for which the user task was dispatched on each CICS TCB under which the task ran. The TCB modes managed by the CICS dispatcher are: QR, RO, CO, FO, SZ, RP, SL, SP, SO, EP, L8, L9, S8, TP, T8, X8, X9, and D2. Be aware that, for each CICS release, new TCB modes might be added to this list, or obsolete TCB modes might be removed. For more information about dispatch time and CPU time, see Transaction dispatch time and CPU time.	DFHTASK S008
USRCPUT_BT		
USRCPUT_CT		
SUSPTIME_TOD	Total elapsed wait time for which the user task was suspended by the dispatcher	DFHTASK S014
SUSPTIME_BT		
SUSPTIME_CT		
DISPWTT_TOD	Elapsed time for which the user task waited for redispach. This time is the aggregate of the wait times between each event completion and user-task re-dispatch. This field does not include the elapsed time spent waiting for first dispatch. This field is a component of the task suspend time, SUSPTIME (014), field.	DFHTASK S102
DISPWTT_BT		
DISPWTT_CT		
EXWTTIME_TOD	Accumulated data for exception conditions. The timer component of the clock contains the total elapsed time for which the user waited on exception conditions. The period count equals the number of exception conditions that have occurred for this task. For more information on exception conditions, see Exception class data: Listing of data fields. For more information on clocks, see Clocks and timestamp.	DFHCICS S103
EXWTTIME_BT		
EXWTTIME_CT		
TCIOWTT_TOD	Elapsed time for which the user task waited for input from the terminal operator after issuing a RECEIVE request. For more information, see Clocks and timestamp and Transaction wait (suspend) times. This field is a component of the task suspend time, SUSPTIME (014), field.	DFHTERM S009
TCIOWTT_BT		
TCIOWTT_CT		
FCIOWTT_TOD	Elapsed time in which the user task waited for file I/O. For more information, see Clocks and timestamp, and Transaction wait (suspend) times.	DFHFILE S063
FCIOWTT_BT		
FCIOWTT_CT		
JCIOWTT_TOD	Elapsed time for which the user task waited for journal (logstream) I/O. For more information, see Clocks and timestamp, and Transaction wait (suspend) times.	DFHJOUR S010
JCIOWTT_BT		
JCIOWTT_CT		
TSIOWTT_TOD	Elapsed time for which the user task waited for VSAM temporary storage I/O. For more information, see Clocks and timestamp, and Transaction wait (suspend) times.	DFHTEMP S011
TSIOWTT_BT		
TSIOWTT_CT		
IRIOWTT_TOD	Elapsed time for which the user task waited for control at this end of an MRO link. For more information, see Clocks and timestamp and Transaction wait (suspend) times. This field is a component of the task suspend time, SUSPTIME (014), field.	DFHTERM S100
IRIOWTT_BT		
IRIOWTT_CT		

Table 7. Fields generated by CICS with default dictionary (continued)

Field name	Description	Dictionary entry ID
TDIOWTT_TOD	Elapsed time in which the user waited for VSAM transient data I/O. For more information, see Clocks and timestamp and Transaction wait (suspend) times. This field is a component of the task suspend time, SUSPTIME (014), field.	DFHDEST S101
TDIOWTT_BT		
TDIOWTT_CT		
PCLOADTM_TOD	Elapsed time in which the user task waited for fetches from DFHRPL or dynamic LIBRARY concatenations. Only fetches for programs with installed program definitions or autoinstalled as a result of application requests are included in this figure. However, installed programs in the LPA are not included (because they do not incur a physical fetch from a library). For more information about program load time, see Clocks and timestamp, and Program load time.	DFHPROG S115
PCLOADTM_BT		
PCLOADTM_CT		
DSPDELAY_TOD	The elapsed time waiting for first dispatch. This field is a component of the task suspend time, SUSPTIME (014), field. For more information, see Clocks and timestamp.	DFHTASK S125
DSPDELAY_BT		
DSPDELAY_CT		
TCLDELAY_TOD	The elapsed time waiting for first dispatch, which was delayed because of the limits set for the transaction class of this transaction, TCLSNAME (166), being reached. For more information, see Clocks and timestamp. This field is a component of the first dispatch delay, DSPDELAY (125), field.	DFHTASK S126
TCLDELAY_BT		
TCLDELAY_CT		
MXTDELAY_TOD	The elapsed time waiting for the first dispatch, which was delayed because of the limits set by the system parameter, MXT, being reached. The field is a component of the first dispatch delay, DSPDELAY (125), field.	DFHTASK S127
MXTDELAY_BT		
MXTDELAY_CT		
ENQDELAY_TOD	The elapsed time waiting for a CICS task control local enqueue. For more information, see Clocks and timestamp. This field is a component of the task suspend time, SUSPTIME (014), field.	DFHTASK S129
ENQDELAY_BT		
ENQDELAY_CT		
LU61WTT_TOD	The elapsed time for which the user task waited for I/O on a LUTYPE6.1 connection or session. This time also includes the waits incurred for conversations across LUTYPE6.1 connections, but not the waits incurred because of LUTYPE6.1 syncpoint flows. For more information, see Clocks and timestamp and Transaction wait (suspend) times. This field is a component of the task suspend time, SUSPTIME (014), field.	DFHTERM S133
LU61WTT_BT		
LU61WTT_CT		
LU62WTT_TOD	The elapsed time for which the user task waited for I/O on a LUTYPE6.2 (APPC) connection or session. This time also includes the waits incurred for conversations across LUTYPE6.2 (APPC) connections, but not the waits incurred because of LUTYPE6.2 (APPC) syncpoint flows. For more information, see Clocks and timestamp and Transaction wait (suspend) times. This field is a component of the task suspend time, SUSPTIME (014), field.	DFHTERM S134
LU62WTT_BT		
LU62WTT_CT		
RMITIME_TOD	The total elapsed time spent in the CICS Resource Manager Interface (RMI). For more information, see Clocks and timestamp, Transaction wait (suspend) times, and RMI elapsed and suspend time.	DFHTASK S170
RMITIME_BT		
RMITIME_CT		
RMISUSP_TOD	The total elapsed time that the task was suspended by the CICS dispatcher while in the CICS Resource Manager Interface (RMI). For more information, see Clocks and timestamp, Transaction wait (suspend) times, and RMI elapsed and suspend time. The field is a component of the task suspend time, SUSPTIME (014), field and also the RMITIME (170) field.	DFHTASK S171
RMISUSP_BT		
RMISUSP_CT		
SZWAIT_TOD	Elapsed time in which the user task waited for all FEPI services. For more information, see Clocks and timestamp, and Transaction wait (suspend) times.	DFHFEPI S156
SZWAIT_BT		
SZWAIT_CT		
RLSWAIT_TOD	Elapsed time in which the user task waited for RLS file I/O. For more information, see Clocks and timestamp, and Transaction wait (suspend) times.	DFHFILE S174
RLSWAIT_BT		
RLSWAIT_CT		
SYNCTIM_TOD	Total elapsed time for which the user task was dispatched and was processing syncpoint requests	DFHSYNC S173
SYNCTIM_BT		
SYNCTIM_CT		
RLSCLPUT_TOD	For RLS requests issued from the QR TCB: The RLS File Request CPU (SRB) time field (RLSCLPUT) is the SRB CPU time this transaction spent processing RLS file requests. This field should be added to the transaction CPU time field (USRCLPUT) when considering the measurement of the total CPU time consumed by a transaction. Also, this field cannot be considered a subset of any other single CMF field (including RLSWAIT). This is because the RLS field requests execute asynchronously under an MVS SRB which can be running in parallel with the requesting transaction. It is also possible for the SRB to complete its processing before the requesting transaction waits for the RLS file request to complete. For RLS requests issued from an open TCB: There is no RLSCLPUT field for applications that are running on an open TCB mode because the requests are completed on the same TCB on which the application is running. In this case, the CPU time for the request is already accumulated in the USRCLPUT field. Note that system initialization parameters FCQRONLY and FORCEQR can both influence the TCB under which the RLS requests are issued. See System initialization parameter descriptions and summary for details.	DFHFILE S175
RLSCLPUT_BT		
RLSCLPUT_CT		

Table 7. Fields generated by CICS with default dictionary (continued)		
Field name	Description	Dictionary entry ID
LMDELAY_TOD	The elapsed time that the user task waited to acquire a lock on a resource. A user task cannot explicitly acquire a lock on a resource, but many CICS modules lock resources on behalf of user tasks using the CICS lock manager (LM) domain.	DFHTASK S128
LMDELAY_BT		
LMDELAY_CT		
WTEXWAIT_TOD	The elapsed time that the user task waited for one or more ECBs, passed to CICS by the user task using the EXEC CICS WAIT EXTERNAL ECBLIST command, to be posted by the MVS POST command. The user task can wait on one or more ECBs. If it waits on more than one, it is dispatchable as soon as one of the ECBs is posted. For more information, see Clocks and timestamp and Transaction wait (suspend) times. This field is a component of the task suspend time, SUSPTIME (014), field.	DFHTASK S181
WTEXWAIT_BT		
WTEXWAIT_CT		
WTCEWAIT_TOD	The elapsed time that the user task waited for one of these events: One or more ECBs, passed to CICS by the user task using the EXEC CICS WAITCICS ECBLIST command, to be posted by the MVS POST command. The user task can wait on one or more ECBs. If it waits on more than one, it is dispatchable as soon as one of the ECBs is posted. Completion of an event initiated by the same or by another user task. The event is usually be the posting, at the expiration time, of a timer-event control area provided in response to an EXEC CICS POST command. The EXEC CICS WAIT EVENT command provides a method of directly giving up control to some other task until the event being waited on is completed.	DFHTASK S182
WTCEWAIT_BT		
WTCEWAIT_CT		
ICDELAY_TOD	The elapsed time that the user task waited as a result of issuing one of the following commands: An interval control EXEC CICS DELAY command for a specified time interval. An interval control EXEC CICS DELAY command for a specified time of day to expire. An interval control EXEC CICS RETRIEVE command with the WAIT option specified.	DFHTASK S183
ICDELAY_BT		
ICDELAY_CT		
GVUPWAIT_TOD	The elapsed time that the user task waited as a result of giving up control to another task. A user task can give up control in many ways. Some examples are application programs that use one or more of the following EXEC CICS API or SPI commands: The EXEC CICS SUSPEND command. This command causes the issuing task to give up control to another task of higher or equal dispatching priority. Control is returned to this task as soon as no other task of a higher or equal priority is ready to be dispatched. The EXEC CICS CHANGE TASK PRIORITY command. This command immediately changes the priority of the issuing task and causes the task to give up control for it to be dispatched at its new priority. The task is not redispached until tasks of higher or equal priority, and that are also dispatchable, have been dispatched. The EXEC CICS DELAY command with INTERVAL (0). This command causes the issuing task to give up control to another task of higher or equal dispatching priority. Control is returned to this task as soon as no other task of a higher or equal priority is ready to be dispatched. The EXEC CICS POST command requesting notification that a specified time has expired. This command causes the issuing task to give up control so that CICS has the opportunity to post the time-event control area. The EXEC CICS PERFORM RESETTIME command to synchronize the CICS date and time with the MVS system date and time of day. The EXEC CICS START TRANSID command with the ATTACH option.	DFHTASK S184
GVUPWAIT_BT		
GVUPWAIT_CT		
TSSHWAIT_TOD	Elapsed time that the user task waited for an asynchronous shared temporary storage request to a temporary storage data server to complete. For more information, see Clocks and timestamp, and Transaction wait (suspend) times.	DFHTEMP S178
TSSHWAIT_BT		
TSSHWAIT_CT		
IMSWAIT_TOD	The elapsed time during which the user task waited for DBCTL to service the IMS requests issued by the user task. This field value is zero if IMS supports the open transaction environment (OTE).	DFHDATA S186
IMSWAIT_BT		
IMSWAIT_CT		
DB2RDYQW_TOD	The elapsed time during which the user task waited for a Db2 thread to become available.	DFHDATA S187
DB2RDYQW_BT		
DB2RDYQW_CT		
DB2CONWT_TOD	The elapsed time during which the user task waited for a Db2 connection to become available for use with the user task's open TCB.	DFHDATA S188
DB2CONWT_BT		
DB2CONWT_CT		
DB2WAIT_TOD	Reserved field, returns zero	DFHDATA S189
DB2WAIT_BT		
DB2WAIT_CT		
CFDTWAIT_TOD	Elapsed time in which the user task waited for a data table access request to the Coupling Facility Data Table server to complete. For more information, see Clocks and timestamp, and Transaction wait (suspend) times.	DFHFILE S176
CFDTWAIT_BT		
CFDTWAIT_CT		
SOIOWTT_TOD	The elapsed time in which the user task waited for inbound socket I/O. For more information, see Clocks and timestamp, and Transaction wait (suspend) times.	DFH SOCK S241
SOIOWTT_BT		
SOIOWTT_CT		
SRVSYWTT_TOD	Total elapsed time in which the user task waited for syncpoint or resynchronization processing using the Coupling Facility data tables server to complete.	DFHSYNC S177
SRVSYWTT_BT		
SRVSYWTT_CT		

Table 7. Fields generated by CICS with default dictionary (continued)

Field name	Description	Dictionary entry ID
SYNCDLY_TOD	The elapsed time in which the user task waited for a syncpoint request to be issued by its parent transaction. The user task was executing as a result of the parent task issuing a CICS BTS run-process or run-activity request to execute a process or activity synchronously. For more information, see Clocks and timestamp, and Transaction wait (suspend) times.	DFHSYNC S196
SYNCDLY_BT		
SYNCDLY_CT		
GNQDELAY_TOD	The elapsed time waiting for a CICS task control global enqueue. For more information, see Clocks and timestamp.	DFHTASK S123
GNQDELAY_BT		
GNQDELAY_CT		
RRMSWAIT_TOD	The elapsed time in which the user task waited indoubt using resource recovery services for EXCI. For more information, see Clocks and timestamp, and Transaction wait (suspend) times.	DFHTASK S191
RRMSWAIT_BT		
RRMSWAIT_CT		
RUNTRWTT_TOD	The elapsed time in which the user task waited for completion of a transaction that executed as a result of the user task issuing a CICS BTS run process, or run activity, request to execute a process, or activity, synchronously. For more information, see Clocks and timestamp, and Transaction wait (suspend) times.	DFHTASK S195
RUNTRWTT_BT		
RUNTRWTT_CT		
QRMODDLY_TOD	The elapsed time for which the user task waited for redispach on the CICS QR TCB. This is the aggregate of the wait times between each event completion. and user-task redispach.	DFHTASK S249
QRMODDLY_BT		
QRMODDLY_CT		
MAXOTDLY_TOD	The elapsed time in which the user task waited to obtain a CICS open TCB, because the region had reached the limit set by the system parameter, MAXOPENTCBS. This applies to L8 and L9 mode open TCBs only. L8 and L9 mode open TCBs are used by OPENAPI application programs, or task-related user exit programs that have been enabled with the OPENAPI option, for example, the CICS-DB2® adapter, when CICS connects to DB2 Version 6 or later and the CICS-MQ adapter, when CICS connects to Websphere MQ Version 6 or later . For more information, see Clocks and timestamp, and Transaction wait (suspend) times.	DFHTASK S250
MAXOTDLY_BT		
MAXOTDLY_CT		
JVMTIME_TOD	The total elapsed time spent in the JVM by the user task. For more information, see JVM elapsed time, suspend time, and cleanup time.	DFHTASK S253
JVMTIME_BT		
JVMTIME_CT		
JVMSUSP_TOD	The elapsed time for which the user task was suspended by the CICS dispatcher while running in the JVM. For more information, see JVM elapsed time, suspend time, and cleanup time.	DFHTASK S254
JVMSUSP_BT		
JVMSUSP_CT		
QRDISPT_TOD	The elapsed time for which the user task was dispatched on the CICS QR TCB. For more information, see Clocks and timestamp.	DFHTASK S255
QRDISPT_BT		
QRDISPT_CT		
QRCPUT_TOD	The processor time for which the user task was dispatched on the CICS QR TCB. For more information, see Clocks and timestamp.	DFHTASK S256
QRCPUT_BT		
QRCPUT_CT		
MSDISPT_TOD	Elapsed time for which the user task was dispatched on each CICS TCB. The CICS TCB modes are used as follows: RO and FO are always used. CO is used if SUBTSKS=1 is specified as a system initialization parameter. SZ is used if FEPI is active. RP is used if the ONC/RPC or CICS Web Interface Feature is installed and active. SO, SL, and SP are used if TCPIP=YES is specified as a system initialization parameter. Mode SL is used by the CICS support for TCP/IP (TCP/IP Service) Listener system transaction CSOL. Mode SO is used to process the CICS support for TCP/IP socket requests issued by or on behalf of the user task. Mode SP is the CICS support for TCP/IP sockets IPT task (Initial Pthread TCB) and also owns all the SSL pthreads (S8 TCBs). D2 is used only in CICS Transaction Server for z/OS, Version 2 Release 2 or later, when CICS is connected to DB2 Version 6 or later, to terminate DB2 protected threads. JM is used for Java shared class cache management when JVMs running in CICS are using a shared class cache.	DFHTASK S257
MSDISPT_BT		
MSDISPT_CT		
MSCPUT_TOD	The processor time for which the user task was dispatched on each CICS TCB. The usage of each CICS TCB is shown in the description for field MSDISPT (field id 257 in group DFHTASK). For more information, see Clocks and timestamp.	DFHTASK S258
MSCPUT_BT		
MSCPUT_CT		
L8CPUT_TOD	The processor time during which the user task was dispatched by the CICS dispatcher domain on a CICS L8 mode TCB. When a transaction invokes an OPENAPI application program defined with EXECKEY=CICS, or a task-related user exit program that has been enabled with the OPENAPI option, CICS allocates a CICS L8 mode TCB to the task. (An L8 mode TCB can also be allocated if the OPENAPI program is defined with EXECKEY=USER, but the storage protection facility is inactive.) Once a task has been allocated an L8 mode TCB, that same TCB remains associated with the task until the transaction is detached.	DFHTASK S259
L8CPUT_BT		
L8CPUT_CT		

Table 7. Fields generated by CICS with default dictionary (continued)		
Field name	Description	Dictionary entry ID
J8CPUT_TOD	The processor time during which the user task was dispatched by the CICS dispatcher domain on a CICS J8 mode TCB. When a transaction invokes a Java program defined with EXECKEY=CICS, that requires a JVM in CICS key, it is allocated and uses a CICS J8 mode TCB. (A J8 mode TCB can also be allocated if the Java program is defined with EXECKEY=USER, but the storage protection facility is inactive.) Once a task has been allocated a J8 mode TCB, that same TCB remains associated with the task until the Java program completes. For more information, see Clocks and timestamp.	DFHTASK S260
J8CPUT_BT		
J8CPUT_CT		
S8CPUT_TOD	The processor time during which the user task was dispatched by the CICS dispatcher domain on a CICS S8 mode TCB. A transaction is allocated a CICS S8 mode TCB when it is using the secure sockets layer (SSL) during client certificate negotiation. The S8 mode TCB remains associated with the same task for the life of the SSL request. For more information, see Clocks and timestamp.	DFHTASK S261
S8CPUT_BT		
S8CPUT_CT		
RODISPT_TOD	The elapsed time during which the user task was dispatched by the CICS dispatcher on the CICS RO mode TCB. The CICS RO mode TCB is used for opening and closing CICS data sets, loading programs, issuing RACF calls, and other functions.	DFHTASK S269
RODISPT_BT		
RODISPT_CT		
KY8DISPT_TOD	The total elapsed time during which the user task was dispatched by the CICS dispatcher on a CICS Key 8 mode TCB: An L8 mode TCB is allocated when a transaction invokes an OPENAPI application program defined with EXECKEY=CICS, or a task-related user exit program that has been enabled with the OPENAPI option. The TCB remains associated with the task until the transaction is detached. A J8 mode TCB is allocated when a transaction invokes a Java program defined with EXECKEY=CICS, that requires a JVM in CICS key. (A J8 mode TCB can also be allocated if the Java program is defined with EXECKEY=USER, but the storage protection facility is inactive.) The TCB remains associated with the task until the Java program completes. An S8 mode TCB is allocated when a transaction is using the secure sockets layer (SSL) during client certificate negotiation. The S8 mode TCB remains associated with the same task for the life of the SSL request. An X8 mode TCB is allocated when a transaction invokes a C or C++ program that was compiled with the XPLINK option, and that is defined with EXECKEY=CICS. The TCB remains associated with the task until the program ends.	DFHTASK S262
KY8DISPT_BT		
KY8DISPT_CT		
ROCPUT_TOD	The processor time during which the user task was dispatched by the CICS dispatcher on the CICS RO mode TCB. The CICS RO mode TCB is used for opening and closing CICS data sets, loading programs, issuing RACF calls, and other functions.	DFHTASK S270
ROCPUT_BT		
ROCPUT_CT		
KY8CPUT_TOD	The processor time during which the user task was dispatched by the CICS dispatcher on a CICS Key 8 mode TCB. The usage of the CICS Key 8 mode TCBs is shown in the description for field KY8DISPT(field id 262 in group DFHTASK).	DFHTASK S263
KY8CPUT_BT		
KY8CPUT_CT		
MAXJTDLY_TOD	The elapsed time in which the user task waited to obtain a CICS JVM TCB (J8 or J9 mode), because the CICS system had reached the limit set by the system parameter, MAXJVMTCBS. The J8 and J9 mode open TCBs are used exclusively by Java programs defined with JVM(YES). For more information, see Transaction wait (suspend) times.	DFHTASK S277
MAXJTDLY_BT		
MAXJTDLY_CT		
RQRWAIT_TOD	The elapsed time during which the request receiver user task CIRR (or user specified transaction id) waited for any outstanding replies to be satisfied. For more information, see Clocks and timestamp, and Transaction wait (suspend) times.	DFHTASK S192
RQRWAIT_BT		
RQRWAIT_CT		
SOOIOWTT_TOD	The total elapsed time the user task waited on outbound sockets. For more information, see Clocks and timestamp, and Transaction wait (suspend) times.	DFH SOCK S299
SOOIOWTT_BT		
SOOIOWTT_CT		
RQPWAIT_TOD	The elapsed time during which the request processor user task CIRP waited for any outstanding replies to be satisfied. For more information, see Clocks and timestamp, and Transaction wait (suspend) times.	DFHTASK S193
RQPWAIT_BT		
RQPWAIT_CT		
OTSINDWT_TOD	The elapsed time in which the user task was dispatched or suspended indoubt (or both) while processing a syncpoint for an Object Transaction Service (OTS) syncpoint request. For more information, see Clocks and timestamp, and Transaction wait (suspend) times.	DFHSYNC S199
OTSINDWT_BT		
OTSINDWT_CT		
JVMITIME_TOD	The elapsed time spent initializing the JVM environment. For more information, see Clocks and timestamp.	DFHTASK S273
JVMITIME_BT		
JVMITIME_CT		
JVMRTIME_TOD	The elapsed time spent in JVM cleanup between uses of the JVM by Java programs. For more information, see Clocks and timestamp, and JVM elapsed time, suspend time, and cleanup time.	DFHTASK S275
JVMRTIME_BT		
JVMRTIME_CT		
PTPWAIT_TOD	The elapsed time in which the user task waited for the 3270 bridge partner transaction to complete. For more information, see Transaction wait (suspend) times.	DFHTASK S285
PTPWAIT_BT		
PTPWAIT_CT		

Table 7. Fields generated by CICS with default dictionary (continued)		
Field name	Description	Dictionary entry ID
KY9DISPT_TOD	The total elapsed time during which the user task was dispatched by the CICS dispatcher on a CICS Key 9 mode TCB: A J9 mode TCB is allocated when a transaction invokes a Java program defined with EXECKEY=USER, that requires a JVM in user key. (If the storage protection facility is inactive, the transaction is allocated a J8 mode TCB instead of a J9 mode TCB.) The TCB remains associated with the task until the Java program completes. An L9 mode TCB is allocated when a transaction invokes an OPENAPI application program defined with EXECKEY=USER. The TCB remains associated with the task until the transaction is detached. An X9 mode TCB is allocated when a transaction invokes a C or C++ program that was compiled with the XPLINK option, and that is defined with EXECKEY=USER. The TCB remains associated with the task until the program ends.	DFHTASK S264
KY9DISPT_BT		
KY9DISPT_CT		
KY9CPUT_TOD	The processor time during which the user task was dispatched by the CICS dispatcher on a CICS Key 9 mode TCB. The usage of the CICS Key 9 mode TCBs is shown in the description for field KY9DISPT(field id 264 in group DFHTASK).	DFHTASK S265
KY9CPUT_BT		
KY9CPUT_CT		
J9CPUT_TOD	The processor time during which the user task was dispatched by the CICS dispatcher domain on a CICS J9 mode TCB. When a transaction invokes a Java program defined with EXECKEY=USER, that requires a JVM in user key, it is allocated and uses a CICS J9 mode TCB. (If the storage protection facility is inactive, a J8 mode TCB is used instead of a J9 mode TCB.) Once a task has been allocated a J9 mode TCB, that same TCB remains associated with the task until the Java program completes.	DFHTASK S267
J9CPUT_BT		
J9CPUT_CT		
DSTCBMWT_TOD	The elapsed time which the user task spent in TCB mismatch waits, that is, waiting because there was no TCB available matching the request, but there was at least one non-matching free TCB. For transactions that invoke a Java program to run in a JVM, this shows the time spent waiting for a TCB of the correct mode (J8 or J9) and JVM profile. How CICS allocates JVMs to applications has more information about how CICS manages TCB mismatch waits for these transactions.	DFHTASK S268
DSTCBMWT_BT		
DSTCBMWT_CT		
DSMMSWWT_TOD	The elapsed time which the user task spent waiting because no TCB pwas available, and none could be created because of MVS storage constraints. For more information about MVS storage constraints, see Dealing with warnings about MVS storage constraints.	DFHTASK S279
DSMMSWWT_BT		
DSMMSWWT_CT		
DSCHMDLY_TOD	The elapsed time in which the user task waited for redispach after a CICS Dispatcher change-TCB mode request was issued by or on behalf of the user task. For example, a change-TCB mode request from a CICS L8 or S8 mode TCB back to the CICS QR mode TCB might have to wait for the QR TCB because another task is currently dispatched on the QR TCB.	DFHTASK S247
DSCHMDLY_BT		
DSCHMDLY_CT		
MAXSTDLY_TOD	The elapsed time in which the user task waited to obtain a CICS SSL TCB (S8 mode), because the CICS system had reached the limit set by the system initialization parameter MAXSSLTCBS. The S8 mode open TCBs are used exclusively by secure sockets layer (SSL) pthread requests issued by or on behalf of a user task. For more information, see Transaction wait (suspend) times.	DFHTASK S281
MAXSTDLY_BT		
MAXSTDLY_CT		
L9CPUT_TOD	The processor time during which the user task was dispatched by the CICS dispatcher domain on a CICS L9 mode TCB. When a transaction invokes an OPENAPI application program defined with EXECKEY=USER, it is allocated and uses a CICS L9 mode TCB. (If the storage protection facility is inactive, an L8 mode TCB is used instead of an L9 mode TCB.) Once a task has been allocated an L9 mode TCB, that same TCB remains associated with the task until the transaction is detached.	DFHTASK S266
L9CPUT_BT		
L9CPUT_CT		
X8CPUT_TOD	The processor time during which the user task was dispatched by the CICS dispatcher domain on a CICS X8 mode TCB. When a transaction invokes a C or C++ program that was compiled with the XPLINK option, and that is defined with EXECKEY=CICS, it is allocated and uses a CICS X8 mode TCB. (An X8 mode TCB can also be allocated if the program is defined with EXECKEY=USER, but the storage protection facility is inactive.) Once a task has been allocated an X8 mode TCB, that same TCB remains associated with the task until the program completes.	DFHTASK S271
X8CPUT_BT		
X8CPUT_CT		
X9CPUT_TOD	The processor time during which the user task was dispatched by the CICS dispatcher domain on a CICS X9 mode TCB. When a transaction invokes a C or C++ program that was compiled with the XPLINK option, and that is defined with EXECKEY=USER, it is allocated and uses a CICS X9 mode TCB. (If the storage protection facility is inactive, an X8 mode TCB is used instead of an X9 mode TCB.) Once a task has been allocated an X9 mode TCB, that same TCB remains associated with the task until the program completes.	DFHTASK S272
X9CPUT_BT		
X9CPUT_CT		
MAXXTDLY_TOD	The elapsed time in which the user task waited to obtain a CICS XP TCB (X8 or X9 mode), because the CICS system had reached the limit set by the system parameter, MAXXPTCBS. The X8 and X9 mode open TCBs are used exclusively by C and C++ programs that were compiled with the XPLINK option. For more information, see Transaction wait (suspend) times.	DFHTASK S282
MAXXTDLY_BT		
MAXXTDLY_CT		
RMITOTAL_TOD	The total elapsed time spent in the CICS Resource Manager Interface (RMI). For more information, see Clocks and timestamp, and RMI elapsed and suspend time.	DFHRMI S001
RMITOTAL_BT		
RMITOTAL_CT		
RMIOOTHER_TOD	The total elapsed time spent in the CICS RMI for resource manager requests other than DB2, DBCTL, EXEC DLI, WebSphere MQ, CICSplex® SM, and CICS TCP/IP socket requests.	DFHRMI S002
RMIOOTHER_BT		
RMIOOTHER_CT		
RMIDB2_TOD	The total elapsed time spent in the CICS RMI for DB2 requests	DFHRMI S003
RMIDB2_BT		
RMIDB2_CT		

Table 7. Fields generated by CICS with default dictionary (continued)		
Field name	Description	Dictionary entry ID
RMIDBCTL_TOD	The total elapsed time spent in the CICS RMI for DBCTL requests	DFHRMI S004
RMIDBCTL_BT		
RMIDBCTL_CT		
RMIEXDLI_TOD	The total elapsed time spent in the CICS RMI for EXEC DLI requests	DFHRMI S005
RMIEXDLI_BT		
RMIEXDLI_CT		
RMIMQM_TOD	The total elapsed time spent in the CICS RMI for WebSphere MQ requests	DFHRMI S006
RMIMQM_BT		
RMIMQM_CT		
RMICPSM_TOD	The total elapsed time spent in the CICS RMI for CICSplex SM requests	DFHRMI S007
RMICPSM_BT		
RMICPSM_CT		
RMITCPIP_TOD	The total elapsed time spent in the CICS RMI for CICS TCP/IP socket requests	DFHRMI S008
RMITCPIP_BT		
RMITCPIP_CT		
PGCSTHWM	Maximum amount (high-water mark), in bytes, of container storage allocated to the user task.	DFHCHNL A329
ONETWKID	The network identifier from which this work request (transaction) originated	DFHCICS C359
WMQREQCT	The total number of MQ requests issued by the user task	DFHDATA A395
WMQGETWT_TOD	The elapsed time the user task waited for WebSphere MQ to service the user task's GETWAIT request. For more information, see Clocks and timestamp, and Transaction wait (suspend) times.	DFHDATA S396
WMQGETWT_BT		
WMQGETWT_CT		
CLIPADDR2	The IP address of the client or Telnet client	DFH SOCK C318
OCLIPADR2	The IP address of the originating client or Telnet client	DFHCICS C372
WBURIMNM	For CICS web support, Atom feeds, and web service applications, the name of the URIMAP resource definition that was mapped to the URI of the inbound request that was processed by this task.	DFHWEBB C380
WBPIPLNM	For web service applications, the name of the PIPELINE resource definition that was used to provide information about the message handlers that act on the service request processed by this task.	DFHWEBB C381
WBATMSNM	For Atom feeds, the name of the ATOMSERVICE resource definition that was used to process this task.	DFHWEBB C382
WBSVCENM	For web service applications, the name of the WEBSERVICE resource definition that was used to process this task.	DFHWEBB C383
WBSVOPNM	For web service applications, the first 64 bytes of the web service operation name.	DFHWEBB C384
WBPROGNM	For CICS web support, the name of the program from the URIMAP resource definition that was used to provide the application-generated response to the HTTP request processed by this task.	DFHWEBB C385
EICTOTCT	The total number of EXEC CICS commands issued by the user task	DFHCICS A402
ECSIGECT	The number of EXEC CICS SIGNAL EVENT commands issued by the user task	DFHCICS A415
ECEFOPCT	The number of event filter operations performed by the user task	DFHCICS A416
ECEVNTCT	The number of events captured by the user task	DFHCICS A417
TIASKTCT	The number of EXEC CICS ASKTIME commands issued by the user task	DFHCICS A405
TITOTCT	The total number of EXEC CICS ASKTIME, CONVERTTIME, and FORMATTIME commands issued by the user task.	DFHCICS A406
BFDGSTCT	The total number of EXEC CICS BIF DIGEST commands issued by the user task	DFHCICS A408
BFTOTCT	The total number of EXEC CICS BIF DEEDIT and BIF DIGEST commands issued by the user task	DFHCICS A409
MLXSSTDL	The total length of the documents that were parsed using the z/OS XML System Services parser	DFHWEBB A412
MLXMLTCT	The number of EXEC CICS TRANSFORM commands issued by the user task	DFHWEBB A413
WSACBLCT	The number of EXEC CICS WSACONTEXT BUILD commands issued by the user task	DFHWEBB A420
WSACGTCT	The number of EXEC CICS WSACONTEXT GET commands issued by the user task	DFHWEBB A421
WSAEPCT	The number of EXEC CICS WSAEPR CREATE commands issued by the user task	DFHWEBB A422
WSATOTCT	The total number of EXEC CICS WS-Addressing commands issued by the user task	DFHWEBB A423
WBSFCRCT	The number of EXEC CICS SOAPFAULT CREATE commands issued by the user task	DFHWEBB A386

Table 7. Fields generated by CICS with default dictionary (continued)		
Field name	Description	Dictionary entry ID
WBSFTOCT	The total number of EXEC CICS SOAPFAULT ADD, CREATE, and DELETE commands issued by the user task.	DFHWEBB A387
WBISSFCT	The total number of SOAP faults received in response to the EXEC CICS INVOKE SERVICE and EXEC CICS INVOKE WEBSERVICE commands issued by the user task.	DFHWEBB A388
WBSREQBL	For web service applications, the SOAP request body length.	DFHWEBB A390
WBSRSPBL	For web service applications, the SOAP response body length.	DFHWEBB A392
T8CPUT_TOD	The processor time during which the user task was dispatched by the CICS dispatcher domain on a CICS T8 mode TCB. T8 mode TCBs are used by a JVM server to perform multithreaded processing. When a thread is allocated a T8 mode TCB, that same TCB remains associated with the thread until the processing completes. This field is a component of the total task CPU time field, USRCPUT (field ID 008 in group DFHTASK), and the task key 8 CPU time field, KY8CPUT (field ID 263 in group DFHTASK).	DFHTASK S400
T8CPUT_BT		
T8CPUT_CT		
MAXTTDLY_TOD	The elapsed time for which the user task waited to obtain a T8 TCB, because the CICS system reached the limit of available threads. The T8 mode open TCBs are used by a JVM server to perform multithreaded processing. Each T8 TCB runs under one thread. The thread limit is 2000 for each CICS region and each JVM server in a CICS region can have up to 256 threads. For more information, see Transaction wait (suspend) times. This field is a component of the task suspend time, SUSPTIME (014), field.	DFHTASK S283
MAXTTDLY_BT		
MAXTTDLY_CT		
MLXSSCTM_TOD	The CPU time taken to convert a document using the z/OS XML System Services parser. This field is a subset of the total CPU time as measured in the USRCPUT field (owner DFHTASK, field ID 008).	DFHWEBB S411
MLXSSCTM_BT		
MLXSSCTM_CT		
JVMTHDWT_TOD	The elapsed time that the user task waited to obtain a JVM server thread because the CICS system had reached the thread limit for a JVM server in the CICS region. This field is a component of the task suspend time, SUSPTIME (014), field. This does not apply to Liberty JVM servers.	DFHTASK S401
JVMTHDWT_BT		
JVMTHDWT_CT		
WMQASRBT_TOD	The WebSphere MQ SRB time this transaction spent processing WebSphere MQ API requests. Add this field to the transaction CPU time field (USRCPUT) when considering the measurement of the total processor time consumed by a transaction. This field is zero for point-to-point messaging activity, but it is nonzero where WebSphere MQ API requests result in publish and subscribe type messaging.	DFHDATA S397
WMQASRBT_BT		
WMQASRBT_CT		
PHNTWKID	The network identifier of the CICS system of an immediately previous task in another CICS system with which this task is associated. See Previous hop data characteristics for more information about previous hop data.	DFHCICS C373
PHAPPLID	The APPLID from previous hop data. This is the APPLID of the CICS system of a previous task in another CICS system with which this task is associated. See Previous hop data characteristics for more information about previous hop data.	DFHCICS C374
PHSTART	The start time of the immediately previous task in another CICS system with which this task is associated. See Previous hop data characteristics for more information about previous hop data.	DFHCICS T375
PHTRANNO	The task number of the immediately previous task in another CICS system with which this task is associated. See Previous hop data characteristics for more information about previous hop data.	DFHCICS P376
PHTRAN	The transaction ID (TRANSID) of the immediately previous task in another CICS system with which this task is associated. See Previous hop data characteristics for more information about previous hop data.	DFHCICS C377
PHCOUNT	The number of times there has been a request from one CICS system to another CICS system to initiate a task with which this task is associated. See Previous hop data characteristics for more information about previous hop data.	DFHCICS A378
OADID	The adapter identifier added to the origin data by the adapter. This field is blank if the task was not started by using an adapter, or if it was and the adapter did not set this value.	DFHCICS C351
OADATA1	The data added to the origin data by the adapter. This field is blank if the task was not started by using an adapter, or if it was and the adapter did not set this value.	DFHCICS C352
OADATA2	The data added to the origin data by using the adapter. This field is blank if the task was not started by using an adapter, or if it was and the adapter did not set this value.	DFHCICS C353
OADATA3	The data added to the origin data by the adapter. This field is blank if the task was not started by using an adapter, or if it was and the adapter did not set this value.	DFHCICS C354
ECSEVCCT	The number of synchronous emission events captured by the user task.	DFHCICS A418
SOCIPHER	Identifies the code for the cipher suite that was selected during the SSL handshake for use on the inbound connection, for example X'0000002F'. For a list of the cipher suites that are supported by CICS and z/OS and their codes, see Cipher suites and cipher suite specification files.	DFH SOCK A320
CECMCHTP	The CEC machine type, in EBCDIC, for the physical hardware environment where the CICS region is running. CEC (central electronics complex) is a commonly used synonym for CPC (central processing complex).	DFHTASK C430
CECMDLID	The CEC model number, in EBCDIC, for the physical hardware environment where the CICS region is running.	DFHTASK C431
MAXTASKS	The MXT or MAXTASKS value, expressed as a number of tasks, for the CICS region at the time the user task was attached.	DFHTASK A433
CURTASKS	The current number of active user transactions in the system at the time the user task was attached	DFHTASK A434

Table 7. Fields generated by CICS with default dictionary (continued)		
Field name	Description	Dictionary entry ID
ACAPPLNM	The 64-character name of the application in the application context data	DFHTASK C451
ACPLATNM	The 64-character name of the platform in the application context data	DFHTASK C452
ACMAJVER	The major version of the application in the application context data, expressed as a 4-byte binary value.	DFHTASK A453
ACMINVER	The minor version of the application in the application context data, expressed as a 4-byte binary value.	DFHTASK A454
ACMICVER	The micro version of the application in the application context data, expressed as a 4-byte binary value.	DFHTASK A455
ACOPERNM	The 64-character name of the operation in the application context data.	DFHTASK C456
SC64CGCT	Number of user-storage GETMAIN requests issued by the user task for storage above the bar, in the CICS dynamic storage area (GCDSA).	DFHSTOR A441
SC64CHWM	Maximum amount (high-water mark) of user storage, rounded up to the next 4K, allocated to the user task above the bar, in the CICS dynamic storage area (GCDSA).	DFHSTOR A442
SC64UGCT	Number of user-storage GETMAIN requests issued by the user task for storage above the bar, in the user dynamic storage area (GUDSA).	DFHSTOR A443
SSC64UHWM	Maximum amount (high-water mark) of user storage, rounded up to the next 4K, allocated to the user task above the bar, in the user dynamic storage area (GUDSA).	DFHSTOR A444
SC64SGCT	Number of storage GETMAIN requests issued by the user task for shared storage above the bar, in the GCDSA or GSDSA.	DFHSTOR A445
SC64GSHR	Amount of shared storage obtained by the user task by using a GETMAIN request above the bar, in the GCDSA or GSDSA. The total number of bytes obtained is rounded up to the next 4096 bytes, and the resulting number of 4K pages is displayed.	DFHSTOR A446
SC64FSHR	Amount of shared storage released by the user task by using a FREEMAIN request above the bar, in the GCDSA or GSDSA. The total number of bytes obtained is rounded up to the next 4096 bytes, and the resulting number of 4K pages is displayed.	DFHSTOR A447
MPPRTXCD	The number of policy task rule thresholds that this task has exceeded. This field is all nulls (0x00 bytes) if no thresholds have been exceeded or if the task has had no task rules applied to it.	DFHCICS A449
CPUTONCP_TOD	The total task processor time on a standard processor for which the user task was dispatched on each CICS TCB under which the task ran. This field is a component of the task CPU time field, USRCPUT (field ID 008 in group DFHTASK). To calculate the task processor time that was spent on a specialty processor (zIIP or zAAP), subtract the time recorded in the CPUTONCP field from the time recorded in the USRCPUT field.	DFHTASK S436
CPUTONCP_BT		
CPUTONCP_CT		
OFFLCPUT_TOD	The total task processor time that was spent on a standard processor but was eligible for offload to a specialty processor (zIIP or zAAP). This field is a component of the task CPU time field, USRCPUT (field ID 008 in group DFHTASK), and also a component of the standard CPU time field, CPUTONCP (field ID 436 in group DFHTASK). To calculate the task processor time spent on a standard processor that was not eligible for offload to a specialty processor, subtract the time recorded in the OFFLCPUT field from the time recorded in the CPUTONCP field.	DFHTASK S437
OFFLCPUT_BT		
OFFLCPUT_CT		
FCXCWTT_TOD	The elapsed time in which the user task waited for exclusive control of a VSAM control interval. This field counts time spent waiting on resource type FCXCSUSP, FCXDSUSP, FCXCPCROT, or FCXDPROT. For more information, see Clocks and timestamp, and Transaction wait (suspend) times.	DFHFILE S426
FCXCWTT_BT		
FCXCWTT_CT		
FCVSWTT_TOD	The elapsed time in which the user task waited for a VSAM string. This field counts time spent waiting on resource type FCPSSUSP or FCSRSUSP. For more information, see Clocks and timestamp, and Transaction wait (suspend) times.	DFHFILE S427
FCVSWTT_BT		
FCVSWTT_CT		
TDILWTT_TOD	The elapsed time for which the user task waited for an intrapartition transient data lock (TDILOCK). For more information, see Clocks and timestamp and Transaction wait (suspend) times. For more information about tasks suspended on resource type TDILOCK, see Resource type TDILOCK: waits for transient data intrapartition requests. This field is a component of the task suspend time, SUSPTIME (014), field.	DFHDEST S403
TDILWTT_BT		
TDILWTT_CT		
TDELWTT_TOD	The elapsed time for which the user task waited for an extrapartition transient data lock (TDELOCK). For more information, see Clocks and timestamp and Transaction wait (suspend) times. For more information about tasks suspended on resource type TDELOCK, see Resource type TDELOCK: waits for transient data extrapartition requests. This field is a component of the task suspend time, SUSPTIME (014), field.	DFHDEST S404
TDELWTT_BT		
TDELWTT_CT		
ROMODDLY_TOD	The elapsed time for which the user task waited for redispach on the CICS RO TCB. This time is the aggregate of the wait times between each event completion and user-task redispach. The ROMODDLY field is a component of the task suspend time, SUSPTIME (014), field, and also the redispach wait, DISPWTT (102), field.	DFHTASK S348
ROMODDLY_BT		
ROMODDLY_CT		
SOMODDLY_TOD	The elapsed time for which the user task waited for redispach on the CICS SO TCB. This time is the aggregate of the wait times between each event completion and user-task redispach. The SOMODDLY field is a component of the task suspend time, SUSPTIME (014), field, and also the redispach wait, DISPWTT (102), field.	DFHTASK S349
SOMODDLY_BT		
SOMODDLY_CT		
ISALWTT_TOD	The elapsed time for which a user task waited for an allocate request for an IPIC session. For more information, see Clocks and timestamp and Transaction wait (suspend) times. This field is a component of the task suspend time, SUSPTIME (014), field.	DFH SOCK S319
ISALWTT_BT		
ISALWTT_CT		

Table 7. Fields generated by CICS with default dictionary (continued)

Field name	Description	Dictionary entry ID
TCALWTT_TOD	The elapsed time for which a user task waited for an allocate request for an MRO (Inter-Region Communication), LU6.1, or LU6.2 session. For more information, see Clocks and timestamp and Transaction wait (suspend) times. This field is a component of the task suspend time, SUSPTIME (014), field.	DFHTERM S343
TCALWTT_BT		
TCALWTT_CT		
TSGETSCT	Number of temporary storage GET requests from shared temporary storage issued by the user task.	DFHTEMP A460
TSPUTSCT	Number of temporary storage PUT requests to shared temporary storage issued by the user task.	DFHTEMP A461
WBJSNRQL	For JSON web service applications, the JSON message request length	DFHWEBB A424
WBJSNRPL	For JSON web service applications, the JSON message response length	DFHWEBB A425
NCGETCT	The total number of requests to a named counter server to satisfy EXEC CICS GET COUNTER and EXEC CICS GET DOUNTER commands issued by the user task.	DFHCICS A464
DSAPTHWT_TOD	The dispatcher allocated pthread wait time. This is the time that the transaction had to wait for a Liberty pthread to be allocated during links to Liberty programs. For more information, see Clocks and timestamp and Transaction wait (suspend) times. This field is a component of the task suspend time, SUSPTIME (014), field.	DFHTASK S429
DSAPTHWT_BT		
DSAPTHWT_CT		
LPARNAME	The name, in EBCDIC, of the logical partition (LPAR) on the processor where the CICS region is running.	DFHTASK C432
PTSTART	The start time of the immediately previous or parent task in the same CICS system with which the task is associated. See Previous transaction data characteristics for more information about previous transaction data.	DFHCICS T480
PTTRANNO	The task number of the immediately previous or parent task in the same CICS system with which the task is associated. See Previous transaction data characteristics for more information about previous transaction data.	DFHCICS P481
PTTRAN	The transaction ID (TRANSID) of the immediately previous or parent task in the same CICS system with which the task is associated. See Previous transaction data characteristics for more information about previous transaction data.	DFHCICS C482
PTCOUNT	The number of times there has been a request from one task to initiate another task in the same CICS system with which this task is associated, such as by a RUN TRANSID or START command. This is effectively the task depth in the local region when using the RUN TRANSID command, or the START command when a new point of origin is not created. See Previous transaction data characteristics for more information about previous transaction data.	DFHCICS A483
ASTOTCT	The total number of EXEC CICS asynchronous API commands that have been issued by the user task. Includes RUN TRANSID, FETCH CHILD, FETCH ANY, and FREE CHILD commands.	DFHTASK A470
ASRUNCT	The number of EXEC CICS RUN TRANSID commands that have been issued by the user task.	DFHTASK A471
ASFCHCT	The number of EXEC CICS FETCH CHILD and EXEC CICS FETCH ANY commands that have been issued by the user task.	DFHTASK A472
ASFRECT	The number of EXEC CICS FREE CHILD commands that have been issued by the user task.	DFHTASK A473
MPSRECT	The number of times that policy system rules have been evaluated for the task.	DFHCICS A466
MPSRACT	The number of times that policy system rules that have evaluated true and have triggered either a message or an event. This field is all nulls (0x00 bytes) if no system rules have been evaluated true.	DFHCICS A467
ASFTCHWT_TOD	The elapsed time that the user task waited for a child task as a result of issuing an EXEC CICS FETCH CHILD or EXEC CICS FETCH ANY command which was not completed. For more information, see Clocks and timestamp and Transaction wait (suspend) times. This field is a component of the task suspend time, SUSPTIME (014), field.	DFHTASK S475
ASFTCHWT_BT		
ASFTCHWT_CT		
ASRNATWT_TOD	The elapsed time that the user task was delayed as a result of asynchronous child task limits managed by the asynchronous services domain. For more information, see Clocks and timestamp and Transaction wait (suspend) times. This field is a component of the task suspend time, SUSPTIME (014), field.	DFHTASK S476
ASRNATWT_BT		
ASRNATWT_CT		
NJSAPPNM	Name of Node.js application from which the task was started.	DFHWEBB C419
SOCONMSG	Indicates whether the task processed the first message for establishing a new connection for a client. This field helps you measure how often a new socket connection is created. Y Indicates that the task processed the first message from the client. N Indicates that the task processed a subsequent message from the client.	DFH SOCK C344
WBURIOPN_TOD	The total elapsed time that the user task processed WEB OPEN URIMAP requests that it issued.	DFHWEBB S339
WBURIOPN_BT		
WBURIOPN_CT		
WBURIRCV_TOD	The total elapsed time that the user task processed WEB RECEIVE requests and the receiving side of WEB CONVERSE requests that it issued. The sessions these requests targeted to were opened by the WEB OPEN URIMAP command.	DFHWEBB S393
WBURIRCV_BT		
WBURIRCV_CT		

Table 7. Fields generated by CICS with default dictionary (continued)

Field name	Description	Dictionary entry ID
WBURISND_TOD	The total elapsed time that the user task processed WEB SEND requests and the sending side of WEB CONVERSE requests that it issued. The sessions these requests targeted to were opened by the WEB OPEN URIMAP command.	DFHWEBC S394
WBURISND_BT		
WBURISND_CT		
WBSVINVK_TOD	The total elapsed time that the user task processed INVOKE SERVICE requests for WEBSERVICEs.	DFHWEBC S379
WBSVINVK_BT		
WBSVINVK_CT		
XSVFYPWD_TOD	The total elapsed time that the user task spent verifying passwords, password phrases, PassTickets, and MFA tokens.	DFHTASK S435
XSVFYPWD_BT		
XSVFYPWD_CT		
XSVFYKER_TOD	The total elapsed time that the user task spent verifying Kerberos tokens (KERBEROS).	DFHTASK S439
XSVFYKER_BT		
XSVFYKER_CT		
XSVFYBAS_TOD	The total elapsed time that the user task spent verifying basic authentication tokens (BASICAUTH).	DFHTASK S438
XSVFYBAS_BT		
XSVFYBAS_CT		
XSVFYJWT_TOD	The total elapsed time that the user task spent verifying JSON web tokens (JWT).	DFHTASK S440
XSVFYJWT_BT		
XSVFYJWT_CT		
SMMVSSWT_TOD	The time that the user task waited because MVS user region or extended user region was short in storage.	DFHTASK S274
SMMVSSWT_BT		
SMMVSSWT_CT		

The following tables provide fields generated by OMEGAON products:

Table 8. Fields collected from OMEGAMON XE for CICS OMEGBSC sections

Field name	Description	Dictionary entry ID
OMEG_PROD1	Internal OMEGAMON® use only	OMEBSC C001
OMEG_PROD2	Internal OMEGAMON use only	
OMEG_PROD3	Internal OMEGAMON use only	
OMEG_PROD4	Internal OMEGAMON use only	
OMEG_PROD5	Internal OMEGAMON use only	
OMEG_PROD6	Internal OMEGAMON use only	
OMEG_GMTOF	GMT offset, used to calculate Local timestamp	
OMEG_FLAGS	Flag values used for umbrella support and 'CONVTIME=(NOIOWAIT,NOIRWAIT)'	
OMEG_VRSN	Version of OMEGAMON used	
OMEG_WRN		
OMEG_UMBR_TRAN	Umbrella support, transaction ID	
OMEG_UMBR_USR	Umbrella SUPPORT, PROGRAM NAME	
OMEG_WAIT_RSRC	User supplied 'Waiting resource name"	
OMEG_WAIT_TYPE	User supplied 'Waiting resource type"	
OMEG_WORK_AREA	Umbrella support, user work area	
OMEG_IDMS_CNT	IDMS support, total number of requests	
OMEG_IDMS_CLK	IDMS support, total time of requests	
OMEG_ADAB_CNT	ADABAS support, total number of requests	
OMEG_ADAB_CLK	ADABAS support, total time of requests	
OMEG_SUPR_CNT	SUPRA support, total number of requests	
OMEG_SUPR_CLK	SUPRA support, total time of requests	
OMEG_DATA_CNT	DATAACOM support, total number of requests	
OMEG_DATA_CLK	DATAACOM support, total time of requests	
OMEG_VSAM_CNT	Total VSAM calls	

Table 9. Fields collected from OMEGAMON XE for CICS OMEGDB2 sections

Field name	Description	Dictionary entry ID
DB2_SCTN_SIZE	Size of DB2 section	OMEGDB2 C001
DB2_OPN_CNT	Count Open CSR	
DB2_OPN_CLCK	Time Open CSR	
DB2_CLO_CNT	Count Close CSR	
DB2_CLO_CLCK	Time Close CSR	
DB2_FET_CNT	Count Fetches	
DB2_FET_CLCK	Time Fetches	
DB2_SEL_CNT	Count Selects	
DB2_SEL_CLCK	Time Selects	
DB2_INS_CNT	Count Inserts	
DB2_INS_CLCK	Time Inserts	
DB2_UPD_CNT	Count updates	
DB2_UPD_CLCK	Time updates	
DB2_DEL_CNT	Count deletes	
DB2_DEL_CLCK	Time deletes	
DB2_PRE_CNT	Count prepares	
DB2_PRE_CLCK	Time prepares	
DB2_DES_CNT	Count describes	
DB2_DES_CLCK	Time describes	
DB2_EXE_CNT	Count executes	
DB2_EXE_CLCK	Time executes	
DB2_EXI_CNT	Count EXEC immed	
DB2_EXI_CLCK	Time EXEC immed	
DB2_MIS_CNT	Count MISC	
DB2_MIS_CLCK	Time MISC	

Table 10. Fields collected from OMEGAMON XE for CICS OMEGDLI sections

Field name	Description	Dictionary entry ID
DLI_SCTN_SIZE	Size of OMEGDLI section	OMEGDLI C001
DLI_SCH_CNT	Count Schedules	
DLI_SCH_CLCK	Time in Schedule	
DLI_TER_CNT	Count for Terminates	
DLI_TER_CLCK	Time in Terminates	
DLI_GU_CNT	Count For GU	
DLI_GU_CLCK	Time in GU	
DLI_GN_CNT	Count Get Updates	
DLI_GN_CLCK	Time get Updates	
DLI_GNP_CNT	Count Get Nexts	
DLI_GNP_CLCK	Time Get Nexts	
DLI_GHU_CNT	Count GHU	
DLI_GHU_CLCK	Time GHU	
DLI_GHN_CNT	Count GHN	
DLI_GHN_CLCK	Time GHN	
DLI_GPN_CNT	Count GHN Parents	
DLI_GPN_CLCK	Time GHN Parents	
DLI_INS_CNT	Count for Inserts	
DLI_INS_CLCK	Time in Inserts	
DLI_DEL_CNT	Count for Deletes	
DLI_DEL_CLCK	Time in Deletes	
DLI_REP_CNT	Count for Replaces	
DLI_REP_CLCK	Time in Replaces	

Table 11. Fields collected from OMEGAMON XE for CICS OMEGMQ sections

Field name	Description	Dictionary entry ID
MQ_SCTN_SIZE	The size of MQ section	CANMQ C001
MQ_TOT_CNT	The total count for MQ requests	
MQ_TOT_CLCK	The sum time of MQ requests	
MQ_OPN_CNT	The count of MQ Open requests	
MQ_OPN_CLCK	The sum time of MQ Open requests	
MQ_CLO_CNT	The count of MQ Close requests	
MQ_CLO_CLCK	The sum time of MQ Close requests	
MQ_GET_CNT	The count of MQ Get requests	
MQ_GET_CLCK	The sum time of MQ Get requests	
MQ_PUT_CNT	The count of MQ Put requests	
MQ_PUT_CLCK	The sum time of MQ Put requests	
MQ_PU1_CNT	The count of MQ Put1 requests	
MQ_PU1_CLCK	The sum time of MQ Put1 requests	
MQ_INQ_CNT	The count of MQ Inquire requests	
MQ_INQ_CLCK	The sum time of MQ Inquire requests	
MQ_SET_CNT	The count of MQ Set requests	
MQ_SET_CLCK	The sum time of MQ Set requests	
MQ_MIS_CNT	The count of MQ Misc requests	
MQ_MIS_CLCK	The sum time of MQ Misc requests	

Table 12. Fields collected from OMEGAMON XE for CICS OMEGWLM sections

Field name	Description	Dictionary entry ID
WLM_START	The time for WLM collection	CANWLMSC C001
WLM_SCNAME	The service class name for WLM collection	
WLM_SCTOKEN	The service token for WLM collection	

Table 13. Fields collected from OMEGAMON XE for CICS OMEGCICS sections

Field name	Description	Dictionary entry ID
CICS_OMEG_IDNT	OMEGAMON ID	OMEGCICS C001
CICS_SCTN_SIZE	The length of OMEGCICS section	
CICS_RSRVED	Reserved	
CICS_OMEG_VRSN	OMEGAMON version	
CICS_RLIM_WRN	The type of RLIM warning received	
CICS_USR_WORK	User work area	
CICS_IDMS_CLCK	IDMS wait time/count	
CICS_IDMS_CNT		
CICS_ADAB_CLCK	ADABAS wait time/count	
CICS_ADAB_CNT		
CICS_SUPR_CLCK	SUPRA wait time/count	
CICS_SUPR_CNT		
CICS_DATA_CLCK	DATACOMM wait time/count	
CICS_DATA_CNT		
CICS_USER_CLCK	User defined events wait time/count	
CICS_USER_CNT		

Table 14. Fields collected from OMEGAMON XE for CICS OMEGUEVNT sections

Field name	Description	Dictionary entry ID
UE_SCTN_SIZE	The size of the user section	CANUE1 C001
UE_TOT_CNT	Total count of user event	
UE_TOT_CLCK	Total elapsed time	
UE_F1_CNT	Count of user event #1	
UE_F1_CLCK	Elapsed time of user event #1	
UE_F2_CNT	Count of user event #2	
UE_F2_CLCK	Elapsed time of user event #2	
UE_F3_CNT	Count of user event #3	
UE_F3_CLCK	Elapsed time of user event #3	
UE_F4_CNT	Count of user event #4	
UE_F4_CLCK	Elapsed time of user event #4	
UE_F5_CNT	Count of user event #5	
UE_F5_CLCK	Elapsed time of user event #5	
UE_F6_CNT	Count of user event #6	
UE_F6_CLCK	Elapsed time of user event #6	
UE_F7_CNT	Count of user event #7	
UE_F7_CLCK	Elapsed time of user event #7	
UE_F8_CNT	Count of user event #8	
UE_F8_CLCK	Elapsed time of user event #8	
UE_F9_CNT	Count of user event #9	
UE_F9_CLCK	Elapsed time of user event #9	
UE_F10_CNT	Count of user event #10	
UE_F10_CLCK	Elapsed time of user event #10	

Creating an application data stream definition

From the Common Data Provider tab in the IBM Z Common Data Provider Configuration Tool, you can create an application data stream definition to use in a policy.

Procedure

To create an application data stream, complete the following steps:

1. In the Configuration Tool, click the **MANAGE CUSTOM DATA STREAM DEFINITIONS** button.
2. In the resulting **Manage Custom Data Stream Definitions** window, click the **Create application data stream definition** box.
3. In the resulting **Define Application Data Stream** window, provide values for the following fields:

Name

Specifies the name of the data stream.

This name is converted to uppercase characters when it is saved.

The **Name** value must contain only alphanumeric characters and underscores. The maximum length is 243 characters.

Group

In the Configuration Tool, the data stream is included in the list of categorized data streams under the main category **Customer Data Streams**. For the hierarchy under **Customer Data Streams**, you must specify the group and subgroup under which you want to include the data stream. The value of **Group** specifies the group. The following example illustrates the hierarchy, and indicates that MYGROUP is specified as the **Group** value:

- **Customer Data Streams**

- **MYGROUP**

The **Group** value is case-insensitive. For example, if you specify MyGroup as the value, and a group that is named MYGROUP exists under the main category **Customer Data Streams**, the Configuration Tool includes the data stream under MYGROUP, and does not create the category MyGroup.

The **Group** value must contain only alphanumeric characters and underscores. The maximum length is 243 characters.

Subgroup

In the Configuration Tool, the data stream is included in the list of categorized data streams under the main category **Customer Data Streams**. For the hierarchy under **Customer Data Streams**, you must specify the group and subgroup under which you want to include the data stream. The value of **Subgroup** specifies the subgroup. The following example illustrates the hierarchy, and indicates that MYSUBGROUP is specified as the **Subgroup** value:

- **Customer Data Streams**

- **MYGROUP**

- **MYSUBGROUP**

The **Subgroup** value is case-insensitive. For example, if you specify MySubGroup as the value, and a subgroup that is named MYSUBGROUP exists under the group, the Configuration Tool includes the data stream under MYSUBGROUP, and does not create the category MySubGroup.

The **Subgroup** value must contain only alphanumeric characters and underscores. The maximum length is 243 characters.

Tags

Specifies the source or format of the data in the data stream. You can use tags to remind you of the source or format.

If you specify multiple values, you must list one tag per line, as shown in the following example:

```
CSV
Split
```

Any tags that you specify are shown in the Configuration Tool on the data stream and transform nodes.

If the data is being sent in split format, specify the tag Split.

If the data is in CSV format, specify the tag CSV.

The **Tags** value is case-sensitive and must contain only alphanumeric characters and underscores.

4. Click **OK**.

The data stream is created and available to be included in policies.

What to do next

Add the custom data stream to your policy. For information about creating or updating a policy, see the following information:

- [“Creating a policy” on page 36](#)
- [“Updating a policy” on page 43](#)

Use the Open Streaming API to send your application data to the Data Streamer and stream it to analytics platforms. For more information, see [Chapter 7, “Sending user application data to the Data Streamer,” on page 283](#).

Deleting a custom data stream definition

To delete a custom data stream definition (for either a System Data Engine data stream or an application data stream), you must delete the associated definition file from the Configuration Tool working directory.

Before you begin

Before you delete a custom data stream definition, verify that the associated data stream is not used in a policy. If it is, delete the data stream from the policy.

What happens if you do not delete a data stream before deleting its definition: If the definition for a data stream that is used in a policy is deleted, the data stream remains visible in the policy for the duration of the session with the Configuration Tool. To make the deletion take effect in the user interface, the user must log off the current session, and log on to the Configuration Tool again. If a user then opens a policy that contains the deleted data stream, the following message HB06511W is shown, and the data stream is removed from the policy:

```
HB06511W Definitions are missing for the following data streams:
- SMF_Z30
Please ensure all *.streams.json files used to create this policy
are located within the policy file directory, then restart the
Configuration Tool. All data streams without a matching stream
definition have been removed from the workspace. If you save the
policy now, the unknown data streams will not be included in the
saved policy file.
```

Procedure

To delete a custom data stream definition, complete the following steps:

1. In the Configuration Tool working directory, find the data stream definition file that you want to delete. The data stream definition file is named `data_stream_name.streams.json`. For example, if the name of the data stream is `SMF_CUST_030`, the data stream definition file is `SMF_CUST_030.streams.json`.
2. To delete the data stream definition file, run the following command:

```
rm data_stream_name.streams.json
```

For example, if you want to delete `SMF_CUST_030.streams.json`, run the following command:

```
rm SMF_CUST_030.streams.json
```

Securing communications between the Data Streamer and its subscribers

To secure communications between the IBM Z Common Data Provider Data Streamer and its subscribers, you must choose a streaming protocol that supports Transport Layer Security (TLS) when you configure a subscriber in a policy. You must also configure the Data Streamer and its subscribers to use TLS.

Before you begin

For more information about the streaming protocols, see [“Subscriber configuration” on page 235](#). The streaming protocols that support TLS contain either SSL or HTTPS in the name except for Kafka subscribers. For example, to secure communications between the Data Streamer and the Data Receiver, select the streaming protocol `CDP Data Receiver SSL` rather than `CDP Data Receiver`.

For Kafka subscribers, whether to enable secure communication between the Data Streamer and Kafka is not controlled by the protocol you select. Instead, edit the `kafka.properties` file in the Data Streamer working directory and set `security.protocol=SSL` or `SASL_SSL` to enable secure communication the Data Streamer and Kafka. The streaming protocol for Kafka subscribers, with or without secure communication enabled, is `CDP Kafka`.

Tip: Transport Layer Security (TLS) is the cryptographic protocol that provides secure communications for your connections. Because the Secure Sockets Layer (SSL) protocol is the predecessor to TLS, the term *Secure Sockets Layer*, or *SSL*, is often used generically to refer to TLS encryption.

About this task

The TLS protocol is provided by the IBM Java Runtime Environment that is installed on the z/OS system where the IBM Z Common Data Provider runs, and on the distributed system where the Data Receiver runs. Use Java 8 because by default, it uses the TLS 1.2 protocol, which is the most recent TLS protocol version.

You can setup secure communication between the IBM Z Common Data Provider Data Streamer and its subscribers using self-signed certificate or non self-signed certificate.

Securing communications using self-signed certificate

To secure communications between the IBM Z Common Data Provider Data Streamer and its subscribers, you must choose a streaming protocol that supports Transport Layer Security (TLS) when you configure a subscriber in a policy. You must also configure the Data Streamer and its subscribers to use TLS with self-signed certificate.

Before you begin

For more information about the streaming protocols, see [“Subscriber configuration” on page 235](#). The streaming protocols that support TLS contain either SSL or HTTPS in the name except for Kafka subscribers. For example, to secure communications between the Data Streamer and the Data Receiver, select the streaming protocol `CDP Data Receiver SSL` rather than `CDP Data Receiver`.

For Kafka subscribers, whether to enable secure communication between the Data Streamer and Kafka is not controlled by the protocol you select. Instead, edit the `kafka.properties` file in the Data Streamer working directory and set `security.protocol=SSL` or `SASL_SSL` to enable secure communication the Data Streamer and Kafka. The streaming protocol for Kafka subscribers, with or without secure communication enabled, is `CDP Kafka`.

Tip: Transport Layer Security (TLS) is the cryptographic protocol that provides secure communications for your connections. Because the Secure Sockets Layer (SSL) protocol is the predecessor to TLS, the term *Secure Sockets Layer*, or *SSL*, is often used generically to refer to TLS encryption.

About this task

The TLS protocol is provided by the IBM Java Runtime Environment that is installed on the z/OS system where the IBM Z Common Data Provider runs, and on the distributed system where the Data Receiver runs. Use Java 8 because by default, it uses the TLS 1.2 protocol, which is the most recent TLS protocol version.

The following scripts for configuring secure communications are provided in the target library `/usr/lpp/IBM/zcdp/v2r1m0/DS/LIB`:

- `setupDataStreamerSSL.sh`
- `importCertificate.sh`

Important: The Java keystore and the script `importCertificate.sh` is the only supported method for handling certificates to secure communications between the Data Streamer and subscribers. RACF keyring is not supported.

Procedure

To configure the Data Streamer and its subscribers to use TLS with self-signed certificate, complete the following steps:

1. For each Data Receiver that must use secure communications with the Data Streamer, complete the following steps on the Data Receiver system.

a) Set the following environment variables:

JAVA_HOME

The Java installation directory on the Data Receiver system.

CDPDR_HOME

The Data Receiver working directory that is described in [“Setting up a working directory and an output directory for the Data Receiver”](#) on page 113.

CDPDR_PATH

The Data Receiver output directory that is described in [“Setting up a working directory and an output directory for the Data Receiver”](#) on page 113.

For Linux systems

```
export JAVA_HOME=/java_installation_directory
export CDPDR_HOME=/dr_working_directory
export CDPDR_PATH=/dr_output_directory
```

For Windows systems

```
JAVA_HOME=/java_installation_directory
CDPDR_HOME=/dr_working_directory
CDPDR_PATH=/dr_output_directory
```

b) Download the `setupDataReceiverSSL.sh` (for Linux systems) or `setupDataReceiverSSL.bat` (for Windows systems) file from the IBM Z Common Data Provider system by using a binary protocol.

c) Move or copy the `setupDataReceiverSSL.sh` or `setupDataReceiverSSL.bat` file into the `CDPDR_HOME` directory where the `DataReceiver.jar` file is located.

d) Run the script `setupDataReceiverSSL.sh` or `setupDataReceiverSSL.bat`, as shown in the following command.

This script configures the Data Receiver to use TLS to communicate with the Data Streamer. This script requires Java Runtime Environment (JRE) 8.

For Linux systems

```
cd CDPDR_HOME
./setupDataReceiverSSL.sh datareceiver_hostname
datareceiver_ip_address datareceiver_cert_alias
keystore_password
```

For Windows systems

```
cd CDPDR_HOME
setupDataReceiverSSL.bat datareceiver_hostname
datareceiver_ip_address datareceiver_cert_alias
keystore_password
```

The following variables are used in the command:

datareceiver_hostname

The fully qualified host name of the Data Receiver.

datareceiver_ip_address

The IP address of the Data Receiver.

datareceiver_cert_alias

The alias name for the public certificate of the Data Receiver. This name must be used in importing the Data Receiver public certificate to the Data Streamer truststore. The alias is defined when a certificate is imported into a certificate store. When you run the Java **keytool**

command to reference a certificate, you must specify the alias. Aliases of all certificates in a key store can be listed using the **keytool** command:

```
keytool -v -list -keystore cdp.jks
```

The **keytool** command will prompt for the Java key store password.

keystore_password

The password that you want to use for the Data Receiver keystore.

The following files are created in the *CDPDR_HOME* directory:

passStore

Contains a secret key for password encryption.

cdp.properties

Contains the encrypted password for the Data Receiver keystore.

cdp.jks

Contains the public certificate and private key pair for the Data Receiver.

cdp.cert

The Data Receiver public certificate, which must be imported to the Data Streamer keystore.

2. For each Humio subscriber that must use secure communications with the Data Streamer, generate a public certificate and private key pair, a truststore containing the public certificates to trust, and a keystore containing the public certificate and private key, and then configure the Humio subscriber to use them. For any other subscriber that must use secure communications with the Data Streamer, generate a public certificate and private key pair, and configure the subscriber to use them.

For information about how to do this configuration, see the Humio, Logstash, Splunk, or other third-party documentation.

3. Transfer the public certificate files from the subscriber systems to the Data Streamer system using a binary mode file transfer. Give each certificate file a unique name.

Important: If a Data Streamer is to send data to more than one subscriber, the public certificate file names must be unique to avoid conflict.

4. On each Data Streamer system that must use secure communications with subscribers, set the following environment variables:

JAVA_HOME

The Java installation directory on the Data Streamer system.

