4. Using the DB2 Analytics Accelerator Loader ISPF interface, enable SMF as a rule:
   a. On the main menu, select **Server administration**.
   b. On the Administer Accelerator Loader server panel, select **Manage Rules**.
   c. On the Event Facility (SEF) Control panel, select **SEF Rule Management**.
   d. On the Event Facility (SEF) Ruleset Entry Profile panel, accept the default value of "*" (asterisk) in the field **Display Only the Ruleset Named** and press Enter.
   f. On the Event Facility (SEF) Event Procedure List panel, in the S column beside member HLVSMFT2, type "B" and press Enter.

**Related concepts:**
- “Rules and events” on page 223
  You can create one or more rules for each event. Within each rule, you specify an action to take in response to the event.
- “Creating SMF records from the server” on page 291
  The System Management Facility (SMF) provides ways to record usage information in SMF data sets.

### Accessing data on a remote system

DB2 Analytics Accelerator Loader provides Inter Data Communications to enable access to remote data sources that are not directly accessible from the local system; for example, data from an IMS database on a remote LPAR.

### Accessing data on a remote system

DB2 Analytics Accelerator Loader provides Inter Data Communications (IDC) to enable access to remote data sources that are not directly accessible from the local system.

IDC is a TCP/IP-based protocol that allows one Accelerator Loader server to communicate with another Accelerator Loader server. You can set up multiple servers on multiple LPARs in your z System infrastructure. Once IDC is configured, the Accelerator Loader server can access data and other resources on one or more LPARs.

For example, consider a z System infrastructure that has four LPARs configured, LPARs A - D. If Adabas is running on LPAR D and you need to access Adabas from LPAR A, then you can use IDC to enable communication between those LPARs. To use IDC, you must perform the following set-up tasks for LPAR A and LPAR D:

- Install the Accelerator Loader server.
- Configure parameters on the local Accelerator Loader server and the remote server so that communications is established.

Once IDC is enabled on an LPAR, the server on that LPAR can communicate with servers on other LPARs to access the data sources. However, no IDC can occur between LPARs until the server is installed and configured.

The Accelerator Loader server allows data virtualization to run on a z System Integrated Information Processor (zIIP) specialty engine for significantly reduced...
MIPS capacity usage. Therefore, a configured server can access data sources on another LPAR with a configured server and use the zIIP processor as it processes data.

**Configuring access to data on a remote system**

To configure access from a local Accelerator Loader server to data on a remote system via Inter Data Communications (IDC), specify values for parameters in the Accelerator Loader server parameter file, `hlvidIN00`.

**Before you begin**

- The configuration of the PORT and APPLNAME in the local servers `hlvidIN00` file must match the OEPORTNUMBER and PASSTICKETAPPNAME in the remote servers `hlvidIN00`.
- When using PassTickets, the USERID must be the same on both LPARs.

**Procedure**

1. In data set `hlq.SHĽEXEC`, locate member `hlvidIN00`, where `hlvid` represents the name of the local Accelerator Loader server started task that was customized by using Tools Customizer.
2. Use `DEFINE DATABASE TYPE(SERVER)` in the local Accelerator Loader server `hlvidIN00` parameter file to indicate a remote Accelerator Loader server. Specify values for the following parameters. All parameters are required.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Valid values</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLNAME</td>
<td>1 - 8 characters</td>
</tr>
<tr>
<td>The APPLNAME (application name) that the remote server uses for PassTicket authentication. IDC uses PassTickets for authorization on the remote server. APPLNAME is required to enable PassTicket support. If APPLNAME is not specified in the definition statement, no default value is provided and PassTicket access is disabled.</td>
<td></td>
</tr>
<tr>
<td>DDFSTATUS</td>
<td>ENABLE</td>
</tr>
<tr>
<td>The DDF activation status can be altered online by using the ISPF 4-DB2 dialog panels.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DISABLE</td>
</tr>
<tr>
<td></td>
<td>DDF endpoint is not used.</td>
</tr>
<tr>
<td>LOCATION</td>
<td>1 - 16 characters</td>
</tr>
<tr>
<td>The DB2 location name for IDC connection.</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Valid values</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------</td>
</tr>
<tr>
<td>NAME</td>
<td>The four-character ID of the remote Accelerator Loader server. The ID was defined when the remote server was configured on the remote LPAR. The remote server name becomes the “DB2 subsystem ID” for the local server’s request.</td>
</tr>
<tr>
<td>PORT</td>
<td>The IP port number that is defined for DRDA access in this DDF endpoint.</td>
</tr>
<tr>
<td>TYPE</td>
<td>The type of DDF endpoint, which is a remote server.</td>
</tr>
</tbody>
</table>