CDP_HOME

The Data Streamer working directory that is described in [“Configuring the Data Streamer”](#) on page 118.

CDP_DATASTREAMER

The directory that contains the `setupDataStreamerSSL.sh` script and `DataStreamer.jar` file

5. On each Data Streamer system that must use secure communications with subscribers, run the script `setupDataStreamerSSL.sh`, as shown in the following command, where *keystore_password* represents the password that you want to use for the Data Streamer keystore.

This script configures the Data Streamer to use TLS to communicate with subscribers.

```
/usr/lpp/IBM/cdpz/v2r1mo/DS/LIB/setupDataStreamerSSL.sh keystore_password
```

The following files are created in the *CDP_HOME* directory:

passStore

Contains a secret key for password encryption.

cdp.properties

Contains the encrypted password for the Data Streamer truststore.

cdp.jks

Keystore to contain the public certificates for the subscribers.

6. On each Data Streamer system, run the script `importCertificate.sh` for each subscriber public certificate, as shown in the following command.

This script imports the public certificate for the subscriber into the Data Streamer keytool.

```
/usr/lpp/IBM/zcdp/v2r1m0/DS/LIB/importCertificate.sh cdp.cert  
subscriber_cert_alias
```

The following variables are used in the command:

cdp.cert

The fully qualified path (including the file name) for the file where the public certificate for the subscriber is stored.

subscriber_cert_alias

The alias name for the public certificate of the subscriber. The alias name is used with the **keytool** command to reference certificates and keys in the Java key store.

Tip: For Kafka subscribers, you can import the subscriber’s public certificate into a separate truststore and configure it in the `kafka.properties` file. Otherwise, the default Data Streamer truststore will be used.

Securing communications using non self-signed certificate

To secure communications between the IBM Z Common Data Provider Data Streamer and its subscribers, you must choose a streaming protocol that supports Transport Layer Security (TLS) when you configure a subscriber in a policy. You must also configure the Data Streamer and its subscribers to use TLS with non self-signed certificate.

Before you begin

For more information about the streaming protocols, see [“Subscriber configuration” on page 235](#). The streaming protocols that support TLS contain either SSL or HTTPS in the name except for Kafka subscribers. For example, to secure communications between the Data Streamer and the Data Receiver, select the streaming protocol `CDP Data Receiver SSL` rather than `CDP Data Receiver`.

For Kafka subscribers, whether to enable secure communication between the Data Streamer and Kafka is not controlled by the protocol you select. Instead, edit the `kafka.properties` file in the Data Streamer working directory and set `security.protocol=SSL` or `SASL_SSL` to enable secure communication the Data Streamer and Kafka. The streaming protocol for Kafka subscribers, with or without secure communication enabled, is `CDP Kafka`.

To setup secure communication between Data Streamer and Data Receiver using non self-signed certificate, please make sure OpenSSL command is available and the following files are generated:

- `ca.crt`: CA certificate file
- `server.crt`: server certificate file
- `server.key`: private key of the server certificate

Tip: Transport Layer Security (TLS) is the cryptographic protocol that provides secure communications for your connections. Because the Secure Sockets Layer (SSL) protocol is the predecessor to TLS, the term *Secure Sockets Layer*, or *SSL*, is often used generically to refer to TLS encryption.

About this task

The TLS protocol is provided by the IBM Java Runtime Environment that is installed on the z/OS system where the IBM Z Common Data Provider runs, and on the distributed system where the Data Receiver runs. Use Java 8 because by default, it uses the TLS 1.2 protocol, which is the most recent TLS protocol version.

The following scripts for configuring secure communications are provided in the target library `/usr/lpp/IBM/zcdp/v2r1m0/DS/LIB`:

- `setupDataStreamerSSL_CA.sh`
- `importCertificate_CA.sh`

Procedure

To configure the Data Streamer and its subscribers to use TLS with non self-signed certificate, complete the following steps:

1. For each Data Receiver that must use secure communications with the Data Streamer, complete the following steps on the Data Receiver system.
 - a) Set the following environment variables:

JAVA_HOME

The Java installation directory on the Data Receiver system.

CDPDR_HOME

The Data Receiver working directory that is described in [“Setting up a working directory and an output directory for the Data Receiver”](#) on page 113.

CDPDR_PATH

The Data Receiver output directory that is described in [“Setting up a working directory and an output directory for the Data Receiver”](#) on page 113.

For Linux systems

```
export JAVA_HOME=/java_installation_directory
export CDPDR_HOME=/dr_working_directory
export CDPDR_PATH=/dr_output_directory
```

For Windows systems

```
JAVA_HOME=/java_installation_directory
CDPDR_HOME=/dr_working_directory
CDPDR_PATH=/dr_output_directory
```

- b) Download the `setupDataReceiverSSL_CA.sh` (for Linux systems) or `setupDataReceiverSSL_CA.bat` (for Windows systems) file from the IBM Z Common Data Provider system by using a binary protocol.
- c) Move or copy the `setupDataReceiverSSL_CA.sh` or `setupDataReceiverSSL_CA.bat` file into the `CDPDR_HOME` directory where the `DataReceiver.jar` file is located.
- d) Run the script `setupDataReceiverSSL_CA.sh` or `setupDataReceiverSSL_CA.bat`, as shown in the following command.

This script configures the Data Receiver to use TLS to communicate with the Data Streamer. This script requires Java Runtime Environment (JRE) 8.

For Linux systems

```
cd CDPDR_HOME
./setupDataReceiverSSL_CA.sh datareceiver_cert_path
datareceiver_key_path keystore_password
```

For Windows systems

```
cd CDPDR_HOME
setupDataReceiverSSL_CA.bat datareceiver_cert_path
datareceiver_key_path keystore_password
```

The following variables are used in the command:

datareceiver_cert_path

The path of the Data Receiver public certificate.

datareceiver_key_path

The path of the Data Receiver private key.

keystore_password

The password that you want to use for the Data Receiver keystore.

The following files are created in the `CDPDR_HOME` directory:

passStore

Contains a secret key for password encryption.

cdp.properties

Contains the encrypted password for the Data Receiver keystore.

cdp.jks

Contains the public certificate and private key pair for the Data Receiver.

- For each Humio subscriber that must use secure communications with the Data Streamer by using non self-signed certificate, generate a public certificate and private key pair signed by CA, a truststore containing the CA certificates to trust, and a keystore containing the CA certificates, public certificates and private key, and then configure the Humio subscriber to use them. For any other subscriber that must setup secure communications with the Data Streamer by using non self-signed certificate, generate a public certificate and private key pair signed by CA, and configure the subscriber to use them.

For information about how to do this configuration, see the Humio, Logstash, Splunk, or other third-party documentation.

- Transfer the CA certificate from the subscriber systems to the Data Streamer system using a binary mode file transfer. Give each CA certificate file a unique name.

Important: If a Data Streamer is to send data to more than one subscriber using different CA certificates, the CA certificate file names must be unique to avoid conflict.

- On each Data Streamer system that must use secure communications with subscribers with non self-signed certificate, set the following environment variables:

JAVA_HOME

The Java installation directory on the Data Streamer system.

CDP_HOME

The Data Streamer working directory that is described in [“Configuring the Data Streamer”](#) on page 118.

CDP_DATASTREAMER

The directory that contains the `setupDataStreamerSSL_CA.sh` script and `DataStreamer.jar` file

- On each Data Streamer system that must use secure communications with subscribers, run the script `setupDataStreamerSSL_CA.sh` to configure the Data Streamer to use TLS to communicate with subscribers, as shown in the following command, where `keystore_password` represents the password that you want to use for the Data Streamer keystore.

```
/usr/lpp/IBM/cdpz/v2r1m0/DS/LIB/setupDataStreamerSSL_CA.sh keystore_password
```

The following files are created in the `CDP_HOME` directory:

passStore

Contains a secret key for password encryption.

cdp.properties

Contains the encrypted password for the Data Streamer truststore.

- On each Data Streamer system, run the script `importCertificate_CA.sh` for each subscriber CA certificate, as shown in the following command.

This script creates Data Streamer keystore named with `cdp.jks` in `CDP_HOME` directory and imports the CA certificate for the subscriber into the Data Streamer keystore.

```
/usr/lpp/IBM/zcdp/v2r1m0/DS/LIB/importCertificate_CA.sh CA_cert_path
CA_cert_alias
```

The following variables are used in the command:

CA_cert_path

The fully qualified path (including the file name) of the CA certificate for the subscriber.

CA_cert_alias

The alias name for the CA certificate of the subscriber. The alias name is used with the **keytool** command to reference certificates and keys in the Java key store.

In this process, you must input the keystore password, which must be the same as the password specified in step 5.

Tip: For Kafka subscribers, you can import the subscriber’s CA certificate into a separate truststore and configure it in the `kafka.properties` file. Otherwise, the default Data Streamer truststore will be used.

Preparing the IBM Z Common Data Provider and the target destinations to stream and receive data

You must prepare your target destinations to receive the z/OS operational data from IBM Z Common Data Provider Data Streamer. The preparation steps differ depending on the target destination.

About this task

For the Data Streamer to stream z/OS operational data to a target destination, you must define the streaming protocol for that target destination in the policy. [Table 15 on page 103](#) lists common target destinations with the required streaming protocols and associated information for preparing the target destination to receive z/OS operational data from the Data Streamer. For more information about defining streaming protocols for sending z/OS operational data from the Data Streamer to its subscribers, see [“Subscribers to a data stream or transform” on page 34](#).

Table 15. Common target destinations with the required streaming protocols and associated information

Target destination	Streaming protocol that must be defined in the policy	Steps for preparing target destination to receive data from the Data Streamer
IBM Z Operations Analytics	One of the following protocols: <ul style="list-style-type: none">• IZOA on IOA-LA via Logstash• IZOA on IOA-LA via Logstash SSL	Install the <code>ioaz</code> Logstash output plugin and the Logstash version that are provided with IBM Z Operations Analytics. The Logstash version that is provided with IBM Z Operations Analytics is optimized for use with Linux on z Systems. For more information, see the IBM Z Operations Analytics documentation .

Table 15. Common target destinations with the required streaming protocols and associated information (continued)

Target destination	Streaming protocol that must be defined in the policy	Steps for preparing target destination to receive data from the Data Streamer
Splunk	One of the following protocols: <ul style="list-style-type: none"> • IZOA on Splunk via Data Receiver • IZOA on Splunk via Data Receiver SSL • CDP Splunk via Data Receiver • CDP Splunk via Data Receiver SSL 	Complete the steps that are described in “Preparing to send data to Splunk” on page 105.
	One of the following protocols: <ul style="list-style-type: none"> • CDP Splunk via HEC via HTTP • CDP Splunk via HEC via HTTPS • IZOA on Splunk via HEC via HTTP • IZOA on Splunk via HEC via HTTPS 	Complete the steps that are described in “Preparing to send data to Splunk via the HTTP Event Collector” on page 107.
Humio	One of the following protocols: <ul style="list-style-type: none"> • CDP Humio via HTTP • CDP Humio via HTTPS 	Complete the steps that are described in “Preparing to send data to Humio” on page 110.

Table 15. Common target destinations with the required streaming protocols and associated information (continued)

Target destination	Streaming protocol that must be defined in the policy	Steps for preparing target destination to receive data from the Data Streamer
Elasticsearch	One of the following protocols: <ul style="list-style-type: none"> • IZOA on Elasticsearch via Logstash • IZOA on Elasticsearch via Logstash SSL • CDP Elasticsearch via Logstash • CDP Elasticsearch via Logstash SSL 	Complete the steps that are described in “Preparing to send data to Elasticsearch” on page 109.
Other target destinations	One of the following protocols based on your requirements: <ul style="list-style-type: none"> • CDP Logstash • CDP Logstash SSL • CDP Generic HTTP • CDP Generic HTTPS • CDP Kafka 	Complete the configuration for one of the following subscribers based on the protocol you choose: <ul style="list-style-type: none"> • Logstash receiver, as described in “Configuring a Logstash receiver” on page 117 • Generic HTTP subscriber, as described in “Subscribers to a data stream or transform” on page 34 • Kafka, as described in “Configuring Kafka” on page 118.

Preparing to send data to Splunk

To send data from IBM Z Common Data Provider to Splunk, you can use either the IBM Z Common Data Provider Data Receiver, or the HTTP Event Collector (HEC) function in Splunk. Prepare your environment based on the method you choose.

Before you begin

Determine which method to use to send data to Splunk.

- Send data to Splunk by using the Data Receiver.
- Send data by using the HTTP Event Collector.

Sending data by using the Data Receiver has lower CPU usage and smaller data size ingested to Splunk. But you must configure and run an IBM Z Common Data Provider Data Receiver on the system where the

Splunk Enterprise server or heavy forwarder is installed. You must also install the IBM Z Common Data Provider Buffered Splunk Ingestion App in Splunk. This is the recommended for Splunk ingestion.

Sending data by using the HTTP Event Collector provides quick end-to-end implementation and does not need the Data Receiver and the Buffered Splunk Ingestion App. However, this method will increase the data ingestion size, the cost, and the CPU usage on mainframe. Consider using the method if you are not using IZOA, and there are limitations on your ability to install or update the Splunk buffered ingestion app.

Preparing to send data to Splunk via the Data Receiver

To send data from IBM Z Common Data Provider to Splunk, configure and run an IBM Z Common Data Provider Data Receiver on the system where the Splunk Enterprise server or heavy forwarder is installed. In Splunk, you must also install the IBM Z Common Data Provider Buffered Splunk Ingestion App.

Procedure

In preparation for sending data to Splunk, complete the following steps:

1. Configure the Data Receiver, as described in [“Configuring the Data Receiver”](#) on page 113.

Important: The Data Receiver working directory and output directory must also be available to Splunk. If you want to set these directories as environment variables, verify that the Data Receiver working directory is assigned to the environment variable `CDPDR_HOME`, and that the Data Receiver output directory is assigned to the environment variable `CDPDR_PATH`, as described in [“Setting up a working directory and an output directory for the Data Receiver”](#) on page 113. If you do not want to change your system environment variables, you can specify `CDPDR_HOME` and `CDPDR_PATH` in `SPLUNK_HOME/etc/splunk-launch.conf`.

2. Start the Data Receiver, as described in [“Running the Data Receiver”](#) on page 271.

3. Define a policy with the Data Receiver as the subscriber.

For more information, see [“Subscribers to a data stream or transform”](#) on page 34.

4. From the IBM Z Common Data Provider `/usr/lpp/IBM/zcdp/v2r1m0/DS/LIB` directory, download the IBM Z Common Data Provider Buffered Splunk Ingestion App (which is a part of your SMP/E installation package) in binary mode.

The following files contain different versions of the App based on the intended platform on which Splunk runs.

Platform on which Splunk runs	File name for Buffered Splunk Ingestion App
UNIX	<code>ibm_cdpz_buffer_nix.spl</code>
Windows	<code>ibm_cdpz_buffer_win.spl</code>
Cloud	<code>ibm_cdpz_buffer_cloud.spl</code>

5. To install the Buffered Splunk Ingestion App in Splunk, complete the following steps:

- a) Log in to Splunk.
- b) Click the gear icon that is next to the word "Apps."
- c) Select **Install app from file**.
- d) Browse for the file that you downloaded in step [“4”](#) on page 106, and select that file.
- e) When you are prompted, select **Enable now**.

Important: If you are sending data to Splunk Cloud via the Data Receiver, you must first install `ibm_cdpz_buffer_nix.spl` or `ibm_cdpz_buffer_win.spl` on the forwarder where the Data Receiver is installed, and then install `ibm_cdpz_buffer_cloud.spl` on the Splunk Cloud instance.

If you are using a Splunk heavy forwarder, you do not have to index the data locally. You can use the system, sysplex, and host attributes to route the data to an appropriate indexer.

If you want to split the indexing locally, you can refine the monitor stanzas in the `inputs.conf` file by extending them to add the sysplex component of the file name. Then, duplicate the monitor stanza for each sysplex from which you want to ingest data, and change the index value on the monitor stanzas to indicate the index in which the data is to be kept. These indexes must be created within Splunk. If you update the IBM Z Common Data Provider Buffered Splunk Ingestion App, this customization is deleted.

Results

You can see the data that is loaded into Splunk by using a simple search. For example, the following search shows you all ingested z/OS SYSLOG events in the `zosdex` index:

```
index=zosdex sourcetype=zOS-SYSLOG-Console
```

If you expand an event, you can see the individual fields for which extraction rules are set.

The following search example shows you the z/OS SYSLOG messages that are issued by the CICS35 job that is running on your production sysplex and are in the `zosdex` index:

```
index=zosdex sysplex=PRODPLEX jobname=CICS35 sourcetype=zOS-SYSLOG-Console
```

You can also use Splunk analytics tools to analyze the data, or write your own deep analysis tools.

Tip: Currently the Buffered Splunk Ingestion App supports only the following log data types for indexing:

- SYSLOG
- SMF
- IMS
- RMF III
- CICS EYULOG
- CICS MSGUSR
- WebSphere SYSOUT
- WebSphere SYSPRINT
- USS Syslogd
- NetView Netlog
- zSecure

Searches for other types of data will not yield any results, although the data is in the output directory that is specified by the environment variable `CDPDR_PATH`. To use this data in the IBM Z Common Data Provider, you can edit the Buffered Splunk Ingestion App, which is installed in the directory `SPLUNK_HOME/etc/apps/ibm_cdpz_buffer/`. More dashboards and indexing capabilities are available in IBM Z Operations Analytics. For more information, see the official documentation at the [IBM Knowledge Center](#).

Splunk indexers can generally ingest data up to 300GB per day. Further data volumes require multiple indexers and search heads. See recommendations of Splunk on scaling and capacity planning for more information.

Preparing to send data to Splunk via the HTTP Event Collector

To send data to Splunk directly via the HTTP Event Collector (HEC), you must enable HEC in Splunk and create an HEC token that allows an application to communicate with Splunk without using user credentials.

About this task

The following steps are based on the operations in Splunk Enterprise version 7.3.0. For more information, see the topic *Set up and use HTTP Event Collector in Splunk Web* in the Splunk documentation of your version.

Procedure

To send data with Splunk HTTP Event Collector (Splunk HEC), complete the following steps.

1. Log on your Splunk server.
2. Go to **Settings > Data Inputs > HTTP Event Collector > Global Settings**.
3. Edit the Global Settings.
 - a) Click the **Enabled** button for the **All Tokens** option.
 - b) If you want to send data to Splunk via HTTPS, click the **Enable SSL** check box.
To send data to Splunk via HTTPS, you must configure the Data Streamer to use Transport Layer Security (TLS). For more information, see [“Securing communications between the Data Streamer and its subscribers”](#) on page 96.
 - c) In the **HTTP Port Number** field, specify a port number for the HEC to listen on.
 - d) Click **Save**.
4. Go to **Settings > Data Inputs**.
5. Click **+Add New** in the **HTTP Event Collector** row to create a new HEC token.
 - a) In the **Name** field, specify a name for the HEC token.
 - b) If you want to replace the source name for events that this input generates, specify the value in the **Source name override** field.
 - c) Click **Next**.
 - d) In the **Index** section, select the index in which Splunk stores the HEC event data.
It is suggested that you use a test index to verify your data before pushing it to a production index.

Note:

Source type

The source type is determined by the Data Streamer. Any option you choose for Input Settings will be overridden by the Data Streamer.

App context

Application contexts are folders within your Splunk instance that contains configurations for the specific data domain.

- e) Click **Review** and confirm all settings are correct.
- f) Click **Submit** to create the HEC token.

Results

The HEC token is created. You can use the token to send data to Splunk via the HTTP Event Collector. Take note of the HEC token value for creating policies in the IBM Z Common Data Provider Configuration Tool. When configuring the subscriber in your policy, always select the default option of 12 threads unless instructed otherwise by support.

Sample dashboards on Splunk

The IBM Z Common Data Provider provides several sample Splunk dashboards to visualize mainframe data.

With the sample Splunk dashboards you can visualize mainframe operational data like SYSLOG and SMF to view application performance and potential error conditions for CICS, Db2, MQ and z/OS. For more information about the dashboards, see [Dashboards - IBM Common Data Provider for z Systems on Splunkbase](#).

By default these sample Splunk dashboards use an index of zosdex, which is the index that the Data Receiver and the IBM Z Common Data Provider Buffered Splunk Ingestion App send data to. To view data that is ingested to another index by the HTTP Event Collector, update the `macro.conf` file under the `ibm_cdpz_dashboards` folder in the Splunk apps directory.

Preparing to send data to Elasticsearch

To send data from IBM Z Common Data Provider to Elasticsearch, configure Logstash by using the Logstash configuration files that are provided by IBM Z Common Data Provider.

About this task

The IBM Z Common Data Provider Elasticsearch ingestion kit contains the Logstash configuration files that are provided by IBM Z Common Data Provider.

Tip: The *Elastic Stack* (formerly known as the *ELK Stack*) is a collection of the popular open source software tools Elasticsearch, Logstash, Kibana, and Beats.

Procedure

In preparation for sending data to Elasticsearch, complete the following steps:

1. From the IBM Z Common Data Provider `/usr/lpp/IBM/zcdp/v2r1m0/DS/LIB` directory, download the IBM Z Common Data Provider Elasticsearch ingestion kit, which is in the `ibm_cdpz_ELK.tar.gz` file, in binary mode.
2. Extract the Elasticsearch ingestion kit to access the Logstash configuration files.
3. Create a new directory under the Logstash installation directory and copy the Logstash configuration files that you need for your environment to the new directory.

Table 16 on page 109 indicates the prefixes that are used in the file names for the Logstash configuration files in the IBM Z Common Data Provider Elasticsearch ingestion kit. The file name prefix is an indication of the configuration file content.

Prefix in file name of Logstash configuration file	Content of configuration file with this prefix
B_	Input stage
E_	Preparation stage
H_	Field name annotation stage
N_	Timestamp resolution stage
Q_	Output stage

The following descriptions further explain the Logstash configuration files in the IBM Z Common Data Provider Elasticsearch ingestion kit:

B_CDPz_Input.lsh file

This file contains the input stage that specifies the TCP/IP port on which Logstash listens for data from the Data Streamer. Copy this file to your Logstash configuration directory. You might need to edit the port number after you copy the file.

E_CDPz_Index.lsh file

This file contains the preparation stage. Copy this file to your Logstash configuration directory.

Files with H_ prefix in file name

Each of these files contains a unique field name annotation stage that maps to a unique data stream that IBM Z Common Data Provider can send to Logstash. To your Logstash configuration directory, copy the H_ files for only the data streams that you want to send to Elasticsearch.

Files with N_ prefix in file name

Each of these files contains a unique timestamp resolution stage that maps to a unique data stream that IBM Z Common Data Provider can send to Logstash. To your Logstash configuration directory, copy the N_ files for only the data streams that you want to send to Elasticsearch.

Q_CDPz_Elastic.lsh file

This file contains an output stage that sends all records to a single Elasticsearch server. Copy this file to your Logstash configuration directory.

After you copy the file, edit it to add the name of the host to which the stage is sending the indexing call. The default name is `localhost`, which indexes the data on the server that is running the ingestion processing. Change the value of the **hosts** parameter rather than the value of the **index** parameter. The **index** value is assigned during ingestion so that the data for each source type is sent to a different index. The host determines the Elasticsearch farm in which the data is indexed. The index determines the index in which the data is held.

To split data according to `sysplex`, you can use the `[sysplex]` field in an `if` statement that surrounds an appropriate Elasticsearch output stage.

4. In the script for starting Logstash, specify the directory that you created in step “3” on page 109.

5. Define a policy with the Logstash as the subscriber.

For more information, see “Subscribers to a data stream or transform” on page 34.

6. Start Logstash and Elasticsearch.

If the activation is successful, IBM Z Common Data Provider starts sending data to Elasticsearch.

What to do next

The data ingestion rate can be up to 480GB per day with one Elasticsearch node and one Logstash instance. If you need to send more data to Elasticsearch, and additional ingestion volume is needed, Elasticsearch should be scaled horizontally to include more nodes.

Sample dashboards on Elastic Stack

IBM Z Common Data Provider provides Sample Insight® Dashboards for Elastic Stack to demonstrate how to use mainframe operational data that is streamed by IBM Z Common Data Provider from a z/OS-based IT operations environment. It enables customers to identify, isolate, and resolve problems across their enterprise from a single interface.

You can download the IBM Z Common Data Provider Sample Insight Dashboards for Elastic Stack here:

- For Elastic Stack 5.x and 6.x versions, download [CDPz-Sample-Dashboards-ELK-1.6.0-20180423-0254.zip](#)
- For Elastic Stack 7.x versions, download [CDPz-Sample-Dashboards-ELK7x-1.7.0-20200206-0001.zip](#)

The dashboards show subsystem information on performance and message indicators based on IBM CICS Transaction Server, IBM Db2 and IBM MQ subsystems. These dashboards can be used out of the box for immediate value from your operational data or they can be used as a starting place to create your own specific Kibana dashboards. Sample data is included in the sample download for validation before IBM Z Common Data Provider data is available.

To install the sample dashboards and sample data, use the instructions in the following PDF file which includes prerequisite information, detailed installation and validation instructions, and information on how to explore and modify the Kibana sample dashboards.

- [CDPz-Sample-Dashboards-ELK-20180427.pdf](#)

You can load the sample data, or configure the IBM Z Common Data Provider to send SMF 30 and SYSLOG or Operlog data to Elastic Stack. After that, you can start to use the sample dashboards on Elastic Stack.

Preparing to send data to Humio

To send data from IBM Z Common Data Provider to Humio, you can configure IBM Z Common Data Provider to send data to Humio directly, or via Logstash.

Before you begin

Determine which method to use for sending data to Humio.

- Sending data to Humio via Logstash.
- Sending data to Humio directly.

Preparing to send data to Humio via Logstash

To send data from IBM Z Common Data Provider to Humio via Logstash, configure Logstash by using the Logstash configuration files that are provided by IBM Z Common Data Provider.

About this task

The Logstash configuration files that are provided in the IBM Z Common Data Provider Elasticsearch ingestion kit can be used for preparing sending data to Humio.

Procedure

In preparation for sending data to Humio, complete the following steps:

1. From the IBM Z Common Data Provider `/usr/lpp/IBM/zcdp/v2r1m0/DS/LIB` directory, download the IBM Z Common Data Provider Elasticsearch ingestion kit, which is in the `ibm_cdpz_ELK.tar.gz` file, in binary mode.
2. Extract the Elasticsearch ingestion kit to access the Logstash configuration files.
3. Create a new directory under the Logstash installation directory and copy the Logstash configuration files that you need for your environment to the new directory.

The column Prefix in file name of Logstash configuration file in the following table indicates the prefixes that are used in the file names for the Logstash configuration files in the IBM Z Common Data Provider Elasticsearch ingestion kit. The file name prefix is an indication of the configuration file content.

<i>Table 17. Mapping of the prefix that is used in a Logstash configuration file name to the content of the file</i>	
Prefix in file name of Logstash configuration file	Content of configuration file with this prefix
B_	Input stage
E_	Preparation stage
H_	Field name annotation stage
N_	Timestamp resolution stage
Q_	Output stage

The following descriptions further explain the Logstash configuration files in the IBM Z Common Data Provider Elasticsearch ingestion kit:

B_CDPz_Input.lsh file

This file contains the input stage that specifies the TCP/IP port on which Logstash listens for data from the Data Streamer. Copy this file to your Logstash configuration directory. You can specify the following input and filter.

```
input {
  tcp {
    port => 8080
  }
}
```

E_CDPz_Index.lsh file

This file contains the preparation stage. Copy this file to your Logstash configuration directory.

Files with H_ prefix in file name

Each of these files contains a unique field name annotation stage that maps to a unique data stream that IBM Z Common Data Provider can send to Logstash. To your Logstash configuration directory, copy the H_ files for only the data streams that you want to send to Elasticsearch.

Files with N_ prefix in file name

Each of these files contains a unique timestamp resolution stage that maps to a unique data stream that IBM Z Common Data Provider can send to Logstash. To your Logstash configuration directory, copy the N_ files for only the data streams that you want to send to Elasticsearch.

Q_CDPz_Elastic.1sh file

This file contains an output stage that sends all records to a single Elasticsearch server. Copy this file to your Logstash configuration directory.

After you copy the file, edit it according to the following example:

```
output {
  elasticsearch {
    hosts => [ "humio_url/api/v1/ingest/elastic-bulk" ]
    user => "humio_user"
    password => "ingest_token"
  }
}
```

humio_url

The Humio server URL, for example, `http://localhost:8080`.

humio_user

The Humio user.

ingest_token

The Humio repository token. If you don't already have one, go to the **Setting** tab on the Humio repository UI and click **API Tokens** to enter the token configuration interface. Then click the **Copy** button in the selected token column in the **Tokens** section.

4. In the script for starting Logstash, specify the directory that you created in step “3” on [page 111](#).
5. In the Configuration Tool, define a policy with one of the following protocols.
 - CDP on Elasticsearch via Logstash
 - CDP on Elasticsearch via Logstash SSL

For more information, see “[Subscribers to a data stream or transform](#)” on [page 34](#).

6. Start Logstash and Humio.

If the activation is successful, IBM Z Common Data Provider starts sending data to Humio.

Preparing to send data to Humio directly

To stream data to Humio directly, you must create a Humio repository and set a Humio repository ingest token that allows you to send data to a specific repository.

About this task

The following steps are based on the description in Humio documentation. For more information, see the topic *Ingest Data into Humio* in the Humio documentation.

Procedure

1. Log on to your Humio server.
2. Click **+ADD item icon > Choose Repository** to create a repository.
3. Create your Ingest Token by using one of the following methods.
 - a) **Settings > Ingest > API Tokens > Copy API default Token**
 - b) **Settings > Ingest > API Tokens > New Token > Copy your Token**

Results

The Humio repository and ingest token are created. Take note of the token value for creating policies in the IBM Z Common Data Provider Configuration Tool.

Configuring the Data Receiver

Before you can use the Data Receiver as a subscriber, you must configure it.

Before you begin

For more information about the Data Receiver, see [“Subscribers to a data stream or transform”](#) on page 34.

Setting up a working directory and an output directory for the Data Receiver

You must set up both a working directory and an output directory for the IBM Z Common Data Provider Data Receiver.

About this task

The Data Receiver working directory contains files that are created and used during the operation of the Data Receiver, including the Data Receiver properties and security-related files. The Data Receiver output directory contains output files that the Data Receiver generates based on the data that it receives.

Guidelines for both directories

Use the following guidelines to help you decide which directories to use as the working directory and the output directory:

- The directories must be readable, writable and executable.
- To avoid possible conflicts, do not use the same directory as both the working directory and the output directory for the Data Receiver.

Procedure

To set up the working directory and the output directory, specify a working directory and an output directory on the platform where you plan to run the Data Receiver.

Note: If you want the Data Receiver working directory and output directory to be available to Splunk as environment variables, you must assign the working directory to the environment variable `CDPDR_HOME`, and assign the output directory to the environment variable `CDPDR_PATH`.

If you do not specify a working directory and an output directory, the default working and output directories are used. For more information about the default working and output directories, see [“Updating the Data Receiver properties”](#) on page 114.

Copying the Data Receiver files to the target system

After you set up a working directory for the IBM Z Common Data Provider Data Receiver, you must copy the Data Receiver files to the target system, which is the system on which you plan to run the Data Receiver. All the Data Receiver files are in the package `DataReceiver.tar.gz`.

Procedure

1. Download the package `DataReceiver.tar.gz` from the z/OS system by using a binary protocol.
2. Move or copy the package `DataReceiver.tar.gz` into the working directory you specify (`CDPDR_HOME` directory).
3. Unzip the package `DataReceiver.tar.gz`.

The package `DataReceiver.tar.gz` is in the `/DS/LIB` directory for IBM Z Common Data Provider.

- `DataReceiver.jar`

The Data Receiver is in this file.

- `cdpdr.properties`

The Data Receiver properties are defined in this sample file, and this file must remain in UTF-8 encoding.

- `DataReceiver.sh`
- `DataReceiver.bat`

The `DataReceiver.sh` and `DataReceiver.bat` are used to start, stop or enable dynamic logging for the Data Receiver.

- `dynamiclogging.jar`

This file contains the codes of dynamic logging.

- `setupDataReceiverSSL.sh`
- `setupDataReceiverSSL.bat`

`setupDataReceiverSSL.sh` and `setupDataReceiverSSL.bat` are used to set up the Transport Layer Security (TLS) protocol for the communication between the Data Receiver and the Data Streamer. For more information, see [“Securing communications between the Data Streamer and its subscribers” on page 96](#).

- `log4j2.xml`

`log4j2.xml` is used to configure the default logging of Data Receiver.

- `datareceiver.service`

The `datareceiver.service` is used to run Data Receiver as a service on Linux.

- `importServiceFile.sh`

`importServiceFile.sh` is used to copy service file to systemd configuration directory.

- `DataReceiverService_alias.bat`

`DataReceiverService_alias.bat` is used to create command alias when you run Data Receiver as a Windows service.

- `DataReceiverService.exe`

`DataReceiverService.exe` is used to run Data Receiver as a service on Windows.

Updating the Data Receiver properties

After you copy the IBM Z Common Data Provider Data Receiver files to the target system, you must update the `cdpdr.properties` sample file in the Data Receiver working directory (`CDPDR_HOME` directory).

About this task

In the `cdpdr.properties` sample file, you can customize the following Data Receiver properties:

port

The port on which the Data Receiver listens for data from the Data Streamer. This port must be the same as the port that is defined for the subscriber in the policy file. For more information, see [“Subscriber configuration” on page 235](#).

cycle

The number of output files that can simultaneously exist in the Data Receiver output directory (`CDPDR_PATH` directory). The maximum value is 299, and the minimum value is 3.

The **cycle** property is related to how the Data Receiver manages disk space. For more information about how the Data Receiver manages disk space, see [“Data Receiver process for managing disk space” on page 115](#).

ssl

A y or n specification of whether to use the Transport Layer Security (TLS) protocol for Data Receiver communication with the Data Streamer. If a lowercase value y is used for this property, the Data Receiver enables TLS, and if a lowercase value n is used for this property, the Data Receiver disables TLS.

trace

A y or n specification of whether to activate tracing for the Data Receiver. If a lowercase value y is used for this property, the Data Receiver enables tracing, and if a lowercase value n is used for this property, the Data Receiver disables tracing. Typically, you activate tracing only at the request of IBM Support.

JAVA_HOME

The Java environment on which you run the Data Receiver. You can specify the local Java environment, otherwise, the `DataReceiver.sh` and the `DataReceiver.bat` scripts run the Data Receiver on the default local Java environment.

CDPDR_HOME

The path of the working directory of Data Receiver.

After the Data Receiver is started, the log directory `logs` is created in the `CDPDR_HOME` directory by default.

If `CDPDR_HOME` is not specified, the default path is the directory where the script `DataReceiver.sh` or `DataReceiver.bat` is located. The working directory (`CDPDR_HOME`) and the output directory (`CDPDR_PATH`) for the Data Receiver must be different.

CDPDR_PATH

The path of output directory for Data Receiver. If `CDPDR_PATH` is not specified, the output directory is generated as the default directory in the current `CDPDR_HOME` directory. The working directory (`CDPDR_HOME`) and the output directory (`CDPDR_PATH`) for the Data Receiver must be different.

Procedure

To update the Data Receiver properties, complete the following steps:

1. In the `cdpdr.properties` file in the `CDPDR_HOME` directory, update the property values with your configuration preferences.
2. If you choose to use the TLS protocol for Data Receiver communication with the Data Streamer (`ssl=y` in the `cdpdr.properties` file), also complete the appropriate configuration steps in [“Securing communications between the Data Streamer and its subscribers”](#) on page 96.
3. If you want to run multiple Data Receivers in your environment, you can modify the property values in the `cdpdr.properties` file. You must ensure that each Data Receiver has a unique port value.

Data Receiver process for managing disk space

The IBM Z Common Data Provider Data Receiver limits the number of output files that can simultaneously exist in its output directory (`CDPDR_PATH` directory). To manage these output files, the Data Receiver uses a cyclic process with rolling files, which are a dynamic, sequential set of files that contain a continuous stream of data.

How the process works

The **cycle** property in the Data Receiver properties file defines the number of output files that can simultaneously exist in the Data Receiver output directory.

When the number of output files in the output directory equals the value of the **cycle** property, and a new file is written, then the oldest file is deleted. Each file contains 1 hour of data. Therefore, if the value of the **cycle** property is set to 3 (3 hours), no more than 3 hours of data (in 3 output files) is on disk at a time.

The following points further illustrate this example of how the Data Receiver manages the output files:

- With the value of the **cycle** property set to 3, the suffixes -0, -1, and -2 are appended to the names of the output files.
- At the beginning of each hour, the Data Receiver erases the old data in the next file in the sequence (if it exists) and writes new data to the file (for example, if it last wrote data to the -0 file, it erases the old data in the -1 file and writes new data to the -1 file).
- One hour later, the Data Receiver erases the old data in the -2 file and writes new data to the -2 file.

Important: The target destination must read the output data in a timely manner. In this example, if the target destination does not read the data within 2 hours of when it is written, the data is lost because it is deleted.

Updating the Data Receiver Log4j Configuration File

The IBM Z Common Data Provider Data Receiver uses Apache Log4J 2 as its default logging framework. The configuration file `log4j2.xml` specifies global defaults parameters. You can specify the maximum size and number of log files to be stored on the server.

Before you begin

Make sure that you have the required disk space that you customized in the configuration file `log4j2.xml`.

About this task

To configure the default logging of the IBM Z Common Data Provider Data Receiver that uses Apache Log4J 2, you can customize the following properties value in `log4j2.xml`:

SizeBasedTriggeringPolicy

The size can be specified in bytes, with the suffix KB, MB or GB. The default value is 10MB. Once `SizeBasedTriggeringPolicy` has reached the size you specify, a rollover occurs. When the `SizeBasedTriggeringPolicy` is triggered, the log file is compressed into the `.gz` format file. The archived log file pattern is `DataReceiver_logging-yyyy-MM-dd_i.log.gz`. The `i` is an incremented number.

```
SizeBasedTriggeringPolicy size="10MB"
```

DefaultRolloverStrategy

You can specify the maximum number of the archived log files on the same day. Once the maximum value is reached, older archives are deleted on subsequent rollovers. The default value is 2000.

```
DefaultRolloverStrategy max="2000"
```

IfAccumulatedFileSize

Specifies the threshold accumulated file size. The size can be specified in bytes, with the suffix KB, MB or GB. During every rollover, when the accumulated file size reaches the threshold size, older log files that exceed the threshold size are deleted first, and the most recent log files are kept in the custom size. When any one of the conditions that **IfAccumulatedFileSize** and **IfAccumulatedFileCount** set is met, the deleting action is executed. The default value is 10GB. If the default value is used, there must be at least 10 GB in the server.

```
IfAccumulatedFileSize exceeds="10GB"
```

IfAccumulatedFileCount

Specifies the threshold count exceed which files are deleted. During every rollover, when the accumulated file count reaches the threshold count, older log files that exceed the threshold count are deleted first, and the most recent log files are kept in the custom count. When any one of the conditions that **IfAccumulatedFileSize** or **IfAccumulatedFileCount** set is met, the deleting action is executed. The default value is 1000.

```
IfAccumulatedFileCount exceeds="1000"
```


Procedure

To update the Data Receiver Configuration File `log4j2.xml`, complete the following steps:

1. Copy the `log4j2.xml` to the `CDPDR_HOME` directory.
2. In the `log4j2.xml` file in the `CDPDR_HOME` directory, update the property values with your configuration preferences. You can use the default values. If you use the default values, make sure there is at least 10 GB in your server.

Configuring a Logstash receiver

If you are using a Logstash receiver for target destinations **other than** IBM Z Operations Analytics or Elasticsearch, you must install and configure Logstash on your own. This procedure summarizes how to configure Logstash TCP input plug-in in this situation.

Before you begin

Remember: If you are sending data to IBM Z Operations Analytics or Elasticsearch, the information in this topic does not apply. Instead, complete the Logstash configuration steps as described in [Table 15 on page 103](#).

About this task

On the distributed Logstash system where you want to send z/OS operational data, you must configure the TCP input plug-in to specify the port on which Logstash listens for data from the Data Streamer.

Procedure

1. To configure the TCP input plug-in, specify the following input and filter in the Logstash configuration:

```
input {
  tcp {
    port => 8080
  }
}

filter {
  json {
    source => "message"
  }
}
```

If you want to keep a persistent connection between the Data Streamer and Logstash even when Logstash times out because of inactivity, add the following line under the input section of the Logstash configuration.

```
tcp_keep_alive => true
```

2. If you want to configure a secure data connection for streaming operational data from IBM Z Common Data Provider to Logstash, see the [Logstash documentation](#) for information about how to set up Transport Layer Security for the TCP input plug-in.
3. Optionally, you can configure the Logstash server to optimize the performance.
 - Always set JVM heap minimum (Xms) and maximum (Xmx) to the same value.
 - If events are backing up, slowly scale up the number of pipeline workers to make more CPU threads available.
 - Disable swapping to improve garbage collection times and increase node stability. You can temporarily disable swapping with `sudo swapoff -a`, or permanently disable swapping by editing the `/etc/fstab` file and commenting out all lines that contain the word `swap`.
 - The file descriptor limit (`ulimit -n`) should be set to 65536 or higher.
 - The user thread limit (`ulimit -u`) should be set to at least 4096.

4. Configure the output plug-in as appropriate for your environment.

Configuring Kafka

If you are using Kafka for target destinations, you must install and configure Kafka on your own. This procedure summarizes how to configure Kafka in this situation.

Before you begin

Determine sending data to Kafka in CSV format or Key-value format.

- For CSV format, the header information will be sent to a separate header topic on Kafka only once.
- For Key-Value format, the header information and the data are put in Key-Value pairs and sent to the data topic on Kafka.

Determine if customized topic will be used as data topic on Kafka.

- If yes, all the data will be sent to the same data topic on Kafka.
- If not, different data source types will be sent to different data topics on Kafka.

About this task

You must either create the header topic and data topics on Kafka manually, or set `auto.create.topics.enable=true` in the `server.properties` file to enable auto creation of topics on Kafka. See Kafka official website for more information.

Procedure

1. Create the header topic named `prefix-CSV-HEADER`.
2. Set `cleanup.policy=compact` for the header topic.
3. Create a data topic or data topics depending on whether customized topic is used.
 - If customized topic is used, create a data topic whose name is `prefix-customizedtopic`, for example: `CDP-CUSTOM`.
 - If customized topic is not used, create data topics for all data source types that you want to collect. The topic names are `prefix-datasourcetype`, for example: `CDP-zOS-SMF_080`.
4. Optional: Copy the `kafka.properties` file from the Data Streamer installation directory to the Data Streamer working directory (`CDP_HOME`) and edit it to change the Kafka producer configuration.

Configuring the Data Streamer

The IBM Z Common Data Provider Data Streamer streams operational data to configured subscribers in the appropriate format. It receives the data from the data gatherers (System Data Engine, Log Forwarder, or Open Streaming API), splits it into individual messages when necessary, and sends the data to the subscriber.

Before you begin

The Data Streamer can stream data to both on-platform and off-platform subscribers. To reduce general CPU usage and costs, you can run the Data Streamer on z Systems Integrated Information Processors (zIIPs).

About this task

To configure the Data Streamer, you must create the Data Streamer started task by copying the sample procedure `HBODSPRO` in the `hlq.SHBOSAMP` library, and updating the copy.

If you want to run the Data Streamer as a job rather than a procedure, use the sample job `HBODS001` in the `hlq.SHBOSAMP` library rather than procedure `HBODSPRO`.

The user ID that is associated with the Data Streamer started task must have the appropriate authority to access the IBM Z Common Data Provider program files, which include the installation files and the policy file. It must also have read/execute permissions to the Java libraries in the UNIX System Services file system.

Procedure

To create the started task, complete the following steps:

1. Copy the procedure HBODSPRO in the *hlq*.SHBOSAMP library to a user procedure library.

Tip: You can rename this procedure according to your installation conventions. When the name HBODSPRO is used in the IBM Z Common Data Provider documentation, including in the messages, it means the Data Streamer started task.

2. In your copy of the procedure HBODSPRO, customize the following parameter values for your environment:

`/usr/lpp/IBM/zcdp/v2r1m0/DS/LIB`

Replace this value with the directory where the Data Streamer is installed in your environment. This directory contains the `startup.sh` script for the Data Streamer.

`/etc/cdpConfig/myPolicy.policy`

Replace this value with the policy file path and name for your environment.

`nnnnn`

Replace this value with the port number on which the Data Streamer listens for data from the data gatherers. The default port on which the Data Streamer listens for data is 51401.

Important: All data gatherers must send data to the Data Streamer through this port. If you update this port in the Data Streamer configuration, you must also update it in the configuration for all data gatherers. For more information, see [“Data Streamer port definition” on page 10](#).

`start=w` and `start=c`

This parameter is optional. The letter "w" stands for warm, and the letter "c" stands for cold. When stopping, the Data Streamer saves data that have not been sent to the subscriber in a file buffer so that the data can be sent when the Data Streamer is started again. If you want the Data Streamer to load the data in the file buffer on startup and then send it to the subscriber, use `start=w`. If you want the Data Streamer to ignore the buffered data at startup, use `start=c`. If this parameter is absent, the default value `start=w` takes effect.

`statint=numberh/d`

This parameter is optional. This parameter specifies whether to enable the metrics capture function or not. If it does not exist, or the value of *number* is 0, then the metrics capture function is disabled. If the value of *number* is a non-zero value, then the metrics capture function will be enabled and the statistic interval is *number* h/d. The h/d is the interval time unit, the h means hour, and the d means day.

`host=ip_address`

This parameter is optional. If specified, the Data Streamer will bind to the specified IP address and can accept connections on that IP address only. The value of the host can be a host name, an IPv4 address, or an IPv6 address. The specified IP address, or the resolved IP address of the specified host name, must be a valid IP address that is active on any of the TCP/IP stacks on the LPAR, or a dynamic virtual IP address (DVIPA) which can be activated by any of the TCP/IP stacks on the LPAR. If this parameter is omitted, or is 0.0.0.0, the Data Streamer does a generic `bind()` to accept connection requests from any IP address associated with any TCP/IP stacks in the LPAR.

Note: the Data Streamer can only accept connections from the same LPAR where the Data Streamer runs. You must run the System Data Engine or Log Forwarder in the same LPAR as the Data Streamer.

3. In your copy of the procedure HBODSPRO, set the following environment variables for your environment:

`JAVA_HOME`

Specify the Java installation directory.

CDP_HOME

Specify the location of the Data Streamer working directory. The Data Streamer working directory contains files that are created and used during the operation of the Data Streamer, including the Data Streamer truststore and file buffers.

Guidelines for the working directory

Use the following guidelines to help you decide which directory to use as the working directory:

- The directory must be readable and writable by the user ID that runs the Data Streamer.
- To avoid possible conflicts, do not use a directory that is defined as the Configuration Tool working directory.

Important: Do not update, delete, or move the files in the *CDP_HOME* directory.

TZ

Specify the time zone offset for the Data Streamer. For more information, see the information about the format of the *TZ* environment variable in the [z/OS product documentation in the IBM Knowledge Center](#).

Important: If the value of the *TZ* environment variable is incorrect, the time interval that is set in the **Time Filter** transform is directly affected. For more information about the **Time Filter** transform, see [“Time Filter transform” on page 234](#).

RESOLVER_CONFIG

If a TCP/IP resolver must be explicitly provided, uncomment the *RESOLVER_CONFIG* environment variable, and specify the correct TCP/IP resolver. The Data Streamer must have access to a TCP/IP resolver. For more information, see [“Verifying the search order for the TCP/IP resolver configuration file” on page 151](#).

_BPXK_SETIBMOPT_TRANSPORT

If you want the Data Streamer to have affinity to a certain TCP/IP stack, uncomment the *_BPXK_SETIBMOPT_TRANSPORT* environment variable, and specify that TCP/IP stack.

Important: If the LPAR has multiple TCP/IP stacks, you must specify which stack you want the Data Streamer to use and specify the same TCP/IP stack for the Log Forwarder (as instructed in [“Log Forwarder properties configuration” on page 154](#)) and the System Data Engine (as instructed in [“Creating the System Data Engine started task for streaming SMF data” on page 135](#)). Otherwise, the Log Forwarder or the System Data Engine might be unable to connect to the Data Streamer.

DEFAULT_HEAP

The heap value that is used by the Data Streamer java application by default. *DEFAULT_HEAP=4g* means that the heap size is 4 GB by default. If you want to change the default heap size, uncomment the parameter **DEFAULT_HEAP** and set a new size. The default value is 4GB.

MAXIMUM_HEAP

The maximum heap value that is available to the Data Streamer Java application. *MAXIMUM_HEAP=4g* means that the maximum heap size is 4 GB. If you want to change the maximum heap size, uncomment the parameter **MAXIMUM_HEAP** and set a new size. This value must be no less than the value of **DEFAULT_HEAP**. The default value is 4 GB. For more information about the maximum heap size, see [“The Data Streamer maximum heap size” on page 121](#).

Important:

- In most cases, a heap size of 4 GB is sufficient.
 - If you are running out of memory, you might increase the heap size.
 - If your heap is never filled, you can reduce the heap size to free up storage.
 - For best performance, set the same value for both **DEFAULT_HEAP** and **MAXIMUM_HEAP**.
4. Update your security software, such as the Resource Access Control Facility (RACF), to permit the Data Streamer started task to run in your environment.
 5. Set HBODSPR0 as SYSSTC or other high priority WLM service classes.

File buffer function in the Data Streamer

The Data Streamer can buffer data when its TCP/IP connections with the analytics platforms are not stable. When the Data Streamer is stopped, any data that is buffered and not sent to the analytics platforms is written to one or more files in the UNIX System Services file system. When the Data Streamer is restarted, the UNIX System Services files are read by the new instance of the Data Streamer, and the buffered data is sent to the analytics platforms.

The Data Streamer maximum heap size

To handle large volumes of data, the Data Streamer uses 64-bit virtual storage. You can set the limit of the storage amount when you configure the Data Streamer by specifying the value for the parameter **MAXIMUM_HEAP**. This maximum heap size value is found under the **STDENV DD** statement in the JCL procedure that you use to start the Data Streamer address space. See the following example:

```
//STDENV DD *  
JAVA_HOME=/Java/J8.0_64  
CDP_HOME=/u/dhods/SYSG/zcdp/V2R1M0/DS/data  
TZ=EST5EDT  
DEFAULT_HEAP=4g  
MAXIMUM_HEAP=4g
```

The value for the parameter **MAXIMUM_HEAP** must be no less than the value for the parameter **DEFAULT_HEAP**. In this example, **MAXIMUM_HEAP=4g** means that the limit of virtual storage for the Data Streamer address space is 4 GB.

If your mainframe environment has sufficient real storage to support large usage of virtual storage, you can set the parameter **MAXIMUM_HEAP** to a value higher than 4g. Otherwise, a large **MAXIMUM_HEAP** value might cause high amounts of paging activities between virtual, real, and auxiliary storage.

The data buffer and UNIX System Services file system size

In general, if the TCP/IP connection between the Data Streamer and the analytics platform is stable, set the maximum heap size to 4 GB. However, if the connection is unstable or lost, the Data Streamer might require a larger amount of virtual storage to buffer data.

If the Data Streamer address space is stopped, any data that is buffered and not sent to the analytics platforms is written to one or more files in the UNIX System Services file system on your z/OS LPAR. The amount of data can be as large as the maximum heap size that is specified for the Data Streamer address space. Therefore, you must ensure that the UNIX System Services file system has at least the same available storage as the maximum heap size of the Data Streamer address space. For example, if the value of **MAXIMUM_HEAP** is 4g, the UNIX System Services file system must have at least 4 GB available space to hold the buffered data from the Data Streamer.

Metrics capture function in the Data Streamer

The Data Streamer can enable the metrics capture function to generate a metric statistics report to record how much data have been ingested to various subscribers. The metrics statistics report can be output to the console log, and sent to all the subscribers.

Enabling the metrics capture function

You can enable the metrics capture function by specifying the value for the parameter **statint** to a non-zero value. This parameter is added under the **STDENV DD** statement in the JCL procedure that you use to start the Data Streamer address space. See the following example:

```
//STDPARM DD *  
PGM /bin/sh  
/u/cdpConfig/cdptest/DS/LIB/startup.sh  
/u/cdpConfig/cdptest/cdpusr5/test.policy  
56116 start=c trace=s statint=5h
```

If the value of the parameter `statint` is 0 or 0h or 0d or the parameter does not exist, the metrics capture function is disabled.

The `statint` parameter indicates how long to generate and send reports for captured metrics. The value is *numberh* or *numberd*. h means hours, and d means days. In this example, `statint=5h`, which means the interval is 5 hours.

Note: Only one unit for each interval value is supported for the `statint` parameter. For example, "1d2h" is not supported, but "1d" or "2h" are supported. And the minimum unit interval is 1 hour.

Apart from the `statint` parameter, another parameter `trace` is related to the metrics capture function. If there is `trace=s/y/p`, it will enable the console message for the metrics statistics report.

By default, if the metrics capture function is enabled, the metrics statistics report is sent to all subscribers.

Dynamic setting for metrics capture function

For the Data Streamer, you can dynamically set metrics capture function using the MVS MODIFY command.

To dynamically set the metrics capture function, you can use the following commands:

Action	Command
Set statistics	<ul style="list-style-type: none"> • To disable the metrics capture function, use the following system command: <pre style="background-color: #f0f0f0; padding: 5px;">F HBODSPRO,APPL=SET,STATISTICS,STATINT=0</pre> • To enable the metrics capture function and change the statistics interval, use one of the following system commands according to your needs: <pre style="background-color: #f0f0f0; padding: 5px;">F HBODSPRO,APPL=SET,STATISTICS,STATINT=number h number is non-zero</pre> <pre style="background-color: #f0f0f0; padding: 5px;">F HBODSPRO,APPL=SET,STATISTICS,STATINT=number d number is non-zero</pre> • To enable the output of the metrics statistics report to console log, use the following system command: <pre style="background-color: #f0f0f0; padding: 5px;">F HBODSPRO,APPL=SET, STATISTICS,console=on</pre> • To disable the output of the metrics statistics report to console log, use the following system command: <pre style="background-color: #f0f0f0; padding: 5px;">F HBODSPRO,APPL=SET, STATISTICS,console=off</pre> • To enable the sending of the metrics statistics report to subscribers, use the following system command: <pre style="background-color: #f0f0f0; padding: 5px;">F HBODSPRO,APPL=SET, STATISTICS,subscribers=on</pre> • To disable the sending of the metrics statistics report to subscribers, use the following system command: <pre style="background-color: #f0f0f0; padding: 5px;">F HBODSPRO,APPL=SET, STATISTICS,subscribers=off</pre>
Display statistics	<p>To display the metrics capture function settings for the Data Streamer, use the following system command:</p> <pre style="background-color: #f0f0f0; padding: 5px;">F HBODSPRO,APPL=DISPLAY, STATISTICS</pre>
Clear statistics	<p>To clear all the metrics capture function settings of the Data Streamer, sue the following system command:</p> <pre style="background-color: #f0f0f0; padding: 5px;">F HBODSPRO,APPL=CLEAR, STATISTICS</pre>

Configuring the data gatherer components

The Log Forwarder and the System Data Engine are the primary data gatherer components of IBM Z Common Data Provider.

About this task

The z/OS File System (zFS) or systems that contain the IBM Z Common Data Provider program files (installation files, configuration files, and working files) can be shared among multiple instances of IBM Z Common Data Provider.

If a single directory contains the Log Forwarder configuration files for more than one system, or logical partition (LPAR), each configuration file name must include the names of the sysplex and the system (LPAR) to which the file applies. The file names must use the following conventions, where *SYSNAME* is the name of the system (LPAR) where the Log Forwarder runs, and *SYSPLEX* is the name of the sysplex (or monoplex) in which that system is located. The values of both *SYSPLEX* and *SYSNAME* must be in all uppercase.

- *SYSPLEX.SYSNAME.zlf.conf*
- *SYSPLEX.SYSNAME.config.properties*

If one file system contains the working directories for multiple instances of IBM Z Common Data Provider, the working directory for each Data Streamer or Log Forwarder instance must be uniquely named.


Configuring the Log Forwarder

Before you run the IBM Z Common Data Provider Log Forwarder to gather z/OS log data, you must configure it.

Before you begin

Before you configure the Log Forwarder, the following policy definition tasks, which are done in the Configuration Tool, must be complete:

1. In the Configuration Tool, create one or more policies that include one or more data streams for z/OS log data.

In the Configuration Tool, when you click the **Configure** icon  on a data stream node for data that is gathered by the Log Forwarder, the "**Configure data stream**" window is shown. [“Data stream configuration for data gathered by Log Forwarder” on page 181](#) lists the configuration values that you can update in this window.

2. After you configure the data streams for z/OS log data, click the **LOG FORWARDER** button, which is in the Global Properties section of the **Policy Profile Edit** window, to set the configuration values for your Log Forwarder environment, as described in [“LOG FORWARDER properties: Defining your Log Forwarder environment” on page 154](#).
3. [“Output from the Configuration Tool” on page 32](#) describes the output from the Configuration Tool, which includes the following Log Forwarder files:

• **.zlf.conf file**

Contains environment variables for the Log Forwarder.

• **.config.properties file**

Contains configuration information for the Log Forwarder.

About this task

To configure the Log Forwarder, you must complete the following tasks:

1. Create the Log Forwarder started task, as described in [“Creating the Log Forwarder started task” on page 125](#).

2. Copy the Log Forwarder configuration files to the environment directory, as described in [“Copying the Log Forwarder configuration files to the environment directory”](#) on page 127.
3. If appropriate for your environment, install the user exit for collecting z/OS SYSLOG data, as described in [“Installing the user exit for collecting z/OS SYSLOG data”](#) on page 127.
4. If appropriate for your environment, configure the z/OS NetView message provider for collecting NetView messages, as described in [“Configuring the z/OS NetView message provider for collecting NetView messages”](#) on page 130.

Creating the Log Forwarder started task

You must create the started task for the IBM Z Common Data Provider Log Forwarder by copying the sample procedure HBOPROC in the `hlq.SHBOSAMP` library, and updating the copy.

Procedure

To create the started task, complete the following steps:

1. Copy the procedure HBOPROC in the `hlq.SHBOSAMP` library to a user procedure library.

Tip: You can rename this procedure according to your installation conventions. When the name HBOPROC is used in the IBM Z Common Data Provider documentation, including in the messages, it means the Log Forwarder started task.

2. Update your copy of the HBOPROC procedure, according to the comments in the sample.

Update the following parameters:

/usr/lpp/IBM/zcdp/v2r1m0/LF/samples/startup.sh

Replace this value with the path where the `startup.sh` script is located.

The following path is the default installation path for the `startup.sh` script:

```
/usr/lpp/IBM/zcdp/v2r1m0/LF/samples/startup.sh
```

Change the value if a different installation path was used during the SMP/E installation.

-e

Specifies the environment directory where the Log Forwarder configuration files are located. To indicate the parameter, the option identifier `-e` precedes the directory specification, as shown in the following example:

```
-e /etc/IBM/zcdp/LF
```

The following directory is the default directory that is used if the environment directory is not specified:

```
/usr/lpp/IBM/zcdp/v2r1m0/LF/samples
```

Important: You must copy the Log Forwarder configuration files to the environment directory, as described in [“Copying the Log Forwarder configuration files to the environment directory”](#) on page 127.

-h

Specifies the IP address of the Data Streamer that the Log Forwarder will connect to. Specify this parameter only if the Data Streamer is configured to bind to a specific IP address. Because the Data Streamer and Log Forwarder must be running on the same LPAR, the IP address must be a valid IP address on the LPAR where the Log Forwarder runs. This parameter is optional. If not specified, the Log Forwarder connects to the Data Streamer on the IP address of 127.0.0.1.

3. Verify that the user ID that is associated with the Log Forwarder started task has the required authorities, as described in [“Requirements for the Log Forwarder user ID”](#) on page 126.
4. Update the SAF product that is protecting your system, such as the Resource Access Control Facility (RACF), to permit the Log Forwarder started task to run in your environment.

5. Set HBOPROC as SYSSTC WLM service class.

Requirements for the Log Forwarder user ID

The user ID that is associated with the Log Forwarder started task must have the required authorities for file access and for issuing z/OS console messages.

The following information further describes the required authorities:

- [“File access authority” on page 126](#)
- [“Authority to issue z/OS console messages” on page 126](#)

Tip: The Log Forwarder user ID does *not* require any special MVS authority to run the Log Forwarder.

File access authority

The Log Forwarder user ID must have the appropriate authority to access the Log Forwarder program files, which include the installation files, the configuration files, and the files in the working directory.

Installation file access

The Log Forwarder user ID must have read and execute permissions to the Log Forwarder installation files in the UNIX System Services file system.

Configuration file access

The Log Forwarder user ID must have read permission to the Log Forwarder configuration files in the UNIX System Services file system.

Important: The user ID that configures the Log Forwarder must have read/write permission to the configuration files.

Working directory access

The Log Forwarder user ID must have read and write permissions to the Log Forwarder working directory, which is described in [“Log Forwarder properties configuration” on page 154](#). The Log Forwarder user ID must also have permission to change the permission bits for a file in the Log Forwarder working directory.

Authority to issue z/OS console messages

The Log Forwarder user ID must have the authority to issue z/OS console messages.

If you are using the Resource Access Control Facility (RACF) as your System Authorization Facility (SAF) product, complete one of the following options to assign this authority:

Option 1 if you are using RACF

You can use the HBORACF procedure in the SHBOSAMP library to create a user ID for the Log Forwarder started task (HBOPROC procedure) and associate that user ID with the started task. The documentation that is provided in the HBORACF sample includes more information, and the following steps outline this process:

1. Copy the HBORACF procedure to a user job library.
2. To define a user ID and associate it with the Log Forwarder started task (HBOPROC procedure), update your copy of the HBORACF procedure according to the comments in the sample and the following instructions:
 - If the user ID exists, comment out the ADDUSER statement.
 - If a user ID other than HBOLGF is to be associated with the HBOPROC procedure, change the USER value on the **STDATA** parameter.
3. Submit your updated copy of the HBORACF procedure.

Option 2 if you are using RACF

Complete the following steps:

1. In RACF, add the BPX.CONSOLE resource to the class FACILITY by using the General Resource Profiles option in the RACF Services Option Menu.

2. In the BPX . CONSOLE profile that was created (or updated) in the preceding step, add the user ID that the Log Forwarder started task is associated with, and assign read access to the user ID.
3. Issue the following command to activate your changes:

```
SETROPTS RACLIST(FACILITY) REFRESH
```

Tips:

- The user ID that the HBORACF procedure creates is named HBOLGF. The Log Forwarder started task does not require the user ID to be HBOLGF. This user ID is provided only as a convenience.
- If the SAF product for your environment is not RACF, use the HBORACF sample procedure and the SAF product documentation to create the appropriate definitions in the SAF product.

Copying the Log Forwarder configuration files to the environment directory

After you create the started task for the IBM Z Common Data Provider Log Forwarder, copy the Log Forwarder configuration files from the Configuration Tool working directory to the environment directory that is specified by the `-e` parameter in the Log Forwarder started task.

Procedure

Copy the `.zlf.conf` and `.config.properties` files from the Configuration Tool working directory to the environment directory that is specified by the `-e` parameter in the Log Forwarder started task. The names of the files in the environment directory must be `zlf.conf` and `config.properties`, or `SYSPLEX.SYSTEM.zlf.conf` and `SYSPLEX.SYSTEM.config.properties`, where `SYSPLEX` is the name of the sysplex and `SYSTEM` is the name of the system.

For more information about these files, see [“Output from the Configuration Tool” on page 32](#).

Installing the user exit for collecting z/OS SYSLOG data

If you configure a **z/OS SYSLOG** data stream for gathering z/OS SYSLOG data from a user exit, you must install either the HBOSYSG or HBOMDBG user exit. If you configure only a **z/OS SYSLOG from OPERLOG** data stream for gathering z/OS SYSLOG data, the installation of a user exit is not necessary.

About this task

The HBOSYSG and HBOMDBG user exits, and other modules that are used by these user exits, are provided with IBM Z Common Data Provider and are in the SHBOLPA product library.

All modules in the SHBOLPA library must be added to the system link pack area (LPA). For more information about the LPA, see the *z/OS MVS Initialization and Tuning Guide*.

The following modules are in the SHBOLPA library:

- HBODSRAW (a program call module)
- HBOGDSDL (a program call module)
- HBOGLMSG (a program call module)
- HBOMDBG
- HBOSYSG
- HBOUERQ (a program call module)

You must install the HBOSYSG or HBOMDBG user exit on the appropriate MVS installation exit. [Table 18 on page 128](#) indicates the MVS installation exit on which to install each user exit and describes how to choose which user exit to install.

Both user exits allocate a data space with a minimum size of 100 MB and a maximum size of 500 MB. The data space is used to store z/OS SYSLOG data for retrieval by the z/OS SYSLOG.

Table 18. User exits for collecting z/OS SYSLOG data, with associated MVS installation exits and usage notes

User exit	MVS installation exit on which to install the user exit	Usage note for user exit
HBOSYSG	CNZ_MSGTOSYSLOG	If your z/OS system is not running JES3 with the DLOG option enabled, install this user exit.
HBOMDBG	CNZ_WTOMDBEXIT	If your z/OS system is running JES3 with the DLOG option enabled, install this user exit.

Important: If you are upgrading IBM Z Common Data Provider from version 1.1.0 to version 2.1.0, you must inactivate the existing exit, delete all changed modules, load the new code, and then activate the new exit. For more information about upgrading the user exit, see [“Upgrading user exit from IBM Z Common Data Provider version 1.1.0 to 2.1.0”](#) on page 17.

Procedure

To install the user exit, complete the following steps:

1. To add the load modules to an LPA, complete one of the following actions:

Action	Instruction
Add the SHBOLPA library to the pageable link pack area (PLPA) at system IPL	<p>Add the following statement to an LPALSTxx member:</p> <pre>hlq.SHBOLPA(volume)</pre> <p>Replace hlq with the target library high-level qualifier that is used to install IBM Z Common Data Provider, and replace volume with the volume identifier of the data set.</p>
Add the individual modules in the SHBOLPA library to the dynamic LPA after the system IPL	<p>Issue the following MVS system commands:</p> <pre>SETPROG LPA,ADD,MODNAME=HBOSYSG,DSNAME=hlq.SHBOLPA SETPROG LPA,ADD,MODNAME=HBOMDBG,DSNAME=hlq.SHBOLPA SETPROG LPA,ADD,MODNAME=HBODSRAW,DSNAME=hlq.SHBOLPA SETPROG LPA,ADD,MODNAME=HBOGDSDL,DSNAME=hlq.SHBOLPA SETPROG LPA,ADD,MODNAME=HBGLMSG,DSNAME=hlq.SHBOLPA SETPROG LPA,ADD,MODNAME=HBOUERQ,DSNAME=hlq.SHBOLPA</pre>

2. To install the exit, complete one of the following actions:

Action	Instruction
Install the user exit on an MVS installation exit at system IPL	<p>Add one of the following statements to a PROGxx member:</p> <ul style="list-style-type: none"> EXIT ADD EXITNAME(CNZ_MSGTOSYSLOG) MODNAME(HBOSYSG) EXIT ADD EXITNAME(CNZ_WTOMDBEXIT) MODNAME(HBOMDBG)
Dynamically install the user exit after the system IPL	<p>Issue one of the following MVS commands:</p> <ul style="list-style-type: none"> SETPROG EXIT,ADD,EXITNAME=CNZ_MSGTOSYSLOG,MODNAME=HBOSYSG SETPROG EXIT,ADD,EXITNAME=CNZ_WTOMDBEXIT,MODNAME=HBOMDBG

***manageUserExit* utility for managing the installed user exit**

The HBOSYSG and HBOMDBG user exits create system resources that might need to be managed while they are in operation. The `manageUserExit` utility is a shell script that can be used to manage the system resources. The utility is included in the product samples directory in the hierarchical file system.

The following system resources might need to be managed:

- A data space, which is used to store z/OS SYSLOG data for retrieval by the Log Forwarder.
- Program call modules, which are loaded by the user exit and made available to other programs (such as the Log Forwarder and the `manageUserExit` utility) for interacting with the data space.

manageUserExit.sh description

This utility manages the data space and program call modules that are controlled by the user exit. For example, you can use the utility to complete the following management actions:

- Refresh the data space.
- Refresh the program call modules.
- Delete the data space, unload the program call modules, and uninstall the user exit from the MVS installation exit.

Important: Before you run the `manageUserExit.sh` utility, stop any instances of the Log Forwarder that are gathering z/OS SYSLOG data. This action prevents the Log Forwarder from trying to access or call a system resource that is being deleted. An abend might occur if the Log Forwarder accesses a non-existent data space or calls a non-existent program call module.

manageUserExit.sh details

Format

```
manageUserExit.sh -p[d] [environment_configuration_directory]
```

```
manageUserExit.sh -d[p] [environment_configuration_directory]
```

```
manageUserExit.sh -u [environment_configuration_directory]
```

Parameters

-d

Refreshes the data space by deleting and re-creating it.

For normal operations, refreshing the data space is not needed. However, for example, if you are requested to refresh the data space by IBM Software Support, use this parameter to delete and re-create the data space. All z/OS SYSLOG data that is in the data space before deletion is lost.

-p

Refreshes the program call modules by unloading and reloading from the LPA.

Refreshing the program call modules might be necessary when maintenance is applied. Updates to the modules in the SHBOLPA library must be reloaded by the user exit. Use this parameter to unload the previously loaded program call modules and load the new program call modules.

Tips:

1. Before you refresh the program call modules, the modules must be loaded dynamically into the system LPA. If the program call modules are currently in the dynamic LPA, the user exit must be uninstalled, and the old program call modules must be deleted from the dynamic LPA before the new modules can be reloaded. The user exit must then be reinstalled on the MVS installation exit.
2. If the application of maintenance requires a refresh of the program call modules, the maintenance information specifies that a refresh is necessary.

-u

Deletes the data space, unloads the program call modules, and uninstalls the user exit.

Examples

manageUserExit.sh -pd /etc/IBM/zcdp/LF

This command refreshes both the data space and program call modules. In this example, the directory /etc/IBM/zcdp/LF contains the environment configuration file.

manageUserExit.sh -u

This command uses the *ZLF_CONF* environment variable to find the directory that contains the environment configuration file. It also deletes the data space, unloads the program call modules, and uninstalls the user exit.

Exit values

0

Successful completion

-1

Did not complete successfully

Messages

The utility issues messages to standard output. The messages have the prefix HBOK.

manageUserExit.sh usage notes

The following information describes some tips for using the `manageUserExit.sh` utility:

- To run the `manageUserExit.sh` utility, you must specify at least one parameter.
- Specification of the environment configuration directory is optional. However, if this directory is not specified, the *ZLF_CONF* environment variable must be set, and its value must be the working directory that contains the `zlf.conf` file that is used by the Log Forwarder.

For example, if the `zlf.conf` file is in /etc/IBM/zcdp/LF, either the environment configuration directory or the value of the *ZLF_CONF* environment variable must be this directory.

- The `-p` and `-d` parameters cannot be used with the `-u` parameter.
- The utility requests operations by using a system common storage area. The requested operation does not complete until the user exit is called by a system console message. The requested operations are not run synchronously by the utility.
- The utility can be run even if the user exit is not active or installed. The requested operations are completed when the user exit is activated and is called by a system console message.
- When the utility completes successfully, it indicates only that it made a request of the user exit. A system console message is issued by the user exit when it performs the requested operations.

Configuring the z/OS NetView message provider for collecting NetView messages

If you configure a **NetView Netlog** data stream for gathering NetView for z/OS message data, you must also configure the NetView message provider to monitor and forward NetView for z/OS messages to the Log Forwarder. The NetView message provider is defined in the REXX module HBONETV in the SHBOCLST data set.

About this task

The NetView message provider must be associated with a NetView autotask and can be run as a long-running command to get NetView for z/OS messages. The NetView autotask to which you associate the NetView message provider must have the following permissions:

- Permission to access and edit the configuration directory for the NetView message provider by using the queued sequential access method (QSAM).

- Permission to issue CZR messages by using the **PIPE** and **CNMECZFS** commands
- Permission to use the **LISTVAR** and **PPI** commands

Procedure

To prepare the NetView message provider for use, complete the following steps:

1. Ensure that the HBONETV module is placed in the DSICLD data set that is defined in the NetView procedure. Also, ensure that the NetView autotask has access to the HBONETV module.
2. In the CNMSTYLE member, specify the following information by using common variables:

Information to specify	How to specify	Example entry in CNMSTYLE member
Indication of whether to start the NetView message provider in cold or warm start mode	Specify either of the following values for the <i>COMMON.HBONETV.START</i> variable: <ul style="list-style-type: none"> • C for cold start mode • W for warm start mode, which is the default mode 	<i>COMMON.HBONETV.START = W</i>
Configuration directory for the NetView message provider	For the configuration directory, specify a partitioned data set (PDS) where the Log Forwarder can store some information to keep track of its progress in reading log data. Specify the data set as the value of the <i>COMMON.HBONETV.CONFIG.DIR</i> variable. The default value is <i>USER.CLIST</i> .	<i>COMMON.HBONETV.CONFIG.DIR = USER.CLIST</i>

Configuring the z/OS RMF Distributed Data Server to collect RMF Monitor III reports

If you configure RMF III Report data streams for gathering RMF Monitor III reports, you must also configure the RMF Distributed Data Server (DDS). The Log Forwarder sends HTTP requests to the RMF DDS and retrieves the report data from the HTTP responses. The gatherer interval is the same as the interval that RMF DDS gathers data from RMF.

About this task

The user ID that is associated with the Log Forwarder must have the permissions to send HTTP requests to the RMF Distributed Data server.

Procedure

Depending on whether or not the Log Forwarder user has a password, use one of the following methods to prepare the RMF Distributed Data server.

- If the user ID that is associated with the Log Forwarder has a password, complete the following actions:
 - a) Run the follow command to grant the user ID access to generate PassTicket when accessing the RMF DDS.

```
PERMIT IRRPTAUTH.GPMSERVE.* CLASS(PTKTDATA) ID(user)ACCESS(UPDATE)
SETROPTS RACLIST(PTKTDATA) REFRESH
```

where *user* is the user ID of the Log Forwarder started task procedure.

For more information about configuring PassTicket support for the Distributed Data Server, see [Configuring PassTicket support for the Distributed Data Server](#) in the RMF documentation.

- If the user ID that is associated with the Log Forwarder is a protected user that does not have a password, complete the following actions:
 - a) Specify the host name or TCP/IP address of the Log Forwarder in the RMF Distributed Data Servers HTTP_NOAUTH option to use the HTTP interface without authentication.
 - b) Optional: If you need secured communication, you can set up Application Transparent-Transport Layer Security (AT-TLS) so that the Log Forwarder is authenticated through a client certificate. For more information about how to set up AT-TLS, see the technote for OMEGAMON z/OS agent <http://www-01.ibm.com/support/docview.wss?uid=swg21697224&aid=1>. Though this technote is for OMEGAMON z/OS agent, replace the OMEGAMON z/OS agent information with the Log Forwarder information when you read through this technote.
- If AT-TLS is set up to enable secure communication with RMF DDS, all RMF DDS clients including the Log Forwarder must use secure communication with RMF DDS. An AT-TLS TTLS rule is required to convert the outbound connection from the Log Forwarder to RMF DDS to secure communication. For more information, see the previously mentioned technote for OMEGAMON z/OS agent.

What to do next

After you configure the RMF DDS, create a policy in the Configuration Tool that streams the RMF Monitor III reports data to the subscriber of your choice. For more information about how to create a policy, see “Creating a policy” on page 36. You can find the data stream under the data stream group **IBM Z Common Data Provider > RMF III Reports**.

Creating the Log Forwarder batch job for sending SYSLOG data to the Data Streamer

To run the IBM Z Common Data Provider Log Forwarder in batch mode so that it streams SYSLOG data to the Data Streamer, you must create the job for loading SYSLOG data in batch. You can create this job by using the sample job HBOLFBCH in the *hlq*.SHBOSAMP library, and updating the copy.

Procedure

To create the job, complete the following steps:

1. Copy the job HBOLFBCH in the *hlq*.SHBOSAMP library to a user job library.
2. Update the job card according to your site standards.
3. Update the following HB0IN DD statement. Replace *File Path* with the value of the **File Path** parameter that is specified when the data stream is configured in the Configuration Tool. Replace *Data Set Name* with the name of the data set that you want to collect SYSLOG data from in batch mode. *File Path* is the identifier of the data set.

```
//HB0IN DD *  
File Path, Data Set Name
```

You can collect SYSLOG data from multiple sources in batch mode at one time. For example,

```
//HB0IN DD *  
z0S-SYSLOG-Console_SYSLOG, ZCDP.SYSLOG1  
z0S-SYSLOG-Console_SYSLOG, ZCDP.SYSLOG2
```

4. Update the following parameters:

/usr/lpp/IBM/zcdp/v2r1m0/LF/samples/startup.sh

Replace this value with the path where the `startup.sh` script is located.

The following path is the default installation path for the `startup.sh` script:

```
/usr/lpp/IBM/zcdp/v2r1m0/LF/samples/startup.sh
```

Change the value if a different installation path was used during the SMP/E installation.

-e

Specifies the environment directory where the Log Forwarder configuration files are located. To indicate the parameter, the option identifier `-e` precedes the directory specification, as shown in the following example:

```
-e /etc/IBM/zcdp/LF
```

The following directory is the default directory that is used if the environment directory is not specified:

```
/usr/lpp/IBM/zcdp/v2r1m0/LF/samples
```

Important: You must copy the Log Forwarder configuration files to the environment directory, as described in [“Copying the Log Forwarder configuration files to the environment directory”](#) on page 127.

-b

Specifies that run the Log Forwarder in batch mode. In the batch mode, the Log Forwarder only collects batch data streams and stops after collecting all the archived z/OS SYSLOG data sets.

Note: If you want to run the Log Forwarder in batch mode, do not delete the parameter.

5. If the Data Streamer is configured to bind to a specific IP address, specify the IP address of the Data Streamer by the option identifier `-h`. Because the Data Streamer and Log Forwarder must be running on the same LPAR, the IP address must be a valid IP address on the LPAR where the Log Forwarder runs. In the following example, the Data Streamer is configured to bind to the IP address of 9.30.243.157.

```
//STDPARM DD *  
PGM /bin/sh  
/usr/lpp/IBM/zcdp/v2r1m0/LF/samples/startup.sh  
-e /etc/IBM/zcdp/LF -b -h 9.30.243.157
```

If the IP address is not specified, the Log Forwarder connects to the Data Streamer on the IP address of 127.0.0.1.


Configuring the System Data Engine

If you want to gather System Management Facilities (SMF) data, you must authorize the IBM Z Common Data Provider System Data Engine with the authorized program facility (APF), and configure the System Data Engine to run either as a started task for streaming SMF data, or as a job for loading SMF data in batch.

Before you begin

Before you configure the System Data Engine, the following policy definition tasks, which are done in the Configuration Tool, must be complete:

1. In the Configuration Tool, create one or more policies that include one or more data streams for SMF data.

In the Configuration Tool, when you click the **Configure** icon  on a data stream node for data that is gathered by the System Data Engine, the "**Configure System Data Engine data stream**" window is shown. [“Data stream configuration for data gathered by System Data Engine”](#) on page 228 lists the configuration values that you can update in this window.

2. After you configure the data streams for SMF data, click the **SYSTEM DATA ENGINE** button, which is in the Global Properties section of the **Policy Profile Edit** window, to set the configuration values for your

System Data Engine environment, as described in [“SYSTEM DATA ENGINE properties: Defining your System Data Engine environment”](#) on page 155.

3. [“Output from the Configuration Tool”](#) on page 32 describes the output from the Configuration Tool, which includes the following System Data Engine file:

.sde file

Contains configuration information for the System Data Engine.

Procedure

To configure the System Data Engine, complete the following steps:

1. Authorize the System Data Engine with the authorized program facility (APF), as described in [“Authorizing the System Data Engine with APF”](#) on page 134.
2. Complete the following steps, depending on whether you want to stream the SMF data, or load the SMF data in batch:

Option	Description
Stream SMF data	<ol style="list-style-type: none"> a. Decide which method to use for collecting SMF data, as described in “Deciding which method to use for collecting SMF data” on page 135. b. Create the System Data Engine started task, as described in “Creating the System Data Engine started task for streaming SMF data” on page 135. c. If you want to collect SMF data from the SMF user exit, install the user exit, as described in “Installing the SMF user exit” on page 139. d. If you want to collect IMS data, you must write IMS records to SMF for processing by the System Data Engine. For more information, see “Collecting IMS records by using IMS User Exit” on page 142.
Load SMF data in batch	Create the job for loading SMF data in batch, as described in “Creating the System Data Engine batch job for writing SMF data to data sets” on page 147.

Tip: You must use the z/OS SYS1.PARMLIB member SMFPRMxx (or its equivalent) to enable the collection of each SMF record type that you want to gather.

Authorizing the System Data Engine with APF

For the System Data Engine to gather System Management Facilities (SMF) data, the SHBOLoad and SHBOLLST libraries must be authorized with the authorized program facility (APF).

About this task

To authorize the SHBOLoad and SHBOLLST libraries, a library name and volume ID must be in the list of authorized libraries in the PROGxx member of the SYS1.PARMLIB library.

Procedure

Use one of the following methods to authorize the SHBOLoad and SHBOLLST libraries:

- To include the SHBOLoad and SHBOLLST libraries in APF at system IPL, add the following statement to a PROGxx member:

```
APF ADD DSNAME(hlq.SHBOLoad) VOLUME(volname)
APF ADD DSNAME(hlq.SHBOLLST) VOLUME(volname)
```

- To dynamically add the SHBLOAD and SHBOLLST libraries to APF after system IPL, issue the following MVS command:

```
SETPROG APF,ADD,DSNAME=h1q.SHBLOAD,VOLUME=volname
SETPROG APF,ADD,DSNAME=h1q.SHBOLLST,VOLUME=volname
```

Deciding which method to use for collecting SMF data

IBM Z Common Data Provider can collect System Management Facilities (SMF) data from any one of the following three sources: an SMF in-memory resource (by using the SMF real-time interface), the SMF user exit HBOSMFEX, or the SMF log stream. You must decide which method you want to use, and do the appropriate configuration for that method.

Before you begin

For more information about the SMF user exit, see [“Installing the SMF user exit” on page 139](#).

About this task

Review the following tips, and decide which method you want to use for collecting SMF data:

- If SMF is running in log stream recording mode, collect SMF data from an SMF in-memory resource by using the SMF real-time interface.
If you are running z/OS V2R1 or V2R2, APAR OA49263 must be applied to use the SMF real-time interface.
If you cannot apply APAR OA49263 to z/OS V2R1 or V2R2, use the SMF user exit to collect SMF data.
- If SMF is running in data set recording mode, consider changing the mode to log stream recording mode and collecting SMF data from an SMF in-memory resource by using the SMF real-time interface.
If you cannot run SMF in log stream recording mode, use the SMF user exit to collect SMF data.

Creating the System Data Engine started task for streaming SMF data

To have the IBM Z Common Data Provider System Data Engine stream SMF data to the Data Streamer, you must create the started task for the System Data Engine by copying the sample procedure HBOSMF in the h1q.SHBOSAMP library, and updating the copy.

Procedure

To create the started task, complete the following steps:

1. Copy the procedure HBOSMF in the h1q.SHBOSAMP library to a user procedure library.
Tip: You can rename this procedure according to your installation conventions. When the name HBOSMF is used in the IBM Z Common Data Provider documentation, including in the messages, it means the System Data Engine started task.
2. Update the high-level qualifier to the one for your IBM Z Common Data Provider target libraries that were installed by using SMP/E.
3. To enable the zIIP offloading function to run eligible code on zIIPs, specify **ZIIPOFFLOAD=YES** in the **PARM** parameter in the EXEC statement. If **ZIIPOFFLOAD** is not included in the **PARM** parameter in the EXEC statement, add it according to the following example.

```
//HBOSMFCL EXEC PGM=HBOPDE,REGION=0M,TIME=1440,  
// PARM='SHOWINPUT=NO,ZIIPOFFLOAD=YES'
```

For more information about the zIIP offloading function, see [“Offloading the System Data Engine code to z Systems Integrated Information Processors” on page 151](#).

4. If appropriate for your environment, update the interval value (in minutes or seconds) for IBM_SDE_INTERVAL, which controls how often the System Data Engine processes data.

At regular intervals, the System Data Engine queries the appropriate sources for new data. For example, it queries one of the following sources:

- SMF in-memory resource
- Shared storage to which the SMF user exit writes
- SMF log stream

The default interval for this querying is 1 minute, and the minimum interval is 1 second. After each interval, the System Data Engine sends the new SMF records to the Data Streamer.

This collect processing interval is set on the EVERY clause of the COLLECT statement.

Guidelines for determining the interval value: Changing the interval value can affect the resource consumption of IBM Z Common Data Provider. The parameter value 1 MINUTES has usually shown the best CPU performance. There are cases with very high throughput, however, when this would result in buffering too much data at once. If you want to collect data more frequently, a SDE interval of 30 SECONDS can alleviate this issue without majorly impacting CPU use. If you are changing this parameter to other values, use the following guidelines to help you determine an appropriate interval value:

- Use a large interval value to minimize overhead.
- Use a small interval value to minimize memory.
- The interval value must be small enough to produce data as often as it is required by the subscriber.
- Use an interval value that is a factor of the total time in one day. [Table 19 on page 136](#) lists some example values.
- The value for EVERY must be a positive integer and is limited to a duration that does not exceed one day. Exceeding the following values will cause an error:

```
EVERY 86400 SECONDS
EVERY 1440 MINUTES
```

- If you want to use an interval value that is equal to or greater than 60 seconds, specify that value as a whole number, and specify the time unit in minutes. For example, if you want to set the interval value to 120 seconds, instead set it to 2 minutes. [Table 19 on page 136](#) lists some example values.

Time unit	Example values
Seconds	1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 24, 25, 27, 30, 32, 36, 40, 45, 48, 50, 54
Minutes	1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15

5. If you want to reduce CPU MIPS when you run multiple System Data Engine instances in a single LPAR or reduce physical CPU MIPS on shared CPUs across multiple System Data Engine instances in a sysplex at the beginning of every minute, update the different offset time values (in minutes or seconds) for OFFTIME of each System Data Engine. OFFTIME specifies how long to defer the System Data Engine data collection, and makes each System Data Engine start SMF data collection at different times.

You can set the collect processing offset time on the OFFTIME clause of the COLLECT statement. For example:

```
//HBOIN DD *
SET IBM_SDE_OFFTIME = '4 SECONDS';
SET IBM_SDE_INTERVAL = '60 SECONDS';
// DD *
COLLECT SMF FROM &IBM_RESOURCE
EVERY &IBM_SDE_INTERVAL
OFFTIME &IBM_SDE_OFFTIME;
```

In the example, the offset time is set to 4 seconds, and the System Data Engine starts collecting SMF data at 4 seconds past the integral multiple of one minute.

The setup of OFFTIME must meet the following rules:

- The OFFTIME is optional. The default value is 0.
 - The OFFTIME is available only when the interval is 30 seconds or an integral multiple of one minute.
 - The value of OFFTIME must be a positive integer and be limited to a duration that does not exceed 5 minutes or 300 seconds, otherwise, there occurs an error.
 - The value of OFFTIME is available only when the value does not exceed half of the interval, otherwise, a warning message occurs and the OFFTIME is ignored.
6. Update the port value for IBM_UPDATE_TARGET to specify the TCP/IP port that is configured for the Data Streamer.

Tip: For more information about the Data Streamer port, see [“Configuring the Data Streamer”](#) on page 118.

7. Optional: If the Data Streamer is configured to bind to a specific IP address, set the IP address of the Data Streamer by specifying the following SET statement in the HBOIN DD statement. Because the Data Streamer and System Data Engine must be running on the same LPAR, the IP address must be a valid IP address on the LPAR where the System Data Engine runs.

```
SET IBM_DS_HOST=ip_address
```

The following example assumes the Data Streamer is configured to bind to the IP address of 9.30.243.157.

```
//HBOIN DD *
SET IBM_DS_HOST = '9.30.243.157';
SET IBM_SDE_INTERVAL = '1 MINUTES';
SET IBM_UPDATE_TARGET = 'PORT ppppp';
SET IBM_FILE_FORMAT = 'CSV';
SET IBM_RESOURCE = 'IFASMF.<resource>';
// DD PATH='/etc/cdpConfig/hboin.sde',
// PATHDISP=(KEEP),RECFM=V,LRECL=255,FILEDATA=RECORD
```

8. Replace the value /etc/cdpConfig/hboin.sde with the file path and name of the policy file that you create in the Configuration Tool.
9. Update the value for IBM_RESOURCE to specify only one of the following values as the source from which to collect SMF data.
- The SMF in-memory resource name
 - The keyword EXIT, which indicates that System Data Engine will collect SMF data from the SMF user exit HBOSMFEX.
 - The SMF log stream name

Collecting SMF data using an in-memory resource has some performance improvements over using log stream. If you have the ability to use the in-memory resource, it is recommended to do so. If you are collecting SMF data from user exits, continue to specify IBM_RESOURCE=EXIT.

Remember:

- One System Data Engine instance can collect SMF data from only one SMF in-memory resource. To collect SMF data from multiple SMF in-memory resources, you must have multiple System Data Engine instances running. For example, if you have two SMF in-memory resources, you must have two System Data Engine started tasks running to collect SMF data from the two resources respectively. Also, if the two System Data Engine started tasks are using the same set of policy files, ensure that the collected SMF records exist only in one of the SMF in-memory resources. Otherwise, those overlapping SMF records will be collected and streamed twice.
 - If you want to collect SMF data from the SMF user exit, you must install the user exit, as described in [“Installing the SMF user exit”](#) on page 139.
10. If you want the System Data Engine to use a TCP/IP stack other than the default stack, specify the name of the stack in the HBOTCAFF job step. For example, to use the TCP/IP stack named TPNAME, change the step to the following line:

```
//HBOUCAFF EXEC PGM=BPXTCAFF,PARM=TPNAME
```

Important: If the LPAR has multiple TCP/IP stacks, you must specify which stack you want the System Data Engine to use and specify the same TCP/IP stack for the Data Streamer (as instructed in “Configuring the Data Streamer” on page 118). Otherwise, the System Data Engine might be unable to connect to the Data Streamer.

11. Verify that the user ID that is associated with the System Data Engine started task has the required authorities, as described in “Requirements for the System Data Engine user ID” on page 139.
12. Update the SAF product that is protecting your system, such as the Resource Access Control Facility (RACF), to permit the System Data Engine started task to run in your environment.
13. Set HBOSMF as SYSSTC WLM service class.

The SET statement

The values of many parameters of the System Data Engine started task are specified by using the SET statement. You can find information for these parameters in this section.

Parameters

IBM_SDE_SFM70_LPAR

Specifies whether to send SMF 70 subtype 1 record data for all LPARs or just the LPAR where the System Data Engine is located. The value of this parameter must be 'ALL' or 'LOCAL'. The default value is 'ALL'.

Because the SMF 70 subtype 1 record provides LPAR and CPU data for all LPARs and processors in the CPC, the data volume might be large. If you want to reduce the data volume for SMF 70 subtype 1 record by collecting data for the current LPAR only, specify SET IBM_SDE_SFM70_LPAR = 'LOCAL' in HBOIN of System Data Engine started task procedure.

```
//HBOIN DD *  
SET IBM_SDE_INTERVAL_='60 SECONDS';  
SET IBM_SDE_SFM70_LPAR = 'LOCAL';
```

IBM_SDE_LS_RECORD

Specifies how to retrieve local SMF data from a shared log stream. The value of this parameter must be ALL or LOCAL. The default value is ALL. If you have a shared log stream across a sysplex and want each LPAR to only collect data relevant to its own instead of retrieving all the plex data, you can specify SET IBM_SDE_LS_RECORD = 'LOCAL' in HBOIN of System Data Engine started task procedure as the following:

```
//HBOIN DD *  
SET IBM_SDE_LS_RECORD = 'LOCAL';
```

When local SMF data is retrieved, the workload balance issue and single point of failure for all data can be reduced.

IBM_SDE_ACF2_RTY

Specifies CA-ACF2 SMF non-default record type. The default record type of CA-ACF2 is 230, which is set in the definition file. You can dynamically specify the SMF record type of CA-ACF2 by setting the variable IBM_SDE_ACF2_RTY in HBOIN of System Data Engine started task procedure. For example,

```
//HBOIN DD *  
SET IBM_SDE_ACF2_RTY = '229';
```

If you set the type by SET statement in JCL, it will precede the default record type in the definition file. We use a consistent name for the data stream with a prefix SMF_230_CA for CA-ACF2 record.

IBM_SDE_TSS_RTY

Specifies CA Top Secret SMF non-default record type. The default record type is 231, which is set in the definition file. You can dynamically specify the SMF record type of CA Top Secret by setting the variable `IBM_SDE_TSS_RTY` in HBOIN of System Data Engine started task procedure. For example,

```
//HBOIN DD *  
SET IBM_SDE_TSS_RTY = '200';
```

If you set the type by SET statement in JCL, it will precede the default record type in the definition file. We use a consistent name for the data stream with a prefix `SMF_231_CA` for CA Top Secret record.

Requirements for the System Data Engine user ID

If you are collecting SMF data from an in-memory resource or log stream, the user ID that is associated with the System Data Engine started task must have authority to read the SMF in-memory resource or log stream. Also, if you are collecting SMF data from a log stream, the user ID must have update access to the RACF profile `MVS.SWITCH.SMF` in the `OPERCMD5 RACF` class.

If you are collecting SMF data from the SMF user exit, there are no other requirements for the user ID.

The following information further describes the required authorities:

Authority to read the SMF in-memory resource or log stream

For example, if you are using the Resource Access Control Facility (RACF) as your System Authorization Facility (SAF) product, you must give the System Data Engine user ID read authority to the profile that you set up to secure your SMF in-memory resource or log stream. In the following examples, *IFASMF.resource* represents the name of the SMF in-memory resource or log stream that is being used to gather SMF records, and *userid* represents the System Data Engine user ID.

Tip: *IFASMF.resource* is also described in step “9” on page 137 of “Creating the System Data Engine started task for streaming SMF data” on page 135.

In-memory resource example

```
PERMIT IFA.IFASMF.resource CLASS(FACILITY) ACCESS(READ) ID(userid)
```

Log stream example

```
PERMIT IFASMF.resource CLASS(LOGSTRM) ACCESS(READ) ID(userid)
```

Update access to the RACF profile `MVS.SWITCH.SMF` in the `OPERCMD5 RACF` class (only if you are collecting SMF data from a log stream)

This authority is **not** required to process data from an SMF in-memory resource.

Update access to the RACF profile `MVS.SWITCH.SMF` in the `OPERCMD5 RACF` class is required only if you are collecting SMF data from a log stream so that the user ID can issue the **MVS SWITCH SMF** command. The System Data Engine periodically issues the **MVS SWITCH SMF** command to verify that it is accessing the most up-to-date data from the log stream. To grant the user ID update access to this RACF profile, issue the following commands:

```
PERMIT MVS.SWITCH.SMF CLASS(OPERCMD5) ACCESS(UPDATE) ID(userid)  
SETROPTS RACLIST(OPERCMD5) REFRESH
```

Installing the SMF user exit

You can configure the IBM Z Common Data Provider System Data Engine to collect System Management Facilities (SMF) data from the SMF user exit `HBOSMFEX`, which is provided with IBM Z Common Data Provider. By using the SMF user exit, you can collect streaming SMF data independently of whether SMF is running in log stream recording mode or data set recording mode.

Before you begin

The SMF user exit will not affect any existing exits. Also, make sure that you set SET IBM_RESOURCE = 'EXIT' in the SDE started task.

Important: The SMF types that you define on the **SYS** parameter in z/OS SYS1 . PARMLIB member SMFPRMxx (or its equivalent) do not take effect if you also have **SUBSYS** parameter definitions. Therefore, if you define any subsystems, you must define the associated SMF types, and the MVS installation exits IEFU83, IEFU84, and IEFU85, for each subsystem that is specified by a **SUBSYS** parameter.

About this task

If you want to use the user exit to collect SMF data, install the HBOSMFEX user exit on the following MVS installation exits:

- IEFU83
- IEFU84
- IEFU85

For more information about MVS installation exits, see the z/OS MVS installation exits documentation.

An MVS installation exit does not receive control for records when the writing of the record is suppressed either because of a system failure or because of options that were selected at IPL time or by using the **SET SMF** command.

The HBOSMFEX module is required by the HBOSMFEX user exit and is in the SHBOLPA library.

All modules in the SHBOLPA library must be added to the system link pack area (LPA). For more information about the LPA, see the z/OS MVS initialization and tuning documentation.

Procedure

To install the SMF user exit HBOSMFEX, complete the following steps:

1. To add the load modules to an LPA, complete one of the following actions:

Action	Instruction
Add the SHBOLPA library to the pageable link pack area (PLPA) at system IPL	Add the following statement to an LPALSTxx member, but replace <i>hlq</i> with the target library high-level qualifier that is used to install IBM Z Common Data Provider, and replace <i>volume</i> with the volume identifier of the data set: <pre>hlq.SHBOLPA(volume)</pre>
Add the individual modules in the SHBOLPA library to the dynamic LPA after the system IPL	Issue the following MVS system command: <pre>SETPROG LPA,ADD,MODNAME=HBOSMFEX,DSNAME=hlq.SHBOLPA</pre>

2. To install the exit, complete one of the following actions:

Action	Instruction
Install the user exit on an MVS installation exit at system IPL	Add the following statements to a PROGxx member of library SYS1 . PARMLIB: <pre>EXIT ADD EXITNAME(SYS.IEFU83) MODNAME(HBOSMFEX) EXIT ADD EXITNAME(SYS.IEFU84) MODNAME(HBOSMFEX) EXIT ADD EXITNAME(SYS.IEFU85) MODNAME(HBOSMFEX)</pre>

Action	Instruction
	<p>If you specified any subsystems in an SMFPRMxx member, you must add the exit to those subsystems. For example, for the subsystem JES2, you must add the following statements:</p> <pre data-bbox="602 302 1471 436">EXIT ADD EXITNAME(SYSJES2.IEFU83) MODNAME(HBOSMFEX) EXIT ADD EXITNAME(SYSJES2.IEFU84) MODNAME(HBOSMFEX) EXIT ADD EXITNAME(SYSJES2.IEFU85) MODNAME(HBOSMFEX)</pre>
<p>Dynamically install the user exit after the system IPL</p>	<p>Issue the following MVS commands:</p> <ul data-bbox="602 506 1471 680" style="list-style-type: none"> • SETPROG EXIT,ADD,EXITNAME=SYS.IEFU83,MODNAME=HBOSMFEX • SETPROG EXIT,ADD,EXITNAME=SYS.IEFU84,MODNAME=HBOSMFEX • SETPROG EXIT,ADD,EXITNAME=SYS.IEFU85,MODNAME=HBOSMFEX <p>If you specified any subsystems in an SMFPRMxx member, you must define the exit to those subsystems. For example, for the subsystem JES2, you must issue the following commands:</p> <pre data-bbox="602 814 1471 926">SETPROG EXIT,ADD,EXITNAME=SYSJES2.IEFU83,MODNAME=HBOSMFEX SETPROG EXIT,ADD,EXITNAME=SYSJES2.IEFU84,MODNAME=HBOSMFEX SETPROG EXIT,ADD,EXITNAME=SYSJES2.IEFU85,MODNAME=HBOSMFEX</pre>

Troubleshooting tips:

- To display the status of the SMF user exit, use the following commands:

```
– D PROG,EXIT,EXITNAME=SYS.IEFU83
```

```
– D PROG,EXIT,EXITNAME=SYS.IEFU84
```

```
– D PROG,EXIT,EXITNAME=SYS.IEFU85
```

- If you need to uninstall the user exit, see [“Uninstalling the SMF user exit” on page 141](#).

Uninstalling the SMF user exit

To uninstall the SMF user exit HBOSMFEX from a system, complete this procedure.

Procedure

1. Stop the System Data Engine.
2. Remove the SMF user exit from the MVS installation exits by issuing the following MVS commands:

```
SETPROG EXIT,DELETE,EXITNAME=SYS.IEFU83,MODNAME=HBOSMFEX
SETPROG EXIT,DELETE,EXITNAME=SYS.IEFU84,MODNAME=HBOSMFEX
SETPROG EXIT,DELETE,EXITNAME=SYS.IEFU85,MODNAME=HBOSMFEX
```

3. Remove the SMF user exit from the system link pack area (LPA) by issuing the following MVS command:

```
SETPROG LPA,DELETE,MODNAME=HBOSMFEX,FORCE=YES
```

4. Stop the Data Streamer.
5. To free the 2G above-the-bar storage, and other storage spaces that are used by the SMF user exit, run the sample job HBODSPCE.

Configuring the System Data Engine for collecting IMS records

To collect IMS records, you can either write IMS records to System Management Facilities (SMF) for processing by the System Data Engine, or configure the System Data Engine to collect IMS records from IMS log data sets.

About this task

Determine which method you want to use for collecting IMS records.

If you want to stream the data in near real-time, you can install user exit and write IMS records to SMF to be processed by the System Data Engine.

If the transaction rate of your IMS system is high, and you do not want to install the user exit, you can configure the System Data Engine to collect IMS records directly from IMS log data sets. This method collects the IMS records when log switching occurs.

If you want to collect IMS Performance Analyzer Transaction Index records, use the HBOPIMS utility provided by IBM Z Common Data Provider.

Collecting IMS records by using IMS User Exit

To collect IBM Information Management System (IMS) log data, you can write IMS records to System Management Facilities (SMF) for processing by the System Data Engine.

Before you begin

Before you complete the steps in this procedure, you must complete the configuration steps for SMF data collection. For example, in the System Data Engine started task, verify that the COLLECT statement specifies the correct source of the SMF records (for example, the SMF in-memory resource or the SMF user exit).

Also, if you are collecting SMF data by using the SMF user exit, install the SMF user exit, as described in “Installing the SMF user exit” on [page 139](#). Make sure that you install exit IEFU85 to both SYS level and STC level by running the following MVS commands:

```
SETPROG EXIT,ADD,EXITNAME=SYS.IEFU85,MODNAME=HBOSMFEX  
SETPROG EXIT,ADD,EXITNAME=SYSSTC.IEFU85,MODNAME=HBOSMFEX
```

About this task

By using IMS User Exit, all IMS log records **except for** IMS Performance Analyzer Transaction Index records can be written to SMF. For more information about collecting IMS Performance Analyzer Transaction Index records, see [“Collecting IMS Performance Analyzer Transaction Index records” on page 147](#).

Procedure

To write IMS records to SMF for processing by the System Data Engine, complete the following configuration steps:

1. Update the z/OS SYS1.PARMLIB member SMFPRMxx (or its equivalent) to enable the collection of SMF record type 127.

Add the following lines to your SMFPRMxx member to enable collection of SMF 127 at both SYS level and STC level.

```
SYS(EXITS(IEFU83,IEFU84,IEFU85),TYPE(127))  
SUBSYS(STC,EXITS(IEFU83,IEFU84,IEFU85),TYPE(127))
```

2. If you are collecting SMF data from the SMF in-memory resource, create a new, or update an existing, SMF in-memory resource to include SMF record type 127.

Important: Do not include SMF record type 127 in any SMF log stream definitions.

3. Depending on the type of IMS log records that you want to collect, choose one or both of the following methods for writing IMS log records to SMF, and complete the associated configuration steps:

Option	Description
IMS LOGWRT user exit	For writing all IMS log records, <i>except for</i> IMS Performance Analyzer Transaction Index records, install the IMS LOGWRT user exit, as described in “Installing the IMS LOGWRT user exit” on page 143.
HBOPIMS utility	For writing IMS Performance Analyzer Transaction Index records, run the HBOPIMS utility, as described in “Collecting IMS Performance Analyzer Transaction Index records” on page 147.

Installing the IMS LOGWRT user exit

IBM Z Common Data Provider provides the IMS LOGWRT user exit to write IMS log records to SMF. The System Data Engine reads the IMS log records either from an SMF in-memory resource or from storage that is created by the SMF user exit.

Before you begin

The IMS LOGWRT user exit supports IMS Version 13 or later. Ensure that the SHBOLLST library is APF-authorized.

Procedure

To install the user exit, complete the steps that apply for your installation option.

Installation option	Steps
IMS multi-user exit	<ol style="list-style-type: none"> Add the <i>hlq</i>.SHBOLLST data set to the STEPLIB concatenation of the IMS Control Region. Add the following LOGWRT user exit definition to the IMS PROCLIB member DFSDF<i>xxx</i>: <pre>EXITDEF=(TYPE=LOGWRT,EXITS=(HBOFLGX0))</pre> After the IMS Control Region JCL is updated, recycle the IMS system to activate the LOGWRT user exit.
IMS tools	<p>If IMS tools are implemented for the IMS environment, install the LOGWRT user exit by using the distributed module HBOFLGX0 that is in the SHBOLLST library. This module is specified as EXITNAME (HBOFLGX0). IMS Tools does not require the load library to be inserted into the IMS Control Region STEPLIB JCL.</p> <ol style="list-style-type: none"> Add an IMS tools user exit definition to the IMS PROCLIB member GLXEXIT0, as shown in the following example: <pre>EXITDEF (TYPE (LOGR) EXITNAME (HBOFLGX0) LOADLIB (hlq.SHBOLLST))</pre> To activate the LOGWRT user exit, recycle the IMS system.
Stand-alone exit	<p>The SHBOLLST library contains member DFSFLGX0, which is the member name that IMS searches for during startup. The DFSFLGX0 module loads HBOLGX?0, which writes IMS log records to SMF.</p> <ol style="list-style-type: none"> Add the SHBOLLST library to the STEPLIB concatenation of the IMS Control Region, and verify that the DFSFLGX0 module in this library is concatenated before any other module of the same name. After the IMS Control Region JCL is updated, recycle the IMS system to activate the LOGWRT user exit.

When the LOGWRT user exit initializes successfully, the following message is written to the z/OS console:
HB08101I CDP IMS LOGWRT EXIT ACTIVATED FOR IMSID=iii

Collecting IMS records without using IMS User Exit

If the transaction rate of your IMS system is high, and you do not want to install the IMS LOGWRT user exit, you can use the System Data Engine to collect IMS records directly from IMS log data sets.

About this task

You can collect IMS log records directly from IMS log data sets by using the following methods:

A separate System Data Engine started task

You can use a separate System Data Engine started task to monitor the IMS online log data set (OLDS) status and collect the IMS log records when log switching occurs.

A System Data Engine batch job

You can use a System Data Engine batch job to load IMS log records from the IMS log data sets that are specified in the HBOLOG DD statement.

Creating the System Data Engine started task for collecting IMS records

To have the IBM Z Common Data Provider System Data Engine stream IMS data to the Data Streamer without using the IMS User Exit, you must create the started task for the System Data Engine by copying the sample procedure HB0IMS in the *hlq.SHBOSAMP* library, and updating the copy.

Procedure

1. Copy the procedure HB0IMS in the *hlq.SHBOSAMP* library to a user procedure library.
2. Update HB0 ν rm to the high-level qualifier for your IBM Z Common Data Provider target libraries that were installed by using SMP/E.
3. Update IMS ν rm to the high-level qualifier for your IMS target libraries that were installed by using SMP/E.

Tip: If multiple IMS versions are running in the same LPAR, update IMS ν rm to the high-level qualifier for the target libraries of the lowest IMS version.

4. To enable the zIIP offloading function to run eligible code on zIIPs, specify **ZIIPOFFLOAD=YES** in the **PARM** parameter in the EXEC statement.

```
//HBOSMFCL EXEC PGM=HBOPDE,REGION=0M,TIME=1440,  
//          PARM='SHOWINPUT=NO,ZIIPOFFLOAD=YES'
```

For more information about the zIIP offloading function, see [“Offloading the System Data Engine code to z Systems Integrated Information Processors” on page 151](#).

5. If appropriate for your environment, update the interval value (in minutes or seconds) for IBM_SDE_INTERVAL, which controls how often the System Data Engine processes data.
At regular intervals, the System Data Engine queries the OLDS data set status from the IMS system. Align the interval with the time when IMS OLDS data set switching usually occurs.
This collect processing interval is set on the EVERY clause of the COLLECT statement.
6. Update the port value for IBM_UPDATE_TARGET to specify the TCP/IP port that is configured for the Data Streamer.
7. Optional: If the Data Streamer is configured to bind to a specific IP address, set the IP address of the Data Streamer by specifying the following SET statement in the HBOIN DD statement. Because the Data Streamer and System Data Engine must be running on the same LPAR, the IP address must be a valid IP address on the LPAR where the System Data Engine runs.

```
SET IBM_DS_HOST=ip_address
```

The following example assumes the Data Streamer is configured to bind to the IP address of 9.30.243.157.

```
//HBOIN DD *
SET IBM_DS_HOST = '9.30.243.157';
SET IBM_SDE_INTERVAL = '5 MINUTES';
SET IBM_UPDATE_TARGET = 'PORT ppppp';
SET IBM_FILE_FORMAT = 'CSV';
// DD PATH='/etc/cdpConfig/hboin.sde',
// PATHDISP=(KEEP),RECFM=V,LRECL=255,FILEDATA=RECORD
```

8. Replace the value `/etc/cdpConfig/hboin.sde` with the file path and name of the policy file that you create in the Configuration Tool.
9. If you want the System Data Engine to use a TCP/IP stack other than the default stack, specify the name of the stack in the HBOTCAFF job step. For example, to use the TCP/IP stack named TPNAME, change the step to the following line:

```
//HBOTCAFF EXEC PGM=BPXTCAFF,PARM=TPNAME
```

Important: If the LPAR has multiple TCP/IP stacks, you must specify which stack you want the System Data Engine to use and specify the same TCP/IP stack for the Data Streamer (as instructed in [“Configuring the Data Streamer”](#) on page 118). Otherwise, the System Data Engine might be unable to connect to the Data Streamer.

10. Verify that the user ID that is associated with the System Data Engine started task has the required authorities as described in [“Requirements for the System Data Engine user ID for collecting IMS records”](#) on page 145.

Requirements for the System Data Engine user ID for collecting IMS records

If you are collecting data from IMS, the user ID that is associated with the System Data Engine started task must have authority to read the IMS RECON and online log data sets (OLDS). Also, if authorization control for DBRC API requests is established, the user ID must have read access to the security resource profiles for the STARTDBRC, STOPDBRC, and QUERY TYPE=OLDS API requests.

The following information further describes the required authorities:

Authority to read the RECON data sets and Online Log data sets (OLDS)

For example, if you are using the Resource Access Control Facility (RACF) as your System Authorization Facility (SAF) product, you must give the System Data Engine user ID read authority to the profiles for the IMS RECON and online log data sets.

```
PERMIT hlq.RECON* CLASS(DATASET) ACCESS(READ) ID(userid)
PERMIT hlq.OLP* CLASS(DATASET) ACCESS(READ) ID(userid)
```

hlq is the high-level qualifier of the RECON and online log data sets.

Authority to issue the DBRC API requests

For example, if you are using the Resource Access Control Facility (RACF) to protect the DBRC API requests, you must give the System Data Engine user ID read authority to the following security resource profiles.

```
PERMIT hlq.STDBRC CLASS(FACILITY)
ACCESS(READ) ID(userid)
PERMIT hlq.LIST.LOG.ALLOLDS CLASS(FACILITY)
ACCESS(READ) ID(userid)
```

hlq is the high-level qualifier of the resource name.

Creating the System Data Engine batch job for writing IMS log data to data sets

Create a batch job for the System Data Engine to run in batch mode so that the output is written to a file instead of being streamed to the Data Streamer. You can create this job based on the sample job HBOJBIMS in the *hlq.SHBOSAMP* library.

Procedure

1. Copy the sample job HBOJBIMS from *hlq.SHBOSAMP* to a user job library.
2. Update the job card according to your environment.

- To enable the zIIP offloading function to run eligible code on zIIPs, specify **ZIIPOFFLOAD=YES** in the **PARM** parameter in the EXEC statement.

```
//HBOSMFCL EXEC PGM=HBOPDE,REGION=OM,TIME=1440,
//          PARM='SHOWINPUT=NO,ZIIPOFFLOAD=YES'
```

For more information about the zIIP offloading function, see “Offloading the System Data Engine code to z Systems Integrated Information Processors” on page 151.

- Update the following STEPLIB DD statement to specify the *hlq*.SHBLOAD data set.

```
//STEPLIB DD DISP=SHR,DSN=HBOvrm.SHBLOAD
```

- Update the following HBOIN DD statements to specify the *hlq*.SHBODEFS data set members.

```
//HBOIN DD DISP=SHR,DSN=HBOvrm.SHBODEFS(HBOCCSV)
//      DD DISP=SHR,DSN=HBOvrm.SHBODEFS(HBOLLSMF)
//      DD DISP=SHR,DSN=HBOvrm.SHBODEFS(HBORSIMS)
//      DD DISP=SHR,DSN=HBOvrm.SHBODEFS(HBOUSIMS)
```

- For each IMS log record type that you want to collect, create a DD statement to receive the output. Change *IMSxxxxx* to the DD name of the IMS log record type. Refer to the SET IBM_FILE statements in the *hlq*.SHBODEFS(HBOUSIMS) member for the output DD names of IMS log record types.

```
//IMSxxxxx DD SYSOUT=*,RECFM=V,LRECL=32756
```

The following example shows the DD statements for receiving the output for IMS log record types x07 and x08.

```
//* Sample COLLECT statement for processing log stream data
//*
// DD *
COLLECT IMS
  COMMIT AFTER END OF FILE;
/*
//HBOLOG DD DISP=SHR,DSN=stored.imsdata
//HBOOUT DD SYSOUT=*
//HBODUMP DD SYSOUT=*
//IMS07 DD SYSOUT=*, RECFM=V,LRECL=32756 for record type 07
//IMS08 DD SYSOUT=*, RECFM=V,LRECL=32756 for record type 08
```

Creating the System Data Engine batch job for sending IMS data to the Data Streamer

Create a batch job for the System Data Engine to run in batch mode so that the output is streamed to the Data Streamer instead of being written to a file. You can create this job based on the sample job HBOJBIM2 in the *hlq*.SHBOCNTL library.

Procedure

- Copy the sample job HBOJBIM2 from *hlq*.SHBOCNTL to a user job library.
- Update the job card according to your environment.
- If affinity with a specific TCP/IP stack is needed, add the name of the stack to the end of the HBOTCAFF job step like the following example:

```
//HBOTCAFF EXEC PGM=BPXTCAFF,PARM=TPNAME
```

- To enable the zIIP offloading function to run eligible code on zIIPs, specify **ZIIPOFFLOAD=YES** in the **PARM** parameter in the EXEC statement.

For more information about the zIIP offloading function, see “Offloading the System Data Engine code to z Systems Integrated Information Processors” on page 151.

- Update the following STEPLIB DD statement to specify the *hlq*.SHBLOAD data set.

```
//STEPLIB DD DISP=SHR,DSN=HBOvrm.SHBLOAD
```

- Update the port value for **IBM_UPDATE_TARGET** to specify the TCP/IP port that is configured for the Data Streamer.

- Optional: If the Data Streamer is configured to bind to a specific IP address, set the IP address of the Data Streamer by specifying the following SET statement in the HBOIN DD statement. Because the Data Streamer and System Data Engine must be running on the same LPAR, the IP address must be a valid IP address on the LPAR where the System Data Engine runs.

```
SET IBM_DS_HOST=ip_address
```

The following example assumes the Data Streamer is configured to bind to the IP address of 9.30.243.157.

```
//HBOIN DD *
SET IBM_DS_HOST = '9.30.243.157';
SET IBM_UPDATE_TARGET = 'PORT ppppp';
SET IBM_FILE_FORMAT = 'CSV';
// DD PATH='/etc/cdpConfig/hboin.sde',
// PATHDISP=(KEEP),RECFM=V,LRECL=255,FILEDATA=RECORD
```

- Replace the value `/etc/cdpConfig/hboin.sde` with the file path and name of the policy file that you create in the Configuration Tool.
- Update the HBOLOG DD to specify the IMS log file name.

Collecting IMS Performance Analyzer Transaction Index records

IMS Performance Analyzer batch reporting can create specialized extract files for IMS Transaction Index and IMS Connect Transaction Index records. IBM Z Common Data Provider provides the HBOPIMS utility for reading IMS Transaction Index and IMS Connect Transaction Index records from the extract files and writing the records to SMF for processing by the System Data Engine.

Procedure

To write the IMS Transaction Index records (x'CA01') or IMS Connect Transaction Index records (x'CA20') to SMF record type 127, subtype 1000, customize and run the HBOJIMS JCL in the `hlq.SHBOSAMP` data set on the system where the System Data Engine is running.

The comments in the JCL job include instructions for customizing and running the job.

Creating the System Data Engine batch job for writing SMF data to data sets

To run the IBM Z Common Data Provider System Data Engine in batch mode so that it writes its output to a file, rather than streaming it to the Data Streamer, you must create the job for loading SMF data in batch. You can create this job by using the sample job HBOJBCOL in the `hlq.SHBOSAMP` library, and updating the copy.

Procedure

To create the job, complete the following steps:

- Copy the job HBOJBCOL in the `hlq.SHBOSAMP` library to a user job library.
- Update the job card according to your site standards.
- To enable the zIIP offloading function to run eligible code on zIIPs, specify **ZIIPOFFLOAD=YES** in the **PARM** parameter in the EXEC statement.

```
//HBOSMFCL EXEC PGM=HBOPDE,REGION=0M,TIME=1440,
// PARM='SHOWINPUT=NO,ZIIPOFFLOAD=YES'
```

For more information about the zIIP offloading function, see [“Offloading the System Data Engine code to z Systems Integrated Information Processors” on page 151](#).

- Update the following STEPLIB DD statement to refer to the `hlq.SHBOLoad` data set:

```
//STEPLIB DD DISP=SHR,DSN=HBOvim.LOAD
```

- For each SMF record type that you want to collect, update the following control statements, which are provided by the HBOIN DD statement, and change the variable *nnn* to the appropriate SMF record type value, for example, 030, 080, or 110.

```
//HBOIN DD DISP=SHR,DSN=HBOvrm.SHBODEFS(HBOCCSV)
// DD DISP=SHR,DSN=HBOvrm.SHBODEFS(HBOLLSMF)
// DD DISP=SHR,DSN=HBOvrm.SHBODEFS(HBORSnnn)
// DD DISP=SHR,DSN=HBOvrm.SHBODEFS(HBOUSnnn)
```

Most of the control statements that are required to run the System Data Engine are provided in the *hlq*.SHBODEFS data set and must not be changed.

Each member in the HBOIN DD concatenation specifies a task that the System Data Engine must do. The last statement in the HBOIN DD concatenation must be a COLLECT control statement, which initiates the processing of the input data by the System Data Engine.

The following example shows the control statements for SMF record types 80 and SMF_110_1_KPI:

```
//* CONTROL STATEMENTS
//*
//HBOIN DD DISP=SHR,DSN=hlq.m1q.SHBODEFS(HBOCCSV)
// DD DISP=SHR,DSN=hlq.m1q.SHBODEFS(HBOLLSMF)
// DD DISP=SHR,DSN=hlq.m1q.SHBODEFS(HBOTCIFI) for type 110_1_KPI
// DD DISP=SHR,DSN=hlq.m1q.SHBODEFS(HBORS110) for type 110_1_KPI
// DD DISP=SHR,DSN=hlq.m1q.SHBODEFS(HBOU110I) for type 110_1_KPI
// DD DISP=SHR,DSN=hlq.m1q.SHBODEFS(HBORS080) for type 80
// DD DISP=SHR,DSN=hlq.m1q.SHBODEFS(HBOUS080) for type 80
```

- For each SMF record type that you specify for collection, add a DD statement, such as the following statement, to receive the output, and change the variable *nnn* to the appropriate SMF record type value, for example, 030, 080, or 110.

```
//SMFnnn DD SYSOUT=*,RECFM=VB,LRECL=32756
```

The following example shows the DD statements for receiving the output for SMF record types 80 and SMF_110_1_KPI:

```
//* Sample COLLECT statement for processing log stream data
//*
// DD *
COLLECT SMF
  COMMIT AFTER END OF FILE;
/*
//HBOLOG DD DISP=SHR,DSN=stored.smfdata
//HBOOUT DD SYSOUT=*
//HBODUMP DD SYSOUT=*
//SMF080 DD SYSOUT=* for type 80
//SMF110 DD SYSOUT=* for type 110_1_KPI
//SMF11001 DD SYSOUT=* for type 110_1_KPI
//SMF110FC DD SYSOUT=* for type 110_1_KPI
//SMF110TX DD SYSOUT=* for type 110_1_KPI
//SMF1101I DD SYSOUT=* for type 110_1_KPI
```

Creating the System Data Engine batch job for sending SMF data to the Data Streamer

To run the IBM Z Common Data Provider System Data Engine in batch mode so that it streams its output to the data streamer, rather than writing it to a file, you must create the job for loading SMF data in batch. You can create this job by using the sample job HBOJBC02 in the *hlq*.SHBOCNTL library, and updating the copy.

Procedure

To create the job, complete the following steps:

- Copy the job HBOJBC02 in the *hlq*.SHBOCNTL library to a user job library.

- Update the job card according to your site standards.
- If affinity with a specific TCP/IP stack is needed, add the name of the stack to the end of the HBOTCAFF job step like the following example:

```
//HBOTCAFF EXEC PGM=BPXTCAFF,PARM=TPNAME
```

- To enable the zIIP offloading function to run eligible code on zIIPs, specify **ZIIPOFFLOAD=YES** in the **PARM** parameter in the EXEC statement.
For more information about the zIIP offloading function, see [“Offloading the System Data Engine code to z Systems Integrated Information Processors” on page 151](#).
- Update the following STEPLIB DD statement to refer to the *hlq*.SHBOLoad data set:

```
//STEPLIB DD DISP=SHR,DSN=HBOvrm.LOAD
```

- Update the port value for **IBM_UPDATE_TARGET** to specify the TCP/IP port that is configured for the Data Streamer.
- Optional: If the Data Streamer is configured to bind to a specific IP address, set the IP address of the Data Streamer by specifying the following SET statement in the HBOIN DD statement. Because the Data Streamer and System Data Engine must be running on the same LPAR, the IP address must be a valid IP address on the LPAR where the System Data Engine runs.

```
SET IBM_DS_HOST=ip_address
```

The following example assumes the Data Streamer is configured to bind to the IP address of 9.30.243.157.

```
//HBOIN DD *
SET IBM_DS_HOST = '9.30.243.157';
SET IBM_UPDATE_TARGET = 'PORT ppppp';
SET IBM_FILE_FORMAT = 'CSV';
// DD PATH='/etc/cdpConfig/hboin.sde',
// PATHDISP=(KEEP),RECFM=V,LRECL=255,FILEDATA=RECORD
```

- Replace the value */etc/cdpConfig/hboin.sde* with the file path and name of the policy file that you create in the Configuration Tool.
- Update the HBOLoad DD statement to specify the SMF log file name.

Creating the System Data Engine batch job for writing DCOLLECT data to data sets

To run the IBM Z Common Data Provider System Data Engine in batch mode so that it writes its output to a file, rather than streaming it to the Data Streamer, you must create the job for loading DCOLLECT data in batch. You can create this job by using the sample job HBOJBDCO in the *hlq*.SHBOSAMP library, and updating the copy.

Procedure

To create the job, complete the following steps:

- Copy the job HBOJBDCO in the *hlq*.SHBOSAMP library to a user job library.
- Update the job card according to your site standards.
- If the job must have an affinity to a specific TCP/IP stack, add the name of the stack to the end of the HBOTCAFF job step, for example:

```
//HBOTCAFF EXEC PGM=BPXTCAFF,PARM=TPNAME
```

- To enable the zIIP offloading function to run eligible code on zIIPs, specify **ZIIPOFFLOAD=YES** in the **PARM** parameter in the EXEC statement.
For more information about the zIIP offloading function, see [“Offloading the System Data Engine code to z Systems Integrated Information Processors” on page 151](#).
- Update the following STEPLIB DD statement to refer to the *hlq*.SHBOLoad data set.

```
//STEPLIB DD DISP=SHR,DSN=HBOvrm.LOAD
```

6. The output DD statements related to 18 record types are already specified in the sample job you copied. For more information about the 18 record types, see [“DCOLLECT Data stream reference” on page 176](#). If you add a new DCOLLECT record type in the future, you must add DD statements related to the new record type. Refer to the SET IBM_FILE statements in HBOUDCOL for more information.
7. Update the HB0LOG DD statement to specify the DCOLLECT log file name.

Creating the System Data Engine batch job for sending DCOLLECT data to the Data Streamer

To run the IBM Z Common Data Provider System Data Engine in batch mode so that it streams its output to the data streamer, rather than writing it to a file, you must create the job for loading DCOLLECT data in batch. You can create this job by using the sample job HBOJBDC2 in the *hlq*.SHBOCNTL library, and updating the copy.

Procedure

To create the job, complete the following steps:

1. Copy the sample job HBOJBDC2 from *hlq*.SHBOCNTL library to a user job library.
2. Update the job card according to your environment.
3. If you want to specify a TCP/IP stack for your environment, add the name of the stack to the end of the HBOTCAFF job step, for example:

```
//HBOTCAFF EXEC PGM=BPXTCAFF,PARM=TPNAME
```

4. To enable the zIIP offloading function to run eligible code on zIIPs, specify **ZIIPOFFLOAD=YES** in the **PARM** parameter in the EXEC statement.

For more information about the zIIP offloading function, see [“Offloading the System Data Engine code to z Systems Integrated Information Processors” on page 151](#).

5. Update the following STEPLIB DD statement to refer to the *hlq*.SHBOLOAD data set:

```
//STEPLIB DD DISP=SHR,DSN=HBOvrm.SHBOLOAD
```

6. Update the port value for **IBM_UPDATE_TARGET** to specify the TCP/IP port that is configured for the Data Streamer.
7. Optional: If the Data Streamer is configured to bind to a specific IP address, set the IP address of the Data Streamer by specifying the following SET statement in the HBOIN DD statement. Because the Data Streamer and System Data Engine must be running on the same LPAR, the IP address must be a valid IP address on the LPAR where the System Data Engine runs.

```
SET IBM_DS_HOST=ip_address
```

The following example assumes the Data Streamer is configured to bind to the IP address of 9.30.243.157.

```
//HBOIN DD *  
SET IBM_DS_HOST = '9.30.243.157';  
SET IBM_UPDATE_TARGET = 'PORT ppppp';  
SET IBM_FILE_FORMAT = 'CSV';  
// DD PATH='/etc/cdpConfig/hboin.sde',  
// PATHDISP=(KEEP),RECFM=V,LRECL=255,FILEDATA=RECORD
```

8. Replace the value */etc/cdpConfig/hboin.sde* with the file path and name of the policy file that you create in the Configuration Tool.
9. Update the HB0LOG DD statement to specify the DCOLLECT log file name.

Offloading the System Data Engine code to z Systems Integrated Information Processors

The System Data Engine can offload most of the code to run on z Systems Integrated Information Processors (zIIPs). This operation frees the general-purpose processors (GCPs) for other work. It can also reduce software licensing costs.

About this task

Not all System Data Engine code is eligible to run on zIIPs. The System Data Engine must switch to task mode for the portion of code that must run in task mode, and switch back to Service Request Block (SRB) when that portion of code finishes running. The synchronization between task and enclave SRB creates additional overhead at the address space level. As a result, the total CPU time (GCPs plus zIIPs) when zIIP offloading is enabled is slightly higher than the total CPU time when zIIP offloading is not enabled. Therefore, if the zIIP capacity on the logical partition is insufficient, do not enable the zIIP offloading function.

Procedure

- To activate the zIIP offloading function, specify **ZIIPOFFLOAD=YES** in the **PARM** parameter in the EXEC statement of the System Data Engine started task procedure or batch job JCL as shown in the following example.

```
//HBOSMFCL EXEC PGM=HBOPDE,REGION=0M,TIME=1440,
//          PARM='SHOWINPUT=NO,ZIIPOFFLOAD=YES'
//STEPLIB DD DISP=SHR,DSN=HB0vrm.LOAD
//HBOIN DD *
SET IBM_SDE_INTERVAL = '1 MINUTES';
SET IBM_UPDATE_TARGET = 'PORT 61001';
SET IBM_FILE_FORMAT = 'CSV';
SET IBM_RESOURCE = 'IFASMF.SYS01.PERF';
// DD PATH='/etc/cdpConfig/SYS01.sde',
// PATHDISP=(KEEP),RECFM=V,LRECL=255,FILEDATA=RECORD
// DD *
COLLECT SMF FROM &IBM_RESOURCE
EVERY &IBM_SDE_INTERVAL;
/*
//HBOOUT DD SYSOUT=*
//HBODUMP DD SYSOUT=*
```

If **ZIIPOFFLOAD=YES** is specified but no zIIPs are online when the System Data Engine address space is started, no work is offloaded to zIIPs.

The Resource Measurement Facility (RMF) provides information on zIIP usage to help you identify when to add more zIIPs to the logical partition. Also, fields in SMF Type 30 records allow you to know how much time is spent on zIIPs, and how much time is spent on running zIIP eligible work on GCPs. A high CPU time consumed on GCPs by work that is eligible for a zIIP indicates high contention on the zIIP processors. In this case, you must add more zIIPs to the logical partition, or specify **ZIIPOFFLOAD=NO** to disable the zIIP offloading function of the System Data Engine.

Verifying the search order for the TCP/IP resolver configuration file

Before you start IBM Z Common Data Provider, verify that the z/OS environment is set up correctly so that IBM Z Common Data Provider can access the TCP/IP resolver configuration file.

About this task

The IBM Z Common Data Provider Data Streamer and Log Forwarder are z/OS UNIX System Services programs. They use TCP/IP functions that require access to the TCP/IP resolver configuration file. This access is provided by using a resolver search order. The resolver search order for z/OS UNIX System Services programs is documented in the topic about resolver configuration files in the *z/OS Communications Server: IP Configuration Guide*.

The following list summarizes the resolver search order:

1. GLOBALTCPIPDATA statement
2. The `RESOLVER_CONFIG` environment variable in the Data Streamer procedure or job and in the Log Forwarder properties (which are part of the global properties that you can define for data streams in a policy).
 - Tip:** For information about this environment variable configuration, see the following topics:
 - “Configuring the Data Streamer” on page 118
 - “Log Forwarder properties configuration” on page 154
3. `/etc/resolv.conf` file
4. SYSTCPD DD statement in the Log Forwarder and Data Streamer started tasks
5. `userid.TCPIP.DATA`, where `userid` is the user ID that is associated with the Log Forwarder and Data Streamer started tasks
6. `SYS1.TCPPARMS(TCPDATA)`
7. `DEFAULTTCPIPDATA`
8. `TCPIP.TCPIP.DATA`

Procedure

Verify that the resolver configuration file is available to the Data Streamer and the Log Forwarder by using one of the search order mechanisms.

Configuration reference for managing policies

This reference contains information that is useful in creating and updating policies. It includes information about the global properties that you can define for a policy, the icons on each node in a policy, the correlation between SMF record types and the associated SMF data stream names, and the configuration values that you can update for each data stream, transform, or subscriber.







Reference information	Area of Configuration Tool where the relevant configuration is done
“Global properties that you can define for all data streams in a policy” on page 153	Policy Profile Edit window in the Global Properties section
“Icons on each node in a policy” on page 180	Policy Profile Edit window in the graph
“SMF data stream reference” on page 158	Window that is shown when you click the Add Data Stream icon  DATA STREAM in the Policy Profile Edit window
“SMF_110_1_KPI data stream content” on page 176	Window that is shown when you click the Add Data Stream icon  DATA STREAM in the Policy Profile Edit window
“Data stream configuration for data gathered by Log Forwarder” on page 181	Window that is shown when you click the Configure icon  on a data stream node for data that is gathered by the Log Forwarder
“Data stream configuration for data gathered by System Data Engine” on page 228	Window that is shown when you click the Configure icon  on a data stream node for data that is gathered by the System Data Engine

Table 20. Configuration reference information for managing policies (continued)

Reference information	Area of Configuration Tool where the relevant configuration is done
“Transform configuration” on page 229	Window that is shown when you click the Transform icon  on a data stream or transform node
“Subscriber configuration” on page 235	Window that is shown when you click a Subscribe icon  on a data stream or transform node

Global properties that you can define for all data streams in a policy

When you create or edit a policy in the IBM Z Common Data Provider Configuration Tool, the buttons **SYSTEM**, **LOG FORWARDER**, **SYSTEM DATA ENGINE**, and **SCHEDULES** are shown in the Global Properties section of the **Policy Profile Edit** window. You can use these buttons to define properties that apply to all data streams (or all data streams from a certain type of data gatherer) in the policy.

Tips:

- The **LOG FORWARDER** button is available only after you define a data stream for z/OS log data, which is gathered by the Log Forwarder. Use this button to set, or verify, the Log Forwarder properties.
- The **SYSTEM DATA ENGINE** button is available only after you define a data stream for SMF or IMS data, which is gathered by the System Data Engine. Use this button to set, or verify, the System Data Engine properties.

SYSTEM properties: Defining alternative host names for source systems

When you create or edit a policy in the IBM Z Common Data Provider Configuration Tool, you can use the **SYSTEM** button to define alternative host names for the source systems from which IBM Z Common Data Provider collects data. The Data Streamer then uses these alternative host names in the associated data records that it sends to subscribers.

About this task

Example of using alternative host names

If the host name for a source system is `abc.host.com`, and you define an alternative host name of `def.host.com` for this source system, the Data Streamer changes the host name in the associated data records to `def.host.com` before it sends the records to the subscriber.

Reasons why you might want to define alternative host names

The host name for a source system can sometimes change. If you know, for example, that a source system interchangeably uses `ghi.host.com` and `jkl.host.com` as its host name, you can define `ghi.host.com` to be the alternative host name for `jkl.host.com`. Then, the host name is always reported as `ghi.host.com` so that the data at the target destination can easily be correlated to the correct source system.

You might also have other reasons for defining alternative host names.

Procedure

To define alternative host names, complete the following steps:

1. In the Global Properties section of the **Policy Profile Edit** window, click **SYSTEM**.
2. Click **ADD SYSTEM**.
3. In the **System name** field, type the current host name.
4. In the **Remapped host name** field, type the alternative host name.

5. Repeat steps “2” on page 153 to “4” on page 153 for each source system for which you want to define alternative host names.
6. Click **OK**.

LOG FORWARDER properties: Defining your Log Forwarder environment

In the IBM Z Common Data Provider Configuration Tool, after you define a data stream for z/OS log data (which is gathered by the Log Forwarder), use the **LOG FORWARDER** button to set the configuration values for your Log Forwarder environment.

About this task

For more information about the Log Forwarder configuration values, see [“Log Forwarder properties configuration” on page 154](#).

For more information about configuring the Log Forwarder, see [“Configuring the Log Forwarder” on page 124](#).

Procedure

To define your Log Forwarder environment, complete the following steps:

1. In the Global Properties section of the **Policy Profile Edit** window, click **LOG FORWARDER**.
2. In the "**Configure Log Forwarder properties**" window, update the configuration values for your environment, and click **OK**.

Log Forwarder properties configuration

This reference lists the configuration values that you can update in the "**Configure Log Forwarder properties**" window of the IBM Z Common Data Provider Configuration Tool.

Port

The port on which the Data Streamer listens for data from the Log Forwarder.

Tip: For more information about the Data Streamer port, see [“Configuring the Data Streamer” on page 118](#).

Discovery Interval

In the process of streaming data, the number of minutes that the Log Forwarder waits before it checks for a new log file in the data stream. This value applies to all data streams from the Log Forwarder, although it can be overridden on some individual streams.

The value must be an integer in the range 0 - 5. A value of 0 specifies that the Log Forwarder only checks for a new log file once when the data gatherer is started. The default value is 1.

Pattern Discovery Interval

In the process of streaming data, the number of minutes that the Log Forwarder waits before it checks for new data sources that match wildcard specifications. This value applies to all data streams from the Log Forwarder, although it can be overridden on some individual streams.

The value must be an integer in the range 0 - 60. The default value is 1.

JRELIB

The fully qualified path to a set of native libraries that are required by the Java Runtime Environment (31-bit). The default value is `/usr/lib/java_runtime`.

JRELIB64

The fully qualified path to a set of native libraries that are required by the Java Runtime Environment (64-bit). The default value is `/usr/lib/java_runtime64`.

REGJAR

The fully qualified path to the `ifaedjreg.jar` file, which provides access to z/OS product registration services. The default value is `/usr/include/java_classes/ifaedjreg.jar`.

RESOLVER_CONFIG

The TCP/IP resolver configuration file that the Log Forwarder must use.

The Log Forwarder is a z/OS UNIX System Services program. It uses TCP/IP functions that require access to the TCP/IP resolver configuration file.

For more information, see [“Verifying the search order for the TCP/IP resolver configuration file” on page 151.](#)

TZ

The time zone offset for the Log Forwarder and all data streams from the Log Forwarder.

ZLF_JAVA_HOME

The Java installation directory.

ZLF_HOME

The Log Forwarder installation directory.

ZLF_WORK

The Log Forwarder working directory, which contains files that are created and used during the operation of the Log Forwarder. For example, it includes files that contain information about the state of the Log Forwarder and its progress in collecting data.

Guidelines for the working directory

Use the following guidelines to help you decide which directory to use as the working directory:

- The working directory must be in a different physical location from the working directory for any other Log Forwarder instance.
- The directory must be readable and writable by the user ID that runs the Log Forwarder.
- To avoid possible conflicts, do not use a directory that is defined as the Configuration Tool working directory.

Important: Do not update, delete, or move the files in the Log Forwarder working directory.

ZLF_LOG

The directory for the `logging.properties` file.

ZLF_WAS_PLUGINS_ROOT

The IBM WebSphere Application Server installation root directory for Web Server Plug-ins. This directory contains the `com.ibm.hpel.logging.jar` file that is used to retrieve log data from High Performance Extensible Logging (HPEL).

ZLF_GATHERER

The directory for use by data gatherers from a third party organization.

Transport Affinity (environment variable `_BPXK_SETIBMOPT_TRANSPORT`)

The TCP/IP stack to which the Log Forwarder must have affinity. If no value is specified, the Log Forwarder has affinity to the default TCP/IP stack.

Important: If the LPAR has multiple TCP/IP stacks, you must specify which stack you want the Log Forwarder to use and specify the same TCP/IP stack for the Data Streamer (as instructed in [“Configuring the Data Streamer” on page 118](#)). Otherwise, the Log Forwarder might be unable to connect to the Data Streamer.

SYSTEM DATA ENGINE properties: Defining your System Data Engine environment

In the IBM Z Common Data Provider Configuration Tool, after you define a data stream for SMF data (which is gathered by the System Data Engine), use the **SYSTEM DATA ENGINE** button to set the configuration values for your System Data Engine environment.

About this task

For more information about configuring the System Data Engine, see [“Configuring the System Data Engine” on page 133.](#)

Procedure

1. In the Global Properties section of the **Policy Profile Edit** window, click **SYSTEM DATA ENGINE**.
2. In the "**Configure System Data Engine properties**" window, update the following configuration values for your environment, and click **OK**.

USER Concatenation

This value is relevant only if you are using custom System Data Engine data streams. It is required as part of enabling the Configuration Tool to support custom System Data Engine data streams. For more information, [“Creating a System Data Engine data stream definition” on page 46](#).

The value must be the name of the USERDEFS data set that contains the custom System Data Engine definitions. This data set is also referenced in the `concat.s.json` file, which is in the working directory for the IBM Z Common Data Provider Configuration Tool.

CDP Concatenation

This value must be the name of the SHBODEFS data set that is installed with IBM Z Common Data Provider in your environment. This data set is also referenced in the `concat.s.json` file, which is in the working directory for the IBM Z Common Data Provider Configuration Tool.

IZOA Concatenation

This value is relevant only if you are using IBM Z Operations Analytics. It is required as part of enabling the Configuration Tool to support SMF data that is destined for IBM Z Operations Analytics. For more information, see the [IBM Z Operations Analytics documentation](#).

The value must be the name of the SGLASAMP data set that is installed with IBM Z Operations Analytics in your environment. This data set is also referenced in the `concat.s.json` file, which is in the working directory for the IBM Z Common Data Provider Configuration Tool.

Tip: The `concat.s.json` file is created in the Configuration Tool working directory when you save the first policy that you create. By default, any new policies that are created use the same `concat.s.json` file.

SCHEDULES properties: Defining time intervals for filtering operational data

When you create or edit a policy in the IBM Z Common Data Provider Configuration Tool, you can use the **SCHEDULES** button to define time intervals for filtering the operational data that IBM Z Common Data Provider collects. For example, you might want to define time intervals to filter data according to the expected peak demand for your applications.

About this task

To define a time interval for filtering the data, you must first define a schedule, which can contain one or more time interval definitions. You can define multiple schedules.

Important: The schedules that you define are used in filtering data streams only if, when you configure the data streams, you select the **Time Filter** transform in the "**Transform data stream**" window. For more information about transform types, see [“Transform configuration” on page 229](#).

Procedure

- In the Global Properties section of the **Policy Profile Edit** window, click **SCHEDULES**, and complete one or more of the following actions, depending on what you want to do.
Any previously defined schedules are shown in the **Schedule** list.

Action	Instruction
Create or edit a schedule	To edit a schedule, select it from the Schedule list. To define a new time interval in a schedule, click ADD , and complete the following steps:

Action	Instruction
	<ol style="list-style-type: none"> 1. In the Edit name field, type the name for the schedule that you want to contain this time interval. 2. To set the time interval for this schedule, either type the time information in the From and to fields, or use the slider to adjust the time. 3. To add another time interval for this schedule, click ADD WINDOW, and repeat the previous step. 4. To save the schedule, click APPLY.
Delete a schedule	<p>Select the schedule from the Schedule list, and click DELETE.</p> <p>Restriction: The DELETE button is not available if a schedule is assigned to a transform for a data stream.</p>

Groups of data streams in the Configuration Tool

This reference lists and describes the data stream groups in the "**Select data stream**" window. In the window, you can expand and select data streams from these groups: Starter Sets, Z Common Data Provider, IBM Z Operations Analytics and Custom Data Streams.

Starter Sets

Starter Sets includes commonly used data streams. These data streams are categorized into various data stream units based on some z/OS components and subsystems. This group includes one subgroup: Z Common Data Provider. You can select basic sets of data streams as a unit.

Table 21. Subgroup and data streams of Starter Sets

Subgroup	Data stream unit	Description of data stream unit
Z Common Data Provider	General z/OS system monitoring	z/OS system log data, common address space work and RMF CPU activity
	Security	z/OS system log data and RACF processing
	CICS	CICS MSGUSR and EYULOG log information and information about key performance indicators (KPIs) for CICS Transaction Server for z/OS monitoring
	WebSphere Application Server	WAS Request Activity and Data from the SYSOUT and SYSPRINT job log
	Db2	Db2 Statistics - system services, database services, Dynamic ZPARMS, Buffer Manager Group Buffer Pool, System Storage Usage and aggregated accounting statistics
	IMS	IMS and IMS CPI-CI program start and termination
	DCOLLECT Data	Data set, Volume and SMS-Configuration information

Z Common Data Provider

This group includes all the data streams that IBM Z Common Data Provider supports.

IBM Z Operations Analytics

This group includes all the data streams that IBM Z Operations Analytics supports.

Custom Data Streams

This group includes the data streams that are customized through System Data Engine language.

SMF data stream reference


For each System Management Facilities (SMF) record type, this reference lists the name of the data stream that IBM Z Common Data Provider uses to collect the data and includes a brief description of the data stream content. In the Configuration Tool, these SMF data stream names are shown in the **"Select data stream"** window, which opens when you click the **Add Data Stream** icon  **DATA STREAM** in the **Policy Profile Edit** window.

Table 22 on page 159 provides the following information:

Column 1

The SMF record type

Column 2

The subtype of the SMF record. In either of the following situations, no subtype is indicated:

- The stream applies to all subtypes of the respective SMF record.
- The respective SMF record has no subtypes.

Column 3

The name of the data stream to which the SMF data is written

Column 4

A brief description of the content of the SMF data stream

Table 22. Data stream names that IBM Z Common Data Provider uses to collect SMF data			
Type	Subtype	Data stream name	Description of data stream content
0		SMF_000	IPL
2		SMF_002	Dump header
3		SMF_003	Dump trailer
4		SMF_004	Step termination. Note: consider using SMF 30 since this data is included in SMF 30
		SMF_004_DEVICE	Step termination device data. Note: consider using SMF 30 since this data is included in SMF 30
5		SMF_005	Job termination. Note: consider using SMF 30 since this data is included in SMF 30
		SMF_005_ACCOUNTING	Job termination accounting data. Note: consider using SMF 30 since this data is included in SMF 30
6		SMF_006	JES2/JES3/PSF/External writer
7		SMF_007	SMF data lost
8		SMF_008	I/O configuration at IPL
		SMF_008_ONLINE	Data for online devices at IPL
9		SMF_009	VARY device ONLINE
		SMF_009_DEVICE	Data for each device varied online
10		SMF_010	Allocation recovery
		SMF_010_DEVICE	Data for each device made available
11		SMF_011	VARY device OFFLINE
		SMF_011_DEVICE	Data for each device varied offline
14		SMF_014	INPUT or RDBACK data set activity
		SMF_014_UCB	UCB information
15		SMF_015	OUTPUT, UPDAT, INOUT, or OUTIN data set activity
		SMF_015_UCB	UCB information
16		SMF_016	DFSORT statistics
		SMF_016_SORTIN	SORTIN data set information
		SMF_016_SORTOUT	SORTOUT data set information
		SMF_016_OUTFIL	OUTFIL data set information
17		SMF_017	Scratch data set status
		SMF_017_VOLUMEXT	Volume information
18		SMF_018	Rename data set status
		SMF_018_VOLUMEXT	Volume information
19		SMF_019	Direct access volume
20		SMF_020	Job initiation. Note: consider using SMF 30 since this data is included in SMF 30
		SMF_020_ACCOUNTING	Job accounting information. Note: consider using SMF 30 since this data is included in SMF 30
21		SMF_021	Error statistics by volume
22		SMF_022	Configuration
23		SMF_023	SMF status
24		SMF_024	JES2 spool offload
		SMF_024_PRODUCT	JES2 product information
		SMF_024_SPOOLOFF	Statistics for spool offload devices
		SMF_024_JOBSEL	Job selection criteria
		SMF_024_SYSSSEL	SYSOUT selection criteria
		SMF_024_SYSAFF	System affinity information

Table 22. Data stream names that IBM Z Common Data Provider uses to collect SMF data (continued)

Type	Subtype	Data stream name	Description of data stream content
25		SMF_025	JES3 device allocation
26		SMF_026	JES2/JES3 job purge
30		SMF_030	Common address space work. If you have configured this data stream, you can use this data stream. Otherwise, SMF_030_V2 is recommended.
		SMF_030_V2	Common address space work (more fields than SMF_030). Some new fields were added to this data stream, and the descriptions of some fields were updated. You are recommended to use this data stream.
		SMF_030_EXCP	I/O information for a specific DD Name/Device address pair for the address space
		SMF_030_ACCOUNTING	User accounting information for the address space
		SMF_030_OPENMVS	z/OS UNIX process information
		SMF_030_ARM	Information related to a batch job or started task that registers as an element of automatic restart management
		SMF_030_USAGE	Product ID information and usage data
		SMF_030_ENCLAVE	Remote system data for each system that executed work under a multisystem enclave
		SMF_030_COUNTER	Hardware Instrumentation Services (HIS) counters
		SMF_030_ZEDC	zEDC usage statistics section
32		SMF_032	TSO user work accounting
		SMF_032_IDENTIF	Identification section
		SMF_032_TSOCOMMAND	TSO/E command segment
33	1	SMF_033	APPC/MVS TP accounting
		SMF_033_ACS	TP usage accounting
		SMF_033_TPS	TP usage scheduler data
34		SMF_034	TS-step termination. Note: consider using SMF 30 since this data is included in SMF 30
		SMF_034_DEVICE	EXCP section. Note: consider using SMF 30 since this data is included in SMF 30
35		SMF_035	LOGOFF. Note: consider using SMF 30 since this data is included in SMF 30
		SMF_035_ACCOUNT	Accounting information. Note: consider using SMF 30 since this data is included in SMF 30
36		SMF_036	Integrated Catalog Facility Catalog
37		SMF_037_HW	NetView Hardware Monitor
		SMF_037_ETHERNET	Ethernet LAN data
		SMF_037_TEXT	Text message data
38	1	SMF_038_1	NetView Command Authorization Table
	2	SMF_038_2	NetView Task Resource Utilization Data
	3	SMF_038_3	NetView Span Authorization Table
	4	SMF_038_4	NetView Command Statistics
		SMF_038_4_CMDS	NetView Command Statistics Command Data Section
39	1 - 7	SMF_039_1_TO_7	NetView Session Monitor
	8	SMF_039_8	NetView Session Monitor
40		SMF_040	Dynamic DD. Note: consider using SMF 30 since this data is included in SMF 30
		SMF_040_DEVICE	EXCP section. Note: consider using SMF 30 since this data is included in SMF 30
41	1 - 3	SMF_041	Data-in-virtual Access/Unaccess
		SMF_041_VLF	VLF statistics

Table 22. Data stream names that IBM Z Common Data Provider uses to collect SMF data (continued)

Type	Subtype	Data stream name	Description of data stream content
42	1	SMF_042_1	DFSMS - BMF performance statistics
		SMF_042_STOR_CLASS	Storage class summary
	2	SMF_042_2	DFSMS - DFP cache control unit statistics
		SMF_042_UNIT_CACHE	Control unit cache section
		SMF_042_VOL_STATUS	Volume status section
	3	SMF_042_3	DFSMS - DFP SMS configuration statistics
		SMF_042_EVNT_AUDIT	Event audit section
	4	SMF_042_4	DFSMS - DFP concurrent copy session statistics
		SMF_042_CONC_COPY	Concurrent copy session section
		SMF_042_EAVCC_VOL	EAV concurrent copy volume section
	5	SMF_042_5	DFP Storage Class statistics
		SMF_042_STOR_RESP	Storage class response time
	6	SMF_042_6	DFP Data Set statistics
		SMF_042_6_X	DFP Data Set statistics from record procedure
11	SMF_042_11	DFP Extended Remote Copy (XRC) Session Statistics	
14	SMF_042_14	ADSM Server statistics	
43	2	SMF_043_JES2	JES2 start
	5	SMF_043_JES3	JES3 start
45	2	SMF_045_JES2	JES2 withdrawal
	5	SMF_045_JES3	JES3 stop
47	2	SMF_047_JES2	JES2 SIGNON/start line (BSC only)
	5	SMF_047_JES3	JES3 SIGNON/start line/LOGON
48	2	SMF_048_JES2	JES2 SIGNOFF/stop line (BSC only)
	5	SMF_048_JES3	JES3 SIGNOFF/stop line/LOGOFF
49	2	SMF_049_JES2	JES2 integrity (BSC only)
	5	SMF_049_JES3	JES3 integrity
50		SMF_050	VTAM® tuning statistics for SNA controllers
		SMF_050_CTC	VTAM tuning statistics for CTC adapters
		SMF_050_MPC_GROUP	VTAM tuning statistics for MPC groups
		SMF_050_MPC_CHL	VTAM tuning statistics for MPC subchannels
		SMF_050_TCP	VTAM tuning statistics for TCP connections
		SMF_050_ROCE	VTAM tuning statistics for RoCE connections
52		SMF_052	JES2 LOGON/start line (SNA only)
53		SMF_053	JES2 LOGOFF/stop line (SNA only)
54		SMF_054	JES2 integrity (SNA only)
55		SMF_055	JES2 network SIGNON
56		SMF_056	JES2 network integrity
57	2	SMF_057_JES2	JES2 network SYSOUT transmission
	5	SMF_057_JES3	JES3 networking transmission
58		SMF_058	JES2 network SIGNOFF
59		SMF_059	MVS/BDT file-to-file transmission
60		SMF_060	VSAM volume data set updated
61		SMF_061	Integrated Catalog Facility Define Activity
62		SMF_062	VSAM component or cluster opened
		SMF_062_ONLINE	Online volume information
64		SMF_064	VSAM component or cluster status
		SMF_064_EXTENT	Extent information section

Table 22. Data stream names that IBM Z Common Data Provider uses to collect SMF data (continued)

Type	Subtype	Data stream name	Description of data stream content
65		SMF_065	Integrated Catalog Facility Delete Activity
66		SMF_066	Integrated Catalog Facility Alter Activity
70	1	SMF_070	RMF CPU activity
		SMF_070_CPU_V2	CPU data section Note: This data stream supersedes SMF_070_CPU.
		SMF_070_CPU	CPU data section. Note: This data stream is superseded by SMF_070_CPU_V2.
		SMF_070_BCT	PR/SM partition data section
		SMF_070_BPD_V2	PR/SM logical processor data section Note: This data stream supersedes SMF_070_BPD.
		SMF_070_BPD	PR/SM logical processor data section. Note: This data stream is superseded by SMF_070_BPD_V2.
		SMF_070_INS	CPU identification section
		SMF_070_LCD	Logical core data section
		SMF_070_TRG	Tenant Resource Group data section
	2	SMF_070_2	RMF Cryptographic Hardware Activity
		SMF_070_PCICA	Cryptographic Accelerator Data Section.
		SMF_070_PCICC	Cryptographic CCA coprocessor data section
		SMF_070_PKCS11	Cryptographic PKCS11 coprocessor data section
71	1	SMF_071	RMF paging activity
		SMF_071_SWAP	Swap placement section

Table 22. Data stream names that IBM Z Common Data Provider uses to collect SMF data (continued)

Type	Subtype	Data stream name	Description of data stream content
72	1	SMF_072_1	RMF workload activity
		SMF_072_PGP	Performance Group Period data section
	2	SMF_072_2	RMF storage data
		SMF_072_2_DATA	Performance Group data section
		SMF_072_2_SWAP_RSN	Swap reason data section
	3	SMF_072_3	RMF goalmode workload activity
		SMF_072_SSS	Service class served data section
		SMF_072_SCS	Service/Report Class period data section
		SMF_072_WRS	Work Manager/Resource Manager state section
		SMF_072_DNS	Resource delay type names section
	4	SMF_072_4	RMF Goalmode delay and storage frame data
		SMF_072_4_DATA	Service class period data section
		SMF_072_4_SWAP_RSN	Swap reason data section
	5	SMF_072_5	RMF system suspend locks and GRS data
		SMF_072_CMS_LOCK	CMS lock data section
		SMF_072_ENQ_LOCK	CMS Enqueue/Dequeue lock data section
		SMF_072_LATCH_LOCK	CMS latch lock data section
		SMF_072_SMF_LOCK	CMS SMF lock data section
		SMF_072_LOCK_TYPE	Local lock data section
		SMF_072_LOCK_OWNER	CML lock owner data section
SMF_072_LOCK_RQSTR		CML lock requestor data section	
SMF_072_LATCH_CRTR		Latch creator data section	
SMF_072_OFFSET_LR		Latch requestor data section	
SMF_072_GRS_ENQ		GRS Enqueue step data section	
SMF_072_ENQ_SYS		GRS Enqueue system data section	
SMF_072_ENQ_SYSS		GRS Enqueue systems data section	
SMF_072_GRS_QSCAN	GRS QScan statistics data section		
73	1	SMF_073	RMF channel path activity
		SMF_073_CHAN_PATH	Channel path data section
		SMF_073_EXT_CHAN	Extended channel path data section

Table 22. Data stream names that IBM Z Common Data Provider uses to collect SMF data (continued)

Type	Subtype	Data stream name	Description of data stream content
74	1	SMF_074_1	RMF device activity
		SMF_074_DEV_DATA	Device data section
	2	SMF_074_2	RMF XCF activity
		SMF_074_SYS_DATA	System data section
		SMF_074_PATH_DATA	Path data section
		SMF_074_MBR_DATA	Member data section
	3	SMF_074_3	RMF OPENMVS kernel activity
		SMF_074_OMVS_DATA	Control data section
	4	SMF_074_4	RMF XES/CF activity
		SMF_074_CONN_DATA	Connectivity data section
		SMF_074_STRUC_DATA	Structure data section
		SMF_074_RQST_DATA	Request data section
		SMF_074_PROC_DATA	Processor utilization data section
		SMF_074_CACHE_DATA	Cache data section
		SMF_074_REMOTE_FAC	Remote facility data section
		SMF_074_CHAN_PATH	Channel path data section
		SMF_074_SC_MEMDATA	Storage class memory data section
		SMF_074_ACFDS	Asynchronous CF Duplexing Data Section
	5	SMF_074_5	RMF Cache activity
		SMF_074_CACHE_DEV	Cache device data section
		SMF_074_XDEV	Cache device data section extension
		SMF_074_CCU_STATUS	Cache control unit status section
		SMF_074_RAID_RANK	RAID Rank/Extent Pool data section
	6	SMF_074_6	RMF Hierarchical file system activity
		SMF_074_HFS_GLOBAL	HFS global data section
		SMF_074_HFS_BUFFER	HFS global buffer section
		SMF_074_HFS	HFS file system section
	7	SMF_074_7	RMF FICON® Director Statistics
		SMF_074_FCD_GLOBAL	FCD global data section
		SMF_074_FCD_PORT	FCD port data section
		SMF_074_FCD_CONN	FCD connector data section
	8	SMF_074_8	RMF Enterprise Storage Server® (ESS) Link Statistics
		SMF_074_ESS_LINK	Link statistics section
		SMF_074_EXT_POOL	Extent pool statistics section
		SMF_074_RANK_STATS	Rank statistics section
		SMF_074_RANK_ARRAY	Rank array data section
		SMF_074_SILS_ARRAY	Synchronous I/O Link Statistics Section
	9	SMF_074_9	RMF Monitor III PCIE Statistics
		SMF_074_PCIE_FUNC	PCIE function data section
		SMF_074_DMA_00	PCIE function type data section for format x'00'
		SMF_074_DMA_01	PCIE function type data section for format x'01'
		SMF_074_DMA_02	PCIE function type data section for format x'02'
		SMF_074_DMA_03	PCIE function type data section for format x'03'
		SMF_074_DMA_04	PCIE function type data section for format x'04'
		SMF_074_HWAC	Hardware accelerator data section
		SMF_074_HWAC_COMP	Hardware accelerator compression data section
		SMF_074_SIOL	Synchronous I/O Link data section
SMF_074_SIOR		Synchronous I/O response time distribution data section	

Table 22. Data stream names that IBM Z Common Data Provider uses to collect SMF data (continued)

Type	Subtype	Data stream name	Description of data stream content
74	10	SMF_074_10	Storage Class Memory (SCM) Statistics
		SMF_074_EADM_DEV	SCM device (subchannel) information section
		SMF_074_SCMC	SCM configuration measurement section
75	1	SMF_075	RMF page/swap data set activity
		SMF_075_PAGE_SWAP	Page Data Set data section
76		SMF_076	RMF trace activity
		SMF_076_PRODUCT	RMF product section
		SMF_076_TRCCTRL	Trace control section
		SMF_076_TRCDATA	Trace data section
		SMF_076_VARDATA	Variable trace data section
77	1	SMF_077	RMF enqueue activity
		SMF_077_ENQ	Enqueue data section
78	2	SMF_078_2	RMF virtual storage activity
		SMF_078_VSPA	Virtual Storage Private Area data section
		SMF_078_VSPASS	Virtual Storage Private Area subpool section
	3	SMF_078_3	RMF I/O queuing activity for the 3090, 9021, 9121, and 9221 processors
		SMF_078_HYPERPAV	HyperPAV data section
		SMF_078_IOQDATA3	I/O Queuing data section
79		SMF_079	RMF Monitor II activity
		SMF_079_ASDDATA	ASD and ASDJ data section
		SMF_079_ARDDATA	ARD and ARDJ data section
		SMF_079_SRCSDATA	SRCS data section
		SMF_079_SPAGDATA	SPAG data section
		SMF_079_ASRMDATA	ASRM and ASRMJ data section
		SMF_079_SENQRDATA	SENQR data section
		SMF_079_SENQDATA	SENQ data section
		SMF_079_TRXDATA	TRX data section
		SMF_079_DEVICEDATA	Device data section
		SMF_079_DDMNDATA	DDMN data section
		SMF_079_PGSPDATA	PGSP control section
		SMF_079_PGSP_DATA	PGSP data set section
		SMF_079_CHANNELCTL	Channel path control section
		SMF_079_IOCONFIGQ	I/O Queuing global section
		SMF_079_IOQ_DATAS	I/O Queuing data section
		SMF_079_LONG_LOCK	IMS long lock data section
		80 (for RACF)	
SMF_080_RELOCATE	RACF relocate section		
SMF_080_XRELOCATE	RACF extended relocate section		
80 (for CA Top Secret)		SMF_080_CA_16	CA Top Secret security-related activity
		SMF_080_CA_REL	CA Top Secret security-related audit and logging information
81		SMF_081	RACF initialization
		SMF_081_RELOCATE	RACF relocate section

Table 22. Data stream names that IBM Z Common Data Provider uses to collect SMF data (continued)

Type	Subtype	Data stream name	Description of data stream content	
82	1	SMF_082_1	Initialization/Options Refresh	
	7	SMF_082_7	Operational Key Entry	
	8	SMF_082_8	CKDS Refresh	
	9	SMF_082_9	CKDS Update	
	13	SMF_082_13	PKDS Update	
	14	SMF_082_14	Master Key Entry	
	15	SMF_082_15	Retained Key Create/Delete	
	16	SMF_082_16	TKE Command Request/Reply	
	18	SMF_082_18	Cryptographic Coprocessor Configuration	
	19	SMF_082_19	PCIXCC Timing	
	20	SMF_082_20	Cryptographic Coprocessor Times	
	21	SMF_082_21	Sysplex Group Change	
	22	SMF_082_22	Trusted Block Processing	
	23	SMF_082_23	TKDS Update	
	24	SMF_082_24	Duplicate Tokens Found	
	24	SMF_082_24_LAB	Duplicate Tokens Found	
	25	SMF_082_25	Key Token Authorization Checking	
	25	SMF_082_25_LABTP	Key Token Authorization Checking	
	26	SMF_082_26	PKDS Refresh	
	27	SMF_082_27	PKA Key Management Extensions	
	27	SMF_082_27_PKAL	PKA Key Management Extensions	
	27	SMF_082_27_SYML	PKA Key Management Extensions	
	28	SMF_082_28	High Performance Encrypted Key	
	28	SMF_082_28_SYML	High Performance Encrypted Key	
	29	SMF_082_29	TKE Workstation Audit	
	30	SMF_082_30	KDS Archived/Inactive Checking	
	31	SMF_082_31	Cryptographic Usage Statistics	
	31	SMF_082_31_TRPL	Cryptographic Usage Statistics	
	40	SMF_082_40	CCA Symmetric Key Lifecycle Event	
	41	SMF_082_41	CCA Asymmetric Key Lifecycle Event	
	42	SMF_082_42	PKCS#11 Object Lifecycle Event	
	43	SMF_082_43	Regional Cryptographic Server Configuration	
	44	SMF_082_44	CCA Symmetric Key Usage Event	
	45	SMF_082_45	CCA Asymmetric key Usage Event	
	46	SMF_082_46	PKCS#11 Key Usage Event	
	47	SMF_082_47	PKCS#11 No Key Usage Event	
	48	SMF_082_48	Compliance Warning Event	
	83		SMF_083	RACF audit record for data sets

Table 22. Data stream names that IBM Z Common Data Provider uses to collect SMF data (continued)

Type	Subtype	Data stream name	Description of data stream content
84	1	SMF_084_1	JES3 monitoring facility (JMF) FCT (Function Control Table) analysis
	2	SMF_084_2	JES3 monitoring facility (JMF) FCT summary and highlight
		SMF_084_JES3_WAIT	JES3 wait analysis section
	3	SMF_084_3	JES3 monitoring facility (JMF) Spool data management
	4	SMF_084_4	JES3 monitoring facility (JMF) Resqueue cellpool, JCT, and control block utilization
	5	SMF_084_5	JES3 monitoring facility (JMF) Job analysis
	6	SMF_084_6	JES3 monitoring facility (JMF) JES3 hot spot analysis
	7	SMF_084_7	JES3 monitoring facility (JMF) JES3 internal reader DSP analysis
	8	SMF_084_8	JES3 monitoring facility (JMF) JES3 SSI response time analysis
	9	SMF_084_9	JES3 monitoring facility (JMF) JES3 SSI destination queue analysis
	10	SMF_084_10	JES3 monitoring facility (JMF) JES3 Workload Manager Analysis
21	SMF_084_21	JES2 memory and resource usage	
	SMF_084_21_MEM	JES2 memory usage section	
	SMF_084_21_RES	JES2 resource usage section	
85		SMF_085	OAM record
		SMF_085_ARRAY	Volume array section
88		SMF_088	System logger
89		SMF_089	Product Usage Data
		SMF_089_USAGE_DATA	Usage data section
		SMF_089_PROD_ISECT	Product intersection data section
		SMF_089_STATE_DATA	State data section
90		SMF_090	System status
		SMF_090_SMFDATASET	SMF data set section
		SMF_090_SUBSYSTEM	Subsystem record section
		SMF_090_SUBPARM	Subsystem parameter section
92		SMF_092	OpenMvs File System Activity
94	1	SMF_094	34xx tape library data server statistics
	2	SMF_094_2	Volume Pool Statistics
98	1	SMF_098_1	High-frequency throughput statistics
		SMF_098_LK_SD	Spin lock detail section
		SMF_098_LK_SS	Suspend lock summary section
		SMF_098_SS_LD	Suspend lock detail section
		SMF_098_LC_CM	Local and CML lock detail section
		SMF_098_WK_UN	Work unit section
	1024	SMF_098_1024	CICS WIC Main record
		SMF_098_1024_B1	CICS WIC Aggregate bucket1 section
		SMF_098_1024_B2	CICS WIC Aggregate bucket2 section
		SMF_098_1024_IX	CICS WIC Exceptional job index section
		SMF_098_1024_JOB	CICS WIC Exceptional job section
	1025	SMF_098_1025	IMS WIC Main record
		SMF_098_1025_B1	IMS WIC Aggregate bucket1 section
		SMF_098_1025_B2	IMS WIC Aggregate bucket2 section
		SMF_098_1025_IX	IMS WIC Exceptional job index section
		SMF_098_1025_JOB	IMS WIC Exceptional job section

Table 22. Data stream names that IBM Z Common Data Provider uses to collect SMF data (continued)

Type	Subtype	Data stream name	Description of data stream content
99		SMF_099	System resource manager decisions
	1	SMF_099_REASM_INFO	Reassembly area information
		SMF_099_AAT	Trace table entry section
		SMF_099_SS	System state information section
		SMF_099_PP	System paging plot information section
		SMF_099_PT	Priority table entry section
		SMF_099_RG	Resource group entry section
		SMF99_S1_GENRES	Generic resource entry section
		SMF99_S1_SL	Software licensing information
		SMF99_S1_SLT	Software licensing table information
		SMF99_S1_ZE	ZE information section
		SMF99_S1_BP	Buffer pool section
	2	SMF99_S2_CLS	Class data section
		SMF99_S2_XMEM	Cross memory delay entry section
		SMF99_S2_SERVER	Server data entry section
		SMF99_S2_SDATA	Server sample data entry section
		SMF99_S2_QDATA	Queue server data entry section
		SMF99_S2_ASESP	Address space expanded storage access policy section
	3	SMF99_S3_CLS	Class data section
		SMF99_S3_PPRP	Period paging rate plot section
		SMF99_S3_RUA	Ready user average plot section
		SMF99_S3_SWP	Swap delay plot section
		SMF99_S3_PAS	Proportional aggregate speed plot section
		SMF99_S3_QMPLP	Queue delay plot section
		SMF99_S3_QRUAP	Queue ready user average plot section
		SMF99_S3_AINS	Active server instances plot section
		SMF99_S3_ASTR	VS plot for active server instances section
		SMF99_S3_TSTR	VS plot for total server instances section
		SMF99_S3_QSTP	Queue service time plot section
	4	SMF99_S4_IOPT	Device cluster priority table section
		SMF99_S4_IOPLOT	I/O plot information section
	5	SMF99_S5_MON	Monitored address space information
	6	SMF99_S6_PDS	Period data section
	7	SMF99_S7_PAV	PAV device section
	8	SMF99_S8_LPAR	LPAR data entry section
		SMF99_S8_PT	Priority table entry section
		SMF99_S8_IOSUB	I/O subsystems samples data section
		SMF99_S8_ICPU	LPAR CPU data for a partition in an LPAR cluster section
		SMF99_S8_SYSH	SYSH CPU plot section
	9	SMF99_S9_SUBS	Channel path data entry section
		SMF99_S9_PLOT	I/O subsystem plot section
		SMF99_S9_CHAN	Channel path data entry section
	10	SMF99_SA_CPUD	CPU data section
		SMF99_SA_PCHGO	Processor speed change (old)
		SMF99_SA_PCHGN	Processor speed change (new)
	11	SMF99_SB_DATA	Capacity group data section
		SMF99_SB_CECS	CEC service data section

Table 22. Data stream names that IBM Z Common Data Provider uses to collect SMF data (continued)

Type	Subtype	Data stream name	Description of data stream content
100	0	SMF_100_0	Db2 statistics, system services
		SMF_100_ADDR_SPACE	Address space data section
		SMF_100_DEST	Instrumentation destination data section
		SMF_100_INST	Instrumentation data section
		SMF_100_LATCH_MGR	Latch manager data section
		SMF_100_STRGE_MGR9	Storage manager data section (Db2 V9 and below)
		SMF_100_STRGE_MGR	Storage manager data section (Db2 V10 and above)
		SMF_100_DDF9	Distributed data facility section (Db2 V9 and below)
		SMF_100_DDF	Distributed data facility section (Db2 V10 and above)
	1	SMF_100_1	Db2 statistics - database services
		SMF_100_BIND	Bind data section (DSNDQTST)
		SMF_100_BUFF_MGR9	Buffer manager data section (DSNDQBST - Db2 V9 and below)
		SMF_100_BUFF_MGR	Buffer manager data section (DSNDQBST - Db2 V10 and above)
		SMF_100_DATA_MGR9	Data manager data section (DSNDQIST - Db2 V9 and below)
		SMF_100_DATA_MGR	Data manager data section (DSNDQIST - Db2 V10 and above)
		SMF_100_BUFF_POOL9	Buffer manager group buffer pool (Db2 V9 and below)
		SMF_100_BUFF_POOL	Buffer manager group buffer pool (Db2 V10 and above)
		SMF_100_SERV_CNTL	Service controller locking statistics
		SMF_100_IDAA_DATA	IDAA data section
		SMF_100_SIMUL_BP	Simulated Buffer Pool section
		2	SMF_100_2
	3	SMF_100_3	Db2 statistics - Buffer Manager Group Buffer Pool
		SMF_100_3BUFF_POOL	Buffer manager group buffer pool
	4	SMF_100_4	Db2 System Storage [®] Usage
		SMF_100_DB2_STRGE9	Db2 system storage usage (Db2 V9 and below)
		SMF_100_DB2_STRGE	Db2 system storage usage (Db2 V10 and above)
		SMF_100_STOR_THRD	Thread information (QW02252)
		SMF_100_STOR_CMN	Shared and common storage summary (QW02253)
		SMF_100_STOR_STMT	Statement cache and shareable statement detail (QW02254)
		SMF_100_STOR_POOL	Pool details (QW02255)
		SMF_100_STOR_IRLM	IRLM storage information (QW02256)
	5	SMF_100_5	DB2 aggregated accounting statistics
		SMF_100_QW0369_2	Connection types(QW0369_2)
		SMF_100_INSTRMET	Instrumentation data
		SMF_100_CLASSOVF	Class 3 overflow data

Table 22. Data stream names that IBM Z Common Data Provider uses to collect SMF data (continued)

Type	Subtype	Data stream name	Description of data stream content	
101	0	SMF_101	Db2 accounting	
		SMF_101_BUFFER_31	Buffer manager accounting block (DSNDQBAC)	
		SMF_101_DIST9	Distributed data facility statistics (DSNDQLAC - Db2 V9 and below)	
		SMF_101_DIST	Distributed data facility statistics (DSNDQLAC - Db2 V10 and above)	
		SMF_101_QMDA	Distributed QMD accounting data (DSNDQMDA)	
		SMF_101_IFI	Distributed IFI accounting data (DSNDQIFA)	
		SMF_101_PACKAGE	Package accounting data (DSNDQPAC)	
		SMF_101_QWAR	Rollup accounting correlation block (DSNDQWAR)	
		SMF_101_BUFFER_MGR	Buffer manager group buffer pool accounting information (DSNDQBGA)	
		SMF_101_GLBL_LOCK	Service controller global locking accounting block (DSNDQTGA)	
		SMF_101_DATA_SHARE	Data sharing accounting data (DSNDQWDA)	
		SMF_101_IDAA_ACCT	Accelerator services accounting block (DSNDQ8AC)	
		1	SMF_101_1	Db2 accounting IFCID 239
	SMF_101_1_PACKAGE		Package accounting data (DSNDQPAC)	
	SMF_101_SQL_ACC		SQL accounting data (DSNDQXPK)	
	SMF_101_BUFMGR_ACC		Buffer manager accounting block (DSNDQBAC)	
	SMF_101_LOCK_ACC		Lock manager accounting block (DSNDQTXA)	
	102		SMF_102	Db2 system initialization parameters
			SMF_102_SYS_PARM	System initialization parameters (DSNDQWPZ)
SMF_102_INI_PARM			Log initialization parameters (DSNDQWPZ)	
SMF_102_ARCH_PARM			Archive initialization parameters (DSNDQWPZ)	
SMF_102_SYS_PARM8			System parameters (DSNDQWPZ - Db2 V8)	
SMF_102_SYS_PARM9			System parameters (DSNDQWPZ - Db2 V9)	
SMF_102_SYS_PARM10			System parameters (DSNDQWPZ - Db2 V10)	
SMF_102_SYS_PARM11			System parameters (DSNDQWPZ - Db2 V11)	
SMF_102_SYS_PARM12			System parameters (DSNDQWPZ - Db2 V12)	
SMF_102_DDF_START			DDF start control information (DSNDQWPZ)	
SMF_102_DATA_SHARE			Group initialization parameters for data sharing (DSNDQWPZ)	
SMF_102_DSNHDECP			DSNHDECP parameters (DSNDQWPZ)	
103			1	INT_103_01
	2	INT_103_02	Internet Connection Secure Server performance record	
104		SMF_104	RMF Distributed Platform Performance Data	
		SMF_104_METRICS	Metric section	

Table 22. Data stream names that IBM Z Common Data Provider uses to collect SMF data (continued)

Type	Subtype	Data stream name	Description of data stream content
110	0	SMF_110_0	CICS journaling record
	1	SMF_110_1_KPI	Information about key performance indicators (KPIs) for CICS Transaction Server for z/OS monitoring
		SMF_110_1	CICS Transaction Server for z/OS monitoring data
		SMF_110_1_FIELD	Field connectors
		SMF_110_1_DICT	Dictionary data
		SMF_110_1_5	CICS Monitoring data
		SMF_110_1_6	CICS Monitoring data
		SMF_110_E	Monitoring exception data
	2	SMF_110_2	CICS statistics
		SMF_110_2_10	CICS statistics - Transaction manager (Global)
		SMF_110_2_11	CICS statistics - Transaction manager (Transaction)
		SMF_110_2_12	CICS statistics - Transaction manager (Transaction class)
		SMF_110_2_16	CICS statistics - FEPI Pool Statistics
		SMF_110_2_17	CICS statistics - FEPI Connection Statistics
		SMF_110_2_18	CICS statistics - Target Statistics
		SMF_110_2_19	CICS statistics - Storage manager domain subpool
		SMF_110_2_20	CICS statistics - Storage manager task subpool
		SMF_110_2_20_TASK	CICS statistics - Storage manager task subpool
		SMF_110_2_21	CICS statistics - VTAM global statistics
		SMF_110_2_23	CICS statistics - Autoinstall Statistics
		SMF_110_2_24	CICS statistics - Autoinstall Global Statistics
		SMF_110_2_25	CICS statistics - Loader public program
		SMF_110_2_28	CICS statistics - DBCTL (Global)
		SMF_110_2_29	CICS statistics - Storage manager DSA
		SMF_110_2_29_BDY	CICS statistics - Storage manager DSA
		SMF_110_2_30	CICS statistics - Global loader
		SMF_110_2_30_DSA	CICS statistics - Global loader DSA
		SMF_110_2_31	CICS statistics - Loader public library
		SMF_110_2_31_DSN	CICS statistics - Loader public library DSN
		SMF_110_2_32	CICS statistics - Loader private library
		SMF_110_2_32_DSN	CICS statistics - Loader private library DSN
		SMF_110_2_34	CICS statistics - Terminal Statistics
		SMF_110_2_36	CICS statistics - Loader private program
		SMF_110_2_39	CICS statistics - LSR Pool Statistics
		SMF_110_2_39_DBUF	CICS statistics - LSR Pool Statistics
		SMF_110_2_39_IBUF	CICS statistics - LSR Pool Statistics
		SMF_110_2_40	CICS statistics - LSR Pool File Statistics
		SMF_110_2_42	CICS statistics - TD Queue (Resource)
		SMF_110_2_45	CICS statistics - TD Queue (Global)
		SMF_110_2_46	CICS statistics - Security (Global)
		SMF_110_2_48	CICS statistics - TSQUEUE
		SMF_110_2_52	CICS statistics - ISC/IRC system entries
		SMF_110_2_54	CICS statistics - ISC connection
		SMF_110_2_61	CICS statistics - User (Global)
		SMF_110_2_62	CICS statistics - Dispatcher status
		SMF_110_2_62_TCB	CICS statistics - Dispatcher status
		SMF_110_2_62_POOL	CICS statistics - Dispatcher status
SMF_110_2_63		CICS statistics - Table Manager Global Statistics	
SMF_110_2_63_TABLE	CICS statistics - Table Manager Global Statistics		
SMF_110_2_64	CICS statistics - Dispatcher TCB (Global)		

Table 22. Data stream names that IBM Z Common Data Provider uses to collect SMF data (continued)

Type	Subtype	Data stream name	Description of data stream content
111		SMF_111	CICS TS for z/OS Statistics
113	1	SMF_113_1	Hardware capacity delta statistics
		SMF_113_1_SCDS	Short counters section
		SMF_113_1_LCDS	Long counters section
	2	SMF_113_2	Hardware capacity reporting and statistics
		SMF_113_2_CSS	Counter set section
		SMF_113_2_CDS	Counter data section
114	1	SMF_114_1	System Automation Tracking
115	1	MQS_115_1_V2	MQSeries® log manager statistics Note: This data stream supersedes MQS_115_1, MQS_115_QSST and MQS_115_QJST.
		MQS_115_1	MQSeries log manager statistics Note: This data stream is superseded by MQS_115_1_V2.
		MQS_115_QSST	Storage manager statistics section Note: This data stream is superseded by MQS_115_1_V2.
		MQS_115_QJST	Log manager statistics section Note: This data stream is superseded by MQS_115_1_V2.
	2	MQS_115_2_V2	MQSeries information statistics Note: This data stream supersedes MQS_115_2, MQS_115_QMST, MQS_115_QLST, MQS_115_Q5ST and MQS_115_QTST.
		MQS_115_2	MQSeries information statistics Note: This data stream is superseded by MQS_115_2_V2.
		MQS_115_QMST	Message manager statistics section Note: This data stream is superseded by MQS_115_2_V2.
		MQS_115_QPST	Buffer manager statistics section
		MQS_115_QLST	Lock manager statistics section Note: This data stream is superseded by MQS_115_2_V2.
		MQS_115_Q5ST	Db2 manager statistics section Note: This data stream is superseded by MQS_115_2_V2.
		MQS_115_QTST	Data manager statistics section Note: This data stream is superseded by MQS_115_2_V2.
		MQS_115_QESD	Shared message data sets section
	201	MQS_115_201	Data Manager Page Set
		MQS_115_QIS1	Data Manager Page Set Statistics section
	215	MQS_115_215	Buffer manager
		MQS_115_QPST215	Buffer Manager Buffer Pool Statistics section
	231	MQS_115_231_V2	Channel initiator statistics data Note: This data stream supersedes MQS_115_231 and MQS_115_QCCT.
		MQS_115_231	Channel initiator statistics data. Note: This data stream is superseded by MQS_115_231_V2.
		MQS_115_QCCT	CHINIT Control info section Note: This data stream is superseded by MQS_115_231_V2.
		MQS_115_QCT_DSP	Dispatcher tasks
		MQS_115_QCT_ADP	Adapter tasks
		MQS_115_QCT_SSL	SSL tasks
		MQS_115_QCT_DNS	DNS task

Table 22. Data stream names that IBM Z Common Data Provider uses to collect SMF data (continued)

Type	Subtype	Data stream name	Description of data stream content	
116	0	MQS_116_V2	MQSeries accounting statistics Note: This data stream supersedes MQS_116, MQS_116_QWHS and MQS_116_QMAC.	
		MQS_116	MQSeries accounting statistics. Note: This data stream is superseded by MQS_116_V2.	
		MQS_116_QWHS	Message manager section Note: This data stream is superseded by MQS_116_V2.	
		MQS_116_QMAC	Message manager accounting section Note: This data stream is superseded by MQS_116_V2.	
	1	MQS_116_1_V2	MQSeries thread and queue level accounting statistics Note: This data stream supersedes MQS_116_1 and MQS_116_WTAS.	
		MQS_116_1	MQSeries thread and queue level accounting statistics. Note: This data stream is superseded by MQS_116_1_V2.	
		MQS_116_WTAS	Task-related statistics section Note: This data stream is superseded by MQS_116_1_V2.	
		MQS_116_WQST1	Queue-level accounting statistics section	
	2	MQS_116_2_V2	MQSeries queue level accounting statistics Note: This data stream supersedes MQS_116_2 and MQS_116_QWHS2.	
		MQS_116_2	MQSeries queue level accounting statistics. Note: This data stream is superseded by MQS_116_2_V2.	
		MQS_116_QWHS2	Common MQSeries SMF Header Note: This data stream is superseded by MQS_116_2_V2.	
		MQS_116_WQST2	Queue-level accounting statistics section	
	10	MQS_116_10	Channel statistics	
		MQS_116_QCST	Channel Statistics section	
	117		SMF_117	IBM Integration Bus message flow statistics and accounting data. This data stream is superseded by SMF_117_V2.
			SMF_117_V2	IBM Integration Bus message flow statistics and accounting data. This data stream supersedes SMF_117 and contains all the fields in SMF_117 and a few new fields that are added to IBM Integration Bus V10.
		SMF_117_T1_THREAD	Thread data	
		SMF_117_T2_NODE	Node data	
		SMF_117_T2_TERM	Terminal data	
118	1 - 2	SMF_118_1	TCP/IP API calls	
	3	SMF_118_3	TCP/IP FTP Client Calls	
	4	SMF_118_4	TCP/IP TELNET Client Calls record	
	5	SMF_118_5	TCP/IP General Stats Record	
		SMF_118_5_2	TCP/IP General Stats Record	
	20 - 21	SMF_118_20	TCP/IP TELNET Server Record	
	70 - 75	SMF_118_70	TCP/IP FTP Server	

Table 22. Data stream names that IBM Z Common Data Provider uses to collect SMF data (continued)

Type	Subtype	Data stream name	Description of data stream content
119	1	SMF_119_1	TCP Connection Initiation Record
	2	SMF_119_2	TCP Connection Termination Record
	3	SMF_119_3	FTP Client Transfer Completion Record
	4	SMF_119_4	TCP/IP Profile Information Record
	5	SMF_119_5	TCP/IP Statistics
	6	SMF_119_6	TCP/IP Interface Statistics
		SMF_119_INTERFACE	Interface statistics
		SMF_119_HOME_IP	Home IP address section
	7	SMF_119_7	TCP/IP Server Port Statistics
		SMF_119_TCP_PORT	TCP server port statistics section
		SMF_119_UDP_PORT	UDP server port statistics section
	8	SMF_119_8	TCP/IP Stack Start/Stop
	10	SMF_119_10	UDP Socket Close Record
	11	SMF_119_11	zERT connection detail record
		SMF_119_11_DN	zERT DNs section
	12	SMF_119_12	zERT summary record
		SMF_119_12_DN	zERT summary DNs section
	20	SMF_119_20	TN3270 Server SNA Session Initiation
	21	SMF_119_21	TN3270 Server SNA Session Termination
	22	SMF_119_22	TSO Telnet Client Connection Initiation
	23	SMF_119_23	TSO Telnet Client Connection Termination
	32	SMF_119_32	DVIPA Status Change Record
	33	SMF_119_33	DVIPA Removed Record
	34	SMF_119_34	DVIPA Target Added Record
	35	SMF_119_35	DVIPA Target Removed Record
	36	SMF_119_36	DVIPA Target Server Started Record
	37	SMF_119_37	DVIPA Target Server Ended Record
	48	SMF_119_48	CSSMTP Configuration record
	49	SMF_119_49	CSSMTP Connection Record
	50	SMF_119_50	CSSMTP Mail Record
	51	SMF_119_51	CSSMTP Spool File Record
	52	SMF_119_52	CSSMTP Statistical Record
	70	SMF_119_70	FTP Server Transfer Completion
	72	SMF_119_72	FTP Server Logon Failure
73	SMF_119_73	IPSec IKE Tunnel Activation/Refresh	
74	SMF_119_74	IPSec IKE Tunnel Deactivation/Expire	
75 - 80	SMF_119_75_80	IPSec Dynamic/Manual Tunnel Activation/Refresh/Deactivate Add/Remove	
94 - 98	SMF_119_94_98	OpenSSH Server/Client Connection Started, Transfer Completion, Login Failure	

Table 22. Data stream names that IBM Z Common Data Provider uses to collect SMF data (continued)

Type	Subtype	Data stream name	Description of data stream content
120	9	SMF_120_9	WAS Request Activity Record
		SMF_120_9_CLA	Classification data section
		SMF_120_9_SEC	Security data section
		SMF_120_9_CPU	CPU usage breakdown section
		SMF_120_9_USR	User data section
	10	SMF_120_10	Outbound request record
	11	SMF_120_11	HTTP requests in liberty
		SMF_120_11_USD	User data section
		SMF_120_11_CLS	Classification section
	12	SMF_120_12	Batch job in liberty
		SMF_120_12_SC5	Accounting section
		SMF_120_12_SC7	Reference names section
		SMF_120_12_SC8	User data section
123	1	SMF_123_01_V2	The z/OS Connect EE audit interceptor records request activity to the SMF data store on z/OS operating systems.
		SMF_123_01	The z/OS Connect EE audit interceptor records request activity to the SMF data store on z/OS operating systems. Superseded by SMF_123_01_V2
		SMF_123_01_REQDATA	Request data section in the z/OS Connect EE audit interceptor records.
127	1000	SMF_IMS_01	IMS input message appeared in message queue
		SMF_IMS_01_TXT	IMS input message text section
		SMF_IMS_03	IMS output message appeared in message queue
		SMF_IMS_03_TXT	IMS output message text section
		SMF_IMS_07	IMS program termination
		SMF_IMS_08	IMS program start
		SMF_IMS_0A07	IMS CPI-CI program termination
		SMF_IMS_0A08	IMS CPI-CI program start
		SMF_IMS_10	IMS security violation
		SMF_IMS_56FA	IMS transaction level statistics
		SMF_IMS_5901	IMS Fast Path input message
		SMF_IMS_5903	IMS Fast Path output message
		SMF_IMS_CA01	IMS Transaction Index
		SMF_IMS_CA20	IMS Connect Transaction Index
		SMF_IMS_F9	IMS program
		SMF_IMS_FA	IMS transaction
		SMF_IMS_FA_DBTLR	IMS transaction database trailer
		194	
230		SMF_230_CA_16	CA ACF2 security-related activity
		SMF_230_CA_16_T1	For CA ACF2 record version 0 or 1, part of command trace information
		SMF_230_CA_16_T2	For CA ACF2 record version 2, part of command trace information
		SMF_230_CA_16_SNEN	Part of CA ACF2 distributed database sense information
231		SMF_231_CA_16	CA Top Secret security events for UNIX System Services
		SMF_231_CA_EXT	CA Top Secret audit records for UNIX System Services

Restriction: The following four fields, which have a length that is greater than 4096 bytes, are not included in the streams:

- ACEMFREC
- ACLMFARE
- ACRMFRUL
- ACWMFDATA

Restriction: The only supported event code type is SMF80EVT =70, which is OMVS TRACE (70).

DCOLLECT Data stream reference

For each DFSMS Data Collection Facility (DCOLLECT) record type, this reference lists the name of the data stream that IBM Z Common Data Provider uses to collect the data and includes a brief description of the data stream content. In the Configuration Tool, these DCOLLECT data stream names are shown in the


"Select data stream" window, which opens when you click the **Add Data Stream** icon  **DATA STREAM** in the **Policy Profile Edit** window.

Table 23 on page 176 provides the following information:

Column 1

The DCOLLECT record type

Column 2

The name of the data stream to which the DCOLLECT data is written

Column 3

A brief description of the content of the DCOLLECT data stream

Table 23. Data stream names that IBM Z Common Data Provider uses to collect DCOLLECT data

Type	Data stream name	Description of data stream content
D	DCOLLECT_D	Active Data Set Record
A	DCOLLECT_A	VSAM Association Information
V	DCOLLECT_V	Volume Information
M	DCOLLECT_M	Migrated Data Set Information
B	DCOLLECT_B	Backup Data Set Information
C	DCOLLECT_C	DASD Capacity Planning Information
T	DCOLLECT_T	Tape Capacity Planning Information
DC	DCOLLECT_DC	Data Class construct information
SC	DCOLLECT_SC	Storage Class construct information
MC	DCOLLECT_MC	Management Class construct information
BC	DCOLLECT_BC	Base Configuration Information
SG	DCOLLECT_SG	Storage Group construct Information
VL	DCOLLECT_VL	Storage Group volume Information
AG	DCOLLECT_AG	Aggregate Group information
DR	DCOLLECT_DR	OAM Drive Record information
LB	DCOLLECT_LB	OAM Library Record Information
CN	DCOLLECT_CN	Cache Names from the Base Configuration information
AI	DCOLLECT_AI	Accounting Information from the ACS routines

SMF_110_1_KPI data stream content

SMF_110_1_KPI records in the **SMF_110_1_KPI** data stream contain information about key performance indicators (KPIs) for CICS Transaction Server for z/OS monitoring.

Data stream definition in Configuration Tool

To select the **SMF_110_1_KPI** data stream in the IBM Z Common Data Provider Configuration Tool, complete the following steps:

1. In the "Select data stream" window, expand **CICS Transaction Server**.
2. Select the **SMF_110_1_KPI** check box.

Fields in the SMF_110_1_KPI data stream

In the following table, the column that is titled "Corresponding SMF field" indicates the name of the SMF field that corresponds to the field name in the data stream.

Table 24. Fields in the **SMF_110_1_KPI** data stream

Field name	Description	Corresponding SMF field
Time	The time that the record was written to SMF	SMFMNTME
Date	The date that the record was written to SMF	SMFMNDTE
MVS_SYSTEM_ID	The system ID, which is also known as the SMF ID	SMFMNSID
START_TIMESTAMP	The start time of the transaction. The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.	START
STOP_TIMESTAMP	The stop time of the transaction. The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.	STOP
ELAPSED_TIME	The elapsed time of the transaction, which is derived by the System Data Engine (the stop time minus the start time)	Not applicable
CICS_SPEC_APPLID	CICS Transaction Server for z/OS specific application ID	SMFMNSPN
CICS_GEN_APPLID	CICS Transaction Server for z/OS generic application ID	SMFMNPRN
JOB_NAME	CICS Transaction Server for z/OS job name	SMFMNJBN
PGM_NAME	CICS Transaction Server for z/OS program name. The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.	PGMNAME
TRANSACTION_ID	CICS Transaction Server for z/OS transaction ID. The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.	TRAN
TRANSACTION_NUM	CICS Transaction Server for z/OS transaction number. The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.	TRANNUM
ORIG_ABEND_CODR	Original CICS Transaction Server for z/OS abend code. The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.	ABCODEO
CURR_ABEND_CODE	Current CICS Transaction Server for z/OS abend code. The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.	ABCODEC
CICS_USER	Current CICS Transaction Server for z/OS user ID. The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.	USERID
SYNCPPOINTS	The total number of syncpoint requests that are issued by the user task. The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.	SPSYNCCT

Table 24. Fields in the **SMF_110_1_KPI** data stream (continued)

Field name	Description	Corresponding SMF field
TERM_WAIT	<p>After the user task issued a RECEIVE request, the elapsed time during which the user task waited for input from the terminal operator.</p> <p>The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.</p>	TCIOWTT
DISPATCH_TIME	<p>The total elapsed time during which the user task was dispatched on each CICS task control block (TCB) under which the task ran. The TCB modes that are managed by the CICS dispatcher are: QR, RO, CO, FO, SZ, RP, SL, SP, SO, EP, J8, J9, L8, L9, S8, TP, T8, X8, X9, JM, and D2.</p> <p>The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.</p>	USRDISPT
CPU_TIME	<p>The total processor time during which the user task was dispatched by the CICS dispatcher domain on each CICS TCB under which the task ran.</p> <p>The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.</p>	USRCPUT
RLS_CPU_TIME	<p>The amount of CPU time in which the transaction was processing record-level sharing (RLS) file requests.</p> <p>Tip: For a measurement of the total CPU time that is used by a transaction, add this RLS_CPU_TIME value to the CPU_TIME value (SMF field USRCPUT).</p> <p>The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.</p>	RLSCPUT
SUSP_TIME	<p>The time in which the user task was suspended by the CICS dispatcher.</p> <p>The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.</p>	SUSPTIME
SYNCTIME	<p>The time in which the user task was dispatched and was processing syncpoint requests.</p> <p>The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.</p>	SYNCTIME
DISP_CICS_USER	<p>The dispatch time that CICS Transaction Server for z/OS gives to a user task, which is the total elapsed time during which the user task is dispatched by the CICS dispatcher domain on a CICS Key 8 mode TCB.</p> <p>The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.</p>	KY8DISPT




Table 24. Fields in the **SMF_110_1_KPI** data stream (continued)

Field name	Description	Corresponding SMF field
JAVA_CPU_TIME	<p>This field is a composite field that indicates one of the following elements:</p> <ul style="list-style-type: none"> • The amount of CPU time that this task used when it was dispatched on the J8 TCB Mode • The number of times that this task was dispatched on the J8 TCB Mode • An indication that the mode is used by Java applications <p>The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.</p>	J8CPUT
L8_TCB_DISP_TIME	<p>This field is a composite field that indicates one of the following elements:</p> <ul style="list-style-type: none"> • The amount of CPU time that this task used when it was dispatched on the L8 TCB Mode • The number of times that this task was dispatched on the L8 TCB Mode • An indication that the mode is used by programs that are defined with CONCURRENCY=THREADSAFE when they issue Db2 requests <p>The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.</p>	L8CPUT
S8_TCB_DISP_TIME	<p>This field is a composite field that indicates one of the following elements:</p> <ul style="list-style-type: none"> • The amount of CPU time that this task used when it was dispatched on the S8 TCB Mode • The number of times that this task was dispatched on the S8 TCB Mode • An indication that the mode is used for making secure socket calls <p>The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.</p>	S8CPUT
RMI_TIME	<p>The time in which the task was external to CICS Transaction Server for z/OS (for example, in Db2 or MQ).</p> <p>The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.</p>	RMITIME
RMI_SUSP_TIME	<p>The time in which the user task was suspended by the CICS dispatcher while it was in the CICS Resource Manager Interface (RMI).</p> <p>The field name corresponds to the CICS Transaction Server for z/OS Dictionary nickname.</p>	RMISUSP




Icons on each node in a policy

This reference describes the icons that are shown on each data stream, transform, and subscriber node that you define in a policy. It also indicates where you can find more information about configuring data streams, transforms, and subscribers.

Data stream node




Icon	Window that opens when you click icon	More information
Configure icon 	One of the following windows opens, depending on whether the data is gathered by the Log Forwarder or the System Data Engine: <ul style="list-style-type: none"> • Configure Log Forwarder data stream • Configure System Data Engine data stream 	<ul style="list-style-type: none"> • “Data stream configuration for data gathered by Log Forwarder” on page 181 • “Data stream configuration for data gathered by System Data Engine” on page 228
Transform icon 	<ul style="list-style-type: none"> • Transform data stream 	<ul style="list-style-type: none"> • “Transform configuration” on page 229
Subscribe icon 	<ul style="list-style-type: none"> • Subscribe to a data stream 	<ul style="list-style-type: none"> • “Adding a subscriber for a data stream or transform” on page 44 • “Subscriber configuration” on page 235

Transform node

Icon	Window that opens when you click icon	More information
Configure icon 	One of the following windows opens: <ul style="list-style-type: none"> • Configure Transcribe transform • Configure Splitter transform • Configure Filter transform 	<ul style="list-style-type: none"> • “Transform configuration” on page 229
Transform icon 	<ul style="list-style-type: none"> • Transform data stream 	<ul style="list-style-type: none"> • “Transform configuration” on page 229
Subscribe icon 	<ul style="list-style-type: none"> • Subscribe to a transform 	<ul style="list-style-type: none"> • “Adding a subscriber for a data stream or transform” on page 44 • “Subscriber configuration” on page 235

Subscriber node

Table 27. Icons on each subscriber node in a policy

Icon	Window that opens when you click icon	More information
Configure icon 	<ul style="list-style-type: none"> • Configure subscriber 	<ul style="list-style-type: none"> • “Adding a subscriber for a data stream or transform” on page 44 • “Subscriber configuration” on page 235
Export icon 	<ul style="list-style-type: none"> • Export 	<ul style="list-style-type: none"> • “Exporting and importing subscribers” on page 45
Subscribe icon 	<ul style="list-style-type: none"> • Update the subscriptions of this subscriber 	<ul style="list-style-type: none"> • “Adding a subscriber for a data stream or transform” on page 44 • “Subscriber configuration” on page 235

Data stream configuration for data gathered by Log Forwarder

This reference lists the configuration values that you can update in the "**Configure Log Forwarder data stream**" window. The fields that are shown in this window are based on the source from which the Log Forwarder collects data for the data stream.

The Log Forwarder gathers z/OS log data from the following sources:

- Job log, which is output that is written to a data definition (DD) by a running job
- z/OS UNIX log file, including the UNIX System Services system log (syslogd)
- Entry-sequenced Virtual Storage Access Method (VSAM) cluster
- z/OS system log (SYSLOG)
- IBM Tivoli NetView for z/OS messages
- IBM WebSphere Application Server for z/OS High Performance Extensible Logging (HPEL) log
- IBM Resource Measurement Facility (RMF) Monitor III reports

[Table 28 on page 182](#) summarizes which data streams come from which sources.

Table 28. Correlation between the sources from which the Log Forwarder gathers data and the data streams that can be defined for those sources

Source	Data streams
Job log	<ul style="list-style-type: none"> • “Generic z/OS Job Output data stream” on page 188 • “CICS EYULOG data stream” on page 192 • “CICS EYULOG DMY data stream” on page 194 • “CICS EYULOG YMD data stream” on page 196 • “CICS User Messages data stream” on page 198 • “CICS User Messages DMY data stream” on page 201 • “CICS User Messages YMD data stream” on page 203 • “WebSphere SYSOUT data stream” on page 209 • “WebSphere SYSPRINT data stream” on page 213 • “WebSphere USS Sysout data stream” on page 212 • “WebSphere USS Sysprint in Distributed Format data stream” on page 218 • “WebSphere SYSPRINT in Distributed Format data stream” on page 215 • WebSphere Liberty Console log
z/OS UNIX log file	<ul style="list-style-type: none"> • “Generic ZFS File data stream” on page 186 • “USS Syslogd data stream” on page 207 • “WebSphere USS Sysprint data stream” on page 217 • WebSphere Liberty log • WebSphere Liberty USS Console log
Entry-sequenced VSAM cluster	<ul style="list-style-type: none"> • “Generic VSAM Cluster data stream” on page 182
z/OS SYSLOG	<ul style="list-style-type: none"> • “z/OS SYSLOG data stream” on page 224 • “z/OS SYSLOG from ARCHIVE data stream” on page 225
IBM Tivoli NetView for z/OS messages	<ul style="list-style-type: none"> • “NetView Netlog data stream” on page 205
IBM WebSphere Application Server for z/OS HPEL log	<ul style="list-style-type: none"> • “WebSphere HPEL data stream” on page 208
IBM Resource Measurement Facility Monitor III reports	<ul style="list-style-type: none"> • “RMF III Report data streams” on page 206

Generic VSAM Cluster data stream

This reference lists the configuration values that you can update in the "**Configure Log Forwarder data stream**" window for the **Generic VSAM Cluster** data stream. It also describes why you might want to define paired data sets for this data stream.

Data collection from paired data sets

For the **Generic VSAM Cluster** data stream, the Log Forwarder can gather log data from a logical pair of data sets, called *paired data sets*. The use of paired data sets prevents an individual data set from getting too large and makes the process of pruning old log data from the system much easier.

With paired data sets, data is logged to only one data set in the pair at a time. When that data set exceeds some threshold (for example, the data set surpasses a specified size, or a specified time interval passes), the data in the other data set is deleted, and logging switches to that other data set. This switching between each data set in the pair is repeated continuously as each threshold is exceeded.

When you define a **Generic VSAM Cluster** data stream, you can specify either a single data set (in the **Data Set Name** field) or two data sets (one in the **Data Set Name** field, and the other in the **Paired Data Set Name** field) that are logically paired. If you specify two data sets, the contents of both data sets are associated with the same data stream. Both data sets must be entry-sequenced Virtual Storage Access Method (VSAM) clusters. At least one of the data sets must be allocated before the Log Forwarder is started.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Data Set Name

The name of the entry-sequenced VSAM cluster that contains the data to be gathered. This name must be in the format *x.y.z*.

Tip: If you want to delete and redefine a VSAM ESDS data set whose data is gathered by the Log Forwarder, after deleting the old data set, wait at least 60 seconds (1 minute) before redefining and writing data to the new data set, so that the Log Forwarder can collect all the records in the new data set.

Paired Data Set Name

The name of the entry-sequenced VSAM cluster that, together with the cluster that is specified in the **Data Set Name** field, contains the data to be gathered. This name must be in the format *x.y.z*.

Tip: If you want to delete and redefine a VSAM ESDS data set whose data is gathered by the Log Forwarder, after deleting the old data set, wait at least 60 seconds (1 minute) before redefining and writing data to the new data set, so that the Log Forwarder can collect all the records in the new data set.

For more information about the use of paired data sets, see [“Data collection from paired data sets” on page 182](#).

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see [“Subscriber configuration” on page 235](#).

Data Source Type

A value that the subscriber can use to uniquely identify the type and format of the streamed data.

File Path

A unique identifier that represents the data origin.

Time Format

The format for the time in a generic data stream. This value is used to extract and parse the time in each record. You can select a pre-defined time format from the drop list or select **Other** to specify a time format if there is no matching one. For all the supported symbols in the time format, refer to the [Table 29 on page 183](#) table.

Symbol	Meaning	Example
G	era	AD

<i>Table 29. Support symbols (continued)</i>		
Symbol	Meaning	Example
yyyy	year-of-era	2019
yy	year-of-era	19
D	day-of-year	189
MMMM	month-of-year	July
MMM	month-of-year	Jul
MM	month-of-year	6
M	month-of-year	6
dd	day-of-month	1
d	day-of-month	1
Q	quarter-of-year	1; 01
YYYY	week-based-year	2019
YY	week-based-year	19
ww	week-of-week-based-year	5
w	week-of-week-based-year	5
W	week-of-month	1; 01
EEEE	day-of-week	Tuesday
EEE	day-of-week	Tue
e	localized day-of-week	1; 01
a	am-pm-of-day	AM; PM
hh	clock-hour-of-am-pm (1-12)	5
h	clock-hour-of-am-pm (1-12)	5
kk	clock-hour-of-am-pm (1-24)	05; 15
k	clock-hour-of-am-pm (1-24)	5; 15
KK	hour-of-am-pm (0-11)	9
K	hour-of-am-pm (0-11)	9
HH	hour-of-day (0-23)	5
H	hour-of-day (0-23)	5
mm	minute-of-hour	6
m	minute-of-hour	6
ss	second-of-min	2
s	second-of-min	2
SSS	fraction-of-sec	111
SSSSSS	fraction-of-sec	111111
A	milli-of-day	1234

<i>Table 29. Support symbols (continued)</i>		
Symbol	Meaning	Example
n	nano-of-second	987654321
N	nano-of-day	1234000000
V	time-zone ID	America/Los_Angeles
z	time-zone name	PST; -08:00; -08:30:00
Z	zone-offset	+0000; -0800;
x	zone-offset	-08; -0830
X	zone-offset	Z; -08; -0830;

Time Zone

If the timestamp in the collected data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone, which is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration”](#) on page 154.

The value must be in the format *plus_or_minusHHMM*, where *plus_or_minus* represents the + or - sign, *HH* represents two digits for the hour, and *MM* represents two digits for the minute.

Examples:

If you want this time zone	Specify this value
Coordinated Universal Time (UTC)	+0000
5 hours west of UTC	-0500
8 hours east of UTC	+0800

RLS Access

A specification of whether the monitored VSAM ESDS data set is accessed through RLS or not. The default value is No, which represents that the data set is not accessed through RLS. If the monitored VSAM data set is accessed through RLS, select Yes to make sure that IBM Z Common Data Provider can monitor the data in the correct way.

Tip: For non-RLS ESDS, if there is an application which has a long open connection to it (for example CICS), the SHAREOPTIONS must be (4,4) or (4,3) to make sure that IBM Z Common Data Provider can continue collecting data after the data size reaches the CI. Otherwise, the remaining data can only be collected after the application performs a "CLOSE" command to the VSAM data set due to the read/write integrity identified by DFSMS.

For RLS ESDS, there is no such requirement.

For more information on SHAREOPTIONS, see <https://www.ibm.com/docs/en/zos/2.4.0?topic=sharing-cross-region-share-options>.

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via

HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

Generic ZFS File data stream

This reference lists the configuration values that you can update in the "**Configure Log Forwarder data stream**" window for the **Generic ZFS File** data stream.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

File Path

A unique identifier that represents the data origin. The identifier must be the absolute path, including the file name, of a log file that contains the relevant data.

Tip: If you are gathering log data from a rolling z/OS UNIX log, see "[Data collection from a rolling z/OS UNIX log](#)" on page 225 for more information, including how to specify this file path value for a rolling log.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see "[Subscriber configuration](#)" on page 235.

Data Source Type

A value that the subscriber can use to uniquely identify the type and format of the streamed data.

Time Format

The format for the time in a generic data stream. This value is used to extract and parse the time in each record. You can select a pre-defined time format from the drop list or select **Other** to specify a time format if there is no matching one. For all the supported symbols in the time format, refer to the [Table 30 on page 186](#) table.

Symbol	Meaning	Example
G	era	AD
YYYY	year-of-era	2019
yy	year-of-era	19
D	day-of-year	189
MMMM	month-of-year	July
MMM	month-of-year	Jul
MM	month-of-year	6
M	month-of-year	6
dd	day-of-month	1
d	day-of-month	1
Q	quarter-of-year	1; 01
YYYY	week-based-year	2019

<i>Table 30. Support symbols (continued)</i>		
Symbol	Meaning	Example
YY	week-based-year	19
ww	week-of-week-based-year	5
w	week-of-week-based-year	5
W	week-of-month	1; 01
EEEE	day-of-week	Tuesday
EEE	day-of-week	Tue
e	localized day-of-week	1; 01
a	am-pm-of-day	AM; PM
hh	clock-hour-of-am-pm (1-12)	5
h	clock-hour-of-am-pm (1-12)	5
kk	clock-hour-of-am-pm (1-24)	05; 15
k	clock-hour-of-am-pm (1-24)	5; 15
KK	hour-of-am-pm (0-11)	9
K	hour-of-am-pm (0-11)	9
HH	hour-of-day (0-23)	5
H	hour-of-day (0-23)	5
mm	minute-of-hour	6
m	minute-of-hour	6
ss	second-of-min	2
s	second-of-min	2
SSS	fraction-of-sec	111
SSSSSS	fraction-of-sec	111111
A	milli-of-day	1234
n	nano-of-second	987654321
N	nano-of-day	1234000000
V	time-zone ID	America/Los_Angeles
z	time-zone name	PST; -08:00; -08:30:00
Z	zone-offset	+0000; -0800;
x	zone-offset	-08; -0830
X	zone-offset	Z; -08; -0830;

Time Zone

If the timestamp in the collected data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone, which is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration”](#) on page 154.

The value must be in the format *plus_or_minusHHMM*, where *plus_or_minus* represents the + or - sign, *HH* represents two digits for the hour, and *MM* represents two digits for the minute.

Examples:

If you want this time zone	Specify this value
Coordinated Universal Time (UTC)	+0000
5 hours west of UTC	-0500
8 hours east of UTC	+0800

Discovery Interval

In the process of streaming data, the number of minutes that the Log Forwarder waits before it checks for a new log file in the data stream.

The value must be an integer in the range 0 - 5. A value of 0 specifies that the Log Forwarder only checks for a new log file once when the data gatherer is started. The default value is the value that is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration”](#) on page 154.

Encoding

A value that specifies the encoding type of the ZFS File. For all the supported encoding types, refer to the following list.

- Default platform encoding

The encoding type of the platform that runs the Log Forwarder component is used as the encoding type of the ZFS File.

- UTF-8
- UTF-16
- UTF-32
- IBM037
- IBM1047
- US-ASCII

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

Generic z/OS Job Output data stream

This reference lists the configuration values that you can update in the **“Configure Log Forwarder data stream”** window for the **Generic z/OS Job Output** data stream. It also describes how to use wildcard characters in the **Job Name** field for this data stream.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Job Name

The name of the server job from which to gather data. This value can contain wildcard characters.

For information about the use of wildcard characters, see [“Use of wildcard characters in the Job Name field” on page 191.](#)

DD Name

The data definition (DD) name for the job log.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see [“Subscriber configuration” on page 235.](#)

Data Source Type

A value that the subscriber can use to uniquely identify the type and format of the streamed data.

File Path

A unique identifier, such as *jobName/ddName*, that represents the data origin.

Time Format

The format for the time in a generic data stream. This value is used to extract and parse the time in each record. You can select a pre-defined time format from the drop list or select **Other** to specify a time format if there is no matching one. For all the supported symbols in the time format, refer to the [Table 31 on page 189 table.](#)

<i>Table 31. Support symbols</i>		
Symbol	Meaning	Example
G	era	AD
YYYY	year-of-era	2019
yy	year-of-era	19
D	day-of-year	189
MMMM	month-of-year	July
MMM	month-of-year	Jul
MM	month-of-year	6
M	month-of-year	6
dd	day-of-month	1
d	day-of-month	1
Q	quarter-of-year	1; 01
YYYY	week-based-year	2019
YY	week-based-year	19
ww	week-of-week-based-year	5
w	week-of-week-based-year	5
W	week-of-month	1; 01
EEEE	day-of-week	Tuesday
EEE	day-of-week	Tue
e	localized day-of-week	1; 01

<i>Table 31. Support symbols (continued)</i>		
Symbol	Meaning	Example
a	am-pm-of-day	AM; PM
hh	clock-hour-of-am-pm (1-12)	5
h	clock-hour-of-am-pm (1-12)	5
kk	clock-hour-of-am-pm (1-24)	05; 15
k	clock-hour-of-am-pm (1-24)	5; 15
KK	hour-of-am-pm (0-11)	9
K	hour-of-am-pm (0-11)	9
HH	hour-of-day (0-23)	5
H	hour-of-day (0-23)	5
mm	minute-of-hour	6
m	minute-of-hour	6
ss	second-of-min	2
s	second-of-min	2
SSS	fraction-of-sec	111
SSSSSS	fraction-of-sec	111111
A	milli-of-day	1234
n	nano-of-second	987654321
N	nano-of-day	1234000000
V	time-zone ID	America/Los_Angeles
z	time-zone name	PST; -08:00; -08:30:00
Z	zone-offset	+0000; -0800;
x	zone-offset	-08; -0830
X	zone-offset	Z; -08; -0830;

Time Zone

If the timestamp in the collected data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone, which is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration”](#) on page 154.

The value must be in the format *plus_or_minusHHMM*, where *plus_or_minus* represents the + or - sign, *HH* represents two digits for the hour, and *MM* represents two digits for the minute.

Examples:

If you want this time zone	Specify this value
Coordinated Universal Time (UTC)	+0000
5 hours west of UTC	-0500
8 hours east of UTC	+0800

Discovery Interval

In the process of streaming data, the number of minutes that the Log Forwarder waits before it checks for a new log file in the data stream.

The value must be an integer in the range 0 - 5. A value of 0 specifies that the Log Forwarder only checks for a new log file once when the data gatherer is started. The default value is the value that is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration”](#) on page 154.

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

Use of wildcard characters in the Job Name field

In the **Job Name** field for this data stream, you can use the following wildcard characters:

Wildcard character	What the character represents
?	Any single character
*	Any sequence of characters, including an empty sequence

If you use wildcard characters in the job name, the job name value becomes a pattern, and the data stream definition becomes a template. When the Log Forwarder starts, it searches the Job Entry Subsystem (JES) spool for job names that match the pattern, and it creates a separate data stream for each unique job name that it discovers. After the Log Forwarder initialization is complete, the Log Forwarder continues to monitor the job names on the JES spool. As it discovers new job names that match the pattern, it uses the same template to create more data streams.

For example, if the job name value is ABCD????, and the JES spool contains the following jobs, two data streams are created, one for job name ABCD1234 and one for job name ABCDE567:

JOBNAME	JobID
ABCD1234	STC00735
DEFG1234	STC00746
ABCDE567	STC00798
DEFG5678	STC00775
ABCD123	STC00772
DEFG456	STC00794
HBODSPRO	STC00623
HBOPROC	STC00661
SYSLOG	STC00552

Tips:

- To avoid gathering data from job logs that you do not intend to gather from, use a job name pattern that is not too broad.
- The Log Forwarder might discover jobs from other systems if spool is shared between systems or if JES multi-access spool is enabled. Although the data stream does not include data for the jobs that run on other systems, the Log Forwarder creates a data stream for that data. Therefore, ensure that the wildcard pattern does not match jobs that run on other systems.

Each resulting data stream is based on the template and has the same configuration values as the template, with the exception of the following values:

Template field	Value
Job Name	The discovered job name
Data Source Name	The value of the Data Source Name field in the template, with <code>_jobName_ddName</code> appended to that value. The <code>jobName</code> is the discovered job name, and the <code>ddName</code> is the DD name for the job log.
File Path	The value of the File Path field in the template, with <code>/jobName/ddName</code> appended to that value. The <code>jobName</code> is the discovered job name, and the <code>ddName</code> is the DD name for the job log.

CICS EYULOG data stream

This reference lists the configuration values that you can update in the "**Configure Log Forwarder data stream**" window for the **CICS EYULOG** data stream. It also describes how to use wildcard characters in the **Job Name** field for this data stream. The source for the **CICS EYULOG** data stream uses the date format "month day year" (MDY) in the timestamp.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Job Name

The name of the server job from which to gather data. This value can contain wildcard characters.

For information about the use of wildcard characters, see ["Use of wildcard characters in the Job Name field"](#) on page 193.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see ["Subscriber configuration"](#) on page 235.

File Path

A unique identifier, such as `jobName/ddName`, that represents the data origin.

Time Zone

If the timestamp in the collected data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone, which is defined in the Log Forwarder properties, as described in ["Log Forwarder properties configuration"](#) on page 154.

The value must be in the format `plus_or_minusHHMM`, where `plus_or_minus` represents the + or - sign, `HH` represents two digits for the hour, and `MM` represents two digits for the minute.

Examples:

If you want this time zone	Specify this value
Coordinated Universal Time (UTC)	+0000
5 hours west of UTC	-0500
8 hours east of UTC	+0800

Discovery Interval

In the process of streaming data, the number of minutes that the Log Forwarder waits before it checks for a new log file in the data stream.

The value must be an integer in the range 0 - 5. A value of 0 specifies that the Log Forwarder only checks for a new log file once when the data gatherer is started. The default value is the value that is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration”](#) on page 154.

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

Use of wildcard characters in the Job Name field

In the **Job Name** field for this data stream, you can use the following wildcard characters:

Wildcard character	What the character represents
?	Any single character
*	Any sequence of characters, including an empty sequence

If you use wildcard characters in the job name, the job name value becomes a pattern, and the data stream definition becomes a template. When the Log Forwarder starts, it searches the Job Entry Subsystem (JES) spool for job names that match the pattern, and it creates a separate data stream for each unique job name that it discovers. After the Log Forwarder initialization is complete, the Log Forwarder continues to monitor the job names on the JES spool. As it discovers new job names that match the pattern, it uses the same template to create more data streams.

For example, if the job name value is CMAS5*, and the JES spool contains the following jobs, two data streams are created, one for job name CMAS53 and one for job name CMAS5862:

JOBNAME	JobID
CMAS43	STC00586
CMAS482	STC00588
CMAS53	STC00587
CMAS5862	STC00589
CMAS61	STC00590

JOBNAME	JobID
CMAS62	STC00600
HBODSPRO	STC00623
HBOPROC	STC00661
SYSLOG	STC00552

Tips:

- To avoid gathering data from job logs that you do not intend to gather from, use a job name pattern that is not too broad.
- The Log Forwarder might discover jobs from other systems if spool is shared between systems or if JES multi-access spool is enabled. Although the data stream does not include data for the jobs that run on other systems, the Log Forwarder creates a data stream for that data. Therefore, ensure that the wildcard pattern does not match jobs that run on other systems.

Each resulting data stream is based on the template and has the same configuration values as the template, with the exception of the following values:

Template field	Value
Job Name	The discovered job name
Data Source Name	The value of the Data Source Name field in the template, with <code>_jobName_EYULOG</code> appended to that value. The <code>jobName</code> is the discovered job name.
File Path	The value of the File Path field in the template, with <code>/jobName/EYULOG</code> appended to that value. The <code>jobName</code> is the discovered job name.

CICS EYULOG DMY data stream

This reference lists the configuration values that you can update in the "**Configure Log Forwarder data stream**" window for the **CICS EYULOG DMY** data stream. It also describes how to use wildcard characters in the **Job Name** field for this data stream. The source for the **CICS EYULOG DMY** data stream uses the date format "day month year" (DMY) in the timestamp.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Job Name

The name of the server job from which to gather data. This value can contain wildcard characters.

For information about the use of wildcard characters, see ["Use of wildcard characters in the Job Name field"](#) on page 195.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see ["Subscriber configuration"](#) on page 235.

File Path

A unique identifier, such as `jobName/ddName`, that represents the data origin.

Time Zone

If the timestamp in the collected data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone, which is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration” on page 154](#).

The value must be in the format *plus_or_minusHHMM*, where *plus_or_minus* represents the + or - sign, *HH* represents two digits for the hour, and *MM* represents two digits for the minute.

Examples:

If you want this time zone	Specify this value
Coordinated Universal Time (UTC)	+0000
5 hours west of UTC	-0500
8 hours east of UTC	+0800

Discovery Interval

In the process of streaming data, the number of minutes that the Log Forwarder waits before it checks for a new log file in the data stream.

The value must be an integer in the range 0 - 5. A value of 0 specifies that the Log Forwarder only checks for a new log file once when the data gatherer is started. The default value is the value that is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration” on page 154](#).

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

Use of wildcard characters in the Job Name field

In the **Job Name** field for this data stream, you can use the following wildcard characters:

Wildcard character	What the character represents
?	Any single character
*	Any sequence of characters, including an empty sequence

If you use wildcard characters in the job name, the job name value becomes a pattern, and the data stream definition becomes a template. When the Log Forwarder starts, it searches the Job Entry Subsystem (JES) spool for job names that match the pattern, and it creates a separate data stream for each unique job name that it discovers. After the Log Forwarder initialization is complete, the Log Forwarder continues to monitor the job names on the JES spool. As it discovers new job names that match the pattern, it uses the same template to create more data streams.

For example, if the job name value is CMAS5*, and the JES spool contains the following jobs, two data streams are created, one for job name CMAS53 and one for job name CMAS5862:

JOBNAME	JobID
CMAS43	STC00586
CMAS482	STC00588
CMAS53	STC00587
CMAS5862	STC00589
CMAS61	STC00590
CMAS62	STC00600
HBODSPRO	STC00623
HBOPROC	STC00661
SYSLOG	STC00552

Tips:

- To avoid gathering data from job logs that you do not intend to gather from, use a job name pattern that is not too broad.
- The Log Forwarder might discover jobs from other systems if spool is shared between systems or if JES multi-access spool is enabled. Although the data stream does not include data for the jobs that run on other systems, the Log Forwarder creates a data stream for that data. Therefore, ensure that the wildcard pattern does not match jobs that run on other systems.

Each resulting data stream is based on the template and has the same configuration values as the template, with the exception of the following values:

Template field	Value
Job Name	The discovered job name
Data Source Name	The value of the Data Source Name field in the template, with <i>_jobName_EYULOG</i> appended to that value. The <i>jobName</i> is the discovered job name.
File Path	The value of the File Path field in the template, with <i>/jobName/EYULOG</i> appended to that value. The <i>jobName</i> is the discovered job name.

CICS EYULOG YMD data stream

This reference lists the configuration values that you can update in the "Configure Log Forwarder data stream" window for the **CICS EYULOG YMD** data stream. It also describes how to use wildcard characters in the **Job Name** field for this data stream. The source for the **CICS EYULOG YMD** data stream uses the date format "year month day" (YMD) in the timestamp.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Job Name

The name of the server job from which to gather data. This value can contain wildcard characters.

For information about the use of wildcard characters, see "[Use of wildcard characters in the Job Name field](#)" on page 197.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see [“Subscriber configuration”](#) on page 235.

File Path

A unique identifier, such as *jobName/ddName*, that represents the data origin.

Time Zone

If the timestamp in the collected data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone, which is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration”](#) on page 154.

The value must be in the format *plus_or_minusHHMM*, where *plus_or_minus* represents the + or - sign, *HH* represents two digits for the hour, and *MM* represents two digits for the minute.

Examples:

If you want this time zone	Specify this value
Coordinated Universal Time (UTC)	+0000
5 hours west of UTC	-0500
8 hours east of UTC	+0800

Discovery Interval

In the process of streaming data, the number of minutes that the Log Forwarder waits before it checks for a new log file in the data stream.

The value must be an integer in the range 0 - 5. A value of 0 specifies that the Log Forwarder only checks for a new log file once when the data gatherer is started. The default value is the value that is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration”](#) on page 154.

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

Use of wildcard characters in the Job Name field

In the **Job Name** field for this data stream, you can use the following wildcard characters:

Wildcard character	What the character represents
?	Any single character
*	Any sequence of characters, including an empty sequence

If you use wildcard characters in the job name, the job name value becomes a pattern, and the data stream definition becomes a template. When the Log Forwarder starts, it searches the Job Entry

Subsystem (JES) spool for job names that match the pattern, and it creates a separate data stream for each unique job name that it discovers. After the Log Forwarder initialization is complete, the Log Forwarder continues to monitor the job names on the JES spool. As it discovers new job names that match the pattern, it uses the same template to create more data streams.

For example, if the job name value is CMAS5*, and the JES spool contains the following jobs, two data streams are created, one for job name CMAS53 and one for job name CMAS5862:

JOBNAME	JobID
CMAS43	STC00586
CMAS482	STC00588
CMAS53	STC00587
CMAS5862	STC00589
CMAS61	STC00590
CMAS62	STC00600
HBODSPRO	STC00623
HBOPROC	STC00661
SYSLOG	STC00552

Tips:

- To avoid gathering data from job logs that you do not intend to gather from, use a job name pattern that is not too broad.
- The Log Forwarder might discover jobs from other systems if spool is shared between systems or if JES multi-access spool is enabled. Although the data stream does not include data for the jobs that run on other systems, the Log Forwarder creates a data stream for that data. Therefore, ensure that the wildcard pattern does not match jobs that run on other systems.

Each resulting data stream is based on the template and has the same configuration values as the template, with the exception of the following values:

Template field	Value
Job Name	The discovered job name
Data Source Name	The value of the Data Source Name field in the template, with <i>_jobName_EYULOG</i> appended to that value. The <i>jobName</i> is the discovered job name.
File Path	The value of the File Path field in the template, with <i>/jobName/EYULOG</i> appended to that value. The <i>jobName</i> is the discovered job name.

CICS User Messages data stream

This reference lists the configuration values that you can update in the "**Configure Log Forwarder data stream**" window for the **CICS User Messages** data stream. It also describes how to use wildcard

characters in the **Job Name** field for this data stream. The source for the **CICS User Messages** data stream uses the date format "month day year" (MDY) in the timestamp.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Job Name

The name of the server job from which to gather data. This value can contain wildcard characters.

For information about the use of wildcard characters, see [“Use of wildcard characters in the Job Name field” on page 200.](#)

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see [“Subscriber configuration” on page 235.](#)

File Path

A unique identifier, such as *jobName/ddName*, that represents the data origin.

Time Zone

If the timestamp in the collected data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone, which is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration” on page 154.](#)

The value must be in the format *plus_or_minusHHMM*, where *plus_or_minus* represents the + or - sign, *HH* represents two digits for the hour, and *MM* represents two digits for the minute.

Examples:

If you want this time zone	Specify this value
Coordinated Universal Time (UTC)	+0000
5 hours west of UTC	-0500
8 hours east of UTC	+0800

Discovery Interval

In the process of streaming data, the number of minutes that the Log Forwarder waits before it checks for a new log file in the data stream.

The value must be an integer in the range 0 - 5. A value of 0 specifies that the Log Forwarder only checks for a new log file once when the data gatherer is started. The default value is the value that is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration” on page 154.](#)

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via

HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

Use of wildcard characters in the Job Name field

In the **Job Name** field for this data stream, you can use the following wildcard characters:

Wildcard character	What the character represents
?	Any single character
*	Any sequence of characters, including an empty sequence

If you use wildcard characters in the job name, the job name value becomes a pattern, and the data stream definition becomes a template. When the Log Forwarder starts, it searches the Job Entry Subsystem (JES) spool for job names that match the pattern, and it creates a separate data stream for each unique job name that it discovers. After the Log Forwarder initialization is complete, the Log Forwarder continues to monitor the job names on the JES spool. As it discovers new job names that match the pattern, it uses the same template to create more data streams.

For example, if the job name value is *CMAS5**, and the JES spool contains the following jobs, two data streams are created, one for job name *CMAS53* and one for job name *CMAS5862*:

JOBNAME	JobID
CMAS43	STC00586
CMAS482	STC00588
CMAS53	STC00587
CMAS5862	STC00589
CMAS61	STC00590
CMAS62	STC00600
HBODSPRO	STC00623
HBOPROC	STC00661
SYSLOG	STC00552

Tips:

- To avoid gathering data from job logs that you do not intend to gather from, use a job name pattern that is not too broad.
- The Log Forwarder might discover jobs from other systems if spool is shared between systems or if JES multi-access spool is enabled. Although the data stream does not include data for the jobs that run on other systems, the Log Forwarder creates a data stream for that data. Therefore, ensure that the wildcard pattern does not match jobs that run on other systems.

Each resulting data stream is based on the template and has the same configuration values as the template, with the exception of the following values:

Template field	Value
Job Name	The discovered job name
Data Source Name	The value of the Data Source Name field in the template, with <i>_jobName_MSGUSR</i> appended to that value. The <i>jobName</i> is the discovered job name.

Template field	Value
File Path	The value of the File Path field in the template, with <code>/jobName/MSGUSR</code> appended to that value. The <code>jobName</code> is the discovered job name.

CICS User Messages DMY data stream

This reference lists the configuration values that you can update in the "**Configure Log Forwarder data stream**" window for the **CICS User Messages DMY** data stream. It also describes how to use wildcard characters in the **Job Name** field for this data stream. The source for the **CICS User Messages DMY** data stream uses the date format "day month year" (DMY) in the timestamp.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Job Name

The name of the server job from which to gather data. This value can contain wildcard characters.

For information about the use of wildcard characters, see ["Use of wildcard characters in the Job Name field"](#) on page 202.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see ["Subscriber configuration"](#) on page 235.

File Path

A unique identifier, such as `jobName/ddName`, that represents the data origin.

Time Zone

If the timestamp in the collected data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone, which is defined in the Log Forwarder properties, as described in ["Log Forwarder properties configuration"](#) on page 154.

The value must be in the format `plus_or_minusHHMM`, where `plus_or_minus` represents the + or - sign, `HH` represents two digits for the hour, and `MM` represents two digits for the minute.

Examples:

If you want this time zone	Specify this value
Coordinated Universal Time (UTC)	+0000
5 hours west of UTC	-0500
8 hours east of UTC	+0800

Discovery Interval

In the process of streaming data, the number of minutes that the Log Forwarder waits before it checks for a new log file in the data stream.

The value must be an integer in the range 0 - 5. A value of 0 specifies that the Log Forwarder only checks for a new log file once when the data gatherer is started. The default value is the value that is defined in the Log Forwarder properties, as described in ["Log Forwarder properties configuration"](#) on page 154.

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

Use of wildcard characters in the Job Name field

In the **Job Name** field for this data stream, you can use the following wildcard characters:

Wildcard character	What the character represents
?	Any single character
*	Any sequence of characters, including an empty sequence

If you use wildcard characters in the job name, the job name value becomes a pattern, and the data stream definition becomes a template. When the Log Forwarder starts, it searches the Job Entry Subsystem (JES) spool for job names that match the pattern, and it creates a separate data stream for each unique job name that it discovers. After the Log Forwarder initialization is complete, the Log Forwarder continues to monitor the job names on the JES spool. As it discovers new job names that match the pattern, it uses the same template to create more data streams.

For example, if the job name value is CMAS5*, and the JES spool contains the following jobs, two data streams are created, one for job name CMAS53 and one for job name CMAS5862:

JOBNAME	JobID
CMAS43	STC00586
CMAS482	STC00588
CMAS53	STC00587
CMAS5862	STC00589
CMAS61	STC00590
CMAS62	STC00600
HBODSPRO	STC00623
HBOPROC	STC00661
SYSLOG	STC00552

Tips:

- To avoid gathering data from job logs that you do not intend to gather from, use a job name pattern that is not too broad.
- The Log Forwarder might discover jobs from other systems if spool is shared between systems or if JES multi-access spool is enabled. Although the data stream does not include data for the jobs that run on other systems, the Log Forwarder creates a data stream for that data. Therefore, ensure that the wildcard pattern does not match jobs that run on other systems.

Each resulting data stream is based on the template and has the same configuration values as the template, with the exception of the following values:

Template field	Value
Job Name	The discovered job name
Data Source Name	The value of the Data Source Name field in the template, with <code>_jobName_MSGUSR</code> appended to that value. The <code>jobName</code> is the discovered job name.
File Path	The value of the File Path field in the template, with <code>/jobName/MSGUSR</code> appended to that value. The <code>jobName</code> is the discovered job name.

CICS User Messages YMD data stream

This reference lists the configuration values that you can update in the "**Configure Log Forwarder data stream**" window for the **CICS User Messages YMD** data stream. It also describes how to use wildcard characters in the **Job Name** field for this data stream. The source for the **CICS User Messages YMD** data stream uses the date format "year month day" (YMD) in the timestamp.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Job Name

The name of the server job from which to gather data. This value can contain wildcard characters.

For information about the use of wildcard characters, see ["Use of wildcard characters in the Job Name field"](#) on page 204.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see ["Subscriber configuration"](#) on page 235.

File Path

A unique identifier, such as `jobName/ddName`, that represents the data origin.

Time Zone

If the timestamp in the collected data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone, which is defined in the Log Forwarder properties, as described in ["Log Forwarder properties configuration"](#) on page 154.

The value must be in the format `plus_or_minusHHMM`, where `plus_or_minus` represents the + or - sign, `HH` represents two digits for the hour, and `MM` represents two digits for the minute.

Examples:

If you want this time zone	Specify this value
Coordinated Universal Time (UTC)	+0000
5 hours west of UTC	-0500
8 hours east of UTC	+0800

Discovery Interval

In the process of streaming data, the number of minutes that the Log Forwarder waits before it checks for a new log file in the data stream.

The value must be an integer in the range 0 - 5. A value of 0 specifies that the Log Forwarder only checks for a new log file once when the data gatherer is started. The default value is the value that is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration”](#) on page 154.

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

Use of wildcard characters in the Job Name field

In the **Job Name** field for this data stream, you can use the following wildcard characters:

Wildcard character	What the character represents
?	Any single character
*	Any sequence of characters, including an empty sequence

If you use wildcard characters in the job name, the job name value becomes a pattern, and the data stream definition becomes a template. When the Log Forwarder starts, it searches the Job Entry Subsystem (JES) spool for job names that match the pattern, and it creates a separate data stream for each unique job name that it discovers. After the Log Forwarder initialization is complete, the Log Forwarder continues to monitor the job names on the JES spool. As it discovers new job names that match the pattern, it uses the same template to create more data streams.

For example, if the job name value is CMAS5*, and the JES spool contains the following jobs, two data streams are created, one for job name CMAS53 and one for job name CMAS5862:

JOBNAME	JobID
CMAS43	STC00586
CMAS482	STC00588
CMAS53	STC00587
CMAS5862	STC00589
CMAS61	STC00590
CMAS62	STC00600
HBODSPRO	STC00623
HBOPROC	STC00661
SYSLOG	STC00552

Tips:

- To avoid gathering data from job logs that you do not intend to gather from, use a job name pattern that is not too broad.
- The Log Forwarder might discover jobs from other systems if spool is shared between systems or if JES multi-access spool is enabled. Although the data stream does not include data for the jobs that run on other systems, the Log Forwarder creates a data stream for that data. Therefore, ensure that the wildcard pattern does not match jobs that run on other systems.

Each resulting data stream is based on the template and has the same configuration values as the template, with the exception of the following values:

Template field	Value
Job Name	The discovered job name
Data Source Name	The value of the Data Source Name field in the template, with <code>_jobName_MSGUSR</code> appended to that value. The <code>jobName</code> is the discovered job name.
File Path	The value of the File Path field in the template, with <code>/jobName/MSGUSR</code> appended to that value. The <code>jobName</code> is the discovered job name.

NetView Netlog data stream

This reference lists the configuration values that you can update in the "**Configure Log Forwarder data stream**" window for the **NetView Netlog** data stream.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Domain Name

The name of the NetView domain from which to gather data.

Important: If you define multiple **NetView Netlog** data streams, do not define the same NetView domain name for multiple streams. Each stream must reference a unique domain name.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see "[Subscriber configuration](#)" on page 235.

File Path

A unique identifier that represents the data origin.

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is `Data Source Type_KV`. You can specify the value according to your needs.

RMF III Report data streams

This reference lists the configuration values that you can update in the "Configure Log Forwarder data stream" window for the **RMF III Report** data streams. **RMF III Report** is a group of data streams. Each data stream represents a Resource Measurement Facility (RMF) Monitor III report type.

RMF III report type	Data stream name
CFACT	RMF_CFACT
CFOVER	RMF_CFOVER
CFSYS	RMF_CFSYS
CHANNEL	RMF_CHANNEL
CPC	RMF_CPC
DELAY	RMF_DELAY
DEVR	RMF_DEVR
ENCLAVE	RMF_ENCLAVE
PROCU	RMF_PROCU
SPACEG	RMF_SPACEG
STORC	RMF_STORC
STORF	RMF_STORF
STORR	RMF_STORR
SYSINFO	RMF_SYSINFO
SYSSUM	RMF_SYSSUM
USAGE	RMF_USAGE

Configuration values that you can update

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see "[Subscriber configuration](#)" on page 235.

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

History Data Time Range

Defines the time range of the history data to be collected at ZCDP restart. The unit is hour, and the value must be an integer ranging from 1 to 48. The default value is 2. For example, when the value of this parameter is 2, upon ZCDP restart, the history data of the last 2 hours is collected.

History Data Collection Speed

Defines the speed of the history data collection. The value of this parameter is the ratio of the history data collection speed to the data generation speed. The default value is 4. For example, when the value of this parameter is 4, the history data is collected at four times the speed of the data generation. If you set the value to 30 while the RMF III report gathering interval is 30 seconds, the history data is collected every one second.

Important: The value of this parameter must follow the rules:

- The value cannot be larger than the RMF III report gathering interval in seconds. For example, if your RMF III report gathering interval is 1 minute (which is 60 seconds), the maximum value of History Data Collection Speed is 60.
- The value cannot be larger than 100. For example, if your RMF III report gathering interval is 200 seconds, the maximum value of History Data Collection Speed is 100.
- Setting the speed value too high might impact your DDS performance.

RMF III Report in CSV format

For the **RMF III Report** data streams, the Log Forwarder gathers data from the RMF Distributed Data Server (DDS) and converts the data to CSV format. Each CSV record contains the following parts:

Part	Description
LOCALSTART	Local start time of data range for the report
LOCALEND	Local end time of data range for the report
CAPTION	Additional headings and summary information for the report
ROW	The requested performance data values in the report

The records that are gathered from the same table have the same LOCALSTART, LOCALEND, and CAPTION values.

USS Syslogd data stream

This reference lists the configuration values that you can update in the "**Configure Log Forwarder data stream**" window for the **USS Syslogd Admin**, **USS Syslogd Debug**, and **USS Syslogd Error** data streams.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see "[Subscriber configuration](#)" on page 235.

File Path

A unique identifier that represents the data origin. The identifier must be the absolute path, including the file name, of a log file that contains the relevant data.

Tip: If you are gathering log data from a rolling z/OS UNIX log, see [“Data collection from a rolling z/OS UNIX log” on page 225](#) for more information, including how to specify this file path value for a rolling log.

Time Zone

If the timestamp in the collected data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone, which is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration” on page 154](#).

The value must be in the format *plus_or_minusHHMM*, where *plus_or_minus* represents the + or - sign, *HH* represents two digits for the hour, and *MM* represents two digits for the minute.

Examples:

If you want this time zone	Specify this value
Coordinated Universal Time (UTC)	+0000
5 hours west of UTC	-0500
8 hours east of UTC	+0800

Discovery Interval

In the process of streaming data, the number of minutes that the Log Forwarder waits before it checks for a new log file in the data stream.

The value must be an integer in the range 0 - 5. A value of 0 specifies that the Log Forwarder only checks for a new log file once when the data gatherer is started. The default value is the value that is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration” on page 154](#).

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

WebSphere HPEL data stream

This reference lists the configuration values that you can update in the **“Configure Log Forwarder data stream”** window for the **WebSphere HPEL** data stream.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Log Directory

The HPEL log directory for an application server that you are collecting data from. This HPEL log directory must have a `logdata` subdirectory, and the HPEL log files must be present in the `logdata` subdirectory.

If you are collecting only trace data, do not specify a value in the **Log Directory** field.

If no value is specified in the **Trace Directory** field, a **Log Directory** value is required. Otherwise, the **Log Directory** value is not required.

Trace Directory

The WebSphere Application Server for z/OS HPEL trace directory for an application server that you are collecting data from. This HPEL trace directory must have a `tracedata` subdirectory, and the HPEL trace files must be present in the `tracedata` subdirectory.

If you are collecting only log data, do not specify a value in the **Trace Directory** field.

If no value is specified in the **Log Directory** field, a **Trace Directory** value is required. Otherwise, the **Trace Directory** value is not required.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see [“Subscriber configuration” on page 235](#).

File Path

A unique identifier that represents the data origin. The identifier must be a virtual or physical path that represents the HPEL log data.

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is `Data Source Type_KV`. You can specify the value according to your needs.

WebSphere SYSOUT data stream

This reference lists the configuration values that you can update in the **“Configure Log Forwarder data stream”** window for the **WebSphere SYSOUT** data stream. It also describes how to use wildcard characters in the **Job Name** field for this data stream.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Job Name

The name of the server job from which to gather data. This value can contain wildcard characters.

For information about the use of wildcard characters, see [“Use of wildcard characters in the Job Name field” on page 210](#).

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see [“Subscriber configuration”](#) on page 235.

File Path

A unique identifier, such as *jobName/ddName*, that represents the data origin.

Time Zone

If the timestamp in the collected data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone, which is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration”](#) on page 154.

The value must be in the format *plus_or_minusHHMM*, where *plus_or_minus* represents the + or - sign, *HH* represents two digits for the hour, and *MM* represents two digits for the minute.

Examples:

If you want this time zone	Specify this value
Coordinated Universal Time (UTC)	+0000
5 hours west of UTC	-0500
8 hours east of UTC	+0800

Discovery Interval

In the process of streaming data, the number of minutes that the Log Forwarder waits before it checks for a new log file in the data stream.

The value must be an integer in the range 0 - 5. A value of 0 specifies that the Log Forwarder only checks for a new log file once when the data gatherer is started. The default value is the value that is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration”](#) on page 154.

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

Use of wildcard characters in the Job Name field

In the **Job Name** field for this data stream, you can use the following wildcard characters:

Wildcard character	What the character represents
?	Any single character
*	Any sequence of characters, including an empty sequence

If you use wildcard characters in the job name, the job name value becomes a pattern, and the data stream definition becomes a template. When the Log Forwarder starts, it searches the Job Entry Subsystem (JES) spool for job names that match the pattern, and it creates a separate data stream for each unique job name that it discovers. After the Log Forwarder initialization is complete, the Log

Forwarder continues to monitor the job names on the JES spool. As it discovers new job names that match the pattern, it uses the same template to create more data streams.

For example, if the job name value is BBOS??S, and the JES spool contains the following jobs, two data streams are created, one for job name BBOSABCS and one for job name BBOSDEFS:

JOBNAME	JobID
BBODMGR	STC00586
BBODMGRS	STC00588
BBODMNC	STC00587
BBON001	STC00589
BBOSABC	STC00590
BBOSABC	STC00600
BBOSABCS	STC00592
BBOSABCS	STC00602
BBOSDEF	STC00594
BBOSDEFS	STC00596
BBOSDEFS	STC00598
HBOPROC	STC00661
SYSLOG	STC00552

Tips:

- To avoid gathering data from job logs that you do not intend to gather from, use a job name pattern that is not too broad.
- The Log Forwarder might discover jobs from other systems if spool is shared between systems or if JES multi-access spool is enabled. Although the data stream does not include data for the jobs that run on other systems, the Log Forwarder creates a data stream for that data. Therefore, ensure that the wildcard pattern does not match jobs that run on other systems.

Each resulting data stream is based on the template and has the same configuration values as the template, with the exception of the following values:

Template field	Value
Job Name	The discovered job name
Data Source Name	The value of the Data Source Name field in the template, with <i>_jobName_SYSOUT</i> appended to that value. The <i>jobName</i> is the discovered job name.
File Path	The value of the File Path field in the template, with <i>/jobName/SYSOUT</i> appended to that value. The <i>jobName</i> is the discovered job name.

WebSphere USS Sysout data stream

This reference lists the configuration values that you can update in the "Configure Log Forwarder data stream" window for the **WebSphere USS Sysout** data stream.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see "[Subscriber configuration](#)" on page 235.

File Path

A unique identifier that represents the data origin. The identifier must be the absolute path, including the file name, of a log file that contains the relevant data.

Tip: If you are gathering log data from a rolling z/OS UNIX log, see "[Data collection from a rolling z/OS UNIX log](#)" on page 225 for more information, including how to specify this file path value for a rolling log.

Time Zone

If the timestamp in the collected data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone, which is defined in the Log Forwarder properties, as described in "[Log Forwarder properties configuration](#)" on page 154.

The value must be in the format *plus_or_minusHHMM*, where *plus_or_minus* represents the + or - sign, *HH* represents two digits for the hour, and *MM* represents two digits for the minute.

Examples:

If you want this time zone	Specify this value
Coordinated Universal Time (UTC)	+0000
5 hours west of UTC	-0500
8 hours east of UTC	+0800

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

WebSphere SYSPRINT data stream

This reference lists the configuration values that you can update in the "Configure Log Forwarder data stream" window for the **WebSphere SYSPRINT** data stream. It also describes how to use wildcard characters in the **Job Name** field for this data stream.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Job Name

The name of the server job from which to gather data. This value can contain wildcard characters.

For information about the use of wildcard characters, see ["Use of wildcard characters in the Job Name field"](#) on page 214.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see ["Subscriber configuration"](#) on page 235.

File Path

A unique identifier, such as *jobName/ddName*, that represents the data origin.

Time Zone

If the timestamp in the collected data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone, which is defined in the Log Forwarder properties, as described in ["Log Forwarder properties configuration"](#) on page 154.

The value must be in the format *plus_or_minusHHMM*, where *plus_or_minus* represents the + or - sign, *HH* represents two digits for the hour, and *MM* represents two digits for the minute.

Examples:

If you want this time zone	Specify this value
Coordinated Universal Time (UTC)	+0000
5 hours west of UTC	-0500
8 hours east of UTC	+0800

Discovery Interval

In the process of streaming data, the number of minutes that the Log Forwarder waits before it checks for a new log file in the data stream.

The value must be an integer in the range 0 - 5. A value of 0 specifies that the Log Forwarder only checks for a new log file once when the data gatherer is started. The default value is the value that is defined in the Log Forwarder properties, as described in ["Log Forwarder properties configuration"](#) on page 154.

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

Use of wildcard characters in the Job Name field

In the **Job Name** field for this data stream, you can use the following wildcard characters:

Wildcard character	What the character represents
?	Any single character
*	Any sequence of characters, including an empty sequence

If you use wildcard characters in the job name, the job name value becomes a pattern, and the data stream definition becomes a template. When the Log Forwarder starts, it searches the Job Entry Subsystem (JES) spool for job names that match the pattern, and it creates a separate data stream for each unique job name that it discovers. After the Log Forwarder initialization is complete, the Log Forwarder continues to monitor the job names on the JES spool. As it discovers new job names that match the pattern, it uses the same template to create more data streams.

For example, if the job name value is BBOS??S, and the JES spool contains the following jobs, two data streams are created, one for job name BBOSABCS and one for job name BBOSDEFS:

JOBNAME	JobID
BBODMGR	STC00586
BBODMGRS	STC00588
BBODMNC	STC00587
BBON001	STC00589
BBOSABC	STC00590
BBOSABC	STC00600
BBOSABCS	STC00592
BBOSABCS	STC00602
BBOSDEF	STC00594
BBOSDEFS	STC00596
BBOSDEFS	STC00598
HBOPROC	STC00661
SYSLOG	STC00552

Tips:

- To avoid gathering data from job logs that you do not intend to gather from, use a job name pattern that is not too broad.
- The Log Forwarder might discover jobs from other systems if spool is shared between systems or if JES multi-access spool is enabled. Although the data stream does not include data for the jobs that run on other systems, the Log Forwarder creates a data stream for that data. Therefore, ensure that the wildcard pattern does not match jobs that run on other systems.

Each resulting data stream is based on the template and has the same configuration values as the template, with the exception of the following values:

Template field	Value
Job Name	The discovered job name
Data Source Name	The value of the Data Source Name field in the template, with <code>_jobName_SYSPRINT</code> appended to that value. The <code>jobName</code> is the discovered job name.
File Path	The value of the File Path field in the template, with <code>/jobName/SYSPRINT</code> appended to that value. The <code>jobName</code> is the discovered job name.

WebSphere SYSPRINT in Distributed Format data stream

This reference lists the configuration values that you can update in the "Configure Log Forwarder data stream" window for the **WebSphere SYSPRINT in Distributed Format** data stream. It also describes how to use wildcard characters in the **Job Name** field for this data stream.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Job Name

The name of the server job from which to gather data. This value can contain wildcard characters.

For information about the use of wildcard characters, see "[Use of wildcard characters in the Job Name field](#)" on page 216.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see "[Subscriber configuration](#)" on page 235.

File Path

A unique identifier, such as `jobName/ddName`, that represents the data origin.

Time Zone

If the timestamp in the collected data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone, which is defined in the Log Forwarder properties, as described in "[Log Forwarder properties configuration](#)" on page 154.

The value must be in the format `plus_or_minusHHMM`, where `plus_or_minus` represents the + or - sign, `HH` represents two digits for the hour, and `MM` represents two digits for the minute.

Examples:

If you want this time zone	Specify this value
Coordinated Universal Time (UTC)	+0000
5 hours west of UTC	-0500
8 hours east of UTC	+0800

Discovery Interval

In the process of streaming data, the number of minutes that the Log Forwarder waits before it checks for a new log file in the data stream.

The value must be an integer in the range 0 - 5. A value of 0 specifies that the Log Forwarder only checks for a new log file once when the data gatherer is started. The default value is the value that is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration”](#) on page 154.

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

Use of wildcard characters in the Job Name field

In the **Job Name** field for this data stream, you can use the following wildcard characters:

Wildcard character	What the character represents
?	Any single character
*	Any sequence of characters, including an empty sequence

If you use wildcard characters in the job name, the job name value becomes a pattern, and the data stream definition becomes a template. When the Log Forwarder starts, it searches the Job Entry Subsystem (JES) spool for job names that match the pattern, and it creates a separate data stream for each unique job name that it discovers. After the Log Forwarder initialization is complete, the Log Forwarder continues to monitor the job names on the JES spool. As it discovers new job names that match the pattern, it uses the same template to create more data streams.

For example, if the job name value is BBOS??S, and the JES spool contains the following jobs, two data streams are created, one for job name BBOSABCS and one for job name BBOSDEFS:

JOBNAME	JobID
BBODMGR	STC00586
BBODMGRS	STC00588
BBODMNC	STC00587
BBON001	STC00589
BBOSABC	STC00590
BBOSABC	STC00600
BBOSABCS	STC00592
BBOSABCS	STC00602
BBOSDEF	STC00594
BBOSDEFS	STC00596
BBOSDEFS	STC00598

JOBNAME	JobID
HBOPROC	STC00661
SYSLOG	STC00552

Tips:

- To avoid gathering data from job logs that you do not intend to gather from, use a job name pattern that is not too broad.
- The Log Forwarder might discover jobs from other systems if spool is shared between systems or if JES multi-access spool is enabled. Although the data stream does not include data for the jobs that run on other systems, the Log Forwarder creates a data stream for that data. Therefore, ensure that the wildcard pattern does not match jobs that run on other systems.

Each resulting data stream is based on the template and has the same configuration values as the template, with the exception of the following values:

Template field	Value
Job Name	The discovered job name
Data Source Name	The value of the Data Source Name field in the template, with <code>_jobName_SYSPRINT</code> appended to that value. The <code>jobName</code> is the discovered job name.
File Path	The value of the File Path field in the template, with <code>/jobName/SYSPRINT</code> appended to that value. The <code>jobName</code> is the discovered job name.

WebSphere USS Sysprint data stream

This reference lists the configuration values that you can update in the "Configure Log Forwarder data stream" window for the **WebSphere USS Sysprint** data stream.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see [“Subscriber configuration”](#) on page 235.

File Path

A unique identifier that represents the data origin. The identifier must be the absolute path, including the file name, of a log file that contains the relevant data.

Tip: If you are gathering log data from a rolling z/OS UNIX log, see [“Data collection from a rolling z/OS UNIX log”](#) on page 225 for more information, including how to specify this file path value for a rolling log.

Time Zone

If the timestamp in the collected data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone, which is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration”](#) on page 154.

The value must be in the format *plus_or_minusHHMM*, where *plus_or_minus* represents the + or - sign, *HH* represents two digits for the hour, and *MM* represents two digits for the minute.

Examples:

If you want this time zone	Specify this value
Coordinated Universal Time (UTC)	+0000
5 hours west of UTC	-0500
8 hours east of UTC	+0800

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

WebSphere USS Sysprint in Distributed Format data stream

This reference lists the configuration values that you can update in the "**Configure Log Forwarder data stream**" window for the **WebSphere USS Sysprint in Distributed Format** data stream.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see [“Subscriber configuration”](#) on page 235.

File Path

A unique identifier that represents the data origin. The identifier must be the absolute path, including the file name, of a log file that contains the relevant data.

Tip: If you are gathering log data from a rolling z/OS UNIX log, see [“Data collection from a rolling z/OS UNIX log”](#) on page 225 for more information, including how to specify this file path value for a rolling log.

Time Zone

If the timestamp in the collected data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone, which is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration”](#) on page 154.

The value must be in the format *plus_or_minusHHMM*, where *plus_or_minus* represents the + or - sign, *HH* represents two digits for the hour, and *MM* represents two digits for the minute.

Examples:

If you want this time zone	Specify this value
Coordinated Universal Time (UTC)	+0000
5 hours west of UTC	-0500
8 hours east of UTC	+0800

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

Encoding

A value that specifies the encoding type of the log file. For all the supported encoding types, refer to the following list.

- Default platform encoding

The encoding type of the platform that runs the Log Forwarder component. The encoding type is used as the encoding type of the log file.

- UTF-8
- ISO8859-1

WebSphere Liberty Log data stream

This reference lists the configuration values that you can update in the "**Configure Log Forwarder data stream**" window for the **WebSphere Liberty Log** data stream. You can also use this data stream to collect message.log and trace.log of WebSphere Liberty. If you want to collect message.log and trace.log in one policy, you need to add two **WebSphere Liberty Log** data streams.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see [“Subscriber configuration” on page 235](#).

File Path

A unique identifier that represents the data origin. The identifier must be the absolute path, including the file name, of a log file that contains the relevant data.

Tip: If you are gathering log data from a rolling z/OS UNIX log, see [“Data collection from a rolling z/OS UNIX log” on page 225](#) for more information, including how to specify this file path value for a rolling log.

Time Zone

If the timestamp in the collected data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone, which is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration” on page 154](#).

The value must be in the format *plus_or_minusHHMM*, where *plus_or_minus* represents the + or - sign, *HH* represents two digits for the hour, and *MM* represents two digits for the minute.

Examples:

If you want this time zone	Specify this value
Coordinated Universal Time (UTC)	+0000
5 hours west of UTC	-0500
8 hours east of UTC	+0800

Discovery Interval

In the process of streaming data, the number of minutes that the Log Forwarder waits before it checks for a new log file in the data stream.

The value must be an integer in the range 0 - 5. A value of 0 specifies that the Log Forwarder only checks for a new log file once when the data gatherer is started. The default value is the value that is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration” on page 154](#).

Encoding

A value that specifies the encoding type of the log file. For all the supported encoding types, refer to the following list.

- Default platform encoding

The encoding type of the platform that runs the Log Forwarder component is used as the encoding type of the log file.

- UTF-8
- ISO8859-1

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

WebSphere Liberty Console Log data stream

This reference lists the configuration values that you can update in the **"Configure Log Forwarder data stream"** window for the **WebSphere Liberty Console Log** data stream. You can also use this data stream to collect `system.stdout` and `system.stderr` of WebSphere Liberty. If you want to collect `system.stdout` and `system.stderr` in one policy, you need to add two **WebSphere Liberty Console Log** data streams.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Job Name

The name of the server job from which to gather data. This value can contain wildcard characters.

For information about the use of wildcard characters, see [“Use of wildcard characters in the Job Name field”](#) on page 222.

DD Name

The data definition (DD) name for the job log.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see [“Subscriber configuration”](#) on page 235.

File Path

A unique identifier that represents the data origin. The identifier must be the absolute path, including the file name, of a log file that contains the relevant data.

Tip: If you are gathering log data from a rolling z/OS UNIX log, see [“Data collection from a rolling z/OS UNIX log”](#) on page 225 for more information, including how to specify this file path value for a rolling log.

Time Zone

If the timestamp in the collected data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone, which is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration”](#) on page 154.

The value must be in the format *plus_or_minusHHMM*, where *plus_or_minus* represents the + or - sign, *HH* represents two digits for the hour, and *MM* represents two digits for the minute.

Examples:

If you want this time zone	Specify this value
Coordinated Universal Time (UTC)	+0000
5 hours west of UTC	-0500
8 hours east of UTC	+0800

Discovery Interval

In the process of streaming data, the number of minutes that the Log Forwarder waits before it checks for a new log file in the data stream.

The value must be an integer in the range 0 - 5. A value of 0 specifies that the Log Forwarder only checks for a new log file once when the data gatherer is started. The default value is the value that is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration”](#) on page 154.

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

Use of wildcard characters in the Job Name field

In the **Job Name** field for this data stream, you can use the following wildcard characters:

Wildcard character	What the character represents
?	Any single character
*	Any sequence of characters, including an empty sequence

If you use wildcard characters in the job name, the job name value becomes a pattern, and the data stream definition becomes a template. When the Log Forwarder starts, it searches the Job Entry Subsystem (JES) spool for job names that match the pattern, and it creates a separate data stream for each unique job name that it discovers. After the Log Forwarder initialization is complete, the Log Forwarder continues to monitor the job names on the JES spool. As it discovers new job names that match the pattern, it uses the same template to create more data streams.

For example, if the job name value is ABCD????, and the JES spool contains the following jobs, two data streams are created, one for job name ABCD1234 and one for job name ABCDE567:

JOBNAME	JobID
ABCD1234	STC00735
DEFG1234	STC00746
ABCDE567	STC00798
DEFG5678	STC00775
ABCD123	STC00772
DEFG456	STC00794
HBODSPRO	STC00623
HBOPROC	STC00661
SYSLOG	STC00552

Tips:

- To avoid gathering data from job logs that you do not intend to gather from, use a job name pattern that is not too broad.
- The Log Forwarder might discover jobs from other systems if spool is shared between systems or if JES multi-access spool is enabled. Although the data stream does not include data for the jobs that run on other systems, the Log Forwarder creates a data stream for that data. Therefore, ensure that the wildcard pattern does not match jobs that run on other systems.

Each resulting data stream is based on the template and has the same configuration values as the template, with the exception of the following values:

Template field	Value
Job Name	The discovered job name

Template field	Value
Data Source Name	The value of the Data Source Name field in the template, with <code>_jobName_ddName</code> appended to that value. The <code>jobName</code> is the discovered job name, and the <code>ddName</code> is the DD name for the job log.
File Path	The value of the File Path field in the template, with <code>/jobName/ddName</code> appended to that value. The <code>jobName</code> is the discovered job name, and the <code>ddName</code> is the DD name for the job log.

WebSphere Liberty USS Console Log data stream

This reference lists the configuration values that you can update in the "**Configure Log Forwarder data stream**" window for the **WebSphere Liberty USS Console Log** data stream.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see "[Subscriber configuration](#)" on page 235.

File Path

A unique identifier that represents the data origin. The identifier must be the absolute path, including the file name, of a log file that contains the relevant data.

Tip: If you are gathering log data from a rolling z/OS UNIX log, see "[Data collection from a rolling z/OS UNIX log](#)" on page 225 for more information, including how to specify this file path value for a rolling log.

Time Zone

If the timestamp in the collected data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone, which is defined in the Log Forwarder properties, as described in "[Log Forwarder properties configuration](#)" on page 154.

The value must be in the format *plus_or_minusHHMM*, where *plus_or_minus* represents the + or - sign, *HH* represents two digits for the hour, and *MM* represents two digits for the minute.

Examples:

If you want this time zone	Specify this value
Coordinated Universal Time (UTC)	+0000
5 hours west of UTC	-0500
8 hours east of UTC	+0800

Discovery Interval

In the process of streaming data, the number of minutes that the Log Forwarder waits before it checks for a new log file in the data stream.

The value must be an integer in the range 0 - 5. A value of 0 specifies that the Log Forwarder only checks for a new log file once when the data gatherer is started. The default value is the value that is defined in the Log Forwarder properties, as described in [“Log Forwarder properties configuration” on page 154](#).

Encoding

A value that specifies the encoding type of the log file. For all the supported encoding types, refer to the following list.

- Default platform encoding

The encoding type of the platform that runs the Log Forwarder component is used as the encoding type of the log file.

- UTF-8
- ISO8859-1

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

z/OS SYSLOG data stream

This reference lists the configuration values that you can update in the **"Configure Log Forwarder data stream"** window for the **z/OS SYSLOG** (from user exit) and **z/OS SYSLOG from OPERLOG** data streams.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see [“Subscriber configuration” on page 235](#).

File Path

A unique identifier that represents the data origin.

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

z/OS SYSLOG from ARCHIVE data stream

This reference lists the configuration values that you can update in the "Configure data stream" window for the **z/OS SYSLOG from ARCHIVE** data stream.

Configuration values that you can update

Name

The name that uniquely identifies the data stream to the Configuration Tool. If you want to add more data streams of the same type, you must first rename the last stream that you added.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see ["Subscriber configuration"](#) on page 235.

File Path

A unique identifier that represents the data origin. For the **z/OS SYSLOG from ARCHIVE** data stream, the identifier should not be a real file path. The file path is used as a key to configure archived z/OS SYSLOG data sets to collect in the Log Forwarder batch job. For more information, see ["Creating the Log Forwarder batch job for sending SYSLOG data to the Data Streamer"](#) on page 132. The Log Forwarder link these data sets with corresponding data origin.

The default value for this field is *Data Source Type_Data Source Name*.

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is *Data Source Type_KV*. You can specify the value according to your needs.

Data collection from a rolling z/OS UNIX log

For data streams that come from z/OS UNIX log file sources, IBM Z Common Data Provider can gather log data from rolling z/OS UNIX logs. The use of a rolling log prevents any one log file from getting too large and simplifies the process of pruning older log data from the system.

Tip: The following data streams come from z/OS UNIX log file sources:

- ["Generic ZFS File data stream"](#) on page 186
- ["USS Syslogd data stream"](#) on page 207
- ["WebSphere USS Sysprint data stream"](#) on page 217

Use of a rolling log

A *rolling log* is a dynamic, sequential set of files that contains a continuous stream of log data. A new file is added whenever a previous file exceeds some threshold (for example, the file surpasses a specified size, or a specified time interval passes). Sometimes, older files are pruned (automatically or manually) so that only a defined number of files is retained.

For example, with a rolling log, a new file might be created once a day, or at specified times. The log is a set of logically grouped log files, rather than only one log file. Individual files are differentiated by an index or a timestamp in the file name.

Important: IBM Z Common Data Provider does not gather log data from a rolling log if the following events occurred when the log was rolled:

- The name of a log file was changed to a name that does not match the configured file pattern.
- The contents of a log file were removed.

File path pattern for a rolling log

IBM Z Common Data Provider uses a file path pattern with one or more wildcard characters to identify the log files that must be logically grouped into one logical log (a rolling log) and mapped to the same data source name.

You must determine the appropriate file path pattern for each set of log files that are gathered, and specify this pattern in the **File Path** field when you configure a data stream that comes from a z/OS UNIX log file source. The file path pattern must be as specific as possible so that only the appropriate log files are included.

The following wildcard characters are valid in a file path pattern (in the **File Path** field for a data stream that comes from a z/OS UNIX log file source):

Wildcard character	What the character represents
?	Any single character
*	Any sequence of characters

Example of how to specify the file path pattern: Assume that a rolling log uses the following file naming scheme, where the integer *n* is incremented for each new log file:

- /u/myLogDir/myLogFile.*n*.log

For example, *n* is 1 for the first file, 2 for the second file, and 3 for the third file.

In this example, the following file path pattern matches all of the file path names:

- /u/myLogDir/myLogFile.*.log

The following scenarios provide more examples:

- [“Sample scenario that uses date and time substitution in the JCL cataloged procedure” on page 227](#)
- [“Sample scenario that uses the redirect_server_output_dir environment variable” on page 227](#)

File path pattern utility for verifying file path values for rolling logs

IBM Z Common Data Provider includes a file path pattern utility to help you verify the file path values for any rolling logs. The utility determines which files on the current system are included by each file path pattern.

To run the utility, issue the following command in the logical partition (LPAR) where IBM Z Common Data Provider runs:

```
checkFilePattern.sh configuration_directory
```

The variable *configuration_directory* represents the directory that contains both the data configuration file and the environment configuration file.

The following example further illustrates how to issue the command and includes sample values:

```
/usr/lpp/IBM/zcdp/v2r1m0/LF/samples/checkFilepattern.sh /usr/lpp/IBM/zcdp/v2r1m0/LF
```

Optionally, a data stream identifier can be specified so that the file path for only the specified data stream is checked. The following example shows that the data stream identifier 9 is specified:

```
/usr/lpp/IBM/zcdp/v2r1m0/LF/samples/checkFilepattern.sh /usr/lpp/IBM/zcdp/v2r1m0/LF 9
```

The command response is written to standard output (STDOUT). As shown in the following example, it contains a list of all files that match each file path value:

```
INFO: HBOB021I The file path pattern
/u/myLogDir/BBOCELL.BBONODE.BBOSAPP.BBOSAPPS.???????.SR.?????.?????.SYSPRINT.txt
for data gatherer identifier 5 resolves to the following files:
/u/myLogDir/BBOCELL.BBONODE.BBOSAPP.BBOSAPPS.STC00036.SR.140929.170703.SYSPRINT.txt
/u/myLogDir/BBOCELL.BBONODE.BBOSAPP.BBOSAPPS.STC00158.SR.140929.193451.SYSPRINT.txt
/u/myLogDir/BBOCELL.BBONODE.BBOSAPP.BBOSAPPS.STC00252.SR.141006.134949.SYSPRINT.txt
INFO: HBOB021I The file path pattern
/u/myLogDir/BBOCELL.BBONODE.BBOSAPP.BBOSAPPS.???????.SR.?????.?????.SYSOUT.txt
for data gatherer identifier 7 resolves to the following files:
/u/myLogDir/BBOCELL.BBONODE.BBOSAPP.BBOSAPPS.STC00036.SR.140929.170703.SYSOUT.txt
/u/myLogDir/BBOCELL.BBONODE.BBOSAPP.BBOSAPPS.STC00158.SR.140929.193451.SYSOUT.txt
/u/myLogDir/BBOCELL.BBONODE.BBOSAPP.BBOSAPPS.STC00252.SR.141006.134949.SYSOUT.txt
```

The following example shows the command response that is written for the data stream if no files match a pattern:

```
WARNING: HBOB022W The file path pattern
/u/myLogDir/BBOCELL.BBONODE.BBOSAPP.BBOSAPPS.???????.SR.?????.?????.SYSPRINT.txt
for data gatherer identifier 6 resolves to no files.
```

Sample scenario that uses date and time substitution in the JCL cataloged procedure

Job logs can be redirected to z/OS UNIX files. They can then be rolled by using date and time substitution in the JCL cataloged procedure that is used to start the job. Each time that the job is restarted, a new file is created.

In this scenario, the following SYSOUT DD statement is from a JCL cataloged procedure is used to start a job:

```
//SYSOUT DD PATH='/u/myLogDir/myLog.&LYMMDD.&LHHMMSS..log',
//          PATHOPTS=(OWRONLY,OCREAT),PATHMODE=SIRWXU
```

The variable `&LYMMDD` is replaced by the local date on which the job was started, and the date is in `YYMMDD` format. Similarly, the variable `&LHHMMSS` is replaced by the local time in which the job was started, and the time is in `HHMMSS` format.

To convert a path with date and time variables into a file path pattern for IBM Z Common Data Provider configuration, replace the date and time variables with one or more wildcard characters.

For example, in this scenario, replace `&LYMMDD` with `??????` because the date format `YYMMDD` is always six characters. Similarly, replace `&LHHMMSS` with `??????` because the time format `HHMMSS` is always six characters.

File path pattern for this scenario

Use the following file path pattern for this scenario:

```
/u/myLogDir/myLog.?????.?????.log
```

Sample scenario that uses the `redirect_server_output_dir` environment variable

WebSphere Application Server for z/OS SYSOUT and SYSPRINT logs can also be redirected to z/OS UNIX files and rolled by using the WebSphere environment variable `redirect_server_output_dir`.

A new set of files for SYSOUT and SYSPRINT is created for each server region at the following times:

- Each time that the server job is restarted.
- Each time that the modify command is issued with the **ROLL_LOGS** parameter.

The new files are created in the directory that is specified by the `redirect_server_output_dir` environment variable.

The following file naming conventions are used for the redirected files:

```
cellName.nodeName.serverName.jobName.jobId.asType.date.time.SYSOUT.txt  
cellName.nodeName.serverName.jobName.jobId.asType.date.time.SYSPRINT.txt
```

For each server region, the cell name, node name, server name, job name, and address space type are constant. Only the job ID, date, and time are variable.

To convert one of these file naming convention into a file path pattern for IBM Z Common Data Provider configuration, complete the following steps:

1. Add the absolute path, which is specified in the WebSphere environment variable `redirect_server_output_dir`, to the beginning of the file path pattern.
2. Replace `cellName`, `nodeName`, `serverName`, and `jobName` with the appropriate values.
3. Replace `asType` with CTL (for controller), SR (for servant), or CRA (for adjunct).
4. If you are using JES2, replace `jobId` with `???????`, which matches any eight characters.

If you are using JES3, replace `jobId` with `*`, which matches any sequence of characters. In JES3, `jobId` is sometimes incorrectly populated with the job name rather than the job ID.

5. Replace `date` with `??????`, which matches any six characters.
6. Replace `time` with `??????`, which matches any six characters.

File path pattern for this scenario

The following file path pattern is an example of the pattern to use for SYSPRINT files for the BBOSAPP server that is using JES2:

```
/u/myLogDir/BB0CELL.BB0NODE.BB0SAPP.BB0SAPPS.???????.SR.??????.??????.SYSPRINT.txt
```

Data stream configuration for data gathered by System Data Engine

This reference lists the configuration values that you can update in the "**Configure System Data Engine data stream**" window.

Data Source Name

The name that uniquely identifies the data source to subscribers.

Tip: If you use the **Auto-Qualify** field in the subscriber configuration to fully qualify the data source name, this **dataSourceName** value is automatically updated with the fully qualified data source name. For more information about the values that you can select in the **Auto-Qualify** field, see "[Subscriber configuration](#)" on page 235.

Customized Data Source Type

A specification of whether to customize the data source type for Splunk HEC. The default value is No, which represents that the subscriber uses the default data source type to identify the type and format of the streamed data. If the value is set to Yes, you need to specify the data source type for Splunk HEC in the following **Data Source Type for Splunk HEC** field.

Data Source Type for Splunk HEC

A value that the subscriber can use to uniquely identify the type and format of the streamed data. This field is available only when you set the value of **Customized Data Source Type** to Yes, choose to customize the data source type and the subscriber is CDP Splunk via HEC via HTTP or CDP Splunk via HEC via HTTPS. The default value is `Data Source Type_KV`. You can specify the value according to your needs.

Transform configuration

This reference lists and describes the transforms that you can select in the "**Transform data stream**" window. For each transform, it also lists and describes the field values that you can update in the "**Configure transform**" window.

The two categories of transform are splitter transforms and filter transforms.

Splitter transforms

Based on specified criteria, a splitter transform splits data that is received as one message into multiple messages.

Transforms in this category

- [“CRLF Splitter transform” on page 229](#)
- [“FixedLength Splitter transform” on page 230](#)

Filter transforms

Based on specified criteria, a filter transform discards messages from the data stream.

Transforms in this category

- [“Regex Filter transform” on page 231](#)
- [“Time Filter transform” on page 234](#)

CRLF Splitter transform

The **CRLF Splitter** transform splits a single message in a packet into multiple messages, based on occurrences of a carriage return (CR) character, a line feed (LF) character, or any contiguous string of these two characters. The transform also considers the packet encoding as it determines whether characters in the message are carriage return or line feed characters.

The transform splits data according to the following delimiters, among others:

- CR
- LF
- CRLF
- LFCR
- CRCR
- LFLF
- CRLFCRLF
- LFCRLFCR

Configuration values that you can update

For the **CRLF Splitter** transform, you can update the following field values in the "**Configure Splitter transform**" window:

Inspect

Specifies whether, and at what stage, data packets in the data stream are to be inspected. For example, during transform processing, the data packets can be inspected by printing them to the z/OS console at the input stage, the output stage, or both stages.

You can choose any of the following values. The default value is None. To prevent the sending of large volumes of data to the z/OS console and to the IBM Z Common Data Provider Data Streamer job log, use the default value, unless you are instructed by IBM Software Support to change this value for troubleshooting purposes.

None

Specifies that data packets are not inspected.

Input

Specifies that data packets are printed to the z/OS console before they are processed by the transform.

Output

Specifies that data packets are printed to the z/OS console after they are processed by the transform.

Both

Specifies that data packets are printed to the z/OS console both before and after they are processed by the transform.

Ignore Character

Specifies a character that, if found at the beginning of a data record, causes the record to be ignored and not included in the outgoing data packet.

This field is optional and is blank by default.

FixedLength Splitter transform

The **FixedLength Splitter** transform splits data records that have a fixed record length into multiple messages, based on configuration values that you provide.

Configuration values that you can update

For the **FixedLength Splitter** transform, you can update the following field values in the "**Configure Splitter transform**" window:

Inspect

Specifies whether, and at what stage, data packets in the data stream are to be inspected. For example, during transform processing, the data packets can be inspected by printing them to the z/OS console at the input stage, the output stage, or both stages.

You can choose any of the following values. The default value is None. To prevent the sending of large volumes of data to the z/OS console and to the IBM Z Common Data Provider Data Streamer job log, use the default value, unless you are instructed by IBM Software Support to change this value for troubleshooting purposes.

None

Specifies that data packets are not inspected.

Input

Specifies that data packets are printed to the z/OS console before they are processed by the transform.

Output

Specifies that data packets are printed to the z/OS console after they are processed by the transform.

Both

Specifies that data packets are printed to the z/OS console both before and after they are processed by the transform.

Start Offset

Specifies the starting point of each data record. This value is required.

Fixed Length

Specifies the expected length of the incoming data record. This value is required.

Skip

Specifies the number of bytes from the incoming data record to skip, which means that these bytes are excluded from the output message. This value is required.

Ignore Character

Specifies a character that, if found at the beginning of a data record, causes the record to be ignored and not included in the outgoing data packet.

This field is optional and is blank by default.

Regex Filter transform

The **Regex Filter** transform filters messages in the data stream according to a regular expression (regex) pattern, which you can define. You also define the **Regex Filter** to either accept or deny incoming messages based on the regular expression. For example, if an incoming message contains the regular expression, and you define the **Regex Filter** to deny incoming messages based on the regular expression, the **Regex Filter** then discards any incoming messages that contain the regular expression.

To use the **Regex Filter** transform, you must know how to use regular expressions. The [Oracle documentation about regular expressions](#) is one source of reference information.

Important:

- The use of complex regular expressions can result in increased usage of system resources.
- Between each data stream and each of its subscribers, only one **Regex Filter** transform is supported. If you have multiple regular expressions, combine them in one **Regex Filter** transform.

Configuration values that you can update

For the **Regex Filter** transform, you can update the following field values in the "**Configure Filter transform**" window:

Inspect

Specifies whether, and at what stage, data packets in the data stream are to be inspected. For example, during transform processing, the data packets can be inspected by printing them to the z/OS console at the input stage, the output stage, or both stages.

You can choose any of the following values. The default value is None. To prevent the sending of large volumes of data to the z/OS console and to the IBM Z Common Data Provider Data Streamer job log, use the default value, unless you are instructed by IBM Software Support to change this value for troubleshooting purposes.

None

Specifies that data packets are not inspected.

Input

Specifies that data packets are printed to the z/OS console before they are processed by the transform.

Output

Specifies that data packets are printed to the z/OS console after they are processed by the transform.

Both

Specifies that data packets are printed to the z/OS console both before and after they are processed by the transform.

Regex

Specifies one or more valid regular expressions. At least one regular expression must be defined for the **Regex Filter** transform. You can also select the check box for any of the following expression flags:

Case Insensitive

Enables case-insensitive matching, in which only characters in the US-ASCII character set are matched.

To enable Unicode-aware, case-insensitive matching, select both the **Unicode Case** flag and the **Case Insensitive** flag.

Comments

Permits white space and comments in the regular expression. In this mode, white space is ignored, and any embedded comment that starts with the number sign character (#) is ignored.

Dotall

Enables dotall mode in which the "dot" expression (.) matches any character, including a line terminator.

Multi Line

Enables multiline mode in which the caret expression (^) and the dollar sign expression (\$) match immediately after, or immediately before, a line terminator or the end of the message.

Unicode Case

Enables Unicode-aware case folding.

To enable Unicode-aware, case-insensitive matching, select both the **Unicode Case** flag and the **Case Insensitive** flag.

Unix Lines

Enables UNIX lines mode in which the "dot" expression (.), the caret expression (^), and the dollar sign expression (\$) are interpreted only as the line feed (LF) line terminator.

To define one or more regular expressions in the **Regex** field, complete the following steps:

1. Type a regular expression in the **Regex** field, and optionally, select one or more check boxes to define the matching modes.
2. To add another regular expression, click **ADD REGEX**, and repeat the previous step.

Filter Type

Specifies whether the filter keeps or discards incoming messages that contain the regular expression.

You can choose either of the following values. The default value is Accept.

Accept

Specifies that any messages that contain the regular expression are kept in the data stream.

Deny

Specifies that any messages that contain the regular expression are discarded from the data stream.

Examples

The following examples show how to define a Regular Expression (Regex) to filter data.

Example 1

Regex expression

```
^\$HASP373.+SMF30.+\$
```

Filter transform configuration

Configure Filter transform ✕

Inspect: Both

Regex: `^$HASP373.+SMF30.+$`

Case Inensitive Comments Dotal Multi Line Unicode Case Unix Lines

ADD REGEX

Filter Type: Accept

OK

Figure 3. Configure regex example 1

Input data before Regex Filter

```
IEF404I BPXAS - ENDED - TIME=18.53.28
$HASP373 SMF30 STARTED - INIT A - CLASS A
$HASP373 ZWESISTC STARTED
IEF403I ZWESISTC - STARTED - TIME=18.28.34
$HASP100 ZWESISTC ON STCINRDR
$HASP395 FTPD ENDED - RC=0000
$HASP373 FTPD STARTED
IEF403I SSHD - STARTED - TIME=18.15.53
```

Output data after Regex Filter

```
$HASP373 SMF30 STARTED - INIT A - CLASS A
```

Example 2

Regex expression

```
^(\$HASP373|IEF403I).+(SMF30|ZWESISTC).+$
```

Filter transform configuration

Configure Filter transform ✕

Inspect: Both ▾

Regex: `^\($HASP373|IEF403I$)+(SMF30|ZWESISTC)$`

Case Inensitive Comments Dotall Multi Line Unicode Case Unix Lines

ADD REGEX

Filter Type: Accept ▾

OK

Figure 4. Configure regex example 2

Input data before Regex Filter

```
IEF404I BPXAS - ENDED - TIME=18.53.28
$HASP373 SMF30 STARTED - INIT A - CLASS A
$HASP373 ZWESISTC STARTED
IEF403I ZWESISTC - STARTED - TIME=18.28.34
IEF403I zwesistc initialization completed
$HASP100 ZWESISTC ON STCINRDR
$HASP395 FTPD ENDED - RC=0000
$HASP373 FTPD STARTED
IEF403I SSHD - STARTED - TIME=18.15.53
```

Output data after Regex Filter

```
$HASP373 SMF30 STARTED - INIT A - CLASS A
$HASP373 ZWESISTC STARTED
IEF403I ZWESISTC - STARTED - TIME=18.28.34
IEF403I zwesistc initialization completed
```

Time Filter transform

The **Time Filter** transform filters messages in the data stream according to a specified schedule, which you can define.

This filter discards messages that are not received within a time interval (or time window) that is defined in the schedule.

Configuration values that you can update

For the **Time Filter** transform, you can update the following field values in the "**Configure Filter transform**" window:

Inspect

Specifies whether, and at what stage, data packets in the data stream are to be inspected. For example, during transform processing, the data packets can be inspected by printing them to the z/OS console at the input stage, the output stage, or both stages.

You can choose any of the following values. The default value is None. To prevent the sending of large volumes of data to the z/OS console and to the IBM Z Common Data Provider Data Streamer job log, use the default value, unless you are instructed by IBM Software Support to change this value for troubleshooting purposes.

None

Specifies that data packets are not inspected.

Input

Specifies that data packets are printed to the z/OS console before they are processed by the transform.

Output

Specifies that data packets are printed to the z/OS console after they are processed by the transform.

Both

Specifies that data packets are printed to the z/OS console both before and after they are processed by the transform.

Schedule

To define a new schedule with one or more time intervals, complete the following steps:

1. For this field value, select **Create a new schedule**, and click **OK**.
2. In the **Edit name** field of the resulting **Schedules** window, type the name for the schedule that you want to contain this time interval.
3. To set the time interval for this schedule, either type the time information in the **From** and **to** fields, or use the slider to adjust the time.
4. To add another time interval for this schedule, click **ADD WINDOW**, and repeat the previous step.
5. To save the schedule, click **APPLY**.

For more information about how to define or update schedules in a policy, see [“SCHEDULES properties: Defining time intervals for filtering operational data”](#) on page 156.

Subscriber configuration

This reference lists the configuration values that you can update in the "**Configure subscriber**" window.

Name

The name of the subscriber.

Description

An optional description for the subscriber.

Protocol

The streaming protocol that the Data Streamer uses to send data to the subscriber.

You can choose any of the following values, which are organized under the applicable subscriber:

Logstash**IZOA on IOA-LA via Logstash**

The protocol for sending data to IBM Z Operations Analytics on IBM Operations Analytics - Log Analysis via Logstash, without encryption.

IZOA on IOA-LA via Logstash SSL

The protocol for sending data to IBM Z Operations Analytics on IBM Operations Analytics - Log Analysis via Logstash, with encryption. If you want to have secure communications between the Data Streamer and Logstash, use this value. You must also complete the relevant configuration steps that are described in [“Securing communications between the Data Streamer and its subscribers”](#) on page 96.

IZOA on Elasticsearch via Logstash

The protocol for sending data that is supported by IBM Z Operations Analytics to Elasticsearch via Logstash, without encryption.

IZOA on Elasticsearch via Logstash SSL

The protocol for sending data that is supported by IBM Z Operations Analytics to Elasticsearch via Logstash, with encryption. If you want to have secure communications between the Data Streamer and Logstash, use this value. You must also complete the relevant configuration

steps that are described in [“Securing communications between the Data Streamer and its subscribers”](#) on page 96.

CDP Elasticsearch via Logstash

The protocol for sending data to Elasticsearch via Logstash, without encryption.

CDP Elasticsearch via Logstash SSL

The protocol for sending data to Elasticsearch via Logstash, with encryption. If you want to have secure communications between the Data Streamer and Logstash, use this value. You must also complete the relevant configuration steps that are described in [“Securing communications between the Data Streamer and its subscribers”](#) on page 96.

CDP Logstash

The protocol for sending data to a Logstash subscriber, without encryption.

CDP Logstash SSL

The protocol for sending data to a Logstash subscriber, with encryption. If you want to have secure communications between the Data Streamer and Logstash, use this value. You must also complete the relevant configuration steps that are described in [“Securing communications between the Data Streamer and its subscribers”](#) on page 96.

Data Receiver

IZOA on Splunk via Data Receiver

The protocol for sending data that is supported by IBM Z Operations Analytics to Splunk via Data Receiver, without encryption.

IZOA on Splunk via Data Receiver SSL

The protocol for sending data that is supported by IBM Z Operations Analytics to Splunk via Data Receiver, with encryption. If you want to have secure communications between the Data Streamer and Data Receiver, use this value. You must also complete the relevant configuration steps that are described in [“Securing communications between the Data Streamer and its subscribers”](#) on page 96.

CDP Splunk via Data Receiver

The protocol for sending data to Splunk via Data Receiver, without encryption.

CDP Splunk via Data Receiver SSL

The protocol for sending data to Splunk via Data Receiver, with encryption. If you want to have secure communications between the Data Streamer and Data Receiver, use this value. You must also complete the relevant configuration steps that are described in [“Securing communications between the Data Streamer and its subscribers”](#) on page 96.

HTTP Event Collector (HEC) of Splunk

CDP Splunk via HEC via HTTP

The protocol for sending data to Splunk HTTP Event Collector via HTTP, without encryption.

CDP Splunk via HEC via HTTPS

The protocol for sending data to Splunk HTTP Event Collector via HTTPS, with encryption. If you want to have secure communications between the Data Streamer and Splunk, use this value. You must also complete the relevant configuration steps that are described in [“Securing communications between the Data Streamer and its subscribers”](#) on page 96.

IZOA on Splunk via HEC via HTTP

The protocol for sending data that is supported by IBM Z Operations Analytics to Splunk HTTP Event Collector via HTTP, without encryption.

IZOA on Splunk via HEC via HTTPS

The protocol for sending data that is supported by IBM Z Operations Analytics to Splunk HTTP Event Collector via HTTPS, with encryption. If you want to have secure communications between the Data Streamer and Splunk, use this value. You must also complete the relevant configuration steps that are described in [“Securing communications between the Data Streamer and its subscribers”](#) on page 96.

Important: A cdpkey file is generated in the Configuration Tool working directory to store the key for encrypting the HEC token in the policy file. The cdpkey file must be in the same directory as

the policy files, which means if you copy the policy files to other directories, you must copy the `cdpkey` file together. If the `cdpkey` file is damaged or deleted, you must restart the Configuration Tool to generate a new one. After that, you must provide token values to the Splunk subscribers in the policies again, and save the changes.

You must not send the `cdpkey` file to anyone including IBM personnel.

Generic HTTP or HTTPS subscriber

CDP Generic HTTP

The protocol for a generic HTTP subscriber, which does not provide encryption.

CDP Generic HTTPS

The protocol for a generic HTTPS subscriber, which provides encryption. You must also complete the relevant configuration steps that are described in [“Securing communications between the Data Streamer and its subscribers”](#) on page 96.

Kafka

CDP Kafka

The protocol for sending data to Kafka.

Humio

CDP Humio via HTTP

The protocol for sending data to Humio via HTTP, without encryption.

CDP Humio via HTTPS

The protocol for sending data to Humio via HTTPS, with encryption. If you want to have secure communications between the Data Streamer and Humio, use this value. You must also complete relevant configuration steps that are described in [“Securing communications between the Data Streamer and its subscribers”](#) on page 96.

Tip: For more information about preparing your target destinations to receive data from the IBM Z Common Data Provider Data Streamer, see [“Preparing the IBM Z Common Data Provider and the target destinations to stream and receive data”](#) on page 103.

Host

The host name or IP address of the subscriber.

Port

The port on which the subscriber listens for data from the Data Streamer.

URL Path

This field is available only if the subscriber is a generic HTTP or HTTPS subscriber. It specifies the path that is used to create the URL for the subscriber. For example, if the subscriber **Host** value is `logstash.myco.com`, the **Port** value is 8080, and the **URL Path** value is `/myapp/upload/data`, the following URL is created for the subscriber:

```
http://logstash.myco.com:8080/myapp/upload/data
```

Auto-Qualify

A specification of whether to prepend system names or sysplex names to data source names in the data streams that are sent to the subscriber. The data source name is the value of the **dataSourceName** field in the data stream configuration.

If you use the same policy file for multiple systems within one sysplex, the data source names must be unique across all systems in that sysplex. If you use the same policy file for multiple sysplexes, the data source names must be unique across all systems in all sysplexes. You can use this field to fully qualify these data source names.

You can choose any of the following values. The default value is System.

None

Indicates that the data source name from the **dataSourceName** field in the data stream configuration is used.

System

Specifies that the system name and the data source name are used in the following format:

```
systemName-dataSourceName
```

systemName represents the name of the system on which the IBM Z Common Data Provider runs.

If you use the same policy file for multiple systems within one sysplex, you might want to use the System value.

Sysplex

Specifies that the sysplex name, system name, and data source name are used in the following format:

```
sysplexName-systemName-dataSourceName
```

systemName represents the name of the system on which the IBM Z Common Data Provider runs.

sysplexName represents the name of the sysplex in which the IBM Z Common Data Provider runs.

If you use the same policy file for multiple sysplexes, you might want to use the Sysplex value.

For more information about the **dataSourceName** field in the data stream configuration, see the following topics:

- [“Data stream configuration for data gathered by Log Forwarder” on page 181](#)
- [“Data stream configuration for data gathered by System Data Engine” on page 228](#)

Number of threads

This configuration value is valid only when you choose one of the HEC protocols or CDP Humio protocols. The number of threads that will send data to the subscriber. The default value is 12. The value must range from 1 to 20. For Splunk HEC protocols, generally you don't need to change this value. For CDP Humio protocols, you must change it based on your environment resource. For more information about the environment resources required by Humio, see <https://docs.humio.com/docs/installation/preparation/>.

Token

This configuration value is valid when you choose one of the HEC protocols or one of the CDP Humio protocols. Specifies the token value. For more information about how to create a token value, see [“Preparing to send data to Splunk via the HTTP Event Collector” on page 107](#). For more information about how to create a Humio repository token, see [“Preparing to send data to Humio via Logstash” on page 111](#).

Bootstrap Servers

This configuration value is valid only when you choose Kafka protocol. It specifies the address of Kafka bootstrap servers. It is a comma-separated list of host and port pairs. A host and port pair use : as the separator.

Prefix

This configuration value is valid only when you choose Kafka protocol. It specifies the prefix of topic name. The default value is CDP. See [“Configuring Kafka” on page 118](#) for more information.

Customized Topic

This optional configuration value is valid only when you choose Kafka protocol. It specifies the customized topic name. See [“Configuring Kafka” on page 118](#) for more information.

Format

This configuration value is valid only when you choose Kafka protocol. It specifies the format of data that is sent to the Kafka server. The format can be CSV or Key-Value. The default value is CSV. See [“Configuring Kafka” on page 118](#) for more information.

Compression Type

A specification of whether to compress data before sending to a Humio subscriber. You can choose any of the following values. The default value is None.

None

Indicates that the data will not be compressed before sending to a Humio subscriber.

GZIP

Specifies that data will be GZIP compressed before sending to a Humio subscriber.

Language reference for System Data Engine

You can use the IBM Z Common Data Provider System Data Engine language to specify how you want the System Data Engine to collect and process data. This reference lists and describes the language elements.

In System Data Engine language statements, expressions are used to specify calculations for processing the data in data streams. Simple expressions include a single identifier, a constant, or both, and an operator, but you can also specify more complex calculations. An expression that specifies a value of `true` or `false` is called a *condition*.

Language overview

Before you can use IBM Z Common Data Provider System Data Engine language statements to create custom definitions, you must understand the concept of constants, data types, expressions, conditions, and functions.

Constants

You can specify a value explicitly by writing a string constant, an integer constant, or a floating-point constant.

String constant

A string constant is a sequence of zero or more characters enclosed within apostrophes (`'`). The sequence can contain any characters. You must add one apostrophe (`'`) before the sequence and another apostrophe (`'`) after the sequence. See the following examples:

```
'A 2'  
'a:b'  
''
```

A string constant represents the character string within the enclosing apostrophes. Therefore, the first two constants in the example represent the strings `A 2` and `a : b`. The first string contains a blank in the middle. The last example is a sequence of zero characters that represents an empty string.

A string constant might contain sequences of double-byte characters, each enclosed between shift-out (SO, `x'0E'` in EBCDIC) and shift-in (SI, `x'0F'` in EBCDIC) characters. The apostrophes are single-byte characters and are recognized outside a double-byte sequence.

The maximum length of a string that is represented by a string constant is 254 bytes, which includes any shift-out and shift-in characters that enclose sequences of double-byte characters.

Integer constant

An integer constant is a sequence of one or more digits. See the following examples:

```
-127  
0  
5  
32767  
720176  
0000000015
```

An integer constant represents a whole number in decimal notation. The number must be no greater than 2,147,483,647, and no smaller than -2,147,483,648. The maximum length of a constant is 32 characters.

Floating-point constant

A floating-point constant is a sequence of one or more digits followed by a decimal point with zero or more digits, and optionally followed by an E and a signed or unsigned number of at most 2 digits. See the following examples:

```
25.5
1000.
0.0
37589.33333
15E1
2.5E5
2.2E-1
5.E+22
```

A floating-point constant represents a 64-bit floating-point number of S/390® architecture. The number is represented in decimal notation, with E_{nn} meaning multiplied by 10 to power nn . For example, 2.5E5 means 2.5×10^5 , and 2.2E-1 means 2.2×10^{-1} . The specified value is rounded to the closest value that can be represented as a 64-bit floating-point number.

The number must not exceed 16^{63} - 16^{49} , which is approximately 7.2E75. The smallest value different from 0 is 16^{-65} , which is approximately 54.E-79. The maximum length of a constant is 32 characters.

Integer constants or floating-point constants can represent non-negative numbers only. To represent a negative number, add a minus operator (-) in front of the constant.

Comments

To explain your text, use comments to add explanations that are ignored by the System Data Engine.

Line comment

A line comment is any sequence of characters that start with a double minus sign (--) to the end of the current input line. See the following examples:

```
-- This is a line comment.
-- Another line comment. Notice that it may contain unpaired ' and ".
```

The comment can contain sequences of double-byte characters that are enclosed between shift-out (SO, x'0E' in EBCDIC) and shift-in (SI, x'0F' in EBCDIC) characters. The line must end in a single-byte sequence to end the line comment. If the line ends in a double-byte sequence, the next line is interpreted as starting in the single-byte mode, which usually causes an error.

Block comment

A block comment is any sequence of characters that start with slash asterisk (/*) to the nearest asterisk slash (*). See the following example:

```
/* This is a block comment.
Notice that it can extend over several lines.
It can contain -- and unpaired ' or " */
```

The comment can contain sequences of double-byte characters that are enclosed between shift-out (SO, x'0E' in EBCDIC) and shift-in (SI, x'0F' in EBCDIC) characters. The asterisk and slash that end the comment are single-byte characters and is only recognized outside a double-byte sequence.

Statements

The input in the System Data Engine is a sequence of statements. The statements must be separated by semicolons (;). The semicolons are not considered a part of the statement and are not shown in syntax diagrams.

Data types

The main task of the System Data Engine is to process data. The smallest unit of data is called a value. There are different types of values that can be obtained from a field of a record, stated in your definition, or computed from other values.

The System Data Engine can handle the following types of values:

- Integer numbers
- Floating-point numbers
- Character strings
- Dates
- Times
- Timestamps

The integer numbers and floating-point numbers are called numbers, or numeric values. The dates, times, and timestamps are called date and time values.

Integer numbers

An integer is a number in the range -2,147,483,648 to 2,147,483,647. For more information, see [“Integer constant” on page 239](#).

Floating-point numbers

A floating-point number is a number that can be represented as a 64-bit floating-point number of S/390 architecture. For more information, see [“Floating-point constant” on page 240](#).

Dates

A date value represents a day according to the Gregorian calendar. This value consists of three parts for day, month, and year. The range of the year part is 1 - 9,999. The range of the month part is 1 - 12. The range of the day part is 1 to x, where x depends on the month. All dates are calculated under the condition that the Gregorian calendar was in effect since year 0001.

Times

A time value represents a time of day under a 24-hour clock. This value consists of four parts for hour, minute, second, and microsecond. The range of the hour part is 0 - 24, the range of the minute and second part is 0 - 59, and the range of the microsecond part is 0 - 999,999. If the hour is 24, the other parts must be 0.

Timestamps

A timestamp value represents a day and a time of that day. This value consists of seven parts for year, month, day, hour, minute, second, and microsecond. The year, month, and day parts represent the day as specified under [“Dates” on page 241](#). The hour, minute, second, and microsecond parts represent the time as specified under [“Times” on page 241](#).

Date and time strings

Date and time strings are character strings of a specific format, and are used to write specific date and time values.

To write specific date and time values, you must write expressions explicitly. The following expressions are specific cases of function calls with date and time strings.

```
DATE('2000-06-27')
TIME('10.32.55.123456')
TIMESTAMP('2000-06-27-10.32.55.123456')
```

The character strings '2000-06-27', '10.32.55.123456', and '2000-06-27-10.32.55.123456' are date and time strings.

Date string

Is a character string that represents a date in the format yyyy-mm-dd where yyyy is the year, mm is the month, and dd is the day.

The DATE function converts a date string to a date. The expression like DATE(*date_string*) specifies the result of such conversion. For example, the expression DATE('2000-06-27') specifies the date June 27, 2000.

Time string

Is a character string that represents a time in the format hh.mm.ss.uuuuuu, where hh is the hour, mm is the minute, ss is the second, and uuuuuu is the microsecond.

The TIME function converts a time string to a time. The expression like TIME(*time_string*) specifies the result of such conversion. For example, the expression TIME('10.32.55.123456') specifies the time 10 hours 32 minutes 55.123456 seconds.

Timestamp string

Is a character string that represents a timestamp in the format yyyy-mm-dd-hh.mm.ss.uuuuuu where yyyy, mm, dd, hh, mm, ss, and uuuuuu are as above.

The TIMESTAMP function converts a timestamp string to a timestamp. The expression like TIMESTAMP(*timestamp_string*) specifies the result of such conversion. For example, the expression TIMESTAMP('2000-06-27-10.32.55.123456') specifies the timestamp 10 hours 32 minutes 55.123456 seconds on June 27, 2000.

In some cases you can use a date and time string instead of a date and time value, and the System Date Engine converts the string for you. For example, if CREATION_DATE specifies a date, you can use CREATION_DATE < '2000-06-25'. The System Data Engine converts the date string to a date value and compares the result with the date that is specified by CREATION_DATE.

Operators

You can specify values by using arithmetic operations on numbers, comparisons, and logical operations. These operations are specified by an infix operator (which is an operator between operands), or by a prefix operator (which is an operator in front of operands).

Arithmetic operations

You can apply the prefix operator plus (+) or minus (-) to any numeric value. See the following example:

```
-DOWN_TIME  
+40  
-23.456  
-1E8
```

The result is of the same type as the operand. The prefix plus (+) does not change its operand. The prefix minus (-) reverses the sign of its operand.

You can apply the infix operator plus (+), minus (-), multiply (*), and divide (/) between any pair of numeric values. See the following example:

```
A+B  
N_DATASETS-5  
COUNT*1E-6  
RESP_TIME/60
```

The result depends on the operand types:

- If both operands are integers, the result is an integer. The operation is performed using integer arithmetic. The division is performed so that the remainder has the same sign as the dividend.
- If both operands are floating-point numbers, the result is a floating-point number. The operation is performed using long floating-point operations of S/390.

- If one of the operands is an integer and the other a floating-point number, the integer is converted to a floating-point number. The operation is then performed on the result of the conversion, using floating-point arithmetic. The result is a floating-point number.

The result of dividing an integer by another integer is also an integer. For example, if `RESP_TIME` is an integer less than 60, the result of `RESP_TIME/60` is 0. If you want the exact result, write `RESP_TIME/60.0` instead, which makes the operand on the right the floating-point number 60.0. Then the operand on the left, which is `RESP_TIME`, is converted to a floating-point number, making the result a floating-point number.

For all operators, the result is null if any of the operands is null. If the result is an integer, the result must be within the range of integers. If the result is a floating-point number, the result must be within the range of floating-point numbers. The operand on the right of a divide operator must not be 0.

Comparisons

You can compare two values by using the infix operator equal (`=`), not equal (`<>`), greater than (`>`), less than (`<`), greater than or equal (`>=`), less than or equal (`<=`). The result is a value of true or false. If one of the values in the comparison is null, the result is unknown. See the following example:

```
A>10
JOB_NAME<'ABC'
DATE<>'1993.04-15'
```

Only the following types of comparisons are allowed:

- Numbers with numbers
- Character strings with character strings or date and time values
- Date and time values with character strings or date and time values of the same type

Numbers are compared by their algebraic values. If both numbers are floating-point numbers, they are compared by using long floating-point operation of S/390. Two floating-point numbers are considered equal only if their normalized forms have identical bit configurations.

If one of the numbers is an integer number and the other a floating-point number, the integer number is converted to a floating-point number. The comparison is then performed with the result of the conversion.

Character strings are compared byte by byte, left to right. If the strings are different in length, the comparison is made with a temporary copy of the shorter string that is extended on the right with the necessary number of blanks so that it has the same length as the other string.

Two strings are equal if they are both empty or if all corresponding bytes are equal. Otherwise, their relation is determined by the comparison of the first unequal pair of bytes.

When a character string is compared with a date and time value, it must be a valid date and time string of the corresponding kind. The string is converted to a date and time value and the comparison is performed on the result.

All comparisons of date and time values are chronological. The value that represents the later point of time is considered to be greater.

Because the hour part may range from 0 to 24, certain pairs of different timestamps represent the same time. When such timestamps are compared, the one with a greater date part is considered greater. For example, the result of this comparison is true:

```
TIMESTAMP('1985-02-23-00.00.00.000000')>
TIMESTAMP('1985-02-22-24.00.00.000000')
```

However, the `INTERVAL` calculates the interval between these timestamps as 10.

Logical operations

You can apply the prefix operator `NOT` to any value of true or false. The following table shows the result that is defined for operand `p`:

Table 34. Logical operation NOT

p	NOT p
True	False
False	True
Unknown	Unknown

You can apply the infix operator AND and OR to any pair values of true or false. The following table shows the result that is defined for operand p and q:

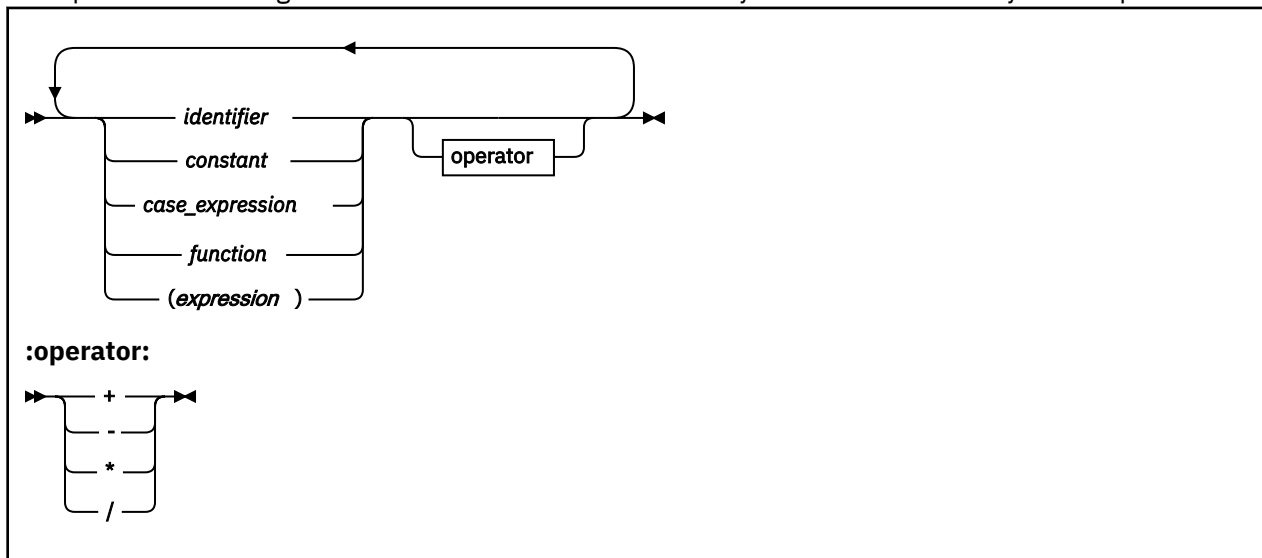
Table 35. Logical operations AND and OR

p	q	p AND q	p OR q
True	True	True	True
True	False	False	True
True	Unknown	Unknown	True
False	True	False	True
False	False	False	False
False	Unknown	False	Unknown
Unknown	True	Unknown	True
Unknown	False	False	Unknown
Unknown	Unknown	Unknown	Unknown

Expressions

In IBM Z Common Data Provider System Data Engine language statements, expressions are used to specify calculations for processing the data.

The following syntax shows the general form of expression that you can use wherever the syntax specifies an expression. The diagram does not reflect all the rules that you must follow when you use operators.



identifier

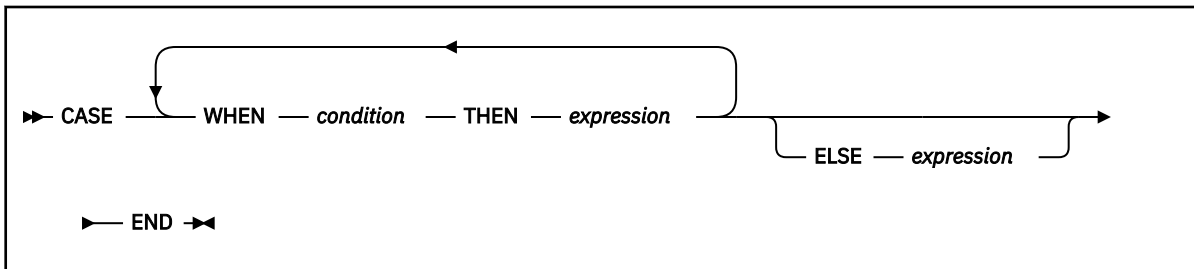
Specifies the name of a value, or the name of something that holds a value. In general, identifiers are used as names of logs, records, and fields within records. An identifier must not exceed 18 bytes.

constant

Specifies a value explicitly. You can specify a value by writing an integer constant, a floating-point constant, or a string constant. For more information, see [“Constants” on page 239](#).

case_expression

Is a case expression that specifies a value that is selected by testing one or more conditions. It has the following syntax.

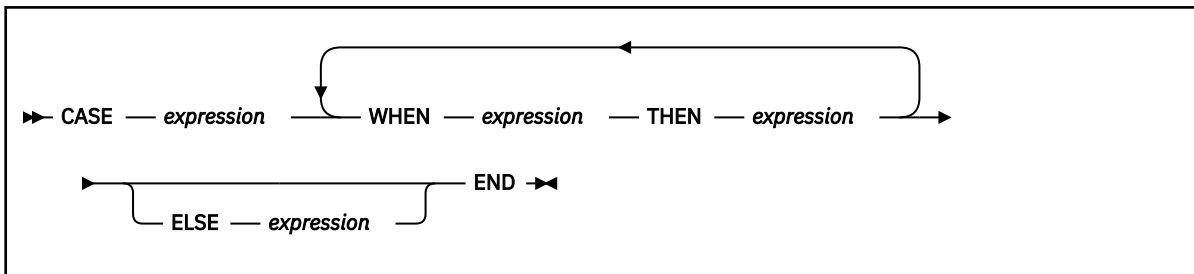


WHEN condition THEN expression

Defines one of the possible cases. The case is applicable if the value of *condition* is true. The result of the case expression is equal to the result of the expression in the applicable case. If several cases are applicable, the result is defined by the first one. If none of the cases is applicable, the result is defined by the ELSE clause. The expressions in all case expressions must specify values of the same type.

ELSE expression

Defines the result of the case expression if no case is applicable. If the ELSE clause is not specified, the result is null. The expression in the ELSE clause must specify a value of the same type as expressions in the case expressions.



CASE expression

The expression is evaluated and the result is compared with results of expressions in the WHEN clauses.

WHEN expression THEN expression

Defines one of the possible cases. The case is applicable if the result of *WHEN expression* is equal to the result of *CASE expression*. The result of the case expression is equal to the result of *THEN expression* in the applicable case. If several cases are applicable, the result is defined by the first one. If none of the cases is applicable, the result is defined by the ELSE clause.

All expressions in the WHEN clause must specify values of the same type as the expression in the CASE clause.

All expressions in the THEN clause must specify values of the same type.

ELSE expression

Defines the result of the case expression if no case is applicable. If the ELSE clause is not specified, the result is null. The expression in the ELSE clause must specify a value of the same type as expressions in the THEN clause.

function

Is a function call. It specifies a value as the result of a function. For more information, see [“Functions” on page 247](#).

(expression)

Specifies the value of expression. You can combine the expressions by using operators and parenthesis.

Operator

Are arithmetic operators. For more information about how to use each operator, see [“Arithmetic operations”](#) on page 242.

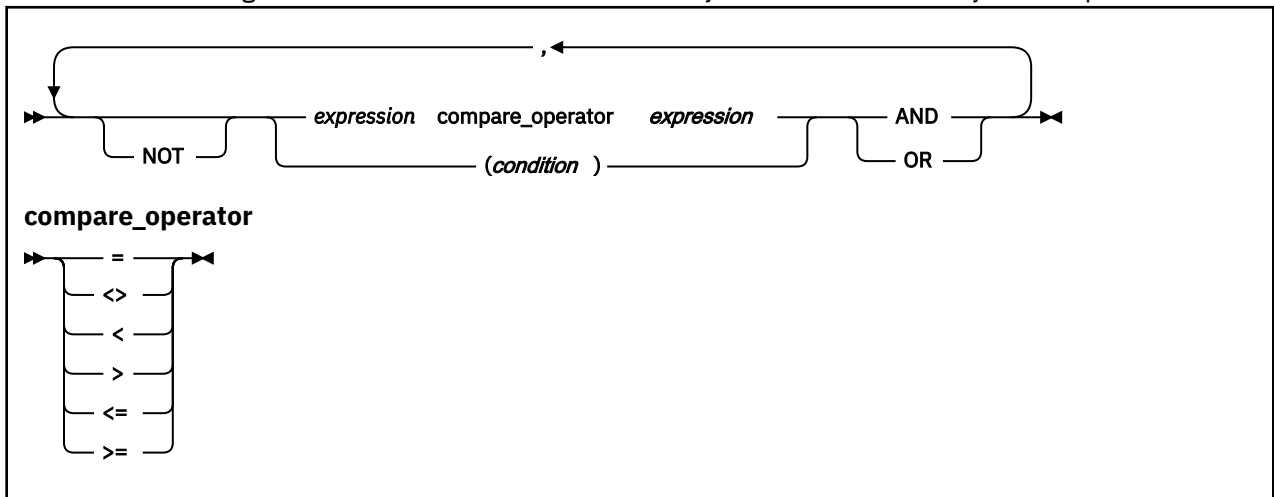
The result of an expression can be one of the following types and it must match the context.

- Integer numbers
- Floating-point numbers
- Character strings
- Dates
- Times
- Timestamps

Conditions

In IBM Z Common Data Provider System Data Engine language statements, an expression that specifies a value of true or false is called a *condition*.

The following syntax shows the general form of expression that you can use wherever the syntax specifies a condition. The diagram does not reflect all the rules that you must follow when you use operators.



expression compare_operator expression

Is a comparison between expressions. The result is a value of true or false. If one of the compared values is null, the result is unknown.

compare operator

Includes equal (`=`), not equal (`<>`), greater than (`>`), less than (`<`), greater than or equal (`>=`), and less than or equal (`<=`). For more information, see [“Comparisons”](#) on page 243.

AND and OR

Specifies the logical relationship between two expressions.

(condition)

You can combine the conditions by using operators and parenthesis. See the following example:

```
(SMF30XXX <> 3 AND SMF30YYY = 'A 2') OR SMF30ZZZ < 2.2E-1
```

Functions

A function call is a special form of expression. You can use a function call directly in the statements whenever the syntax specifies an expression. You can also use it as a part of more complex expressions.

CHAR

The CHAR function obtains a string representation of a date and time value. See the following syntax:

►► CHAR — (— *expression* —) ►◄

The argument *expression* must be a date, a time, or a timestamp. For more information about the date, time, and timestamp values, see [“Data types” on page 241](#).

Assume the following conditions:

- X_DATE has the value May 3, 2000.
- X_TIME has the value 5 hours, 17 minutes, and 34 seconds.
- X_TSTAMP has the value 5 hours, 17 minutes, and 34 seconds on May 3, 2000.

The CHAR function produces the following results:

```
CHAR(X_DATE) = '2000-05-03'  
CHAR(X_TIME) = '05.17.34.000000'  
CHAR(X_TSTAMP) = '2000-05-03-05.17.34.000000'
```

DATE

The DATE function obtains a date from a value.

►► DATE — (— *expression* —) ►◄

The argument *expression* must be a date, a timestamp, a number, or a date string.

The result of this function is a date.

- If the argument is a date, the result is that date.
- If the argument is a timestamp, the result is the date part of that timestamp.
- If the argument is a number, consider the integer part of that number as n. It must be in the range 1 - 3,652,059. The result of the function is the date of the day with sequential number n, counting from January 1, 0001 as day 1.
- If the argument is a date string, the result is the date that is represented by that string.

Assume the following conditions:

- X_DATE has the value April 22, 1993.
- X_TSTAMP has the value 15 hours, 2 minutes, and 1 second on March 6, 1993.
- X_STRING has the value '2000-03-06'.

The DATE function produces the following results:

```
DATE(X_DATE) = April 22, 1993  
DATE(X_TSTAMP) = March 6, 1993  
DATE(727159) = November 23, 1991  
DATE('2000-06-15') = June 15, 2000  
DATE(X_STRING) = March 6, 2000
```

DIGITS

The DIGITS function obtains a character string representation of a number.

►► DIGITS — (— *expression* —) ►◄

The argument *expression* must be an integer.

The result is the string of digits that represents the absolute value of the argument. Leading zeros are not included in the result.

```
DIGITS(754) = '754'  
DIGITS(00054) = '54'  
DIGITS(-54) = '54'
```

INTEGER

The INTEGER function obtains the integer part of a number.

►► INTEGER — (— *expression* —) —►

The argument must be a number.

If the argument is an integer, the result is that integer. If the argument is a floating-point number, the result is the integer part of that number.

```
INTEGER(45) = 45  
INTEGER(-75.3) = -75  
INTEGER(0.0005) = 0
```

INTERVAL

The INTERVAL function obtains the length of a time interval in seconds.

►► INTERVAL — (— *expression* — , — *expression* —) —►

Both arguments must be date and time values of the same type.

The result is a floating-point number. The result is the interval, in seconds, from the instant designated by the first argument to the instant designated by the second argument. If the first argument is later than the second, the result is negative. The result has the maximum precision that is allowed by its floating-point representation. Therefore, results up to 2283 years have a precision of 1 microsecond. Assume the following conditions:

- TME1 has the value of 6 hours, 20 minutes, 29 seconds, and 25000 microseconds.
- TME2 has the value of 18 hours, 25 minutes, 20 seconds.
- DAY1 has the value of March 5, 1993.
- DAY2 has the value of March 8, 1993.
- TS1 has the value of 5 hours on March 5, 1993.
- TS2 has the value of 10 hours, 30 minutes on March 11, 1993.

The INTERVAL function produces the following results:

```
INTERVAL(TME1, TME2) = 43490.975  
INTERVAL(TME2, TME1) = -43490.975  
INTERVAL(DAY1, DAY2) = 259200.0  
INTERVAL(TS1, TS2) = 538200.0
```

LENGTH

The LENGTH function obtains the length of a character string.

►► LENGTH — (— *expression* —) —►

The argument must be a character string.

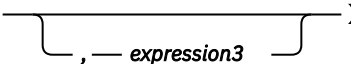
The result of the LENGTH function is an integer. Assume that X_STRING has the value of 'LOG_NAME', the function produces the following results:

```
LENGTH(X_STRING) = 8  
LENGTH('REC_LOG') = 7
```

```
LENGTH(' ') = 1
LENGTH('') = 0
```

SUBSTR

The SUBSTR function obtains a substring of a character or bit string.

►► SUBSTR — (— *expression1* — , — *expression2* — ) ►►

The *expression1* is called string, and must be a character or bit string. The *expression2* is called start, and must be an integer in the range 1 - 254. The *expression3* is called length, and must be an integer in the range 0 to 255-*expression2*.

The result is a character string. If length is specified, the result consists of length bytes of string, starting at the start position. The string is regarded as extended on the right with the necessary number of blanks so that the specified substring exists.

If length is not specified, the result consists of all bytes or bits of string, starting at the position start and extending to the end of string. If start is greater than the length of string, the result is an empty string.

Both start and length are expressed in bytes. The SUBSTR function does not recognize double-byte characters, and the result is not necessarily a well-formed character string.

Assume that C_STR is a character string with the value of 'SUB_REC'. The function produces these results:

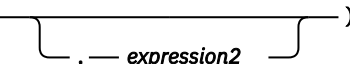
```
SUBSTR(C_STR,3,2) = 'B_'
SUBSTR(C_STR,3) = 'B_REC'
SUBSTR(C_STR,3,10) = 'B_REC'
```

Assume that B_STR is a 1-byte bit string with the value of '01010111'. The function produces these results:

```
SUBSTR(B_STR,3,2) = '01'
SUBSTR(B_STR,3) = '010111'
SUBSTR(B_STR,3,10) = '010111'
```

TIMESTAMP

The TIMESTAMP function obtains a timestamp from a value or a pair of values.

►► TIMESTAMP — (— *expression1* — ) ►►

The result of the function depends on whether *expression1* or *expression2* are specified.

If only one argument is specified, *expression1* must be a timestamp or a timestamp string. The result is a timestamp:

- If *expression1* is a timestamp, the result is that timestamp.
- If *expression1* is a timestamp string, the result is the timestamp represented by that string.

If both arguments are specified, *expression1* must be a date or a date string, and *expression2* must be a time or a time string. The result is a timestamp. It consists of the date and time that is specified by the arguments. Assume the following conditions:

- X_TIME has the value 3 hours, 24 minutes, 20 seconds, and 2 microseconds.
- X_DATE has the value February 11, 1993.
- X_TSTAMP has the value 15 hours, 33 minutes, 25 seconds, and 75 microseconds on June 20, 1993.

The function produces the following results:

```
TIMESTAMP(X_TSTAMP) = 15 hours, 33 minutes, 25 seconds, and 75 microseconds on
```

```

June 20, 1993
TIMESTAMP('1993-04-17-19.01.25.000000') = 19 hours, 1 minute, 25 seconds on
April 17, 1993
TIMESTAMP(X_DATE, X_TIME) = 3 hours, 24 minutes, 20 seconds, and 2 microseconds
on February 11, 1993

```

VALUE

The VALUE function returns the first argument that is not null.

►► VALUE — (— *expression* — , — *expression* — , — *expression* —) ►►

All arguments must have the same data type.

The result has the same data type as the arguments. It is equal to the first argument that is not null. If all arguments are null, the result is null.

Assume the following conditions:

- EXPA has the value of 25.
- EXPB has the value of 50.
- EXPC has a null value.

The function produces the following results:

```

VALUE(EXPA, EXPB, EXPC) = 25
VALUE(EXPC, EXPB, EXPA) = 50
VALUE(EXPB, EXPA) = 50

```

DEFINE RECORD statement

The DEFINE RECORD statement defines a new record type, which is used to create a custom data stream.

- [“Syntax” on page 250](#)
- [“Parameters” on page 251](#)
- [“SECTION clause” on page 251](#)
- [“FIELDS clause” on page 253](#)
- [“Examples” on page 257](#)

Syntax

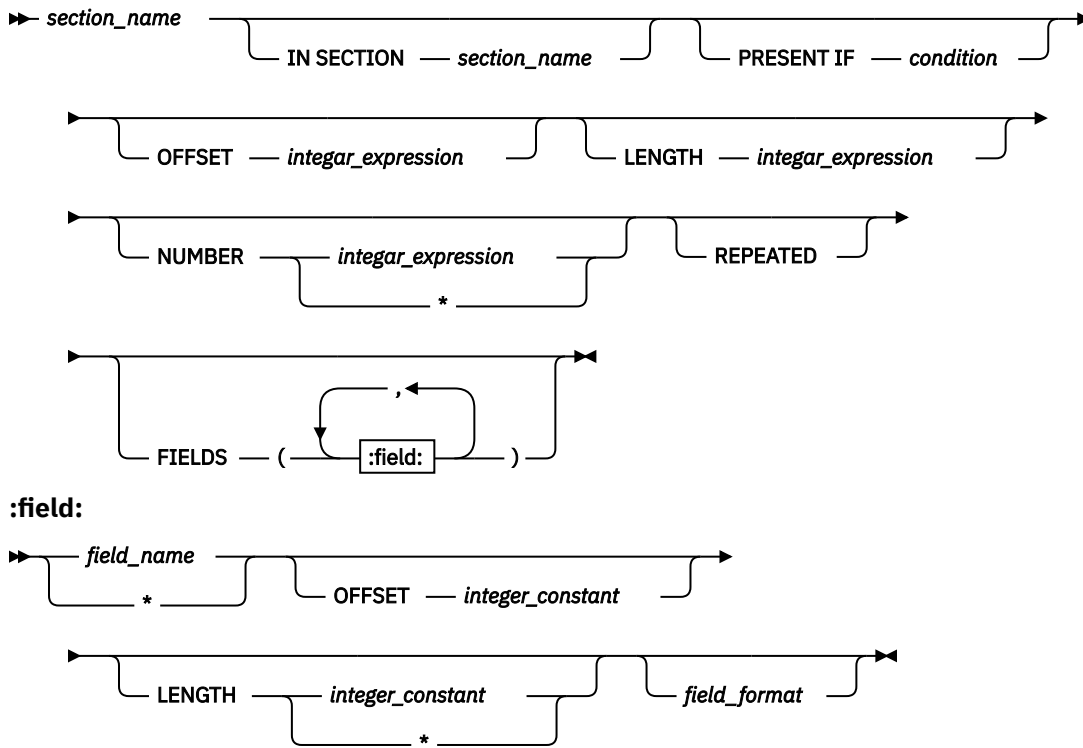
DEFINE RECORD statement

►► DEFINE RECORD — *record_name* — *VERSION — version* — IN LOG SMF — ►►

► IDENTIFIED BY — *condition* — *FIELDS — (— :field: —)* — ►

► *SECTION — :section: —* ►►

:section:



Parameters

DEFINE RECORD *record_name*

Specifies the name of the record type that you are defining to the System Data Engine. The record name must be unique, and cannot both start and end with an asterisk (*). Verify that in the SHBODEFS data set that is used as the CDP concatenation library, and in the data set that is used as the USER concatenation library, no existing definition has the same name.

VERSION *version*

Specifies the version for the definition. The maximum length for this value is 18 characters. You might want to specify this optional value for troubleshooting purposes.

IDENTIFIED BY

Specifies how records of this type are distinguished from each other. If a record meets the condition *condition*, it's identified as the type *record_name*. A specific record might meet the condition of several record definitions. In this case, only one of these definitions (undefined which one) is used by the System Data Engine for processing this record.

condition

Specifies the condition according to which the records are identified. For more information, see [“Conditions” on page 246](#).

Any identifier that is used in the *condition* must be a field name that is defined directly in the record, not in the sections within the record.

SECTION clause and FIELDS clause

Define the sections and fields with options. For more information, see [“SECTION clause” on page 251](#) and [“FIELDS clause” on page 253](#).

SECTION clause

SECTION *section* defines one section. A record can have a maximum of 300 sections.

section_name

Specifies the name of the section. Section names must be unique within a record type.

IN SECTION *section_name*

Specifies that the section that is being defined is a subsection of the section named *section_name*. If you omit the IN SECTION clause, the section is a section of the record. The containing section *section_name* must be an existing section in this record definition.

PRESENT IF

Specifies that the section is optional. The section is absent if the *condition* isn't true. If the *condition* is true while the containing section or record is too short to contain the first byte of the section, the section is also absent.

condition

Specifies the condition for sections to present. For more information, see [“Conditions” on page 246](#).

Any identifier that is used in the *condition* must be a field name in one of the following areas:

- The section that is being defined
- The containing section
- The record
- The previously defined non-repeated subsections

OFFSET

Specifies the offset of the section within the containing section or record.

integer_expression

Specifies an integer value that is defined by an expression. The expression result must be an integer value such as an integer constant or a field that holds an integer value. The value must be no smaller than 0. For more information, see [“Expressions” on page 244](#).

Any identifier that is used in the expression must be a field name in one of the following areas:

- The containing section
- The record
- The previously defined non-repeated subsections

If you omit OFFSET, the section starts at the end of the most recently defined section with the same IN SECTION attribute. That section can't be a repeated section. If no section with the same IN SECTION attribute has been previously defined, an omitted OFFSET means offset 0.

LENGTH

Specifies the length of the section.

integer_expression

Specifies an integer value that is defined by an expression. The expression result must be an integer value such as an integer constant or a field that holds an integer value. The value must be no smaller than 0. For more information, see [“Expressions” on page 244](#).

Any identifier that is used in the expression must be a field name in one of the following areas:

- The section that is being defined
- The containing section
- The record
- The previously defined non-repeated subsections

If you omit LENGTH, the System Data Engine takes the minimum length needed to contain all named fields specified for this section.

If the containing section or record is too short to contain the whole section, the System Data Engine assumes that the section extends up to the end of the containing section or record. If the containing section or record is too short to contain the first byte of the section, the section is absent.

NUMBER

Specifies the number of occurrences of the section.

integer_expression

Specifies an integer value that is defined by an expression. The expression result must be an integer value such as an integer constant or a field that holds an integer value. The value must be no smaller than 0. For more information, see [“Expressions” on page 244](#).

Any identifier that is used in the expression must be a field name in one of the following areas:

- The containing section
- The record
- The previously defined non-repeated subsections

Specifies that the number of occurrences of the section is as many occurrences as the containing section or record can contain.

If you omit NUMBER clause, it has the same behavior as NUMBER 1.

REPEATED

Specifies that the section can occur more than once. If you omit REPEATED, the section can occur only once.

Tip: If the keyword REPEATED is not included, even the value of NUMBER is larger than 1, the section can occur only once.

FIELDS clause

Specifies fields of the record or section. A record, including the fields inside and outside its sections, can have a maximum of 2000 fields. Use this clause to specify one or multiple fields. Multiple *field* entries are separated by commas.

This general rule applies to all fields. The LENGTH and OFFSET (explicit or default) define a field as an area and its length, starting at a specific place in the record or section. If the record or section is too short to contain all bytes of a field, the field is absent and a reference to it produces null value.

However, if you specify LENGTH *, which means the field extends up to the end of the record or section, the previous rule doesn't apply. The field is absent if the record or section is too short to contain the first byte of the field.

field_name

Specifies the name of the field. Field names must be unique within a record type.

OFFSET *integer_constant*

Defines the offset, in bytes, of the field in the record or section.

If you omit OFFSET, the field starts at the end of the field defined just before it. The preceding field can't have LENGTH *. If you omit the OFFSET for the first field in the list, that field begins at offset 0.

LENGTH

Specifies the length of the field.

LENGTH *integer_constant*

Specifies the length of the field in bytes. The allowed lengths depend on the format of the field. See the Length in bytes column in table [Table 36 on page 254](#) for more information.

If you omit LENGTH, the System Data Engine uses the default length depending on the field format. See the Length in bytes column in table [Table 36 on page 254](#) for more information. If there is only one value in the column, it's the default.

LENGTH *

Specifies that the field extends up to the end of the containing record or section.

field_format

Specifies the format of the data in the field. The possible values of *field_format* are listed in [Table 36 on page 254](#), in the Field format column. The Data type column states the data type to which the System Data Engine automatically converts.

If you omit the *field_format*, the field format is HEX.

<i>Table 36. Field formats for the FIELDS clause of the DEFINE RECORD statement</i>			
Field format	Contents	Length in bytes	Data type
BINARY BINARY SIGNED BINARY UNSIGNED	Binary integer represented according to System/390® architecture. The default is SIGNED for lengths 2, 4, and 8, and UNSIGNED for lengths 1 and 3.	1, 2, 3, 4 (default), 8	Integer for SIGNED of length ≤ 4 and UNSIGNED of length ≤ 3; otherwise floating-point
EXTERNAL HEX	A string of bits representing an integer in hexadecimal characters.	2, 4, 8 (default)	String
DECIMAL(<i>p</i> , <i>s</i>) where $1 \leq p \leq 31$ and $0 \leq s \leq p$	Packed decimal number of System/390 architecture, with precision <i>p</i> and scale <i>s</i> . The precision is the total number of decimal digits. Odd <i>p</i> means a signed number; even <i>p</i> means an unsigned number. The scale is the number of digits after the decimal point.	Integer part of (<i>p</i> + 1)/2	Integer if <i>s</i> = 0 and <i>p</i> ≤ 9; otherwise floating-point
ZONED(<i>p</i> , <i>s</i>) where $1 \leq p \leq 31$, and $0 \leq s \leq p$	Unsigned zoned decimal number of System/390 architecture, with precision <i>p</i> and scale <i>s</i> . The precision is the total number of decimal digits. The scale is the number of digits after the decimal point.	<i>p</i>	Integer if <i>s</i> = 0 and <i>p</i> ≤ 9; otherwise floating-point
FLOAT	A floating-point number of System/390 architecture, short (4 bytes) or long (8 bytes).	4, 8 (default)	Floating-point
EXTERNAL FLOAT	A string of characters expressing a floating-point number in the same format used for floating-point constants. Leading and trailing blanks are allowed.	1 to 32 while the default value is 8	Floating-point
CHAR	A string of characters. May include sequences of double-byte characters, enclosed between shift-out and shift-in characters.	1 to 254 while the default value is 1	String
CHAR(<i>n</i>) where $1 \leq n \leq 254$	A string of characters occupying <i>n</i> bytes. May include sequences of double-byte characters, enclosed between shift-out and shift-in characters.	<i>n</i>	String
CHAR(*)	A string of characters, extending up to the end of the containing structure. If the string is longer than 254 bytes, it's truncated. This format is only allowed with LENGTH *.	* (1 to 254)	String

Table 36. Field formats for the *FIELDS* clause of the *DEFINE RECORD* statement (continued)

Field format	Contents	Length in bytes	Data type
VARCHAR	A string of characters including length information. The first two bytes contain the length <i>l</i> of the data as a binary integer; the remaining bytes contain the data itself. The length <i>l</i> may be 0, and can't exceed the length of the field minus 2. The data portion of the string may include sequences of double-byte characters, enclosed between shift-out and shift-in characters.	3 to 256 while the default value is 8	String
BIT	A string of bits. Converted to string of characters "0" and "1" and "1" representing individual bits.	1 to 31 while the default value is 1	String
BIT(<i>n</i>) where $8 \leq n \leq 248$, <i>n</i> multiple of 8	A string of <i>n</i> bits. Converted to string of characters "0" and "1" representing individual bits.	<i>n</i> /8	String
HEX	A string of bits. Converted to string of characters "0" through "F" representing the string in hexadecimal notation.	1 to 127 while the default value is 1	String
DATE(0CYDDDF)	Date in the format <i>0cyydddF</i> (packed), where <i>c</i> indicates the century (0=1900, 1=2000), <i>yy</i> is the year within the century, <i>ddd</i> is the day within the year, and <i>F</i> can have any value. (<i>F</i> is ignored and isn't checked to be a valid decimal sign).	4	Date
DATE(YYYYDDDF)	Date in the format <i>yyyydddF</i> (packed), where <i>yyyy</i> is the year, <i>ddd</i> is the day within the year, and <i>F</i> can have any value. (<i>F</i> is ignored and isn't checked to be a valid decimal sign).	4	Date
DATE(YDDDF)	Date in the format <i>yydddF</i> (packed), where <i>yy</i> is the year, <i>ddd</i> is the day within the year, and <i>F</i> can have any value. (<i>F</i> is ignored and isn't checked to be a valid decimal sign).	3	Date

Table 36. Field formats for the FIELDS clause of the DEFINE RECORD statement (continued)

Field format	Contents	Length in bytes	Data type
DATE(CYYMMDDF)	Date in the format <i>cyyymmddF</i> (packed), where <i>c</i> indicates the century (0=1900, 1=2000), <i>yy</i> is the year within the century, <i>mm</i> is the month, <i>dd</i> is the day of month, and <i>F</i> can have any value. (<i>F</i> is ignored and isn't checked to be a valid decimal sign).	4	Date
DATE(YYMMDD)	Date as character string <i>yyymmdd</i> , where <i>yy</i> is the year, <i>mm</i> is the month, and <i>dd</i> is the day. <i>yy</i> ≥ 50 means year 19 <i>yy</i> ; <i>yy</i> < 50 means year 20 <i>yy</i> .	6	Date
DATE(MMDDYY)	Date as character string <i>mmddyy</i> , where <i>mm</i> is the month, <i>dd</i> is the day, and <i>yy</i> is the year.	6	Date
DATE(MMDDYYYY)	Date as character string <i>mmddyyyy</i> , where <i>mm</i> is the month, <i>dd</i> is the day, and <i>yyyy</i> is the year.	8	Date
TIME(1/100S)	A 32-bit binary integer representing time in hundredths of a second elapsed since hour 0.	4	Time
TIME(HHMMSSTF)	Time in the format <i>hhmmsstF</i> (packed), where <i>hh</i> is hours, <i>mm</i> is minutes, <i>ss</i> is seconds, <i>t</i> is tenths of a second, and <i>F</i> can have any value. (<i>F</i> is ignored and isn't checked to be a valid decimal sign).	4	Time
TIME(OHHMMSSF)	Time in the format <i>OhhmmssF</i> (packed), where <i>hh</i> is hours, <i>mm</i> is minutes, <i>ss</i> is seconds, and <i>F</i> can have any value. (<i>F</i> is ignored and isn't checked to be a valid decimal sign).	4	Time
TIME(HHMMSSTH)	Time in the format <i>hhmmssth</i> (packed), where <i>hh</i> is hours, <i>mm</i> is minutes, <i>ss</i> is seconds, and <i>th</i> is hundredths of a second.	4	Time
TIME(HHMMSSU6)	Time in the format <i>hhmssuuuuuu</i> (packed), where <i>hh</i> is hours, <i>mm</i> is minutes, <i>ss</i> is seconds, and <i>uuuuuu</i> is microseconds.	6	Time
TIME(HHMMSS)	Time as character string <i>hhmmss</i> , where <i>hh</i> is hours, <i>mm</i> is minutes, and <i>ss</i> is seconds.	6	Time

Table 36. Field formats for the FIELDS clause of the DEFINE RECORD statement (continued)

Field format	Contents	Length in bytes	Data type
INTV(MMSSTTF)	Time duration in the format <i>mmssttF</i> (packed), where <i>mm</i> is minutes of duration, <i>ss</i> is seconds, <i>ttt</i> is milliseconds, and <i>F</i> can have any value. The duration is converted to milliseconds and expressed as an integer. (<i>F</i> is ignored and isn't checked to be a valid decimal sign).	4	Integer
INTV(TOD)	Time duration in microseconds. If the difference of System/390 time-of-day(TOD) is smaller than 1970, it is converted to microseconds. Otherwise, the value is 0.	8	floating-point
IPADDR(IPV4)	Convert record data to IPv4 address	4	Dot separated decimal string
IPADDR(IPV6)	Convert record data to IPv6 address	16	Colon separated hex string
TIMESTAMP(TOD)	Date and time in System/390 time-of-day (TOD) clock format: the number of microseconds since the start of year 1900, expressed as a binary number, with the highest bit position representing 2^{51} .	4, 8 (default)	Timestamp

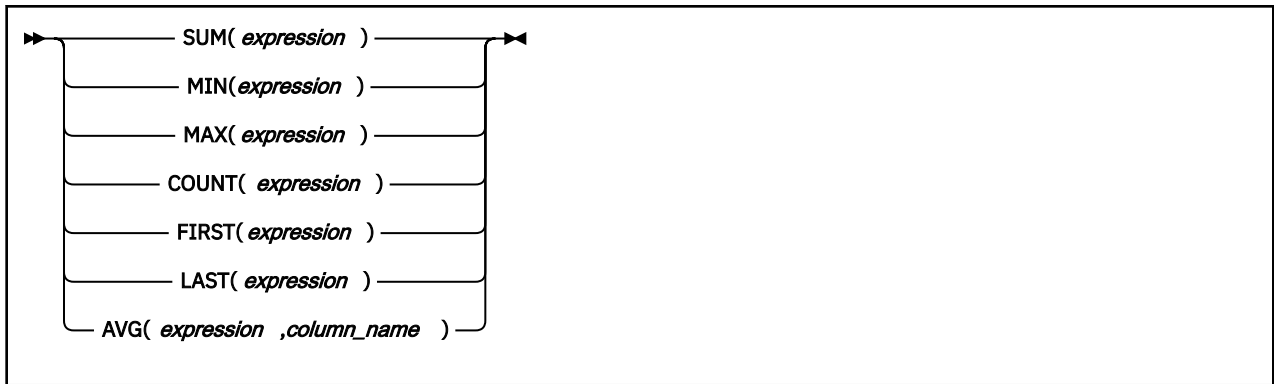
Examples

The following example shows a DEFINE RECORD statement for a simple record.

```

DEFINE RECORD SMF_XXX_CUST
VERSION 'CDP.210'
IN LOG SMF
IDENTIFIED BY SMFXXRTY = 130
FIELDS
(SMFXXLEN OFFSET 0   LENGTH 2 BINARY,
SMFXXSEG OFFSET 2   LENGTH 2 BINARY,
SMFXXFLG OFFSET 4   LENGTH 1 BIT,
SMFXXRTY OFFSET 5   LENGTH 1 BINARY,
SMFXXTME OFFSET 6   LENGTH 4 TIME(1/100S),
SMFXXDTE OFFSET 10  LENGTH 4 DATE(0CYYDDDF),
SMFXXSID OFFSET 14  LENGTH 4 CHAR,
*          OFFSET 18 LENGTH 4 CHAR,
SMFXXSTP OFFSET 22  LENGTH 2 BINARY)
SECTION SUBSYSTEM
OFFSET 24
LENGTH 12
NUMBER 1
FIELDS
(SMFXXSOF OFFSET 0   LENGTH 4 BINARY,
SMFXXSLN OFFSET 4   LENGTH 2 BINARY,
SMFXXSON OFFSET 6   LENGTH 2 BINARY,
SMFXXIOF OFFSET 8   LENGTH 4 BINARY)
SECTION ACCOUNTING
PRESENT IF SMFXXFLG > 0
OFFSET SMFXXSOF
LENGTH SMFXXSLN
NUMBER SMFXXSON
REPEATED

```

Parameters

DEFINE UPDATE *update_name*

Specifies the name of the update that you are defining to the System Data Engine. The update name must be unique. Verify that in the SHB0DEFS data set that is used as the CDP concatenation library, and in the data set that is used as the USER concatenation library, no existing definition has the same name.

Important: The DEFINE UPDATE statements must be included after the DEFINE RECORD statement.

VERSION *version*

Specifies the version for the update. The maximum length for this value is 18 characters. You might want to specify this optional value for troubleshooting purposes.

FROM *record_name*

Specifies the source of the data for the update. It must refer to a previously defined record.

SECTION *section_name*

Controls the processing of repeated sections. This clause specifies that the source of the update is a repeated section *section_name* of the record *record_name*. A section is repeated if the keyword REPEATED is included when it is defined. For more information about how to define a section, see [“SECTION clause” on page 251](#).

If you include this clause, the System Data Engine generates an internal record for each occurrence of the repeated section, and the source of the update is that internal record.

If the record *record_name* has repeated sections, and you omit the SECTION clause, the update can use only the data that is outside of the repeated sections, meaning that the repeated sections are not processed.

WHERE

Specifies that the update applies to only those source records or rows for which the condition that follows WHERE is true.

condition

Specifies the condition for the records or rows according to which the update applies. For more information, see [“Conditions” on page 246](#).

Any identifier that is used in the *condition* must be the name of a field in the source record.

TO &IBM_UPDATE_TARGET; &IBM_CORRELATION; AS &IBM_FILE_FORMAT

These parameters must be included as shown in the syntax.

LET, GROUP BY, and SET clauses

These clauses specify the processing to be performed by the update. This includes advanced functions like GROUP BY(*field_name=expression*) and SET (*field_name=accumulation*) that are not commonly used.

The processing occurs in the order that the clauses are defined. You must specify at least one GROUP BY clause or SET clause.

The first clause in the definition uses the source records from *record_name* as input. Each subsequent clause uses the result of the preceding clause as input.

For more information about these clauses, see the following sections:

- [“LET clause” on page 260](#)
- [“GROUP BY clause” on page 260](#)
- [“SET clause” on page 261](#)

LET clause

Assigns names to expressions that are frequently used in subsequent clauses, which can simplify the DEFINE UPDATE definition and improve the efficiency of the update. For example, if you want to calculate a value from a field, and use that value in several expressions, you can assign the result of the calculation to a name in the LET clause, and refer to that name wherever the result of the calculation is required in the definition.

identifier

Specifies the name that is assigned to the result of the expression. The name can be any identifier that is distinct from the names of fields in the source and names that are defined in the same LET clause.

expression

Specifies the expression to whose result the name is assigned. The expression can use the names that are defined in the same LET clause. For more information about how to use expressions, see [“Expressions” on page 244](#).

Any identifier that is used in this clause must be the name of a field in the source record, or a name that is introduced in one of the preceding clauses.

GROUP BY clause

Organizes records in groups according to specified values. The input to this clause is the source data that is specified in the FROM clause.

The result of GROUP BY processing is groups of input records, such that all records within each group have the same grouping values. All grouping values must be non-null. A row that has a null grouping value is not included in any group.

If you omit the GROUP BY clause, all input records are processed as one group.

Important: You must specify at least one of the following elements:

- GROUP BY *group_by_option*
- SET *set_option*

If the update definition has table definition, the GROUP BY clause must be consistent with that in DEFINE TABLE statement.

GROUP BY RECORD

With this option, all the records and repeated sections are recombined into a single output record. Use GROUP BY RECORD to have each input record produce a single output record. For example, for a section SECTION_A that repeated three times, this option outputs only one record.

GROUP BY NONE

With this option, all the records and repeated sections are treated individually. If you specify GROUP BY NONE, no grouping is performed, and each input record is processed individually. For example, for a section SECTION_A that repeated three times, this option outputs three records.

Tip: If there are no repeated sections, GROUP BY RECORD and GROUP BY NONE have the same behavior.

GROUP BY ALL

With this option, all input records are processed as a single group.

GROUP BY (*field_name* = *expression*)

With this option, all input records are processed according to the following values:

field_name

Specifies the field based on which the records are grouped. The field value cannot be a decimal or a long string.

expression

Specifies one grouping value. For more information about how to use expressions, see [“Expressions” on page 244](#).

Any identifier that is used in an expression in any of the clauses must be the name of a field in the source record, or a name that is introduced in one of the preceding clauses.

You can have multiple (*field_name* = *expression*) entries, separated by commas.

DURATION *integer* SECONDS/MINUTES

Incoming records are organized in memory according to specified grouping value. Special control is required to ensure System Data Engine does not send a partially aggregated record out. DURATION option specifies the minimum amount of time that a temporary group of records will be retained in memory before it is sent to the Data Streamer.

Changing the DURATION value can affect the resource consumption. Follow these guidelines to determine a proper DURATION value:

- DURATION is specified when GROUP BY organizes records according to a specified time, and full data aggregation is required. Specified time means that you might want to aggregate your data within a certain time.
- Use a value that is a multiple of the time that is specified in GROUP BY. For example, if you aggregate data for 3 second intervals, then the DURATION value can be 3 seconds, 6 seconds, 9 seconds, 12 seconds etc.
- The minimum value is 1 second. The maximum value is 60 minutes. A large value may cause storage buffer overrun if the rate of incoming records is high. Specify a value closest to the data aggregation requirement.
- DURATION is not useful if GROUP BY does not aggregate data with any time fields.
- By default, the timestamp of an incoming record is determined by TIMESTAMP (SMFDATE, SMFTIME), where SMFDATE and SMFTIME are the date and time from the SMF record header. For other data like non-SMF records, you can set a variable called *CDP_TIMESTAMP* in the LET clause or GROUP BY clause to specify the field that is used to identify the timestamps of incoming records.
- If you omit DURATION, the System Data Engine might produce partially aggregated records. In streaming mode, all data, aggregated or non - aggregated, in memory are always sent to the data streamer. In batch mode, all data, aggregated or non-aggregated, in memory are sent to the destination at end of the data set or when buffered.



CAUTION: Using the GROUP BY clause and LET clause might cause storage buffer overrun.

SET clause

Summarizes the groups of records that result from the GROUP BY clause. The SET clause produces one record in the target output for each group. In that record, the grouping values are stored in the fields that are specified in the GROUP BY clause. The values of other fields are derived from all the records in the group, as specified in the SET clause.

Important: You must specify at least one of the following elements:

- GROUP BY *group_by_option*
- SET *set_option*

If the update definition has table definition, the GROUP BY clause must be consistent with that in DEFINE TABLE statement.

SET (ALL)

Specifies that an update object is to be created that contains all fields in the source record object. If a section named TRIPLETS is present in the source record object, no fields from that section are included in the generated update object.

If SECTION *section_name* is not specified, fields in repeated sections are also omitted from the generated update object. If SECTION *section_name* is specified, all fields inside and outside of the repeated sections are included in the generated update object.

If GROUP BY RECORD is specified with SET (ALL), all fields of the generated update object are *field_name*=FIRST(*field_name*).

Important: If SET (ALL) is specified, the LET clause is not valid.

SET (*field_name*=*accumulation*)

You can have multiple (*field_name*=*accumulation*) entries, separated by commas.

field_name

Specifies a field of the target output.

accumulation

Specifies how to derive the value to be stored in the field. It can be one of the following expressions:

SUM(*expression*)

Evaluates the expression *expression* for each record in the group. The value of SUM is the sum of all non-null values that are obtained. If the value of *expression* is null for all records in the group, the value of SUM is null. The expression must specify a numerical value.

MIN(*expression*)

Evaluates the expression *expression* for each record in the group. The value of MIN is the least of all non-null values that are obtained. If the value of *expression* is null for all records in the group, the value of MIN is null.

MAX(*expression*)

Evaluates the expression *expression* for each record in the group. The value of MAX is the greatest of all non-null values that are obtained. If the value of *expression* is null for all records in the group, the value of MAX is null.

COUNT(*expression*)

Evaluates the expression *expression* for each record in the group. The value of COUNT is an integer that is the total number of non-null values that are obtained.

FIRST(*expression*)

Evaluates the expression *expression* for each record in the group, in the order in which the records are processed. The value of FIRST is the first non-null value of *expression*. If the value of *expression* is null for all records in the group, the value of FIRST is null.

LAST(*expression*)

Evaluates the expression *expression* for each record in the group, in the order in which the records are processed. The value of LAST is the last non-null value of *expression*. If the value of *expression* is null for all records in the group, the value of LAST is null.

AVG(*expression*,*field_name*)

Evaluates the expression *expression* for each record in the group. The value of AVG is the average, or weighted average, of the values that are obtained, depending on the *field_name* value, which must name a field whose value is specified in the same SET clause. The value of *field_name* must be specified by using either COUNT or SUM. If *field_name* is specified by using COUNT, its value must be equal to the number of non-null values of the expression. The result of AVG is the average of all non-null values of the expression in the group. If the value of expression is null for all records in the group, the value of AVG is null. If *field_name* is specified by using SUM, the result of AVG is the weighted average of all non-null values of

expression in the group. The argument of SUM obtained for the same record is used as the weight. If the value of expression is null for all records in the group, or the sum of all weights is 0, the value of AVG is null. The expression must specify a numeric value. The result of AVG is a floating-point field.

Any identifier that is used in an expression in any of the clauses must be the name of a field in the source record, or a name that is introduced in one of the preceding clauses. For more information about how to use expressions, see [“Expressions” on page 244](#).

Examples

The following example of a DEFINE UPDATE statement specifies how the System Data Engine is to extract data from SMF record type 130 and type 140.

Example 1

```
DEFINE UPDATE SMF_130
  VERSION 'CDP.V210'
  FROM SMF_130
  WHERE (SMF130PNAME='SYSTEMA')
  TO &IBM_UPDATE_TARGET
  &IBM_CORRELATION
  AS &IBM_FILE_FORMAT
  SET(ALL);
```

Example 2

```
DEFINE UPDATE SMF_140
  VERSION CDP.V210'
  FROM SMF_140
  WHERE (SMF140PNAME='SYSTEMA')
  TO &IBM_UPDATE_TARGET
  &IBM_CORRELATION
  AS &IBM_FILE_FORMAT
  LET (C_S_TIME = TIME(SMF140STME),
       C_Q_TIME = TIME(SMF140QTME))
  GROUP BY NONE
  SET (C_SYSTEM_ID = FIRST(SMF140SYSID),
       C_A_SYSTEM_ID = FIRST(SMF140ASYID),
       C_LAST_SEC = SUM(INTERVAL(C_S_TIME,C_Q_TIME)));
```

DEFINE TEMPLATE statement

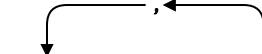
The DEFINE TEMPLATE statement refines the update definition to include only specified fields and to change the order of these fields in the output file.

- [“Syntax” on page 263](#)
- [“Parameters” on page 264](#)
- [“Example” on page 264](#)

Syntax

DEFINE TEMPLATE statement

► DEFINE TEMPLATE — *template_name* — FOR — *update_name* — ORDER — (—



 ◀ *field_name* —) — AS &IBM_FILE_FORMAT — ►

Parameters

DEFINE TEMPLATE *template_name*

Specifies the name of the template. The name of the template definition must be the same as the update definition that it is associated with. The custom template definition replaces the default template definition that is created by the update definition.

Important: The DEFINE TEMPLATE statements must be included after the DEFINE UPDATE statement in the input stream because the field names are verified based on the fields in the update definition.

FOR *update_name*

Specifies the name of the update definition that this template augments.

ORDER (*field_name*,...)

Specifies the order of fields for this template. One *field_name* entry represents a field that is required for this template.

field_name

Specifies the name of the field. The field names must be unique in a template, and they must be names that are already defined in the update definition that this template is augmenting.

Important: Do not specify repeated field names.

AS &IBM_FILE_FORMAT

This parameter must be included as shown in the syntax.

Example

The following example shows a simple DEFINE TEMPLATE statement.

```
DEFINE TEMPLATE SMF_030_CUST FOR SMF_030_CUST
ORDER
(SMF30JBN,
 SMF30PGM,
 SMF30STM,
 SMF30UIF)
AS &IBM_FILE_FORMAT;
```

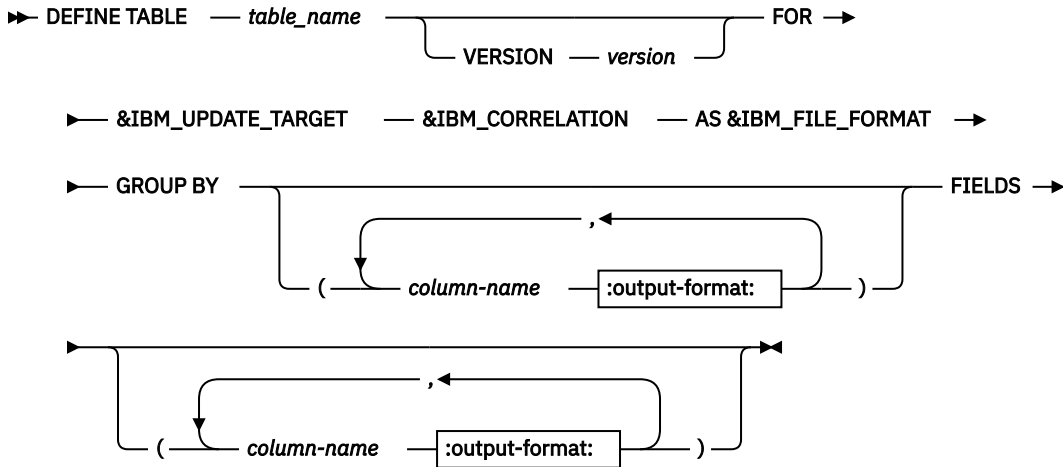
DEFINE TABLE statement

The DEFINE TABLE statement defines how to create a virtual table to connect multiple updates with the same attributes. You can use DEFINE UPDATE statement to connect update definitions to the table definition.

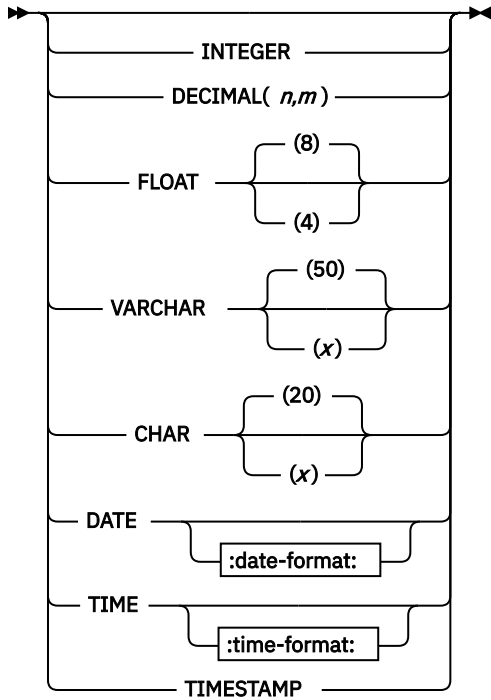
- [“Syntax” on page 265](#)
- [“Parameters” on page 266](#)
- [“Use Scenarios” on page 266](#)
- [“Example” on page 266](#)

Syntax

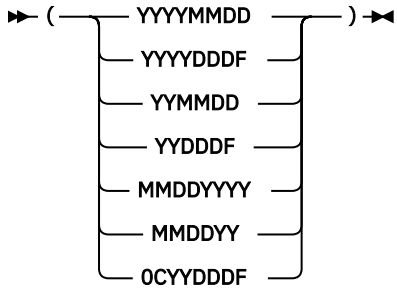
DEFINE TABLE statement



:output-format:



:date-format:



:time-format:



Parameters

DEFINE TABLE *table_name*

Specifies the name of the table that you are defining to the System Data Engine. The table name must be unique and consistent with the customized System Data Engine data stream definition name that is defined in the IBM Z Common Data Provider Configuration Tool.

VERSION *version*

Specifies the version for the table. The maximum length for this value is 18 characters. You might want to specify this optional value for troubleshooting purposes.

FOR &IBM_UPDATE_TARGET; &IBM_CORRELATION; AS &IBM_FILE_FORMAT

These parameters must be included as shown in the syntax.

GROUP BY *column-name*

Specifies the key fields for this table. These fields must be the same fields as any update definition that connects to this table. The name and order of these fields must be identical to the table and any of these updates.

Important: The GROUP BY clause must be consistent with that in DEFINE UPDATE statement.

FILEDS *column-name*

Specifies the fields required for this table. The column names are the names of the fields required for this table. The column-names must be the same as the names that have been defined in the update definitions that connect to this table.

output-format

Specifies the data type of columns to which the System Data Engine automatically converts.

Use Scenarios

This function is used to join fields from both non-repeated and repeated sections in a single record. You can define several update definitions to update the same table. Use one update definition to update the table with the fields from the common section (non-repeated section), and other update definitions to update the table with the fields from repeated sections. Use GROUP BY columns to define the key of row on how these fields can be joined together.



Warning: If you define the table with update definitions of different records, you might not get a complete record. Some fields may be empty due to the missing of some records at the time when the data is flushed.

Example

The following example of DEFINE TABLE statement and DEFINE UPDATE statement specifies how the System Data Engine extracts data from SMF record type 74 subtype 4.

The table A_PM_CF_I is from two update definitions: A_PM_CF update definition is from the record SMF_074_4 repeat section R744SREQ, and A_PM_CF1 update definition is from the main record SMF_074_4.

```
DEFINE TABLE A_PM_CF_I
  FOR &IBM_UPDATE_TARGET
  AS &IBM_FILE_FORMAT
  GROUP BY
  (
    PERIOD_NAME          CHAR (8),
    MVS_SYSTEM_ID        CHAR (4),
  )
  FIELDS
  (
    REQ_ASYNC_NO          FLOAT,
    SYSTEM_NAME           CHAR (8),
    CF_LEVEL              SMALLINT);
```

```
-----
DEFINE UPDATE A_PM_CF
  VERSION 'CDP.V210'
  FROM SMF_074_4 SECTION R744SREQ
  TO TABLE A_PM_CF_I
```

```

&IBM_FILE_FORMAT
LET
(
  T1          = TIMESTAMP (SMF74DAT, SMF74IST) + (
              SMF74INT/2000) SECONDS,
  D1          = DATE (T1),
  P1          = VALUE (PERIOD (SMF74SID, D1, TIME (T1))
                    , '?' )
)
GROUP BY
(
  PERIOD_NAME =
P1,
  MVS_SYSTEM_ID = SMF74SID
)
SET
(
  REQ_ASYNC_NO = SUM (R744SARC)
);
-----

DEFINE UPDATE A_PM_CF1
VERSION 'CDP.V210'
FROM SMF_074_4
WHERE SMF74RAN = 0 OR (SMF74RAN = 1 AND
SMF74RSQ = 1)
TO TABLE A_PM_CF_I
&IBM_FILE_FORMAT
LET
(
  T1          = TIMESTAMP (SMF74DAT, SMF74IST) + (
              SMF74INT/2000) SECONDS,
  D1          = DATE (T1),
  P1          = VALUE (PERIOD (SMF74SID, D1, TIME (T1))
                    , '?' )
)
GROUP BY
(
  PERIOD_NAME =
P1,
  MVS_SYSTEM_ID = SMF74SID
)
SET
(
  SYSTEM_NAME = FIRST (SMF74SNM),
  CF_LEVEL    = FIRST (R744FLVL)
);

```

Configuration scenarios

This section provides scenarios for complicated configuration tasks via examples and use cases.

Binding the Data Streamer to a specific IP address

By default, the Data Streamer does a generic bind() and accepts connection requests from any IP address associated with any TCP/IP stacks in the LPAR. The data gatherers, such as Log Forwarder and System Data Engine, connect to the Data Streamer via either the loop back IP address (127.0.0.1) or primary IP address of the default TCP/IP stack. If required, you can configure the Data Streamer to bind to a specific IP address such as a dynamic virtual IP address (DVIPA). Also, you must properly configure the data gatherers, such as Log Forwarder and System Data Engine, so that they can connect to the specific IP address that the Data Streamer is listening on.

About this task

The IBM Z Common Data Provider requires the Data Streamer, Log Forwarder, and System Data Engine to run on the same LPAR. In this scenario, the Data Streamer is configured to bind to the IP address 9.30.243.157, and is running on the same LPAR as the data gatherers.

Procedure

1. Configure the Data Streamer to bind to a specific IP address.

Use the parameter **host** to specify the IP address that the Data Streamer will bind to. The IP address must be a valid IP address that is active on any of the TCP/IP stacks on the LPAR, or a dynamic virtual IP address (DVIPA) which can be activated by any of the TCP/IP stacks on the LPAR.

The follow example specifies that the Data Streamer will bind to IP address 9.30.243.157 only.

```
//STDPARM DD *
PGM /bin/sh
/usr/lpp/IBM/cdpz/v2r1m0/DS/LIB/startup.sh
/etc/cdpConfig/myPolicy.policy
nnnnn start=w trace=n host=9.30.243.157
/**
```

For more information about configuring the Data Streamer, see [“Configuring the Data Streamer” on page 118](#).

2. Configure the data gathers to connect to the Data Streamer.

If the Data Streamer is configured to bind to a specific IP address, you must configure each data gatherer to be able to connect to the Data Streamer successfully.

- For the Log Forwarder, use the parameter **-h** to specify the Data Streamer IP address. For example,

```
//STDPARM DD *
PGM /bin/sh
/usr/lpp/IBM/zcdp/v2r1m0/LF/samples/startup.sh
-e /usr/lpp/IBM/zcdp/v2r1m0/LF/samples
-h 9.30.243.157
/**
```

For more information about configuring the Log Forwarder, see the following topics:

- [“Creating the Log Forwarder started task” on page 125](#)
- [“Creating the Log Forwarder batch job for sending SYSLOG data to the Data Streamer” on page 132](#)
- For the System Data Engine, add the SET statement SET IBM_DS_HOST=*ip_address* in the HBOIN DD statement to specify the Data Streamer IP address. For example,

```
//HBOIN DD *
SET IBM_DS_HOST = '9.30.243.157';
SET IBM_UPDATE_TARGET = 'PORT ppppp';
SET IBM_FILE_FORMAT = 'CSV';
// DD PATH='/etc/cdpConfig/hboin.sde',
// PATHDISP=(KEEP),RECFM=V,LRECL=255,FILEDATA=RECORD
```

For more information about configuring the System Data Engine, see the following topics:

- [“Creating the System Data Engine started task for streaming SMF data” on page 135](#)
- [“Creating the System Data Engine batch job for sending SMF data to the Data Streamer” on page 148](#)
- [“Creating the System Data Engine batch job for sending DCOLLECT data to the Data Streamer” on page 150](#)
- [“Creating the System Data Engine started task for collecting IMS records” on page 144](#)
- [“Creating the System Data Engine batch job for sending IMS data to the Data Streamer” on page 146](#)
- For user applications using open stream Java API, specify the Data Streamer IP address when initializing the CDPSender object. For example,

```
CDPSender sender = new CDPSender("9.30.243.157",20201);
```

For more information about sending data by using the Java API, see [“Sending data by using the Java API” on page 287](#).

- For user applications using open stream REXX API, specify the Data Streamer IP address in the **hbo.host** parameter in the REXX code. For example,

```

hbo.host = '9.30.243.157'

meta.0 = 8
meta.1 = 'encoding=IBM-1047'
meta.2 = 'sourcetype=HBO-TESTREXX1'
meta.3 = 'sourcename=TESTREXX1'
meta.4 = 'path=hbo/test/rexx/one'
meta.5 = 'sysplex=TESTPLEX'
meta.6 = 'system=SYS1'
meta.7 = 'host=somewhere.com'
meta.8 = 'timezone=+0000'

```

For more information about sending data by using the REXX API, see [“Sending data by using the REXX API”](#) on page 288.

Join data streams for SMF record type 70 subtype 1

In SMF record type 70 subtype 1, some sections, like SMF70BCT (PR/SM partition data) and SMF70BPD (R/SM logical processor data), and SMF_070_LCD(Logical core data) and SMF_070_CPU(CPU data), are bound together. However, when these sections are respectively streamed through different data streams, such correlation is lost. You can join the data streams by using the key fields.

About this task

When you use the data stream SMF_070_BCT (the PR/SM partition data section stream) and SMF_070_BPD (the PR/SM logical processor data section stream) to analyze the LPAR and CPU usage information, the correlation between section SMF70BCT and SMF70BPD in the original SMF record type 70 subtype 1 is lost. Therefore, we added the key field SMF70LPAR, which is the LPAR name, to the new SMF_070_BPD_V2 data stream to indicate the correlation between data stream SMF_070_BPD_V2 and SMF_070_BCT.

Similar issue occurs in data stream SMF_070_LCD (the Logical core data section stream) and SMF_070_CPU (the CPU data section stream). We added the key field SMF70_COREID, which is the core identification, to the new SMF_070_CPU_V2 data stream to indicate the correlation between data stream SMF_070_CPU_V2 and SMF_070_LCD.

To avoid migration issues, new V2 data streams are added while the original data streams remains valid.

Procedure

- You can join data stream SMF_070_BCT and SMF_070_BPD_V2, or join data stream SMF_070_LCD and SMF_070_CPU_V2 by combining the values of the Correlator ID field (Key1) and the values of the new key field (Key2) according to the following examples.
 - Join data stream SMF_070_BCT and SMF_070_BPD_V2 according to the following tables.

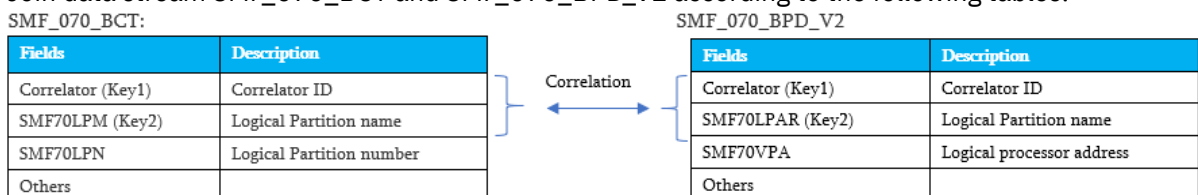


Figure 5. Join data stream SMF_070_BCT and SMF_070_BPD_V2

See the following example.

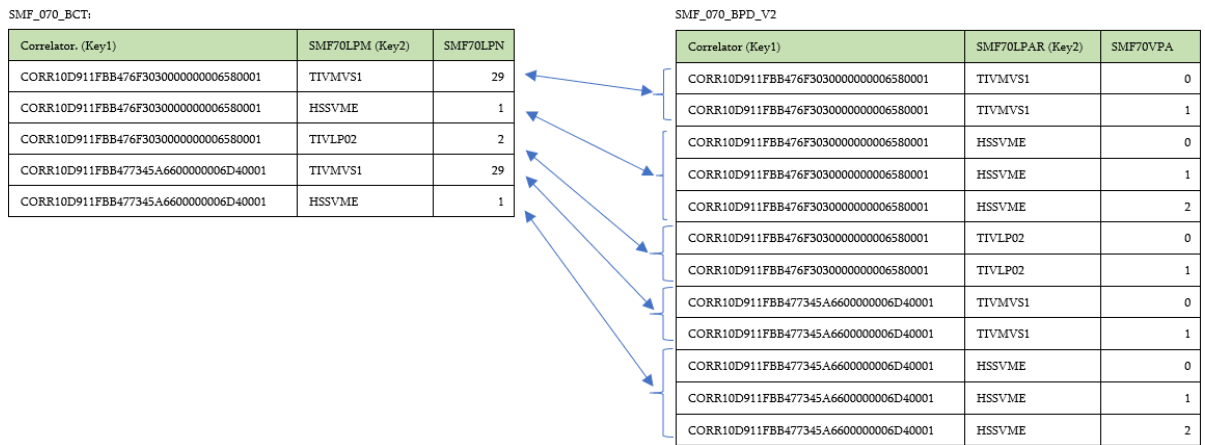


Figure 6. Example of joining data stream SMF_070_BCT and SMF_070_BPD_V2

- Join data stream SMF_070_LCD and SMF_070_CPU_V2 according to the following tables.

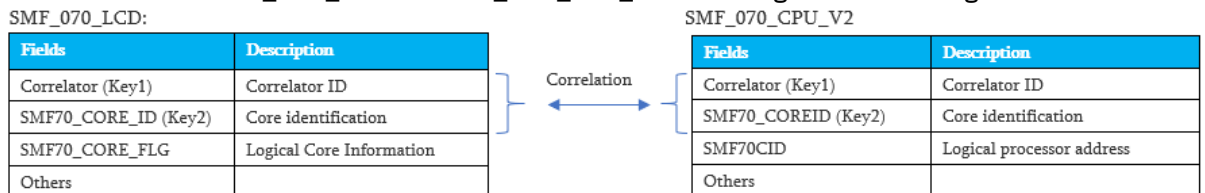


Figure 7. Join data stream SMF_070_LCD and SMF_070_CPU_V2

Chapter 6. Operating Z Common Data Provider

To operate IBM Z Common Data Provider, you must know how to run (for example, start, stop, or pass information to) key components, such as the data gatherer components (Log Forwarder and System Data Engine), the Data Streamer, and the Data Receiver.

Before you begin

Before you start IBM Z Common Data Provider, verify that the z/OS environment is set up correctly for IBM Z Common Data Provider to do the following tasks:

- Access the TCP/IP resolver configuration file.
- Resolve host names.

Search order for the TCP/IP resolver configuration file

For more information, see [“Verifying the search order for the TCP/IP resolver configuration file” on page 151.](#)

Host name resolution

To operate, IBM Z Common Data Provider must determine the fully qualified domain name (FQDN) of the system on which it is running. Therefore, activate the networking and name resolution services that are configured in the system for use by IBM Z Common Data Provider before you start IBM Z Common Data Provider.

About this task

The following lists indicate the best practice for the order in which to start or stop the components:

Order in which to start the components

1. Start the data receiving components, such as the Data Receiver or Logstash.
2. Start the Data Streamer.
3. Start the data gatherer components, such as the Log Forwarder and System Data Engine.

Order in which to stop the components

1. Stop the data gatherer components, such as the Log Forwarder and System Data Engine.
2. Stop the Data Streamer.
3. Stop the data receiving components.

Running the Data Receiver

To operate the Data Receiver, you can run the Data Receiver with scripts or run the Data Receiver as services.

About this task

There are two ways to run the Data Receiver, you must determine how to and on which platform you want to run the Data Receiver and follow corresponding instructions to run the Data Receiver:

- Run the Data Receiver with scripts

You can use the shell script `DataReceiver.sh` or the batch file `DataReceiver.bat` to run the Data Receiver on Linux or Windows. For more information, see [“Running the Data Receiver with scripts” on page 272.](#)

- Run the Data Receiver as services

If you run the Data Receiver as services, you can operate, monitor and manage the service through service managers.

- Run the Data Receiver as systemd services on Linux

You can operate, monitor and manage the service through systemd. For more information, see [“Running the Data Receiver as systemd services on Linux” on page 273.](#)

- Run the Data Receiver as system services on Windows

You can operate, monitor and manage the service through Service Control Manager. For more information, see [“Running the Data Receiver as system services on Windows” on page 274.](#)

Running the Data Receiver with scripts

To operate the IBM Z Common Data Provider Data Receiver, you can run the shell script `DataReceiver.sh` or the batch file `DataReceiver.bat`.

Before you begin

To run the Data Receiver, Java Runtime Environment (JRE) 8 must be installed.

When it runs, the Data Receiver uses the configuration values of the Data Receiver properties in the `cdpdr.properties` file in the Data Receiver working directory (`CDPDR_HOME` directory). If you want to update the property value for the currently running Data Receiver, you must stop the current Data Receiver first, update the property value, and restart the Data Receiver for the updated definitions to take effect. For more information about these configuration values, see [“Updating the Data Receiver properties” on page 114.](#)

After the Data Receiver is started, the log directory `logs` is created in the `CDPDR_HOME` directory by default.

About this task

You can use the commands defined in `DataReceiver.sh` and `DataReceiver.bat` to run the Data Receiver.

Procedure

To operate the Data Receiver, run the following commands:

Platform on which the Data Receiver runs	Instructions
Linux	<ul style="list-style-type: none">• Start the Data Receiver: <pre>./DataReceiver.sh start</pre>• Stop the Data Receiver: <pre>./DataReceiver.sh stop</pre>• Restart the Data Receiver: <pre>./DataReceiver.sh restart</pre><p>The Data Receiver is restarted based on the parameters of current <code>cdpdr.properties</code> file.</p>• Check the status of Data Receiver: <pre>./DataReceiver.sh status</pre>• Turn on the trace level of the running Data Receiver:

Platform on which the Data Receiver runs	Instructions
	<pre data-bbox="641 237 1162 264">./DataReceiver.sh dynamicLogging traceON</pre> <p data-bbox="641 296 1422 323">No need to restart the Data Receiver for this change to take effect.</p> <ul data-bbox="613 338 1252 365" style="list-style-type: none"> • Turn off the trace level of the running Data Receiver: <pre data-bbox="641 396 1175 424">./DataReceiver.sh dynamicLogging traceOFF</pre> <p data-bbox="641 455 1422 483">No need to restart the Data Receiver for this change to take effect.</p>
Windows	<ul data-bbox="613 516 919 543" style="list-style-type: none"> • Start the Data Receiver: <pre data-bbox="641 575 959 602">.\DataReceiver.bat start</pre> <ul data-bbox="613 625 915 653" style="list-style-type: none"> • Stop the Data Receiver: <pre data-bbox="641 684 945 711">.\DataReceiver.bat stop</pre> <ul data-bbox="613 735 946 762" style="list-style-type: none"> • Restart the Data Receiver: <pre data-bbox="641 793 984 821">.\DataReceiver.bat restart</pre> <p data-bbox="641 846 1422 905">The Data Receiver is restarted based on the parameters of current <code>cdpdr.properties</code> file.</p> <ul data-bbox="613 919 1040 947" style="list-style-type: none"> • Check the status of Data Receiver: <pre data-bbox="641 978 971 1005">.\DataReceiver.bat status</pre> <ul data-bbox="613 1029 1252 1056" style="list-style-type: none"> • Turn on the trace level of the running Data Receiver: <pre data-bbox="641 1087 1175 1115">.\DataReceiver.bat dynamicLogging traceON</pre> <ul data-bbox="613 1138 1252 1165" style="list-style-type: none"> • Turn off the trace level of the running Data Receiver: <pre data-bbox="641 1197 1187 1224">.\DataReceiver.bat dynamicLogging traceOFF</pre>

Running the Data Receiver as systemd services on Linux

To run IBM Z Common Data Provider Data Receiver as system services on Linux, you must configure the Data Receiver.

About this task

To configure the Data Receiver to run as systemd services on Linux, you must define the `datareceiver.service` service script, and place the script under the `systemd` folder. Data Receiver uses `systemd` to handle services on Linux.

Procedure

1. Customize the following parameter values in the `datareceiver.service` service script. You can find the service script in the directory that you specified for copying Data Receiver files:

User

The name of the user that runs the service.

Group

The name of the group that is associated with that user.

Environment

The location of the executable Data Receiver shell script. Ensure that the file path is an absolute path.

- Copy the `datareceiver.service` script to the `/etc/systemd/system` folder. If you run multiple agents on the system, use a unique name for the script, for example, `datareceiver-A.service`. And ensure that each Data Receiver service has a unique port value and a separate `CDPDR_HOME` directory, and the file path `<PATH_TO_SERVICE_FILE>` is an absolute path.

```
./importServiceFile.sh <PATH_TO_SERVICE_FILE>/datareceiver.service
```

- Reload systemd manager configuration to make the new service file take effect.

```
sudo systemctl daemon-reload
```

- Optional: Set the Data Receiver service to start automatically at startup.

```
systemctl enable datareceiver.service
```

- To operate the systemd service of Data Receiver on Linux, run the following commands:

Action	Command
Start the Data Receiver service	<code>systemctl start datareceiver</code>
Stop the Data Receiver service	<code>systemctl stop datareceiver</code>
Restart the Data Receiver service	<code>systemctl restart datareceiver</code>
Check the status of Data Receiver service	<code>systemctl status datareceiver</code>

Note: The dynamic logging feature of Data Receiver is not supported via systemd. You must turn on or off the trace level of the running Data Receiver service through the `DataReceiver.sh` script in your working directory, and only activate the tracing at the request of IBM Support.

- You can view the log messages of the service in two ways. If your systemd version is higher than 219, you can see the Data Receiver service log messages directly on the console, and view the log messages via the `journalctl` command. If your systemd version is equal to or lower than 219, you can only view the logs via `journalctl` command. You can use the following `journalctl` command to view log messages. For the following example, `datareceiver.service` is the service that you have configured.

```
journalctl -u datareceiver
```

Running the Data Receiver as system services on Windows

You can set up the IBM Z Common Data Provider Data Receiver to run as system services on Windows.

Before you begin

To install the Data Receiver services, you must have `installutil.exe` on your system and add the path to the `PATH` system variable.

About this task

If you want to use the brief Data Receiver service command alias to run Data Receiver, you must add `DataReceiverService_alias.bat` to Registry first and restart the system for the update to take effect. Then you can install Data Receiver service. After you install the Data Receiver service successfully, run the service to manage Data Receiver.

Procedure

- Log in as administrative user and add `DataReceiverService_alias.bat` to Registry. `DataReceiverService_alias.bat` wraps original Windows NET and SC commands into brief alias so that you can run Data Receiver easier. If you want to use these alias, you must complete the following steps:
 - Open Registry Editor, find `Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Command Processor`, create a new string value, and set a value name, for example, `AutoRun`. Then add the value data to the path of `DataReceiverService_alias.bat`. For example, `<CDPDR_HOME>\DataReceiverService_alias.bat`.
 - Restart the computer for the changes to take effect. When CMD starts, the `DataReceiverService_alias.bat` is executed automatically.
- Install Data Receiver service. You must run Command Prompt as administrator to install the service. If you use as a common user, run the following Runas command to convert to administrator. You must replace the host with your IP address in the command:

```
Runas /profile /user:<host>\Administrator CMD
```

Then enter the directory where the `DataReceiverService.exe` is, and run command alias or original command to install the service.

Action	Command
Use command alias	<code>DataReceiverServiceInstall</code>
Use original command	<code>installutil DataReceiverService.exe</code>

- After the service is installed successfully, operate the Data Receiver with the following commands. You can find the log of Data Receiver service in `<PATH of DataReceiverService.exe>\logs\DataReceiverService_logs`. The name format of the log file is `DataReceiverService_<Action>_<Mon-Day-Year>.log`.

Action	Command alias	Original command
Start the Data Receiver	<code>DataReceiverStart</code>	<code>net start DataReceiverService</code>
Stop the Data Receiver	<code>DataReceiverStop</code>	<code>net stop DataReceiverService</code>
Restart the Data Receiver	<code>DataReceiverRestart</code>	<code>sc control DataReceiverService 129</code>
Check the status of Data Receiver	<code>DataReceiverStatus</code>	<code>sc control DataReceiverService 128</code>

- By default, the Data Receiver service is set to start automatically at startup. If you do not want to start the service automatically at startup, you can change the startup type in services window.
- If you want to uninstall Data Receiver service, you can enter the directory where the `DataReceiverService.exe` is, and run the command alias or original command to uninstall the service:

Action	Command
Use command alias	<code>DataReceiverServiceUninstall</code>

Action	Command
Use original command	<code>installutil /u DataReceiverService.exe</code>

Running the Data Streamer

To start the IBM Z Common Data Provider Data Streamer, you use the Data Streamer started task. When a policy is updated, you must stop and restart the Data Streamer to make the updated definitions take effect.

Before you begin

Create the Data Streamer started task, as described in [“Configuring the Data Streamer”](#) on page 118.

About this task

You use z/OS console commands to control the operation of the Data Streamer and to view information about the current policy.

Troubleshooting tip: After the Data Streamer is started, it should not stop until you stop it. If it does stop without your stopping it explicitly, review the Data Streamer job log output for possible errors.

Procedure

To operate the Data Streamer, issue the following console commands, where *procname* represents the name of the started task (such as HBODSPRO).

Action	z/OS console command
Start the Data Streamer	<code>START <i>procname</i></code>
Stop the Data Streamer	<code>STOP <i>procname</i></code>
View information about the current policy	<code>MODIFY <i>procname</i>,APPL=DISPLAY,POLICY</code> The following message is sample output from the command: <pre>HB06076I The current policy is /etc/cdpConfig/myPolicy.policy.</pre>

Running the Log Forwarder

To start the IBM Z Common Data Provider Log Forwarder, you use the Log Forwarder started task. If you are collecting NetView for z/OS message data, the NetView message provider must also be active. The NetView message provider is started as a started task by using the REXX module HBONETV.

Before you begin

Create the Log Forwarder started task, as described in [“Creating the Log Forwarder started task”](#) on page 125.

If you configure a **NetView Netlog** data stream for gathering NetView for z/OS message data, also configure the NetView message provider to monitor and forward NetView for z/OS messages to the Log Forwarder, as described in [“Configuring the z/OS NetView message provider for collecting NetView messages”](#) on page 130. You can start the REXX module HBONETV from the command line of an existing NetView user ID, or create a new NetView user ID to support the running of this REXX module. Always start the NetView message provider after the Log Forwarder is started for the first time.

If you configure a **z/OS SYSLOG** data stream for gathering z/OS SYSLOG data from a user exit, you must install either the HBOSYSG or HBOMDBG user exit, as described in [“Installing the user exit for collecting z/OS SYSLOG data”](#) on page 127. The HBOSYSG and HBOMDBG user exits create system resources that might need to be managed while they are in operation. The `manageUserExit` utility is a shell script that can be used to manage the system resources. For more information about this utility, see [“manageUserExit utility for managing the installed user exit”](#) on page 129.

About this task

You use z/OS console commands to control the operation of the Log Forwarder, including to start, stop, or view status or configuration information for Log Forwarder data streams.

For more information about Log Forwarder data streams, including the correlation between the sources from which the Log Forwarder gathers data and the data streams that can be defined for those sources, see [“Data stream configuration for data gathered by Log Forwarder”](#) on page 181.

Troubleshooting tip: After the Log Forwarder is started, it should not stop until you stop it. If it does stop without your stopping it explicitly, review the Log Forwarder job log output for possible errors.

Procedure

1. To operate the Log Forwarder, issue the following console commands, where *procname* represents the name of the started task (such as HBOPROC).

Action	z/OS console command
Start the Log Forwarder	<p>Issue one of the following console commands:</p> <p>Warm start</p> <pre>START <i>procname</i></pre> <p>A warm start resumes data collection where it previously stopped.</p> <p>Cold start</p> <pre>START <i>procname</i>,OPT=-C</pre> <p>A cold start starts data collection anew. Any operational data that was written while the Log Forwarder was stopped is not collected.</p> <p>Tip: If the Log Forwarder is shut down for more than a few minutes, you might want to use cold start mode to avoid having to wait for the Log Forwarder to collect accumulated data.</p>
Stop the Log Forwarder	<pre>STOP <i>procname</i></pre>
View the status of all known Log Forwarder data streams	<pre>MODIFY <i>procname</i>,APPL=DISPLAY,GATHERER,LIST</pre>
Start, stop, or view the status or configuration information for an individual data stream	<p>Table 37 on page 278 lists the commands, which vary depending on the source of the data stream.</p>

Table 37. z/OS console commands for starting, stopping, or viewing status or configuration information for individual Log Forwarder data streams

Source of data stream	z/OS console commands for controlling a data stream from this source
Job log	<p>Start the data stream</p> <pre>MODIFY procname,APPL=START,GATHERER,JOBNAME=jobname,DDNAME=ddname</pre> <p>Stop the data stream</p> <pre>MODIFY procname,APPL=STOP,GATHERER,JOBNAME=jobname,DDNAME=ddname</pre> <p>View the status of the data stream</p> <pre>MODIFY procname,APPL=DISPLAY,GATHERER,STATUS,JOBNAME=jobname,DDNAME=ddname</pre> <p>View the configuration information for the data stream</p> <pre>MODIFY procname,APPL=DISPLAY,GATHERER,CONFIG,JOBNAME=jobname,DDNAME=ddname</pre> <p>Usage note: If you used wildcard characters in the job name when you defined the data stream, the values of the JOBNAME and DDNAME parameters in these commands must reference the specific instance of the job log data stream. For example, you must specify JOB0011 or JOB0021 rather than JOB*1.</p>
z/OS UNIX log file	<p>Start the data stream</p> <pre>MODIFY procname,APPL=START,GATHERER,UNIXFILEPATH='UNIXfilepath'</pre> <p>Stop the data stream</p> <pre>MODIFY procname,APPL=STOP,GATHERER,UNIXFILEPATH='UNIXfilepath'</pre> <p>View the status of the data stream</p> <pre>MODIFY procname,APPL=DISPLAY,GATHERER,STATUS,UNIXFILEPATH='UNIXfilepath'</pre> <p>View the configuration information for the data stream</p> <pre>MODIFY procname,APPL=DISPLAY,GATHERER,CONFIG,UNIXFILEPATH='UNIXfilepath'</pre> <p>Usage note: To prevent an error message, the UNIX file path must be enclosed in quotation marks.</p>

Table 37. z/OS console commands for starting, stopping, or viewing status or configuration information for individual Log Forwarder data streams (continued)

Source of data stream	z/OS console commands for controlling a data stream from this source
Entry-sequenced VSAM cluster	<p>Start the data stream</p> <pre>MODIFY procname,APPL=START,GATHERER,DATASET=dataset</pre> <p>Stop the data stream</p> <pre>MODIFY procname,APPL=STOP,GATHERER,DATASET=dataset</pre> <p>View the status of the data stream</p> <pre>MODIFY procname,APPL=DISPLAY,GATHERER,STATUS,DATASET=dataset</pre> <p>View the configuration information for the data stream</p> <pre>MODIFY procname,APPL=DISPLAY,GATHERER,CONFIG,DATASET=dataset</pre> <p>Usage note: <i>dataset</i> represents the name of the dataset.</p>
z/OS SYSLOG	<p>Start the data stream</p> <ul style="list-style-type: none"> • From the user exit: <pre>MODIFY procname,APPL=START,GATHERER,SYSLOG</pre> • From OPERLOG: <pre>MODIFY procname,APPL=START,GATHERER,OPERLOG</pre> <p>Stop the data stream</p> <ul style="list-style-type: none"> • From the user exit: <pre>MODIFY procname,APPL=STOP,GATHERER,SYSLOG</pre> • From OPERLOG: <pre>MODIFY procname,APPL=STOP,GATHERER,OPERLOG</pre> <p>View the status of the data stream</p> <ul style="list-style-type: none"> • From the user exit: <pre>MODIFY procname,APPL=DISPLAY,GATHERER,STATUS,SYSLOG</pre> • From OPERLOG: <pre>MODIFY procname,APPL=DISPLAY,GATHERER,STATUS,OPERLOG</pre> <p>View the configuration information for the data stream</p> <ul style="list-style-type: none"> • From the user exit: <pre>MODIFY procname,APPL=DISPLAY,GATHERER,CONFIG,SYSLOG</pre> • From OPERLOG: <pre>MODIFY procname,APPL=DISPLAY,GATHERER,CONFIG,OPERLOG</pre>

Table 37. z/OS console commands for starting, stopping, or viewing status or configuration information for individual Log Forwarder data streams (continued)

Source of data stream	z/OS console commands for controlling a data stream from this source
IBM Tivoli NetView for z/OS messages	<p>Start the data stream</p> <pre>MODIFY procname,APPL=START,GATHERER,DOMAIN=domain</pre> <p>Stop the data stream</p> <pre>MODIFY procname,APPL=STOP,GATHERER,DOMAIN=domain</pre> <p>View the status of the data stream</p> <pre>MODIFY procname,APPL=DISPLAY,GATHERER,STATUS,DOMAIN=domain</pre> <p>View the configuration information for the data stream</p> <pre>MODIFY procname,APPL=DISPLAY,GATHERER,CONFIG,DOMAIN=domain</pre> <p>Usage note: <i>domain</i> represents the name of the NetView domain.</p>
IBM WebSphere Application Server for z/OS HPEL log	<p>Start the data stream</p> <pre>MODIFY procname,APPL=START,GATHERER,HPELDIRECTORY='hpeldirectory'</pre> <p>Stop the data stream</p> <pre>MODIFY procname,APPL=STOP,GATHERER,HPELDIRECTORY='hpeldirectory'</pre> <p>View the status of the data stream</p> <pre>MODIFY procname,APPL=DISPLAY,GATHERER,STATUS,HPELDIRECTORY='hpeldirectory'</pre> <p>View the configuration information for the data stream</p> <pre>MODIFY procname,APPL=DISPLAY,GATHERER,CONFIG,HPELDIRECTORY='hpeldirectory'</pre> <p>Usage note: <i>hpeldirectory</i> represents the High Performance Extensible Logging (HPEL) log directory. To prevent an error message, the directory must be enclosed in quotation marks.</p>
IBM Resource Measurement Facility Monitor III reports	<p>Start the data stream</p> <pre>MODIFY procname,APPL=START,GATHERER,RMFIIREPORT='reporttype'</pre> <p>Stop the data stream</p> <pre>MODIFY procname,APPL=STOP,GATHERER,RMFIIREPORT='reporttype'</pre> <p>View the status of the data stream</p> <pre>MODIFY procname,APPL=DISPLAY,GATHERER,STATUS,RMFIIREPORT='reporttype'</pre> <p>View the configuration information for the data stream</p> <pre>MODIFY procname,APPL=DISPLAY,GATHERER,CONFIG,RMFIIREPORT='reporttype'</pre> <p>Usage note: <i>reporttype</i> represents the type of the RMF Monitor III report.</p>

2. To run the NetView message provider, complete the following actions as appropriate.

Action	Instructions
Start the NetView message provider	Specify either C (cold start) or W (warm start) as the value for the <i>COMMON.HBONETV.START</i> variable in the CNMSTYLE member, as shown in the following example: COMMON.HBONETV.START = W
Stop the NetView message provider	Change the value of the <i>HBONETV.STOP</i> variable in the CGED panel to YES.

Running the System Data Engine

To start the IBM Z Common Data Provider System Data Engine to have it stream SMF data to the Data Streamer, you use the System Data Engine started task.

Before you begin

Create the System Data Engine started task, as described in [“Creating the System Data Engine started task for streaming SMF data”](#) on page 135.

About this task

You use z/OS console commands to query the status of the System Data Engine and control its operation.

Troubleshooting tip: After the System Data Engine is started, it should not stop until you stop it. If it does stop without your stopping it explicitly, review the System Data Engine job log output for possible errors.

Procedure

To run the System Data Engine, issue the following console commands, where *procname* represents the name of the started task (such as HBOSMF).

Action	z/OS console command
Start the System Data Engine	START <i>procname</i>
Stop the System Data Engine	STOP <i>procname</i>
View System Data Engine status	MODIFY <i>procname</i> , DISPLAY STATUS The status of the System Data Engine is written to the System Data Engine HBOOUT file.

Chapter 7. Sending user application data to the Data Streamer

The IBM Z Common Data Provider Open Streaming API provides an efficient way to gather operational data from your own applications by enabling your applications to be data gatherers. You can use the API to send your application data to the Data Streamer and stream it to analytics platforms.

Before you begin

Create your application stream definition, as described in [“Creating an application data stream definition” on page 94](#).

About this task

For sending your application data to the Data Streamer, IBM Z Common Data Provider provides the Data Transfer Protocol, and Java and REXX APIs that implement the Data Transfer Protocol. When the Data Streamer receives a data packet, it processes and sends the data to subscribers, based on the policy that you define in the Configuration Tool.

The Data Streamer has an open TCP/IP port on which it accepts connections. It accepts connections only from data gatherers that are running on the same system, and using the same TCP/IP stack.

Tip: For more information about the Data Streamer port, see [“Data Streamer port definition” on page 10](#).

Data Transfer Protocol

The Data Transfer Protocol is used to transfer data among IBM Z Common Data Provider components. It uses a binary, self-describing format that is delivered over TCP/IP. Data that is sent by using the Data Transfer Protocol can be split or unsplit.

Header information for data that is sent by using the Data Transfer Protocol

The transmission of data is preceded by the following information, in the following order:

1. A header that has a length of 96 bytes. Headers are listed and described in [Table 38 on page 283](#).
2. One of the following payload structures, which is described in the header:
 - [“Unsplit payload” on page 284](#)
 - [“Split payload” on page 285](#)

Number of bytes	Type	Description	Value
4	Binary	Endianness	0x12345678

Table 38. Headers for data that is sent by using the Data Transfer Protocol (continued)

Number of bytes	Type	Description	Value
8	Text	Header encoding	The name of an encoding to use for all other text in the header, and for metadata in the payload. The encoding must be supported by Java. The name should be padded with blanks to 8 characters and encoded in UTF-8. It is typically one of the following values: <ul style="list-style-type: none"> • IBM037 • IBM1047 • UTF-8
8	Text	Eye catcher	HBOCDP (encoded in the header encoding)
8	Text	Sender identifier	A unique identifier for the sender (encoded in header encoding)
4	Binary	Version	0x00000001
8	Reserved		
4	Binary	Payload type	<ul style="list-style-type: none"> • For unsplit data, 0x00000001 • For split data, 0x00000002
4	Binary	Payload length	The number of bytes in the payload. The maximum value is 2000000000.
48	Reserved		

Unsplit payload

To transmit unsplit data, use the payload format that is shown in [Table 39 on page 284](#).

Table 39. Unsplit payload format

Number of bytes	Type	Description
4	Binary	Number of metadata values
16 times the number of metadata values	Binary	Metadata keyword and value lengths and offsets. For each metadata value, the following information is included: <ul style="list-style-type: none"> • 4 bytes, which is the offset from the beginning of the payload to the metadata keyword • 4 bytes, which is the length of the metadata keyword • 4 bytes, which is the offset from the beginning of the payload to the metadata value • 4 bytes, which is the length of the metadata value

Table 39. Unsplit payload format (continued)

Number of bytes	Type	Description
4	Binary	Offset from the beginning of the payload to the data
4	Binary	Length of data
Variable	Text	<p>Metadata keywords and values.</p> <p>Important: this data must be encoded with the encoding specified in the header.</p> <p>Tip: The lengths and offsets of these keywords and values are previously described in this payload. For more information about the metadata keywords and values, see Table 41 on page 286.</p>
Variable	Text	<p>The data that is being transmitted</p> <p>Important: this data must be encoded with the encoding specified in the metadata.</p>

Split payload

To transmit split data, use the payload format that is shown in [Table 40 on page 285](#).

Table 40. Split payload format

Number of bytes	Type	Description
4	Binary	Number of metadata values
16 times the number of metadata values	Binary	<p>Metadata keyword and value lengths and offsets. For each metadata value, the following information is included:</p> <ul style="list-style-type: none"> • 4 bytes, which are offset from the beginning of the payload to the metadata keyword • 4 bytes, which is the length of the metadata keyword • 4 bytes, which are offset from the beginning of the payload to the metadata value • 4 bytes, which is the length of the metadata value
4	Binary	Number of records in the data
8 times the number of records in the data	Binary	<p>Record offsets and lengths. For each record, the following information is included:</p> <ul style="list-style-type: none"> • 4 bytes, which are offset from the beginning of the payload to the record • 4 bytes, which is the length of the record
4	Binary	Offset from the beginning of the payload to the data
4	Binary	Length of data

Table 40. Split payload format (continued)

Number of bytes	Type	Description
Variable	Text	<p>Metadata keywords and values.</p> <p>Important: this data must be encoded with the encoding specified in the header.</p> <p>Tip: The lengths and offsets of these keywords and values are previously described in this payload. For more information about the metadata keywords and values, see Table 41 on page 286.</p>
Variable	Text	<p>The data that is being transmitted.</p> <p>Important: this data must be encoded with the encoding specified in the metadata.</p> <p>Tip: Based on the lengths and offsets, you can have data in this field that is not included in any record.</p>

Metadata keywords and values

Table 41 on page 286 lists and describes the expected metadata keywords and values.

Table 41. Metadata keywords and values

Keyword	Value	Indication of whether the keyword is required or optional
encoding	<p>The character encoding of the data, which is typically one of the following values:</p> <ul style="list-style-type: none"> • IBM037 • IBM1047 • UTF-8 	Required
path	<p>The path of the data. This value must be the same as the File Path of the data stream that the data is associated with. This value can be found on the Configure data stream dialog in the Configuration Tool.</p>	Required
sourcetype	<p>The source type of the data. This value must be the same as the Data Source Type of the data stream that the data is associated with. This value can be found on the Configure data stream dialog in the Configuration Tool.</p>	Required
sourcename	<p>The source name of the data. This value must be the same as the Data Source Name of the data stream that the data is associated with. This value can be found on the Configure data stream dialog in the Configuration Tool.</p>	Required

Table 41. Metadata keywords and values (continued)

Keyword	Value	Indication of whether the keyword is required or optional
timezone	<p>If the timestamp in the data does not include a time zone, this value specifies a time zone to the target destination. Specify this value if the time zone is different from the system time zone.</p> <p>This value must be in the following format, where <i>plus_or_minus</i> represents the + or - sign, <i>HH</i> represents two digits for the hour, and <i>MM</i> represents two digits for the minute:</p> <pre>plus_or_minusHHMM</pre>	Optional

Sending data by using the Java API

The IBM Z Common Data Provider Java API is a set of Java classes that IBM Z Common Data Provider uses to exchange data internally. You can use these classes to write Java applications that send data to the Data Streamer.

About this task

To send data, the API must have the port number on which the Data Streamer listens for data.

Tip: For more information about the Data Streamer port, see [“Data Streamer port definition”](#) on page 10.

Procedure

To use the Java API to send data to the Data Streamer, complete the following steps:

1. Extract the `/DS/LIB/CDPzLibraries.tar` file, and add the `CdpCommon.jar` and `CdpProtocol.jar` files to the Java build path. Java API documentation is included in the TAR file.
2. As shown in the following example, define a Java class for the sender, where *localhost* is the host where the Data Streamer is running. Specify the IP address of the Data Streamer host if the Data Streamer is configured to bind to a specific IP address. Because the Data Streamer and Java application must be running on the same LPAR, the IP address must be a valid IP address on the LPAR where the Java application runs. The variable *port* is the port number on which the Data Streamer listens for data:

```
CDPSender sender = new CDPSender("localhost",port);
```

3. As shown in the following example, define a variable for identifying the origin of the data in traces and dumps.

The value of *senderName*, which must have a maximum length of 8 characters, is included in the header to identify the origin of the data.

```
String senderName = "SAMPSNDR";
```

4. As shown in the following example, define a Java class for containing the metadata for the data.

This table must contain the metadata keywords and values that are described in [“Metadata keywords and values”](#) on page 286.

```
HashMap<String, String> metadata = new HashMap<String, String>(5);
metadata.put(DictionaryKey.encoding.name(), "IBM1047");
metadata.put(DictionaryKey.path.name(), "APP/MyDataStream");
metadata.put(DictionaryKey.sourcename.name(), "MyDataStream");
```

```
metadata.put(DictionaryKey.sourcetype.name(), "zOS-MyDataStream");
metadata.put(DictionaryKey.timezone.name(), "+0000");
```

5. To send the data to the Data Streamer, complete the following steps that apply, depending on whether you are sending split or unsplit data:

Option	Description
Split data	<p>a. Define a Java class for containing the records to be sent, and for adding records as they are collected, as shown in the following example:</p> <pre>List<String> records = new ArrayList<String>(); records.add(someRecord);</pre> <p>b. Send the data to the Data Streamer, as shown in the following example:</p> <pre>sender.sendType2(senderName, metadata, records);</pre>
Unsplit data	<p>a. Send the data to the Data Streamer, as shown in the following example:</p> <pre>String data = someData; sender.sendType1(senderName, metadata, data);</pre>

Important: The following Java exceptions are thrown by the `sendType2` and `sendType1` methods and must be caught:

IOException

Thrown if an I/O error occurs when connecting to or sending data to the Data Streamer.

IllegalArgumentException

Thrown when the metadata does not contain a value for encoding, or when the length of the sender name is greater than 8 characters.

UnsupportedEncodingException

Thrown when the encoding that is provided in the metadata is not supported by Java.

Sending data by using the REXX API

The IBM Z Common Data Provider REXX API is a set of REstructured eXtended eXecutor (REXX) language functions that can be used to send data to the Data Streamer.

About this task

The sample REXX program HBORS001 in the `hlq.SHBOSAMP` library illustrates how to use the REXX API as described in the following procedure.

To send data, the API must have the port number on which the Data Streamer listens for data.

Tip: For more information about the Data Streamer port, see [“Data Streamer port definition” on page 10](#).

Procedure

To use the REXX API to send data to the Data Streamer, complete the following steps:

1. In your REXX program, include the REXX procedures from the HBORDAPI sample program, which is in the `hlq.SHBOSAMP` library.
2. Optional: If the Data Streamer is configured to bind to a specific IP address, specify the IP address of the Data Streamer in the variable `hbo.host` as shown in the following example. Because the Data Streamer and REXX application must be running on the same LPAR, the IP address must be a valid IP address on the LPAR where the REXX application runs.

```
hbo.host = '9.30.243.157'
```

3. As shown in the following example, define your metadata in a stem variable that is named "*Meta.*". This table must contain the metadata keywords and values that are described in “Metadata keywords and values” on page 286, with one value for each entry in *keyword=value* format.

```
Meta.0 = 5
Meta.1 = 'encoding=IBM1047'
Meta.2 = 'path=APP/MyDataStream'
Meta.3 = 'sourcename=MyDataStream'
Meta.4 = 'sourcetype=zOS-MyDataStream'
Meta.5 = 'timezone=+0000'
```

4. As shown in the following example, define your data in a stem variable that is named "*Data.*":

```
Data.0 = 3
Data.1 = 'Record 1'
Data.2 = 'Record 2'
Data.3 = 'Record 3'
```

5. To send data to the Data Streamer, complete the following steps that apply, depending on whether you are sending data in a single transmission or multiple transmissions:

Option	Description
Single transmission	<p>To connect to the Data Streamer, send the data, and disconnect from the Data Streamer, call <code>HBO_Send_Data</code>, as shown in the following example:</p> <pre>Call HBO_Send_Data port, type, sender</pre>
Multiple transmission	<p>If you have a long running program, you can open a connection to the Data Streamer before you call <code>HBO_Send_Data</code> so that the connection remains open, and you do not have to reconnect to send more data.</p> <p>a. Call <code>HBO_Open</code>, as shown in the following example:</p> <pre>Call HBO_Open port</pre> <p>b. Call <code>HBO_Send_Data</code>, as shown in the following example, which sends the data without connecting to, or disconnecting from, the Data Streamer:</p> <pre>Call HBO_Send_Data port, type, sender</pre> <p>Tip: In this call, the value for <i>port</i> is ignored because the connection to the Data Streamer is already open.</p> <p>c. When the program completes the sending of data, call <code>HBO_Close</code> to disconnect from the Data Streamer.</p> <p>Tip: If you make these calls from multiple, different REXX subroutines, ensure that any procedure statements expose the following variables:</p> <ul style="list-style-type: none">• <code>HBO_Socket</code>• <code>hbo.</code>• <code>ecpref</code>• <code>ecname</code>

The following information describes the variables that are used in the calls:

port

The port number on which the local Data Streamer listens for data.

type

A value of 1 indicates unsplit data, and a value of 2 indicates split data.

sender

An eye catcher, with a maximum length of 8 characters, for identifying the origin of the data in traces and dumps.

Chapter 8. Loading data to IBM Db2 Analytics Accelerator for target destination IBM Z Decision Support

If your target destination is IBM Z Decision Support, you must load the z/OS operational data in batch mode from IBM Z Common Data Provider to IBM Db2 Analytics Accelerator for z/OS for use by IBM Z Decision Support.

About this task

IBM Db2 Analytics Accelerator for z/OS is a high-performance component that is tightly integrated with Db2 for z/OS. It delivers high-speed processing for complex Db2 queries to support business-critical reporting and analytics workloads.

IBM Z Common Data Provider can send System Management Facilities (SMF) data directly to IBM Db2 Analytics Accelerator for z/OS for storage, analytics, and reporting. The data is stored in IBM Db2 Analytics Accelerator for z/OS by using a database schema from IBM Z Decision Support analytics components.

The IBM Z Common Data Provider System Data Engine converts SMF data into data sets that contain the IBM Z Decision Support analytics component tables in Db2 UNLOAD format. The IBM Db2 Analytics Accelerator Loader for z/OS is then used to load the data sets directly into IBM Db2 Analytics Accelerator for z/OS.

By sending data directly to IBM Db2 Analytics Accelerator for z/OS, you gain the following advantages:

- The need to store data in Db2 for z/OS is eliminated.
- More detailed timestamp level records can be stored.
- More CPU work is eliminated from the z/OS system.
- Reporting functions benefit from the high query speeds of IBM Db2 Analytics Accelerator for z/OS.

Configuring IBM Z Decision Support for loading the data

You must configure IBM Z Decision Support in preparation for loading the z/OS operational data in batch mode from IBM Z Common Data Provider to IBM Db2 Analytics Accelerator for z/OS.

Before you begin

Apply the following updates for the following prerequisite software:

IBM Tivoli Decision Support for z/OS Version 1.8.2

APAR PI70968

IBM Db2 Analytics Accelerator for z/OS Version 5.1

One of the following two sets of PTFs are required (either PTF-2 level or PTF-3 level), as indicated:

- PTF-2 level with the following PTFs applied:
 - UI30285
 - UI30337
 - UI30740
 - UI31021
 - UI31148
 - UI31287

- UI31302
- UI31507
- UI31571
- UI31739
- UI32368
- UI32588
- UI32707
- UI32810
- UI35006
- UI35007
- UI35008
- UI35009
- UI35010
- UI35011
- UI35012
- UI37271
- UI37783
- UI37784
- UI37785
- UI37786
- UI37793
- UI37794
- UI37795
- UI37796
- UI38702
- PTF-3 level with the following PTFs applied:
 - UI33493
 - UI33603
 - UI33797
 - UI35501
 - UI36461
 - UI37053
 - UI37534
 - UI39653
 - UI39921
 - UI40892
 - UI41378
 - UI42327
 - UI42328
 - UI42329

IBM Db2 Analytics Accelerator Loader for z/OS Version 2.1

The following PTFs are required:

- UI18415

- UI20963
- UI21883
- UI22759
- UI23712
- UI26834
- UI27815
- UI33956
- UI35108
- UI36231
- UI36343
- UI38008
- UI38201
- UI38202
- UI38810
- UI38811
- UI38939
- UI38943
- UI38973
- UI39437
- UI39451
- UI39454

About this task

IBM Z Decision Support includes an analytics component for each set of tables that are supported in IBM Db2 Analytics Accelerator for z/OS. “IBM Z Decision Support analytics components that can be loaded by the System Data Engine” on page 298 lists these analytics components with their subcomponents and the names of the corresponding base components in IBM Z Decision Support.

Procedure

To configure IBM Z Decision Support for loading the data, complete the following steps:

1. Bind the Db2 plan that is used by IBM Z Decision Support by specifying one of the following BIND options:
 - QUERYACCELERATION(ELIGIBLE)
 - QUERYACCELERATION(ENABLE)

For example, if you use the default plan name DRLPLAN, the following BIND PACKAGE is used to set the query acceleration register as eligible:

```
//SYSTSIN DD *
DSN SYSTEM(DSN)
  BIND PACKAGE(DRLPLAN) OWNER(authid) MEMBER(DRLPSQLX) -
    ACTION(REPLACE) ISOLATION(CS) ENCODING(EBCDIC) -
    QUERYACCELERATION(ELIGIBLE)
  BIND PLAN(DRLPLAN) OWNER(authid) PKLIST(*.DRLPLAN.*) -
    ACTION(REPLACE) RETAIN

  RUN PROGRAM(DSNTIAD) PLAN(DSNTIAxx) -
    LIB('xxxx.RUNLIB.LOAD')
END
```

The SDRLCNTL (DRLJDBIN) job includes sample instructions for binding the plan with QUERYACCELERATION specified.

2. Modify DRLFPROF, which is the IBM Z Decision Support data set that contains user-modified parameters, to reflect the settings to apply when installing new analytics components. The following parameters in DRLFPROF provide support for the IBM Db2 Analytics Accelerator for z/OS:

def_useaot = "YES" | "NO"

"YES"

Means that the table is created as an Accelerator Only table.

"NO"

Means that the table is created in Db2 and can be used either as a Db2 table or as an IDAA_ONLY table. The default value is "NO".

def_accelerator = "xxxxxxxx"

"xxxxxxxx"

The name of the accelerator where the table resides.

def_timeint = "H" | "S" | "T"

"H"

The timestamp for tables is rounded to an hourly interval (similar to tables with a suffix of _H in other components).

"S"

The timestamp for tables is rounded to a seconds interval (similar to tables with a time field rather than a timestamp in other components).

"T"

The timestamp for tables is the actual timestamp in the SMF record (similar to tables with suffix _T). The default value is "T".

3. If you are using IBM Z Decision Support to collect and populate the component tables in Db2 for z/OS, or if you are using IBM Z Decision Support reporting, customize each new lookup table in the IBM Z Decision Support analytics components to reflect the contents of any existing lookup tables in IBM Z Decision Support.

For example, insert the same rows that are currently in the DB2_APPLICATION table into the A_DB2_APPLICATION table.

Table 42 on page 294 lists the lookup table members to customize.

Tip: If you are collecting data only into IBM Db2 Analytics Accelerator for z/OS rather than having the data reside in Db2 for z/OS, the lookup tables are configured in IBM Z Common Data Provider, as described in “Running the System Data Engine to write data in Db2 UNLOAD format” on page 295.

Member	Base component table	Analytics component table
DRLTA2AP	DB2_APPLICATION	A_DB2_APPLICATION
DRLTA2AC	DB2_ACCUMAC	A_DB2_ACCUMAC
DRLTALUG	USER_GROUP	A_USER_GROUP
DRLTALKP	KPM_THRESHOLDS	A_KPM_THRESHOLDS_L
DRLTALW2	MVS_WORKLOAD2_TYPE	A_WORKLOAD2_L
DRLTALDA	MVSPM_DEVICE_ADDR	A_DEVICE_ADDR_L
DRLTALUT	MVSPM_UNIT_TYPE	A_UNIT_TYPE_L
DRLTALMI	MVS_MIPS_T	A_MIPS_L
DRLTALSP	MVS_SYSPLEX	A_SYSPLEX_L

<i>Table 42. IBM Z Decision Support lookup table members to customize (continued)</i>		
Member	Base component table	Analytics component table
DRLTALWL	MVS_WORKLOAD_TYPE	A_WORKLOAD_L
DRLTALW2	MVS_WORKLOAD2_TYPE	A_WORKLOAD2_L
DRLTALTR	MVSPM_TIME_RES	A_TIME_RES_L

4. Install the IBM Z Decision Support analytics components that you want to use into IBM Z Decision Support.

For information about how to install components into IBM Z Decision Support, see the IBM Z Decision Support administration documentation in the [IBM Knowledge Center](#).

5. After the IBM Z Decision Support analytics components and their associated tables are created in IBM Z Decision Support, add them to IBM Db2 Analytics Accelerator for z/OS by using the Data Studio Eclipse application or by using stored procedures.

[Table 43 on page 295](#) lists sample jobs for adding tables to IBM Db2 Analytics Accelerator for z/OS.

<i>Table 43. Sample jobs for adding tables to IBM Db2 Analytics Accelerator for z/OS</i>	
Analytics component	SDRLCNTL member
Analytics - z/OS Performance	DRLJAPMA
Analytics - Db2	DRLJA2DA
Analytics - KPM CICS	DRLJAKCA
Analytics - KPM Db2	DRLJAKDA
Analytics - KPM z/OS	DRLJAKZA

6. To move the contents of the lookup tables into the IBM Db2 Analytics Accelerator for z/OS, modify and submit the SDRLCNTL members that are listed in [Table 44 on page 295](#).

<i>Table 44. Sample jobs for moving lookup table contents to IBM Db2 Analytics Accelerator for z/OS</i>	
Analytics component	SDRLCNTL member
Analytics - z/OS Performance	DRLJAPMK
Analytics - KPM Db2	DRLJAKDK
Analytics - KPM z/OS	DRLJAKZK

Running the System Data Engine to write data in Db2 UNLOAD format

The IBM Z Common Data Provider System Data Engine converts System Management Facilities (SMF) data into data sets that contain the IBM Z Decision Support analytics component tables in Db2 UNLOAD format. The IBM Db2 Analytics Accelerator Loader for z/OS is then used to load the data sets directly into IBM Db2 Analytics Accelerator for z/OS.

Procedure

To run the System Data Engine to write data in Db2 UNLOAD format, complete the following steps:

1. Copy and customize the IBM Z Common Data Provider lookup definition members in [Table 45 on page 296](#) to reflect the contents of the corresponding IBM Z Decision Support lookup tables.
For example, insert the same rows that are currently in the DB2_APPLICATION table into the A_DB2_APPLICATION table.

These lookup tables are used by the System Data Engine when generating the Db2 UNLOAD format for each table. The System Data Engine lookup tables that are defined in these members have the same names as the IBM Z Decision Support analytics component lookup tables.

HB0vrm . SHBODEFS member	Analytics component lookup table	Base component lookup table
HBOTA2AP	A_DB2_APPLICATION	DB2_APPLICATION
HBOTA2AC	A_DB2_ACCUMAC	DB2_ACCUMAC
HBOTALUG	A_USER_GROUP	USER_GROUP
HBOTALKP	A_KPM_THRESHOLDS_L	KPM_THRESHOLDS
HBOTALWL	A_WORKLOAD2_L	MVS_WORKLOAD2_TYPE
HBOTALMI	A_MIPS_L	MVS_MIPS_T
HBOTALSP	A_SYSPLEX_L	MVS_SYSPLEX
HBOTALWL	A_WORKLOAD_L	MVS_WORKLOAD_TYPE
HBOTALW2	A_WORKLOAD2_L	MVS_WORKLOAD2_TYPE
HBOTALDA	A_DEVICE_ADDR_L	MVSPM_DEVICE_ADDR
HBOTALUT	A_UNIT_TYPE_L	MVSPM_UNIT_TYPE
HBOTALTR	A_TIME_RES_L	MVSPM_TIME_RES

2. Run the System Data Engine to generate Db2 UNLOAD format for the tables that are created for the IBM Db2 Analytics Accelerator by the IBM Z Decision Support analytics components.

The HB0vrm . SHBOSAMP members that are listed in [Table 46 on page 296](#) include sample JCL jobs for generating Db2 UNLOAD format for each of the analytics component tables.

HB0vrm . SHBOSAMP member	Analytics component
HBOAPMUN	Analytics - z/OS Performance
HBOA2DUN	Analytics - Db2
HBOAKCUN	Analytics - KPM CICS
HBOAKDUN	Analytics - KPM Db2
HBOAKZUN	Analytics - KPM Z/OS

Each sample includes two steps. The first step deletes output files that are created by a prior run, and the second step allocates output files and generates Db2 UNLOAD format from a data set that contains SMF records. The second COLLECT step uses the following DD names:

- HBOIN provides control statement input to the System Data Engine. It references the following members of HB0vrm . SHBODEFS:
 - HBOLLSMF contains control statements defining the SMF log as input.
 - HBORS* members contain control statements for extracting data from SMF records.
 - HBOT* members contain control statements to define the lookup tables that are used by the System Data Engine.
 - HBOUA* members contain control statements to store the SMF data in Db2 UNLOAD format.
 - The in-stream COLLECT statement initiates System Data Engine processing.

- HBOLOG provides the input to the System Data Engine, which must be a data set that contains SMF records.
- Various UA* DD names refer to the output files to be written in Db2 UNLOAD format. The convention is that the DD name of the file matches the name of the HBOVIM.SHB0DEFS member, without the HBO prefix. Each file that is produced by a definition member is in a sequence (such as 1, 2, 3, or 4). For example, DD name UA2D11 refers to table A_DB2_SYS_PARM_I in Db2 UNLOAD format, which is the first file that is output by definition member HBOUA2D1.

Loading data to IBM Db2 Analytics Accelerator

The IBM Db2 Analytics Accelerator Loader for z/OS is used to load the data that is output from the IBM Z Common Data Provider System Data Engine directly into IBM Db2 Analytics Accelerator for z/OS.

Procedure

Run the IBM Db2 Analytics Accelerator Loader for z/OS by using the Db2 LOAD utility with the following updates:

- A DD statement that enables the loader to intercept the Db2 LOAD utility:

```
//HLODUMMY DD DUMMY
```

- A statement that directs the loader to load data into the IBM Db2 Analytics Accelerator. This statement indicates the name of the accelerator and indicates that the target is an IDAA_ONLY table, as shown in the following example:

```
//SYSIN DD *
LOAD DATA RESUME YES LOG NO INDDN input_data_set_ddname
      IDAA_ONLY ON accelerator-name
      INTO TABLE DRLxx.KPMZ_WORKLOAD_T FORMAT INTERNAL;
```

The DRLVIM.SDRLCNTL members that are listed in [Table 47 on page 297](#) include sample JCL jobs for loading Db2 UNLOAD format data for each of the analytics component tables to IBM Db2 Analytics Accelerator.

DRLVIM.SDRLCNTL member	Analytics component
DRLJAPMD	Analytics - z/OS Performance
DRLJA2DD	Analytics – Db2
DRLJAKCD	Analytics - KPM CICS
DRLJAKDD	Analytics - KPM Db2
DRLJAKZD	Analytics - KPM Z/OS

After the load is complete from the first time that you load an IDAA_ONLY table, the table must be enabled for acceleration in IBM Db2 Analytics Accelerator for z/OS. Tables can be enabled for acceleration by using the Data Studio Eclipse application, or by using stored procedures.

The DRLVIM.SDRLCNTL members that are listed in [Table 48 on page 297](#) include sample JCL jobs for using stored procedures to enable tables for acceleration.

DRLVIM.SDRLCNTL member	Analytics component
DRLJAPME	Analytics - z/OS Performance
DRLJA2DE	Analytics – Db2

Table 48. Sample jobs for enabling tables for acceleration in IBM Db2 Analytics Accelerator (continued)

DRLvzm .SDRLCNTL member	Analytics component
DRLJAKCE	Analytics - KPM CICS
DRLJAKDE	Analytics - KPM Db2
DRLJAKZE	Analytics - KPM Z/OS

Removing tables from IBM Db2 Analytics Accelerator

If you want to uninstall a component in IBM Z Decision Support that has tables that were added to IBM Db2 Analytics Accelerator for z/OS (by using job DRLJAKZA), you must first remove the tables from IBM Db2 Analytics Accelerator for z/OS.

Procedure

To remove tables from IBM Db2 Analytics Accelerator for z/OS, customize and submit one or more of the jobs in Table 49 on page 298.

Table 49. Sample jobs that are provided by IBM Z Decision Support for removing tables from IBM Db2 Analytics Accelerator for z/OS

DRLvzm .SDRLCNTL member	Analytics component
DRLJAPMR	Analytics - z/OS Performance
DRLJA2DR	Analytics – Db2
DRLJAKCR	Analytics - KPM CICS
DRLJAKDR	Analytics - KPM Db2
DRLJAKZR	Analytics - KPM Z/OS

IBM Z Decision Support analytics components that can be loaded by the System Data Engine

This reference lists the analytics components of IBM Z Decision Support that can be loaded by the IBM Z Common Data Provider System Data Engine and used for storing data directly in the IBM Db2 Analytics Accelerator for z/OS.

Table 50 on page 299 lists the analytics components with their subcomponents and the names of the corresponding base components in IBM Z Decision Support.

Table 50. IBM Z Decision Support analytics components that can be loaded by the System Data Engine

Analytics component	Subcomponents	Corresponding base component in IBM Z Decision Support
Analytics - z/OS Performance	<ul style="list-style-type: none"> • Coupling Facility (CF) • Cross System Coupling Facility (XCF) • Open MVS (OMVS) • System • Workload • I/O • Global Storage • Virtual Storage • Device • Cryptography • Application 	MVSPM
Analytics - Db2	<ul style="list-style-type: none"> • Initialization • Address Space • Buffer Pool • Acct and RespTime • Package • Data Sharing • DDF • Storage 	Db2
Analytics - KPM CICS	<ul style="list-style-type: none"> • Monitoring 	CICS Key Performance Metrics
Analytics - KPM Db2	<ul style="list-style-type: none"> • Db2 Accounting Level • Db2 Package • Db2 System Level 	Db2 Key Performance Metrics
Analytics - KPM z/OS	<ul style="list-style-type: none"> • Address Space • LPAR • Storage • Workload • Capture Ratio Workload/LPAR • Channel • Coupling Facility • Hardware Capacity • Problem Determination 	z/OS Key Performance Metrics

Analytics component tables

For each IBM Z Decision Support analytics component that can be loaded by the IBM Z Common Data Provider System Data Engine, this reference lists the associated tables, with the corresponding IBM Z Decision Support base component tables.

The tables are listed for the following analytics components:

- [“Analytics - z/OS Performance component” on page 300](#)
- [“Analytics - Db2 component” on page 302](#)
- [“Analytics - KPM CICS component” on page 302](#)
- [“Analytics - KPM Db2 component” on page 303](#)
- [“Analytics - KPM z/OS component” on page 303](#)

Analytics - z/OS Performance component

<i>Table 51. Tables for Analytics - z/OS Performance component of IBM Z Decision Support, with corresponding base component tables</i>	
Table	Corresponding base component table
A_PM_CF_I	MVSPM_CF_H
A_PM_CF_LINK_I	MVSPM_CF_LINK_H
A_PM_CF_PROC_I	MVSPM_CF_PROC_H
A_PM_CF_REQ_I	MVSPM_CF_REQUEST_H
A_PM_CF_CF_I	MVSPM_CF_TO_CF_H
A_PM_XCF_MEMBER_I	MVSPM_XCF_MEMBER_H
A_PM_XCF_PATH_I	MVSPM_XCF_PATH_H
A_PM_XCF_SYS_I	MVSPM_XCF_SYS_H
A_PM_OMVS_BUF_I	MVSPM_OMVS_BUF_H
A_PM_OMVS_FILE_I	MVSPM_OMVS_FILE_H
A_PM_OMVS_GHFS_I	MVSPM_OMVS_GHFS_H
A_PM_OMVS_HFS_I	MVSPM_OMVS_HFS_H
A_PM_OMVS_KERN_I	MVSPM_OMVS_KERN_H
A_PM_OMVS_MOUNT_I	MVSPM_OMVS_MOUNT_H
A_PM_SYS_CLUST_I	MVSPM_CLUSTER_H
A_PM_SYS_CPU_I	MVSPM_CPU_H
A_PM_SYS_CPUMT_I	MVSPM_CPUMT_H
A_PM_SYS_ENQ_I	MVSPM_ENQUEUE_H
A_PM_SYS_LPAR_I	MVSPM_LPAR_H
A_PM_SYS_SYS_I	MVSPM_SYSTEM_H
A_PM_SYS_PROD_I	MVSPM_PROD_T
A_PM_SYS_PRDINT_I	MVSPM_PROD_INT_T
A_PM_SYS_MSU_I	MVSPM_LPAR_MSU_T

Table 51. Tables for **Analytics - z/OS Performance** component of IBM Z Decision Support, with corresponding base component tables (continued)

Table	Corresponding base component table
A_PM_WL_GOAL_I	MVSPM_GOAL_ACT_H
A_PM_WL_SERVED_I	MVSPM_WLM_SERVED_H
A_PM_WL_STATE_I	MVSPM_WLM_STATE_H
A_PM_WL_WKLD_I	MVSPM_WORKLOAD_H
A_PM_WL_WKLD2_I	MVSPM_WORKLOAD2_H
A_PM_IO_DATASET_I	MVSPM_DATASET_H
A_PM_IO_VOLUME_I	MVSPM_VOLUME_H
A_PM_IO_LCU_I	MVSPM_LCU_IO_H
A_PM_GS_BMF_I	MVSPM_BMF_H
A_PM_GS_CACHE_I	MVSPM_CACHE_H
A_PM_GS_PAGEDS_I	MVSPM_PAGE_DS_H
A_PM_GS_PAGING_I	MVSPM_PAGING_H
A_PM_GS_STORAGE_I	MVSPM_STORAGE_H
A_PM_GS_STORCLS_I	MVSPM_STORCLASS_H
A_PM_GS_SWAP_I	MVSPM_SWAP_H
A_PM_GS_CACHESS_I	MVSPM_CACHE_ESS_H
A_PM_VS_VLF_I	MVSPM_VLF_H
A_PM_VS_CSASQA_I	MVSPM_VS_CSASQA_H
A_PM_VS_PRIVATE_I	MVSPM_VS_PRIVATE_H
A_PM_VS_SUBPOOL_I	MVSPM_VS_SUBPOOL_H
A_PM_DEV_CHAN_I	MVSPM_CHANNEL_H
A_PM_DEV_HSCHAN_I	MVSPM_HS_CHAN_H
A_PM_DEV_AP_I	MVSPM_DEVICE_AP_H
A_PM_DEV_DEVICE_I	MVSPM_DEVICE_H
A_PM_DEV_FICON_I	MVSPM_FICON_H
A_PM_DEV_RAID_I	MVSPM_RAID_RANK_H
A_PM_DEV_ESSLNK_I	MVSPM_ESSLINK_H
A_PM_DEV_ESSEXT_I	MVSPM_ESS_EXTENT_H
A_PM_DEV_ESSRNK_I	MVSPM_ESS_RANK_H
A_PM_DEV_PCIE_I	MVSPM_PCIE_H
A_PM_Cryp_PCI_I	MVSPM_CRYPTOPCI_H
A_PM_Cryp_CCF_I	MVSPM_CRYPTOPCF_H
A_PM_APP_APPL_I	MVSPM_APPL_H

Analytics - Db2 component

*Table 52. Tables for **Analytics - Db2** component of IBM Z Decision Support, with corresponding base component tables*

Table	Corresponding base component table
A_DB2_SYS_PARM_I	DB2_SYS_PARAMETER
A_DB2_DB_I	DB2_DATABASE_T
A_DB2_DB_BIND_I	DB2_DATABASE_T
A_DB2_DB_QIST_I	DB2_DATABASE_T
A_DB2_DB_SYS_I	DB2_SYSTEM_T
A_DB2_BP_I	DB2_BUFFER_POOL_T
A_DB2_USERTRAN_I	DB2_USER_TRAN_H
A_DB2_UT_BP_I	DB2_USER_TRAN_H
A_DB2_UT_SACC_I	DB2_USER_TRAN_H
A_DB2_UT_IDAA_I	DB2_USER_TRAN_H
A_DB2_IDAA_STAT_I	DB2_IDAA_STAT_H
A_DB2_IDAA_ACC_I	DB2_IDAA_ACC_H
A_DB2_IDAA_ST_A_I	DB2_IDAA_STAT_A_H
A_DB2_IDAA_ST_S_I	DB2_IDAA_STAT_S_H
A_DB2_PACK_I	DB2_PACKAGE_H
A_DB2_SHR_BP_I	DB2_BP_SHARING_T
A_DB2_SHR_BPAT_I	DB2_BPATTR_SHR_T
A_DB2_SHR_LOCK_I	DB2_LOCK_SHARING_T
A_DB2_SHR_INIT_I	DB2_SHARING_INIT
A_DB2_SHR_TRAN_I	DB2_US_TRAN_SHAR_H
A_DB2_DDF_I	DB2_USER_DIST_H
A_DB2_SYSTEM_I	DB2_SYSTEM_DIST_T
A_DB2_STORAGE_I	DB2_STORAGE_T
A_DB2_TRAN_IV	DB2_TRANSACTION_D
A_DB2_DATABASE_IV	DB2_DATABASE_T

Analytics - KPM CICS component

*Table 53. Tables for **Analytics - KPM CICS** component of IBM Z Decision Support, with corresponding base component tables*

Table	Corresponding base component table
A_KC_MON_TRAN_I	KPMC_MON_TRAN_H

Analytics - KPM Db2 component

*Table 54. Tables for **Analytics - KPM Db2** component of IBM Z Decision Support, with corresponding base component tables*

Table	Corresponding base component table
A_KD_UT_I	KPM_DB2_USERTRAN_H
A_KD_UT_BP_I	KPM_DB2_USERTRAN_H
A_KD_EU_I	KPM_DB2_ENDUSER_H
A_KD_EU_BP_I	KPM_DB2_ENDUSER_H
A_KD_PACKAGE_I	KPM_DB2_PACKAGE_H
A_KD_SYS_IO_I	KPM_DB2_SYSTEM_T
A_KD_SYS_TCBSRB_I	KPM_DB2_SYSTEM_T
A_KD_SYS_LATCH_I	KPM_DB2_LATCH_T
A_KD_SYS_BP_I	KPM_DB2_BP_T
A_KD_SYS_BP_SHR_I	KPM_DB2_BP_SHR_T
A_KD_SYS_ST_DBM_I	KPM_DB2_STORAGE_T
A_KD_SYS_ST_DST_I	KPM_DB2_STORAGE_T
A_KD_SYS_ST_COM_I	KPM_DB2_STORAGE_T
A_DB_SYS_DB_WF_I	KPM_DB2_DATABASE_T
A_DB_SYS_DB_EDM_I	KPM_DB2_DATABASE_T
A_DB_SYS_DB_SET_I	KPM_DB2_DATABASE_T
A_DB_SYS_DB_LOCK_I	KPM_DB2_LOCK_T

Analytics - KPM z/OS component

*Table 55. Tables for **Analytics - KPM z/OS** component of IBM Z Decision Support, with corresponding base component tables*

Table	Corresponding base component table
A_KPM_EXCEPTION_I	KPM_EXCEPTION_T
A_KZ_JOB_INT_I	KPMZ_JOB_INT_T
A_KZ_JOB_STEP_I	KPMZ_JOB_STEP_T
A_KZ_LPAR_I	KPMZ_LPAR_T
A_KZ_STORAGE_I	KPMZ_STORAGE_T
A_KZ_WORKLOAD_I	KPMZ_WORKLOAD_T
A_KZ_CHANNEL_I	KPMZ_CHANNEL_T
A_KZ_CF_I	KPMZ_CF_T
A_KZ_CF_STRUC_I	KPMZ_CF_STRUCTR_T
A_KZ_CPUMF_I	KPMZ_CPUMF_T
A_KZ_CPUMF1_I	KPMZ_CPUMF1_T

Table 55. Tables for **Analytics - KPM z/OS** component of IBM Z Decision Support, with corresponding base component tables (continued)

Table	Corresponding base component table
A_KZ_CPUMF_PT_I	KPMZ_CPUMF_PT_T
A_KZ_CPUMF1_PT_I	KPMZ_CPUMF1_PT_T
A_KZ_SRM_WKLD_I	KPMZ_SRM_WKLD_T

Analytics component views that are based on multiple tables

In some cases, multiple tables from an IBM Z Decision Support analytics component are combined into a single view. In these cases, the resulting view matches a table from an IBM Z Decision Support base component. This reference lists these analytics component views that are based on multiple tables.

Table 56. IBM Z Decision Support analytics component views that are based on multiple tables

Analytics component	View name	Analytics component tables that are used in view	Base component table on which view is based
Analytics - Db2	A_DB2_USERTRAN_IV	<ul style="list-style-type: none"> • A_DB2_USERTRAN_I • A_DB2_UT_BP_I • A_DB2_UT_SACC_I • A_DB2_UT_IDAA_ 	DB2_USER_TRAN_H
Analytics - Db2	A_DB2_DATABASE_IV	<ul style="list-style-type: none"> • A_DB2_DB_I • A_DB2_DB_BIND_I • A_DB2_DB_QIST_I 	DB2_DATABASE_T
Analytics - KPM Db2	A_KD_USERTRAN_IV	<ul style="list-style-type: none"> • A_KD_UT_I • A_KD_UT_BP_I 	KPM_DB2_USERTRAN_H
Analytics - KPM Db2	A_KD_ENDUSER_IV	<ul style="list-style-type: none"> • A_KD_EU_I • A_KD_EU_BP_I 	KPM_DB2_ENDUSER_H
Analytics - KPM Db2	A_KD_SYSTEM_IV	<ul style="list-style-type: none"> • A_KD_SYS_IO_I • A_KD_SYS_TCBSRB_I 	KPM_DB2_SYSTEM_T
Analytics - KPM Db2	A_KD_STORAGE_IV	<ul style="list-style-type: none"> • A_KD_SYS_ST_DBM_I • A_KD_SYS_ST_DST_I • A_KD_SYS_ST_COM_I 	KPM_DB2_STORAGE_T
Analytics - KPM Db2	A_KD_DATABASE_IV	<ul style="list-style-type: none"> • A_DB_SYS_DB_WF_I • A_DB_SYS_DB_EDM_I • A_DB_SYS_DB_SET_I 	KPM_DB2_DATABASE_T

Chapter 9. Troubleshooting Z Common Data Provider

This reference lists known problems that you might experience in using the IBM Z Common Data Provider and describes known solutions. It also includes information about Log Forwarder and System Data Engine logging and tracing.

Troubleshooting flowchart

Follow the steps in this troubleshooting flowchart to resolve the issue where the subscribers cannot receive data.

Log Forwarder log files

Troubleshooting information is available in log files that are generated by the Log Forwarder.

Log Forwarder log files

Log Forwarder logging information (and tracing information, if tracing is enabled) is sent to the STDERR data set on the HBOPROC job.

Significant Log Forwarder messages, such as the following messages, are also written to the console:

- Startup and shutdown messages are written as information messages.
- Certain errors are written as action messages. These messages are cleared when the error condition is resolved or when the Log Forwarder is stopped.

The level of message information that is provided on the console and in the STDERR data set is the same. However, if stack trace data is available, it is included in only the STDERR data set.

Log Forwarder: enabling tracing

For certain problems, IBM Software Support might request that you enable tracing for the Log Forwarder.

About this task

For the Log Forwarder, you can enable tracing in either of the following ways:

- Enable static tracing by using the logging configuration file
- Enable dynamic tracing by using the MVS **MODIFY** command

Enabling static tracing for the Log Forwarder

For the Log Forwarder, you can enable static tracing by using the logging configuration file.

About this task

The trace settings in the logging configuration file are applied each time that the Log Forwarder is started.

If you do not want to restart the Log Forwarder to enable tracing, use the MVS **MODIFY** command to enable dynamic tracing.

Procedure

To enable static tracing, complete the following steps:

1. Copy the `logging.properties` file from the `samples` directory to a read/write directory.
2. Edit the `logging.properties` file as instructed by IBM Software Support.
3. Update the environment configuration file to set the value of the `ZLF_LOG` environment variable to the directory where you copied the `logging.properties` file in step [“1”](#) on page 308.
4. Restart the Log Forwarder.

What to do next

When the trace settings are no longer needed, return the logging configuration file to its original contents.

Enabling dynamic tracing for the Log Forwarder

For the Log Forwarder, you can enable dynamic tracing by using the MVS **MODIFY** command.

About this task

Trace settings that are changed by using the MVS **MODIFY** command are not persisted and therefore have no effect when the Log Forwarder is restarted.

To configure trace settings that persist each time that the Log Forwarder is started, use the logging configuration file to enable static tracing.

Procedure

To enable dynamic tracing, IBM Software Support might require you to issue one or more of the following commands:

Option	Description
Set trace	<p>To set the trace level for a specific component of the Log Forwarder, issue the following system command:</p> <pre>F HBOPROC,APPL=SET,TRACE,logger,level</pre> <p>The values for <i>logger</i> and <i>level</i> are provided by IBM Software Support. Typically, <i>level</i> is one of the following three values:</p> <p>EVENT The default tracing level. This level provides limited tracing that shows detailed error and warning responses from other applications.</p> <p>DEBUG This level provides moderate tracing that shows values of significant variables at key points in the code path.</p> <p>TRACE This level provides extensive tracing that shows detailed paths through the code, including method entry and exit.</p>
Display trace	<p>a. To display the trace levels for all components of the Log Forwarder, issue the following system command:</p> <pre>F HBOPROC,APPL=DISPLAY,TRACE</pre> <p>The loggers with level values that are explicitly set are the only ones that are displayed. Typically, only the root logger has a level value set, and that value is typically EVENT (the default level). By default, all other loggers inherit their level from the root logger.</p> <p>b. To display the trace level for a specific component of the Log Forwarder, issue the following system command:</p> <pre>F HBOPROC,APPL=DISPLAY,TRACE,logger</pre> <p>The value for <i>logger</i> is provided by IBM Software Support.</p>
Clear trace	<p>a. To clear the trace levels for all components of the Log Forwarder, issue the following system command:</p> <pre>F HBOPROC,APPL=CLEAR,TRACE</pre> <p>This command sets the value of the root logger level to EVENT. It also clears the values for all other loggers so that they inherit their level from the root logger. After the enabled</p>

Option	Description
	<p>trace settings are no longer needed, use this command to return to the default tracing state.</p> <p>b. To clear the trace level for a specific component of the Log Forwarder, issue the following system command:</p> <pre data-bbox="396 344 1472 401">F HBOPROC,APPL=CLEAR,TRACE,Logger</pre> <p>The value for <i>logger</i> is provided by IBM Software Support. After a logger level is cleared, the logger inherits its level from another component.</p>

System Data Engine log files

Troubleshooting information is available in log files that are generated by the System Data Engine.

System Data Engine log files

System Data Engine logging information is in the following data sets:

- The HB00UT data set contains the general messages that are generated by the System Data Engine. There might be error messages that refer to the HBODUMP data set for more information.
- The HBODUMP data set contains diagnostic messages for the errors that the System Data Engine encounters. It also contains the tracing information if the HBODEBUG data set is not in the JCL for the System Data Engine started task.
- The HBODEBUG data set contains the tracing information if this data set exists in the JCL for the System Data Engine started task before you turn on the trace function.

System Data Engine: enabling tracing and statistics data

For certain problems, IBM Software Support might request that you enable tracing and statistics data for the System Data Engine.

About this task

For the System Data Engine, you can enable tracing and statistics data in either of the following ways:

- Enable tracing at startup by modifying the HBOIN DD.
- Enable tracing after startup by using the MVS **MODIFY** command.
- Enable statistics data after startup by using the MVS **MODIFY** command.

Enabling tracing for the System Data Engine at startup

For certain problems, IBM Software Support might request that you enable tracing for the System Data Engine at startup by modifying the HBOIN DD.

Procedure

1. To enable tracing, IBM Software Support might require you to add the trace command `DEBUG LEVEL` to the beginning of the HBOIN DD.

```
//HBOIN DD *
        DEBUG LEVEL
        ...
```

LEVEL

Is the level of tracing to be enabled. The value of level will be provided by IBM Software Support when you must enable tracing. Tracing should not be enabled otherwise.

2. Start the System Data Engine for the trace command to take effect.
3. Verify that the appropriate messages are written in the HBODEBUG data set.

Results

The tracing output is in the HBODEBUG data set if it is in the JCL for the System Data Engine started task, or in the HBODUMP data set if the HBODEBUG data set is not in the JCL for the System Data Engine started task. See the following example for specifying the HBODEBUG DD.

```
//HBODEBUG DD SYSOUT=*
```

What to do next

To disable tracing, use one of the following methods:

- Remove the trace command `DEBUG LEVEL` and recycle the started task.
- Run the following command:

```
F STCNAME, DEBUG CLEAR
```

Ensure that you remove the trace command when the tracing function is not needed.

Enabling tracing for the System Data Engine after startup

For certain problems, IBM Software Support might request that you enable tracing for the System Data Engine after startup by using the MVS **MODIFY** command.

Procedure

1. To enable tracing, IBM Software Support might require you to run a command that is similar to one of the following lines.

```
F JOBNAME, DEBUG LEVEL  
F JOBNAME, DEBUG,LEVEL
```

JOBNAME

Is the name of the job for which you want to enable tracing, for example DEV\$SDE.

LEVEL

Is the level of tracing to be enabled.

The exact syntax of the command will be provided by IBM Software Support when you must enable tracing. Tracing should not be enabled otherwise.

2. Verify that the appropriate messages are written in the HBODEBUG data set after you run each command.

Results

The tracing output is in the HBODEBUG data set if it is in the JCL for the System Data Engine started task, or in the HBODUMP data set if the HBODEBUG data set is not in the JCL for the System Data Engine started task. See the following example for specifying the HBODEBUG DD.

```
//HBODEBUG DD SYSOUT=*
```

What to do next

To disable tracing, run the following command:

```
F STCNAME, DEBUG CLEAR
```

Enabling statistics data for the System Data Engine after startup

For certain problems, IBM Software Support might request that you enable statistics data for the System Data Engine after startup by using the MVS **MODIFY** command.

About this task

For System Data Engine, statistics data shows the number of SMF records that it collects. From the data, you can see whether you have collected the records you want to collect, and how many records you collected.

Procedure

1. To enable statistics data, IBM Software Support might require you to run a command that is similar to the following:

```
F STCNAME, STATS ON
```

STCNAME

The name of System Data Engine started task for which you want to enable statistics data, for example HBOSMF.

2. Verify that the following messages are displayed in the JESMSG LG after you run the command, which indicate that the command is successful.

```
HB05006I STATS ON  
HB05006I STATISTICS IS ON
```

Results

The statistics data output is written in the HB00UT data set of the System Data Engine started task.

What to do next

To disable statistics data, run the following command:

```
F STCNAME, STATS OFF
```

Enabling dynamic tracing for the Data Streamer

For the Data Streamer, you can enable dynamic tracing by using the MVS **MODIFY** command.

About this task

Trace settings that are changed by using the MVS **MODIFY** command are not persistent and therefore have no effect when the Data Streamer is restarted.

Procedure

To enable dynamic tracing, IBM Software Support might require you to issue one or more of the following commands:

Option	Description
Set trace	<ol style="list-style-type: none">a. To set the trace level for a specific Data Streamer, issue the following system command: <pre>F HBODSPRO,APPL=SET,TRACE,level</pre>b. To set the trace level for specific data streams, issue the following system command: <pre>F HBODSPRO,APPL=SET,TRACE,level,DS,loggerName</pre>

Option	Description
	<p>c. To set the trace level for a specific component of the Data Streamer, issue the following system command:</p> <pre data-bbox="396 275 1472 323">F HBODSPRO,APPL=SET,TRACE,level,PKG,loggerName</pre> <p>For more information about the values for <i>level</i>, <i>loggerName</i>, refer to the instructions below this table.</p>
Display trace	<p>a. To display the root trace level for the Data Streamer, issue the following system command:</p> <pre data-bbox="396 491 1472 539">F HBODSPRO,APPL=DISPLAY,TRACE</pre> <p>b. To display the trace level for a specific data stream, issue the following system command:</p> <pre data-bbox="396 596 1472 644">F HBODSPRO,APPL=DISPLAY,TRACE,DS,loggerName</pre> <p>If the level is set for the <i>loggerName</i>, the Data Streamer prints the trace level directly, otherwise it prints the inherited value for the <i>loggerName</i>.</p> <p>c. To display the trace level for a specific component of the Data Streamer, issue the following system command:</p> <pre data-bbox="396 814 1472 863">F HBODSPRO,APPL=DISPLAY,TRACE,PKG,loggerName</pre> <p>If the level is set for the <i>loggerName</i>, the Data Streamer prints the trace level directly, otherwise it prints the inherited value for the <i>loggerName</i>.</p> <p>For more information about the values for <i>loggerName</i>, refer to the instructions below this table.</p>
Clear trace	<p>To clear all the trace settings of the Data Streamer, issue the following system command:</p> <pre data-bbox="358 1100 1472 1148">F HBODSPRO,APPL=CLEAR,TRACE</pre> <p>This command sets the value of the root logger level to EVENT. It also clears the values for all other loggers so that they inherit their level from the root logger. After the enabled trace settings are no longer needed, use this command to return to the default tracing state.</p> <p>Partial clear is not provided because of the inheritable level for logger. You can override a previous trace value you have set. For example, from the previous DEBUG to EVENT. After you use the dynamic tracing, you can clear all the tracing settings.</p>

Level

- EVENT

The default tracing level. This level provides limited tracing that shows detailed error and warning responses from other applications.

- PERF

This level provides some performance information besides event level.

- DEBUG

This level provides moderate tracing that shows values of significant variables at key points in the code path.

LoggerName

- If the logger type is DS for a specific data stream, the *loggerName* can be
 - *dataSourceType*

– *dataSourceType*, *dataSourceName*

The *dataSourceType* and the *dataSourceName* have the same meanings as that of a data stream in the "**Configure data stream**" window in the Configuration Tool.

The following example shows how to set debug on CICS User Messages:

```
F HBODSPRO,APPL=SET,TRACE,DEBUG,DS,zOS-CICS-MSGUSR
```

The following example shows how to set debug on a CICS User Messages data stream named GLASCICS:

```
F HBODSPRO,APPL=SET,TRACE,DEBUG,DS,zOS-CICS-MSGUSR,GLASCICS
```

If there are multiple data streams of one data source type in the policy. The *dataSourceType* affects all the data streams with this data source type. The *dataSourceType*, *dataSourceName* only affects specific data stream.

- If the logger type is PKG for a specific component, the *loggerName* is provided by IBM Software Support.

Configuration Tool issues

You can find solutions to issues that are caused by Configuration Tool errors.

Configuration Tool troubleshooting checklist

When you encounter Configuration Tool errors, go through this checklist to help determine the causes and fix the errors.

- Provide the following information to the support team for problem determination:
 - Environment information
 - z/OS version
 - ZCDP version
 - z/OSMF version or Liberty version
 - Java version
 - Browser version
 - Enable developer mode of the browser, get the content from the Console tab, and check the Network tab for the response information with failed HTTP request.
- Check the Java version. For JRE1.8.0 64bit, make sure the Java version is above SR5 FP22.
- When importing ZCDP plug-in in z/OSMF, make sure the `cdpConfig.properties` file is imported from the working directory which is created during running `savingpolicy.sh` script.
- Firefox and IE are the browsers supported by ZCDP. If you encounter issues in one browser, try the other one and see if the issue persists. By doing this we can confirm whether it is a problem specific to one browser.
- Using MVS command `/D OMVS,0` to query message queue size (IPCMSGQBYTES). Make sure the values is larger than the minimal value 20971520. If not, use the command `SETOMVS IPCMSGQBYTES=20971520` to enlarge message queue size.
- When you encounter errors while running the `savingpolicy.sh` script, check from the following aspects:
 - Whether there is enough available space in the working directory. You can run command `df .` in USS working directory to get the available size.
 - Whether the user ID has sufficient authority to the directory containing the policy files.

Subscribers are greyed out in the Subscribe to a data stream window

Subscribers are greyed out when you select the subscribers in the Subscribe to a data stream window.

Symptom

After you select the data stream and click the Subscribe button of the data stream box, the check boxes of the subscribers in the Subscribe to a data stream window are greyed out and cannot be checked.

Solution

If you are using IBM Z Common Data Provider Version 1.1.0 and Microsoft Internet Explorer 11, consider upgrading your IBM Z Common Data Provider to Version 2.1.0, or changing to another browser.

The Configuration Tool failed to load with the error message SRVE0295E

The Configuration Tool failed to load with the error message SRVE0295E: Error reported: 404.

Symptom

When you try to log in the Configuration Tool, it failed to load with the error SRVE0295E.

Cause

It is caused by the `index.html` file being missing, or the Configuration Tool user had no READ permission to this file. It is probably caused by customer's incorrect operations.

Solution

- Check whether `index.html` exists in the install directory. If not, run `Savingpolicy.sh` again.
- Check the file permissions for the `index.html` file in the install directory and working directory of the Configuration Tool. Make sure that the Configuration Tool user has READ permission.

The Configuration Tool failed to load the policy with the error message HBO6502E regarding message queue size

The Configuration Tool failed to load the policy with the error message HBO6502E Unable to get list of definition.

Symptom

The Configuration Tool failed to load the policy with the error message HBO6502E.

Cause

The message queue size is not big enough.

Solution

- If you get a message like `message queue size 262144 is less than minimum: 20971520`, you can execute the TSO command `SETOMVS IPCMSGQBYTES=20971520` and update dataset `BPXPRMxx IPCMSGQBYTES(20971520)` to enlarge the message queue.
- If the message doesn't show the reason, enable developer mode of the browser and check the Network tab for the information with failed HTTP response.

Transform boxes are missing in the Configuration Tool after upgrading IBM Z Common Data Provider from Version 1.1 to Version 2.1

Transform boxes cannot be found in the Configuration Tool after you upgrade IBM Z Common Data Provider from Version 1.1 to Version 2.1

Symptom

Transform boxes are missing in the Configuration Tool after upgrading IBM Z Common Data Provider from Version 1.1 to Version 2.1.

Cause

It works as design. In 2019 2Q PTF, the Configuration Tool had an update to merge the transform splitter information into data streaming. You no longer have to specify those transforms manually. If you upgrade from a version before 2019 2Q PTF to 2019 2Q PTF or later, you will find that the transform boxes are missing after the upgrade.

The text of the buttons or boxes doesn't appear correctly

The text of the buttons or boxes doesn't appear correctly.

Symptom

The text of the buttons or boxes doesn't appear correctly.

Cause

There might be various causes:

- Java version does not meet the basic requirement. Refer to the KC topic [z/OS system requirements for the required Java version](#).
- Browser preferred language is not English.

Solution

Possible solutions:

- Upgrade Java to the correct version.
- Upgrade IBM Z Common Data Provider to 2020 4Q release or later.

If these solutions do not work, contact IBM support.

No data streams available for creating policies after deploying the Configuration Tool on z/OSMF

After you deploy the Configuration Tool on z/OSMF, the data stream list is empty when you try to create a policy.

Symptom

After you deploy the Configuration Tool on z/OSMF, no data streams are listed when you try to create a policy.

Cause

The file `cdpConfig.properties` was imported incorrectly. It was possible that the file was imported into z/OSMF from the install directory `/usr/lpp/IBM/zcdp/v2r1m0/UI/LIB/` and not from the working directory.

Solution

Perform the following steps to import the `cdpConfig.properties` file again.

1. Run the `savingpolicy.sh` script from the install directory `/usr/lpp/IBM/zcdp/v2r1m0/UI/LIB/`
2. Remove the active `cdpConfig.properties` file from the z/OSMF import manager.
3. Import the `cdpConfig.properties` file into z/OSMF from the working directory that is created by the `savingpolicy.sh` script.

User ID of parameter **AUTHORIZED_USER** is not found

When you run the `defracf.cmd` script, the existing user ID that is specified for the parameter **AUTHORIZED_USER** is not found.

Symptom

The existing user ID that is specified for the parameter **AUTHORIZED_USER** is not found when you run the `defracf.cmd` script.

Cause

This issue happens on systems where zSecure command verifier modifies the standard output of RACF's LISTUSER command. In this instance the `defracf.cmd` script is not able to find an existing ID as configured.

Solution

1. Open the `/var/cdp-uiconfig/cdpui.properties` file.
2. Find the parameter **AUTHORIZED_USER** and delete the user ID to the right of the equal sign.
3. Rerun the `defracf.cmd` script and specify `G0` at the last prompt to run the RACF commands.
4. Manually run the following command:

```
CONNECT user_id GROUP(group_id)
```

Change `user_id` to the user ID that the `defracf.cmd` script couldn't find. Change `group_id` to the RACF group that is assigned to the **AUTHORIZED_GROUP**. This command allows the user specified on the `connect` command to access and use the configuration tool.

Failed to open the Configuration Tool on z/OSMF with the error message **HBO6501W**

The Configuration Tool was successfully deployed on z/OSMF but you are not able to open the Configuration Tool in the browser. It gives error messages HBO6501W and HBO6502E.

Symptom

When you try to open the Configuration Tool on z/OSMF, the follow messages appear:

```
ERROR: There was an error processing your request
HBO6501W Unable to load policy files from host. z/OSMF HTTP response:
status: 500 requested URL: ...Unable to generate unique CeaTso APPTAG
category: 3 rc: 99 reason: 99 details: 0: "No details were included in
```

the response"

```
HB06502E Unable to get list of definition files. z/OSMF HTTP response:
status: 500 requested URL: ...Unable to generate unique CeaTso APPTAG
category: 3 rc: 99 reason: 99 details: 0: "No details were included in
the response"
```

Cause

This issue might be caused by insufficient security. Review the z/OSMF log and look for the following messages to further identify the cause.

```
IZUG1120E: An error occurred. The common event adapter (CEA) component returned
reason code 0x32d and diagnostic codes 0x4, 0x0, 0x0, 0x0.
```

```
IZUG899E: The request could not be completed because an error occurred.
Error: "null" com.ibm.zosmf.util.tso.LauncherException: CEA TSO launcher
called function: CEA_TSO_PING, returned reason code: 0x32d, diagnostic
codes: 0x4, 0x0, 0x0, 0x0
```

CEA_TSO_PING with returned reason code: 0x32d indicates that the user is not authorized to make the request. The cause might be the CEA profiles in the SERVAUTH class not being set up correctly, or the SERVAUTH class being inactive.

Solution

Set up the following CEA profiles in the SERVAUTH class correctly, or activate the SERVAUTH class.

```
CEA.CEADOCMD
CEA.CEAPDWB.CEADELETEINCIDENT
CEA.CEAPDWB.CEAGETINCIDENT
CEA.CEAPDWB.CEASETINCIDENTINFO
CEA.CEAPDWB.CEASETPROBLEMTRACKINGNUMBER
CEA.CEAGETPS
```

These commands are found in job SYS1.SAMPLIB(CFZSEC) and are also described in APAR OA45537: "DOCUMENTATION REQUIRED FOR THE PECEA STEP IN CFZSEC".

The buttons in the Configuration Tool do not show the text but the variable name behind it

The buttons in the Configuration Tool (on z/OSMF) don't show the text but the underlying variable names.

Symptom

When you use the Configuration Tool on IBM z/OS Management Facility (z/OSMF), the buttons don't show the text but the underlying variable names.

Cause

There are several possible reasons:

- The web browser used to visit the Configuration Tool is not supported.
- JRE level is not up to date.
- Missing the file `en.i18.json`.

The following logs can be used to further investigate:

- IZUSVR1 PROC (default z/OSMF PROC) job log
- IZUGx.log

z/OSMF creates its log files at `/var/zosmf/data/logs` in z/OS UNIX System Services. This location varies depending on your z/OSMF user directory, which by default is `/var/zosmf`. Logs are named like

IZUGx.log, where x is a number in the range 0 - 9. A smaller number indicates the log is newer (that is, IZUG0.log is the most recent, and is most likely the log you should investigate).

To make the file readable, run the following command first:

```
iconv -t IBM1047 -f ISO8859-1 /var/zosmf/data/logs/IZUG0.log > /tmp/izug0.ebcdic
```

Solution

Depending on the causes, there are different solutions:

- The web browser used to visit the Configuration Tool is not supported.

The following web browsers are supported by z/OSMF, and are recommended for best results:

- Microsoft Internet Explorer Version 11
- Mozilla Firefox Version 45 or later.

You can run the environment checker tool (https://www.ibm.com/support/knowledgecenter/SSLTBW_2.1.0/com.ibm.zos.v2r1.izua300/IZUHPINFO_BrowserTroubleshooting.htm) to inspect your web browser and workstation operating system for compliance with z/OSMF requirements and recommended settings.

- JRE level is not up to date.

Run the z/OS UNIX System Services command to check the JRE level:

```
java -fullversion
```

Upgrade JRE to latest level. Here is the link to download the JRE: <https://developer.ibm.com/javasdk/downloads/>.

- Missing the file en.i18.json.

Check whether this file is missing in the build directory (where you put your build) and installation directory.

z/OSMF default directory is: /u/userid/cdpConfig/HBOCDEUI/v2r1m0/LIB.

Consider that this path brings you to a symbolic and not a physical file but if you navigate to the link with the z/OS UNIX System Services ISHELL, you get the file location the link represents.

Troubleshooting Liberty issues

Find solutions to the issues that are related to Liberty.

The Configuration Tool failed to load with the CWWK00801E message in the HBOCFGJ job log

When you load the Configuration Tool on Liberty, the page could not be loaded successfully. The CWWK00801E message is in the HBOCFGJ job log.

Symptom

When you load the Configuration Tool on Liberty, the page could not be loaded successfully. The following message is in the HBOCFGJ job log:

```
ERROR ~ CWWK00801E: Unable to initialize SSL connection. Unauthorized access was denied or security settings have expired. Exception is javax.net.ssl.SSLHandshakeException: Client requested protocol TLSv1.1 not enabled or not supported
at com.ibm.jsse2.c.a(c.java:12)
at com.ibm.jsse2.as.a(as.java:257)
at com.ibm.jsse2.as.unwrap(as.java:528)
at javax.net.ssl.SSLEngine.unwrap(SSLEngine.java:5)
```

```
at com.ibm.ws.channel.ssl.internal.SSLConnectionLink.readyInbound(SSLConnectionLink.java:560)
at internal classes"
```

Cause

There might be something wrong with the certification or keyring.

Solution

Run the following RACF commands to verify if the certificate or keyring is correct:

1. Run `RACDCERT ID(HBOSTCID) LISTRING(*)` to get the digital ring information for user HBOSTCID:

```
Ring:
>HBO.Keyring.DFLT<
Certificate Label Name          Cert Owner      USAGE          DEFAULT
-----
HBODefaultCert                ID(HBOSTCID)   PERSONAL       YES
HBOCA                          CERTAUTH       CERTAUTH       NO
```

2. Run `RACDCERT ID(HBOSTCID) LIST` to get the digital certificate information for user HBOSTCID:

```
Label: HBODefaultCert
Certificate ID: 2QjIwtbi48PjxMjC1sSFhoGkk6PDhZmj
Status: TRUST
Start Date: 2020/06/10 00:00:00
End Date: 2023/12/31 23:59:59
Serial Number:
>01<
Issuer's Name:
>CN=CDPz CA Certification_hostname<
Subject's Name:
>CN=hostname<
Signing Algorithm: sha256RSA
Key Type: RSA
Key Size: 2048
Private Key: YES
Ring Associations:
  Ring Owner: HBOSTCID
  Ring:
  >HBO.Keyring.DFLT<
```

3. Run `RACDCERT CERTAUTH LIST(LABEL('HBOCA'))` to get the digital certificate information for CERTAUTH:

```
Label: HBOCA
Certificate ID: 2QiJmZmDhZmjgcjC1sPB
Status: TRUST
Start Date: 2020/06/10 00:00:00
End Date: 2023/12/31 23:59:59
Serial Number:
>00<
Issuer's Name:
>CN=CDPz CA Certification_hostname<
Subject's Name:
>CN=CDPz CA Certification<
Signing Algorithm: sha256RSA
Key Usage: CERTSIGN
Key Type: RSA
Key Size: 2048
Private Key: YES
Ring Associations:
  Ring Owner: HBOSTCID
  Ring:
  >HBO.Keyring.DFLT<
```

If the keyring is incorrect, rerun the following RACF command:

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(HBOSTCID)
ACCESS(READ)
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('CDPz CA Certification_hostname'))
WITHLABEL('HBOCA') TRUST NOTAFTER(2023/12/31)
RACDCERT ID (HBOSTCID) GENCERT SUBJECTSDN(CN('hostname'))
WITHLABEL('HBODefaultCert') SIGNWITH(CERTAUTH LABEL('HBOCA'))
```

```

NOTAFTER(DATE(2023/12/31))
RACDCERT ADDRING(HBO.Keyring.DFLT) ID(HBOSTCID)
RACDCERT ID(HBOSTCID) CONNECT (LABEL('HBOdefaultCert'))
RING(HBO.Keyring.DFLT) DEFAULT)
RACDCERT ID(HBOSTCID) CONNECT (LABEL('HBOCA'))
RING(HBO.Keyring.DFLT) CERTAUTH)

SETROPTS RACLIST(DIGTCERT, DIGTRING) REFRESH

```

Then restart the angel server HBOCFGGA and the Configuration Tool server HBOCFGT.

If the problem still exists, make sure that the keyring name in your RACF is the same as the keyring name that is defined in `server.xml`.

Note:

- The default path of the `server.xml` file is `/var/local/CDPServer/servers/cdp_ui_server/server.xml`.
- The keyring name is case sensitive.

The angel server HBOCFGGA failed to start with RETURN CODE 0000081 REASON CODE 0594003D

The angel server for the Configuration Tool server could not start successfully.

Symptom

The angel server for the Configuration Tool server could not start successfully. The following message is in the HBOCFGGA job log:

```

BPXM047I BPXBATCH FAILED BECAUSE SPAWN (BPX1SPN) OF
WLPDIR /lib/native/zos/s390x/bbgzangl FAILED WITH
RETURN CODE 0000081 REASON CODE 0594003D.

```

Cause

The following causes are possible

- The value of the variable `WLPDIR`, which is the WebSphere Application Server for z/OS Liberty installation path, in HBOCFGGA is incorrect.
- The file extend attributes are incorrect. Issue the command `ls -lE` in the USS directory `WLPDIR/lib/native/zos/s390x` to see the extend attributes of the files in this directory. The following files must have an `a` in their extend attributes.

```

-rwxI-xI-x ap-- 1 ROOT OMVS 4096 May 28 03:37 bbgzackk
-rwxI-xI-x ap-- 1 ROOT OMVS 360448 May 28 03:37 bbgzadm
-rwxI-xI-x ap-- 1 ROOT OMVS 270336 May 28 03:38 bbgzafsm
-rwxI-xI-x aps- 1 ROOT OMVS 233472 May 28 03:38 bbgzangl
-rwxI-xI-x aps- 1 ROOT OMVS 208896 May 28 03:38 bbgzcs1
-rwxI-xI-x ap-- 1 ROOT OMVS 1064960 May 28 03:39 bbgzsafm
-rwxI-xI-x ap-- 1 ROOT OMVS 585728 May 28 03:39 bbgzscfm
-rwxI-xI-x ap-- 1 ROOT OMVS 106496 May 28 03:39 bboacall

```

Solution

For the previously mentioned possible causes, you can try the following solutions

- Make sure the value of `WLPDIR` is correct in HBOCFGGA which is in the user procedure library.
- Run the command `Extattr +a filename` to add `a` to the extend attributes for the files in `WLPDIR/lib/native/zos/s390x`.

The z/OS user ID cannot log on the Configuration Tool successfully

The z/OS user ID cannot log on the Configuration Tool successfully.

Symptom

The z/OS user ID cannot log on the Configuration Tool successfully. The following messages are in the HBOCFG job log:

```
CWWKS1100A: Authentication did not succeed for user ID user_id. An invalid user ID or password was specified.
CWWKS2907E: SAF Service IRRSIA00_CREATE did not succeed because user user_id has insufficient authority to access APPL-ID HBOCFG. SAF return code 0x00000008. RACF return code 0x00000008. RACF reason code 0x00000020.
CWIML4537E: The login operation could not be completed. The specified principal name user_id is not found in the back-end repository.
com.ibm.wsspi.security.wim.exception.PasswordCheckFailedException: CWIML4537E: The login operation could not be completed. The specified principal name user_id is not found in the back-end repository.
```

Cause

The z/OS user ID you used to access the Configuration Tool was not added to the user group AUTHORIZED_GROUP.

Solution

Run the following command to add the user ID to the user group AUTHORIZED_GROUP:

```
CONNECT user_id GROUP(group_id)
```

Change *user_id* to the user ID that you want to use to access and use the Configuration Tool. Change *group_id* to the RACF group that is assigned to the user group AUTHORIZED_GROUP which is created by the `defracf.cmd` script. The default value is HBOUSRGP.

Log Forwarder issues

You can find solutions to issues that are caused by Log Forwarder errors.

The Log Forwarder is not able to gather all data when a VSAM ESDS data set is deleted or redefined

If a VSAM ESDS data set is deleted or redefined, the Log Forwarder might not gather all the records written to the new data set.

Symptom

A VSAM ESDS data set was deleted and then redefined. As a result, not all records written to the new data set were collected by the Log Forwarder.

Cause

When collecting data from a VSAM ESDS data set, the Log Forwarder periodically reads new records beginning with the last file position that was processed in the preceding record collection. If the VSAM ESDS data set is deleted, redefined and written with new records within a short time frame, the Log Forwarder can not detect this is a newly defined data set. If the last file position in the old data set is still valid in the new data set, the records between the beginning of the new file and that file position will be skipped.

If you delete a VSAM dataset, you must wait for at least 60 seconds (1 minute) before redefining and inserting any data into this redefined dataset, so that the Log Forwarder can detect that the dataset has been deleted and redefined. Otherwise, the Log Forwarder is not able to gather all data inserted in the redefined VSAM dataset.

Solution

If you want to delete and redefine a VSAM ESDS data set whose data is gathered by the Log Forwarder, after deleting the old data set, wait at least 60 seconds (1 minute) before redefining and writing data to the new data set. The Log Forwarder can then detect this is a new data set and gather records from the beginning.

The Log Forwarder procedure ended with a message saying out of memory

The Log Forwarder procedure ended with the message JVMDUMP039I.

Symptom

The Log Forwarder procedure ended with the following message.

```
JVMDUMP039I Processing dump event "systhrow", detail "java/lang/OutOfMemoryError"
```

Cause

Not enough memory defined for Java.

Review `startup.sh` for the Log Forwarder and you can find the following line that defines the memory size:

```
JAVA_HOME/bin/java -jar -Dfile.encoding=UTF-8 CDP_BASE/DataStreamer.jar POLICY PORT TRACE
```

It might contain parameters like `-Xms2g -Xmx2g`.

Solution

Add or update the `-Xm` parameters to increase the memory to 4GB.

```
JAVA_HOME/bin/java -Xms4g -Xmx4g -jar -Dfile.encoding=UTF-8 CDP_BASE/DataStreamer.jar POLICY PORT TRACE
```

The Log Forwarder fails to warm start with the error message HBOB003E

The Log Forwarder fails to warm start with the error message HBOB003E saying the z/OS Log Forwarder cannot read progress information in the `persistent_values.dat` file.

Symptom

The Log Forwarder fails to warm start with the following error message:

```
HBOB003E The z/OS Log Forwarder cannot read progress information in the file with path name /usr/lpp/IBM/zscala/persistent_values.dat
```

Cause

The `persistent_values.dat` file cannot be deleted at Log Forwarder warm start.

The `persistent_values.dat` file is created when the Log Forwarder stops for storing resume information for all data streams that the Log Forwarder is collecting. This file is used by the Log Forwarder to know where to pick up its work at warm start. The Log Forwarder will not read this file at cold start. If this file is deleted, a new one will be generated when the Log Forwarder stops.

Solution

Perform the following steps:

1. Make sure that the Log Forwarder user ID has read and write permission to the Log Forwarder working directory.
2. Rename or delete the `persistent_values.dat` file.
3. Cold start the Log Forwarder.

The next time you stop the Log Forwarder, the `persistent_values.dat` file will be created again, and deleted at Log Forwarder warm start.

Log Forwarder cannot gather job logs due to spool files purge

Operations automation and tools that archive JES2 and JES3 spool files can affect how the Log Forwarder gathers job logs. If a JES spool file is purged in a short interval, the Log Forwarder cannot gather job logs.

Symptom

Spool files are purged, and the Log Forwarder cannot gather job logs.

Cause

The spool files might be purged in one of the following ways:

- Spool files are purged automatically in a short interval (seconds).
- All spool files are purged automatically according to output class.
- Software products are used to off-load spool files to another location immediately upon creation, for example, SAVRS (a product of Software Engineering of America (SEA (tm))).

If spool files are purged in a short interval, the following problems might occur and prevent the Log Forwarder from collecting job logs:

- The job log spool file cannot be detected, though the configured filters are matched.
- An error occurs while the spool file is read, or the spool file cannot be read, which results in data missing.
- An abend occurs in the Log Forwarder with the following message in the SYSLOG:

```
IEC292I CLOSE MACRO MAY HAVE BEEN USED WITH INCONSISTENT MODE SPECIFICATIONS
```

and an abend code occurs:

```
SYSTEM COMPLETION CODE=378 REASON CODE=0000001C
```

Solution

When you select data sources during configuration, consider how your operations impact the job logs that you are gathering from the JES spool. For all the job logs that the Log Forwarder is configured to collect, allow files to reside in the spool long enough for the Log Forwarder to gather its contents.

A best practice is to allow files to reside in the spool for a minimum of two minutes before off-loading or purging them.

Log Forwarder user ID has insufficient authority

The Log Forwarder does not operate correctly due to insufficient authority.

Symptom

The following symptoms are possible:

- Messages in the procedure STDERR data set indicate that the `startup.sh` script cannot be found.
- System Authorization Facility (SAF) messages indicate that the user has insufficient authority to complete an operation. For example, the ICH408I message is issued by the Resource Access Control Facility (RACF) for authority issues.

Solution

Verify that the user ID that is associated with the Log Forwarder has the appropriate authority to access the Log Forwarder files and directories.

Log Forwarder cannot generate RACF PassTicket with return code and reason codes `SafRc=8, racfRc=8 racfRsn=16`

Log Forwarder cannot generate RACF PassTicket.

Symptom

A Log Forwarder message states that the RACF PassTicket cannot be generated for a user. The return code and reason codes are `SafRc=8, racfRc=8 racfRsn=16`.

Cause

The user ID that is associated with the Log Forwarder start procedure has insufficient authority to use the RACF PassTicket service.

Solution

Grant the RACF permissions to the user ID to generate PassTickets.

1. In RACF, define a profile in the PTKTDATA class controlling access to the PassTicket services and explicitly set the universal access authority to NONE:

```
RDEFINE PTKTDATA IRRPTAUTH.GPMSERVE.* UACC(NONE)
```

2. The user ID must have update permission to the new profile:

```
PERMIT IRRPTAUTH.GPMSERVE.* CLASS(PTKTDATA) ID(user) ACCESS(UPDATE)
```

where *user* is the user ID associated with the Log Forwarder start procedure.

3. Run the following command to make your changes effective:

```
SETROPTS RACLIST(PTKTDATA) REFRESH
```

Log Forwarder failed to parse the data returned from the RMF Distributed Data Server with an HTTP response code 401

Log Forwarder cannot parse the data returned from the RMF Distributed Data Server.

Symptom

A Log Forwarder message states that it cannot parse the data returned from the RMF Distributed Data Server. The HTTP response code is 401.

Cause

The user ID that is associated with the Log Forwarder start procedure has insufficient authority to send HTTP request to the RMF Distributed Data Server.

Solution

If the user ID that is associated with the Log Forwarder has a password, configure the RMF Distributed Data Server to enable PassTicket.

1. In RACF, activate the RACF class PTKDATA:

```
SETROPTS CLASSACT(PTKDATA)
SETROPTS RACLIST(PTKDATA)
```

2. Define a DDS application profile with an associated encryption key:

```
RDEFINE PTKDATA GPMSERVE SSIGNON(KEYMASKED(key))
```

Where *key* is a user-supplied 16-digit value used to generate the PassTicket. You can specify a value of your choice. Valid characters are integers from 0 to 9 and letters from A to F.

If the user ID that is associated with the Log Forwarder is a protected user that does not have a password, complete the following actions:

1. Specify the host name or TCP/IP address of the Log Forwarder in the RMF Distributed Data Servers HTTP_NOAUTH option to use the HTTP interface without authentication.
2. If you need secured communication, you can set up Application Transparent-Transport Layer Security (AT-TLS) so that the Log Forwarder is authenticated through a client certificate. For more information about how to set up AT-TLS, see the technote for OMEGAMON z/OS agent <http://www-01.ibm.com/support/docview.wss?uid=swg21697224&aid=1>. When you read through the document, replace the OMEGAMON z/OS agent information with the Log Forwarder information.

Log Forwarder gets message HBOD007E and EDC8128I at startup

The Log Forwarder (HBOPROC) started task receives messages HBOD007E and EDC8128I at startup.

Symptom

The Log Forwarder started task HBOPROC receives the following messages at startup.

```
com.ibm.tivoli.unity.systemz.sender.CDPLogSenderImpl run
SEVERE: HBOD007E The z/OS Log Forwarder request to send log data to the Data Streamer for file
path SYSLOG
failed with java.net.ConnectException: EDC8128I Connection refused. (Connection refused)
```

Cause

The Data Streamer is not running.

Solution

Make sure that the Data Streamer is up and running before you start the Log Forwarder.

Log Forwarder gets a HBOPROC CLASS error at startup

At Log Forwarder procedure (HBOPROC) startup, an error regarding class occurred.

Symptom

The Log Forwarder started task HBOPROC receives the following error at startup.

```
Error: Could not find or load main class com.ibm.tivoli.unity.systemz.SystemzLogProvider
```

Cause

ZLF_HOME is pointing to the wrong directory.

Solution

Remove /samples at the end of the directory for ZLF_HOME.

```
EDIT /u/lsant/LFwork/zlf.conf Columns 00001 00072
***** ***** Top of Data *****
000001 JRELIB=/usr/lib/java_runtime
000002 JRELIB64=/usr/lib/java_runtime64
000003 REGJAR=/usr/include/java_classes/ifaedjreg.jar
000004 TZ=EST5EDT
000005 ZLF_HOME=/usr/lpp/IBM/zscala/V1R1/samples
000006 ZLF_JAVA_HOME=/usr/lpp/java/J8.0_64
000007 ZLF_LOG=/u/lsant/LFwork
000008 ZLF_WORK=/u/lsant/LFwork
000009 ERESOLVER_CONFIG=
000010 EZLF_GATHERER=
000011 EZLF_WAS_PLUGINS_ROOT=
000012 E_BPXK_SETIBMOPT_TRANSPORT=
***** ***** Bottom of Data *****
```

BPX messages precede HBO messages in the z/OS SYSLOG

In the z/OS SYSLOG, messages with the prefix BPX precede the Log Forwarder messages, which are messages with the prefix HBO.

Symptom

The first messages that you see when you start the Log Forwarder are the following messages:

```
S HBOPROC
BPXM023I (HBOLGF) HBOA001I The z/OS Log Forwarder started successfully
BPXM023I (HBOLGF) HBOA002I The z/OS Log Forwarder initialization is complete
```

Cause

The user ID that is associated with the Log Forwarder start procedure has insufficient authority.

If the appropriate authority is granted to the Log Forwarder start procedure, the BPX messages do not precede the HBO messages.

Because the BPX messages precede the HBO messages, the z/OS SYSLOG Insight Pack does not recognize the HBO messages.

Solution

If you are using the Resource Access Control Facility (RACF) as your System Authorization Facility (SAF) product, for example, either use the HBORACF sample that is provided in the SHBOSAMP data set, or complete the following steps to resolve this problem:

1. In RACF, add the BPX.CONSOLE resource to the class FACILITY by using the General Resource Profiles option in the RACF Services Option Menu.
2. In the BPX.CONSOLE profile that was created (or updated) in the preceding step, add the user ID that the Log Forwarder start procedure is associated with, and assign READ access to the user ID.
3. Issue the following command to activate your changes:

```
SETROPTS RACLIST(FACILITY) REFRESH
```

After the changes are made in RACF, the z/OS SYSLOG includes the following messages when the Log Forwarder starts:

```
S HBOPROC
HBOA001I The z/OS Log Forwarder started successfully
HBOA002I The z/OS Log Forwarder initialization is complete
```

Log Forwarder message states that PPI issued return code 24

A Log Forwarder message states that the NetView for z/OS program-to-program interface (PPI) issued return code 24.

Cause

The Log Forwarder PPI receiver failed with return code 24, which can indicate that the NetView for z/OS PPI is not active.

Solution

To start the PPI, start the NetView for z/OS subsystem.

NetView message provider HBONETV issues message HBOL004E with return code 15

The NetView message provider, which is the REXX module HBONETV, issues message HBOL004E with return code 15.

Cause

The HBONETV module received return code 15, which indicates that the Log Forwarder program-to-program interface (PPI) receiver buffer for the NetView for z/OS PPI is not yet defined. The HBONETV module stops because the PPI receiver buffer is not defined.

Solution

Start the Log Forwarder before you start the NetView message provider.

NetView message provider HBONETV issues message HBOL006E

The NetView message provider, which is the REXX module HBONETV, was started, but it logged message HBOL006E and did not gather data.

Cause

Possibly, the NetView message provider tried to gather messages by using the CZR pipeline stage. Based on settings in the **DEFAULTS CZBRWAIT** command and on the amount of data in the NetView for z/OS program, the CZR pipeline stage might time out and therefore, not return any data, which causes the HBONETV module to issue message HBOL006E.

Solution

Use one of the following solutions:

- Restart the NetView message provider (the HBONETV module) in cold start mode by specifying C for the *COMMON.HBONETV.START* variable in the CNMSTYLE member, as shown in the following example:

```
COMMON.HBONETV.START = C
```

This action forces a cold start, which causes the provider to try to gather the most recent data.

- Specify a high value for the **CZBRWAIT** command by using the NetView **DEFAULTS** or **OVERRIDE** command, and restart the HBONETV module.

If neither solution resolves the problem, contact IBM Software Support.

System Data Engine issues

You can find solutions to issues that are caused by System Data Engine errors.

System Data Engine does not start

When the System Data Engine is started, it abends with system completion code 047.

Cause

The cause might be one of the following problems:

- The SHBLOAD library is not authorized with the authorized program facility (APF). For the System Data Engine to gather SMF data, the SHBLOAD library must be authorized with APF.
- In the System Data Engine started task, an SMF in-memory resource name is specified as the source from which SMF data is to be gathered (the value for IBM_RESOURCE), but the name is incorrectly coded. Therefore, the System Data Engine assumes that the name is an SMF log stream name.

Solution

To resolve the problem, complete one or more of the following steps, as appropriate:

- Verify that the SHBLOAD library is authorized with APF. For more information, see [“Authorizing the System Data Engine with APF”](#) on page 134.
- In the System Data Engine started task, verify the value for IBM_RESOURCE. For more information, see [“Creating the System Data Engine started task for streaming SMF data”](#) on page 135.

The System Data Engine gets ABEND U006 at startup

The System Data Engine procedure HBOSMF starts and receives ABEND U006.

Symptom

The System Data Engine starts and receives the following errors:

```
IEF403I HBOSMF - STARTED - TIME=14.24.12
BPXP018I THREAD 0E97800000000000, IN PROCESS 50397308, ENDED 557
WITHOUT BEING UNDUBBED WITH COMPLETION CODE 04000006
, AND REASON CODE 00000001.
IEF450I HBOSMF HBOSMF - ABEND=S000 U0006 REASON=00000001 558

HB00181I Product registration is successful.
HB05015A Collect could not be started due to errors in parsing earlier syntax.
HB00000I The System Data Engine abnormally ended. The HBODUMP file contains more information.
Abnormal termination, user completion code = U006
In module HBOPXB10 2019.223 starting at 0E170918
at address 0E170C64
which is at offset 0000034C in the module
```

Cause

The HBOSMF procedure does not have a valid .sde file pointed in the HBOIN DD card. In the following example, the DD card referencing the Test1.sde file was commented out incorrectly.

```
//HBOIN DD *
SET IBM_SDE_INTERVAL = '1 MINUTES';
SET IBM_UPDATE_TARGET = 'PORT 51401';
SET IBM_FILE_FORMAT = 'CSV';
SET IBM_RESOURCE = 'EXIT';
/*SET IBM_RESOURCE = 'IFASMF.<resource>';
/* DD PATH='/var/local/CDPServer/cdpConfig/Test1.sde',
/* PATHDISP=(KEEP),RECFM=V,LRECL=255,FILEDATA=RECORD
/*
```

Solution

Removing the comments from the following lines:

```
// DD PATH='/var/local/CDPServer/cdpConfig/Test1.sde',  
// PATHDISP=(KEEP),RECFM=V,LRECL=255,FILEDATA=RECORD
```

The System Data Engine fails to start with SYSTEM COMPLETION CODE=DC4 REASON CODE=90041620

The System Data Engine cannot start and fails with the message SYSTEM COMPLETION CODE=DC4 REASON CODE=90041620 in the job log.

Symptom

The System Data Engine cannot start and fails with the following message in the job log.

```
SYSTEM COMPLETION CODE=DC4 REASON CODE=90041620
```

Cause

The following explanation to this message can be found at the topic: https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.4.0/com.ibm.zos.v2r4.ieah700/idg36424.htm.

- The parameter list passed to the IARCP64 BUILD service from an unauthorized caller requested an authorized option: system key, common storage, RCT ownership, MEMLIMIT=NO, MOTKN, TYPE=FIXED or DREF.

Action: Either correct the environment such that the caller is authorized, or change the options on IARCP64 BUILD such that it does not request options that require authorization.

Solution

Authorize the System Data Engine with APF according to the instructions in [“Authorizing the System Data Engine with APF”](#) on page 134.

The System Data Engine has messages in its job log saying data fields being null when streaming SMF_030 and SMF_080

When streaming SMF_030 and SMF_080, there are messages in the System Data Engine job log saying data fields being null.

Symptom

When the System Data Engine is streaming SMF_030 and SMF_080, you see the following messages in the job log:

```
Record xxxxxx in the log, of type SMF_030,  
Results in null in data field(s):  
Value for target field SMF30IDT was set to null due to error  
  
COUT Record 000A0000 Length 0  
  
Record xxxxxx in the log, of type SMF_080,  
Results in null in data field(s):  
Value for target field SMF80RSD was set to null due to error  
  
COUT Record 000A0000 Length 0
```

Cause

The problem is a known issue for some records. The result output of the field is blank when the problem occurs. This message is just an informational message, which means that the content in SMF does not fit the format of the required fields. The System Data Engine cannot always predict the contents in SMF records. If these fields have invalid content or not initialized, they are ignored. The System Data Engine handles only the fields that have valid record length with some useful information.

Solution

These messages do not affect the data collection and can be ignored.

The message HBO0308I shows up frequently when collecting data with the System Data Engine

The message HBO0308I shows up frequently when collecting data with the System Data Engine.

Symptom

The message HBO0308I shows up frequently when the System Data Engine is collecting data.

Cause

HBO0308I is an informational message and provides no cause for concern. What this message says is that the specified buffer was filled, and the data sent. This results in more frequent and smaller packets of data flowing into the Data Streamer, which should help prevent the Data Streamer from being overwhelmed by data.

Solution

To avoid this message, you can perform the following tasks:

- Reduce the data collection interval by setting the System Data Engine started task parameter `IBM_SDE_INTERVAL` to a smaller value.
- Increase the buffer size by changing the `COLLECT` statement in the System Data Engine started task.

The SMF data packets that are sent to the target subscriber are very large

When sending SMF data to the target subscriber, the data packet size is very big.

Symptom

The size of the SMF data packets that are sent to the target subscriber is very big.

Cause

The System Data Engine data collection interval is long, which, as a result, creates large data packets.

Solution

Set a shorter data collection interval for the System Data Engine by changing the value of the parameter `IBM_SDE_INTERVAL` in its started task. By default, the interval is 1 minute (`IBM_SDE_INTERVAL = '1 MINUTES'`). You can set a shorter time, for example 30 second:

```
IBM_SDE_INTERVAL = '30 SECONDS'
```

For more information about this parameter, see [“Creating the System Data Engine started task for streaming SMF data”](#) on page 135, step “4” on page 135.

The System Data Engine ends with RC=8 when collecting SMF_110_2 related records

The System Data Engine ends with RC=8 when collecting SMF_110_2 related records.

Symptom

When collecting SMF_110_2 related records, the System Data Engine ends with RC=8 and the following message in the log:

```
can not get the time filed
```

Cause

The System Data Engine is using the record procedure HBO2CIST for handling the SMF_110_2, SMF_110_3, SMF_110_4 and SMF_110_5 related types of records. But the definition of this record procedure is only in the record definition of SMF_110_3 (HBOR1103).

```
DEFINE RECORDPROC HBO2CIST
  VERSION 'CDP.RW066419'
  FOR SMF_110_2, SMF_110_3, SMF_110_4, SMF_110_5;
```

If HBOR1103 is not included in the HBOIN statement when you monitor or collect these types of records, the System Data Engine ends with RC=8.

Solution

The problem can be resolved by adding HBOR1103 in the HBOIN statement of System Data Engine job or start process. For example:

```
//HBOIN DD DISP=SHR,DSN=hldev.SHBODEFS(HBOCCSV)
// DD DISP=SHR,DSN=hldev.SHBODEFS(HBOLLSMF)
// DD DISP=SHR,DSN=hldev.SHBODEFS(HBOY1102)
// DD DISP=SHR,DSN=hldev.SHBODEFS(HBOR1103)
```

Data Streamer is not receiving data from System Data Engine

The System Data Engine started task started, but the Data Streamer is not receiving data from the System Data Engine.

Solution

Complete the following steps to determine and resolve the problem:

1. Verify that the z/OS system and relevant z/OS subsystems are configured to produce the required records, and correct the configuration as appropriate.
2. Verify that the required records are being sent to the correct SMF log stream or in-memory resource. If no records of the required type were written during the problem time period, investigate why.
3. Search HBO0319I in HBOOUT file. If you cannot find statistics data that you want to display, you can run the following command to enable the data to display.

```
F STCNAME,STATS ON
```

To disable the displaying of statistics data, run the following command:

```
F STCNAME,STATS OFF
```

4. Review HBOOUT file HBO0319I messages to confirm that the required records are being recognized by the System Data Engine. If these messages indicate that the required records are not being

recognized by the System Data Engine, use the SMF dump utility to examine the SMF data that is produced during the problem time period to determine if any records of the required type were written.

5. Review HBOOUT file HBO0326I messages to confirm what streams are being produced by the System Data Engine.
6. Confirm that the Data Streamer is correctly configured to send the required records to the subscriber.
7. Check the HBOOUT file for error messages. In the System Data Engine started task, if the port value that is set for IBM_UPDATE_TARGET is incorrect, syntax errors can occur, which results in a failure to define the System Data Engine objects that are required to process and send data. The port number must be the one on which the Data Streamer listens for data from the data gathers. For more information about the port number, see [“Configuring the Data Streamer” on page 118](#).

Stop the System Data Engine, correct the value for IBM_UPDATE_TARGET in the System Data Engine started task, and restart the System Data Engine.

For more information, see [“Creating the System Data Engine started task for streaming SMF data” on page 135](#).

Data Streamer issues

You can find solutions to issues that are caused by Data Streamer errors.

Data Streamer does not start

Message HBO6001I, which indicates that the Data Streamer started successfully, is not present in the IBM Z Common Data Provider log files, which means that the Data Streamer did not start.

Cause

The cause might be one of the following problems:

- The Data Streamer is not connected to its specified port.
- An error occurred in reading or loading the policy file.

Solution

To determine the cause, review the log files for error messages, such as the following messages, for example:

- HBO6004E
- HBO6005E
- HBO6006E

Verify that the Data Streamer port is not closed or in use. To run the Data Streamer on a different port, change the port number. For more information about this port, see [“Data Streamer port definition” on page 10](#).

Verify that the policy file is valid and correctly formatted. For more information about the policy-related files, see the following information:

- [“Setting up a working directory for the Configuration Tool” on page 29](#)
- [“Output from the Configuration Tool” on page 32](#)

Data Streamer gets message HBO6057E at startup

After you start the Data Streamer procedure, you get message HBO6057E.

Symptom

The Data Streamer started task HBODSPRO receives error messages like the following ones at startup.

```
HBO6119I Default Heap:8g Maximum Heap:8g
HBO6057E The file "/etc/izoa/S-datareceiver su z/OZ SYSC2-localhost-8088.txt" cannot be created.
HBO6001I The Data Streamer started successfully in Warm start mode.
HBO6003I Gatherer from IP address /127.0.0.1 on port 53833 connected to the Data Streamer.
HBO6059I S-datareceiver su z/OZ SYSC2-localhost-8088 connected to subscriber datareceiver su
z/OZ SYSC2 at address
HBO6003I Gatherer from IP address /xx.x.xx.xxx on port 54643 connected to the Data Streamer.
HBO6057E The file "/etc/izoa/S-datareceiver su z/OZ SYSC2-localhost-8088.txt" cannot be
created. java.io.IOException: EDC5129I No such file or directory.
```

Cause

The problem is caused by illegal characters in the name of the subscriber that is associated with the Data Receiver.

Solution

Change the Data Receiver name in the policy to remove illegal characters.

To check the Data Receiver name, perform the following steps.

1. Access the Policy in the Configuration Tool.
2. Edit the subscriber that is associated with the Data Receiver.
3. Check the name of the subscriber in the field **Name** of the **Configure subscriber** window.

In this example, the subscriber name is `datareceiver su z/OZ SYSC2` with the illegal character `/`. Rename the subscriber to something similar to `datareceiver_ZOS_SYSC2`. The Data Steamer will start properly afterward.

Data Streamer fails to start with the message JVMJ9VM015W

The Data Streamer cannot be started and produces the message JVMJ9VM015W.

Symptom

The Data Streamer cannot be started and produces the following message:

```
JVMJ9VM015W Initialization error for library j9gc29(2): Failed to instantiate heap; 4G requested
```

Cause

Virtual Storage above the bar is not enough to allocate the Heap needed by JRE environment.

Solution

Use one of the following methods to resolve this issue:

- Increase Virtual Storage above the bar by using UNIX System Services `ulimit` command from its command console where `nnnn` is the number of bytes for the Virtual Storage above the bar:

```
ulimit - A nnnn
```

You can specify 4,294,967,296, which is the number of bytes for 4GB.

- Increase the **MEMLIMIT** parameter in the JCL. You can specify the following value to allow the Data Streamer to allocate an 8GB limit for Virtual Storage above the 2GB bar.:

```
ZLOGOUT EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,MEMLIMIT=8G
```

The Data Streamer issues a message about Java out of memory when a target subscriber remains unresponsive for a long time

When a target subscriber is not available for a long time, the Data Streamer issues message HBO6113W and HBO6114W, and the message JVMDUMP039I appears in the log.

Symptom

When a target subscriber is not available for a long time, the Data Streamer issues message HBO6113W:

```
HBO6113W The percentage of Data Streamer heap storage that is being used is 87.
```

This message states how much of the Java Heap size is used, which is 87% in the above case. The Data Streamer can even start discarding events after the 90% of Heap size is exceeded as per message HBO6114W:

```
HBO6114W Because the Data Streamer was using more than 90 percent of its available memory, bytes of data intended for subscriber SPLUNK have been discarded.
```

The Data Streamer also gets the following message in the log:

```
JVMDUMP039I Processing dump event "systhrow", detail "java/lang/OutOfMemoryError" at 2019/03/28 05:10:37 - please wait.
```

Cause

Data Streamer was not designed to buffer data for long periods of time while a subscriber is unavailable. Although the code changes in PTF UA96562 make it less likely for an out of memory condition, it's still possible. With PTF UA96562 applied, the data discards attempt to prevent the out of memory condition, but the longer the subscriber is down, the greater the chance that other Data Streamer processing, or a new large block of data will cause an out of memory condition.

Solution

Make the Java heap size as large as possible and ensure the subscriber is available as much as possible. The current default size is 4GB. In order to increase (for example to the value of 8GB) the Default and Maximum Heap size the Data Streamer can use, specify the following lines in the JCL (under `//STDENV DD *`):

```
DEFAULT_HEAP=8GB  
MAXIMUM_HEAP=8GB
```

After that, make sure when you start the Data Streamer, the following message is produced:

```
HBO6119I Default Heap:8GB Maximum Heap:8GB
```

If the subscriber will be unavailable for more than several hours, you should consider stopping IBM Z Common Data Provider during the subscriber outage. When the subscriber becomes available, start the Log Forwarder with the warm start option and batch load any SMF data being collected via the System Data Engine batch procedure. You may use the message HBO6113W to automate all this batch load.

Data Streamer is not receiving data from System Data Engine

The System Data Engine started task started, but the Data Streamer is not receiving data from the System Data Engine.

Solution

Complete the following steps to determine and resolve the problem:

1. Verify that the z/OS system and relevant z/OS subsystems are configured to produce the required records, and correct the configuration as appropriate.
2. Verify that the required records are being sent to the correct SMF log stream or in-memory resource. If no records of the required type were written during the problem time period, investigate why.
3. Search HBO0319I in HBOOUT file. If you cannot find statistics data that you want to display, you can run the following command to enable the data to display.

```
F STCNAME,STATS ON
```

To disable the displaying of statistics data, run the following command:

```
F STCNAME,STATS OFF
```

4. Review HBOOUT file HBO0319I messages to confirm that the required records are being recognized by the System Data Engine. If these messages indicate that the required records are not being recognized by the System Data Engine, use the SMF dump utility to examine the SMF data that is produced during the problem time period to determine if any records of the required type were written.
5. Review HBOOUT file HBO0326I messages to confirm what streams are being produced by the System Data Engine.
6. Confirm that the Data Streamer is correctly configured to send the required records to the subscriber.
7. Check the HBOOUT file for error messages. In the System Data Engine started task, if the port value that is set for IBM_UPDATE_TARGET is incorrect, syntax errors can occur, which results in a failure to define the System Data Engine objects that are required to process and send data. The port number must be the one on which the Data Streamer listens for data from the data gathers. For more information about the port number, see [“Configuring the Data Streamer” on page 118](#).

Stop the System Data Engine, correct the value for IBM_UPDATE_TARGET in the System Data Engine started task, and restart the System Data Engine.

For more information, see [“Creating the System Data Engine started task for streaming SMF data” on page 135](#).

The Data Receiver has a high CPU usage

The Data Receiver has a high CPU usage which delays sending data to the target subscriber.

Symptom

The Data Receiver consumes continuously 100% of CPU which delays sending data to the target subscriber.

Solution

One possible reason is the command line debug logging is enabled in your Data Receiver. Check whether you started the Data Receiver with the option `-Djavax.net.debug=all` in the command line like this:

```
java -jar -Dfile.encoding=UTF-8 -Djavax.net.debug=all DataReceiver.jar
```

If yes, remove this option and start the Data Receiver by running the following command to resolve the issue:

```
java -jar -Dfile.encoding=UTF-8 DataReceiver.jar
```

Subscriber is not receiving data

The Data Streamer successfully started and is operational, but a subscriber is not receiving data.

Cause

The cause might be one of the following problems:

- The policy file is incorrect.
- The policy file is referencing an incorrect tag to the file path for a data stream. If the trace is activated, check the logs for message HBO6021E, which indicates that an incorrect tag is present in the policy file.
- The subscriber is subscribed to the wrong streams.
- The parameters that are used to connect to the subscriber are incorrect.
- The Data Streamer cannot connect to, or is trying to reconnect to, the subscriber.
- The data packets are being discarded.
- The data gatherer (Log Forwarder or System Data Engine) is not connected to the Data Streamer.

Solution

To resolve the problem, complete one or more of the following steps, as appropriate:

- Verify that the policy file is correct. Also, verify that the subscriber is subscribed to the correct data streams, and that all parameters that are used to connect to the subscriber are correctly defined.

If the policy file contains data streams that have no subscribers, the logs include message HBO6020E.

If you update the policy file, rerun the Data Streamer with the correct policy file.

- Check the logs to determine whether a connection is established between the Data Streamer and the subscriber. Message HBO6012E indicates that this connection is not established.

Verify that the subscriber host and port is correct and available for connection.

- Check the logs to determine whether a connection is established between the Data Streamer and the data gatherer (Log Forwarder or System Data Engine). Message HBO6003I indicates that this connection is established.

If the connection is established, also verify that the data gatherers are providing output to the Data Streamer.

For more information about troubleshooting the connection between the Data Streamer and the System Data Engine, see [“Data Streamer is not receiving data from System Data Engine”](#) on page 332.

syslogd message problems: inconsistencies in timestamp, or missing or misplaced messages

In the forwarded syslog daemon (syslogd) messages, you notice either that the timestamp is inconsistent, or that some messages are missing or misplaced.

Symptom

When syslogd messages are sorted by time in the output, some messages are not shown in the expected order based on the timestamp.

Cause

On the z/OS system, the TZ variable for individual applications might not be correctly set to log data to syslogd.

Solution

In the syslogd file from which you want to collect data, check the timestamp of the messages that are missing or misplaced in the output. If the timestamp is inconsistent in the syslogd file, the *TZ* variable for individual applications is probably not correctly set to log data to syslogd.

To configure syslogd to accurately record timestamp, see the information about [Starting and stopping syslogd](#) in the *z/OS* product documentation in the IBM Knowledge Center.

Logstash gets JSON parse error messages when receiving data from IBM Z Common Data Provider

Logstash gets JSON parse error messages when receiving data from IBM Z Common Data Provider

Symptom

Logstash gets the following error messages when receiving data from IBM Z Common Data Provider

```
[ERROR][logstash.codecs.json      ][main] JSON parse error, original data now in message field
{:error=>#<LogStash::Json::ParserError: Unexpected end-of-input: was expecting closing quote
for a string value

[ERROR][logstash.codecs.json      ][main] JSON parse error, original data now in message field
{:error=>#<LogStash::Json::ParserError: Unrecognized token 'RMF': was expecting ('true',
'false' or 'null')

[ERROR][logstash.codecs.json      ][main] JSON parse error, original data now in message field
{:error=>#<LogStash::Json::ParserError: Invalid numeric value: Leading zeroes not allowed
```

Solution

Edit the Logstash pipeline configuration file and change from

```
input {
  tcp {
    port => 8080
    codec => "json"
  }
}
```

to

```
input {
  tcp {
    port => 8080
    codec => "json_lines"
  }
}
```

The Logstash configuration file is in the `/etc/logstash/conf.d` directory and is named `cdpz_input.conf`.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)[®] are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