**Example**

An Accelerator Loader server is set up on LPAR DEV1 to send requests to server VDBA on LPAR RS28 (whose DNS host name is also RS28).

The local server *hlvidIN00* parameter file is configured as follows:

```
DEFINE DATABASE TYPE(SERVER)
   ,
   "NAME(VDBA)"
   ,
   "LOCATION(RS2BVDBA)"
   ,
   "DDFSTATUS(ENABLE)"
   ,
   "DOMAIN(RS28)"
   ,
   "PORT(1260)"
   ,
   "APPLNAME(VDBARS28)"
```

The remote server *hlvidIN00* parameter file must be configured as follows:

```
MODIFY PARM NAME(DEPONUMBER) VALUE(1260)*
MODIFY PARM NAME(PASSTICKETAPPNAME) VALUE(VDBSRS28)*
```

The PassTicket application name must be defined to RACF on both systems in the PTKTDATA class as follows:

```
CLASS       NAME
---       -----
PTKTDATA   VDBARS28

LEVEL      OWNER  UNIVERSAL ACCESS    YOUR ACCESS   WARNING
---        ---     ---------------     ---------------
00       TSSAL    READ               READ            NO
```

**Compressing data with zEDC**

IBM z Systems Data Compression (zEDC) is an accelerated compression solution that provides high performance, low latency compression with minimal system overhead.

Accelerator Loader server uses zEDC to optimize cross-platform exchange of data by reducing network flow between the two servers when using IDC to load from a remote server.

zEDC uses an industry standard compression library that provides efficient performance with large sequential files. If you have the zEDC compression hardware device installed, you can enable the Accelerator Loader server to use it to...
compress data. If you need to trace zEDC calls for diagnostic purposes, you can enable tracing of the calls to the Server Trace facility.

**Enabling zEDC**

**Before you begin**

For system requirements to support zEDC, see the IBM documentation at https://www.ibm.com/support/knowledgecenter/SSLTBW_2.1.0/com.ibm.zos.v2r1.ieac100/RequirementsForZEnterpriseDataCompr.htm.

**Procedure**

1. Set NETWORKBUFFERSIZE on the local and remote Accelerator Loader server to a value between ZEDCMINDATASIZE and 1048512.
2. Set parameters in the hlvidIN00 configuration member as follows. For descriptions of the parameters, see the table that follows this step.

```c
/* Enable zEDC support. */
if 1 = 1 then
  do
    "MODIFY PARM NAME(ZEDCCOMPRESSION) VALUE(YES)"
    "MODIFY PARM NAME(ZEDCMINDATASIZE) VALUE(8192)"
  end
  if 1 = 1 then
    do
      "MODIFY PARM NAME(TRACEZEDCCOMPRESSION) VALUE(NO)"
      "MODIFY PARM NAME(TRACEFULLZEDC) VALUE(NO)"
    end
```

**Table 15. Parameters for enabling zEDC**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Valid values</th>
</tr>
</thead>
<tbody>
<tr>
<td>NETWORKBUFFERSIZE</td>
<td>Controls the size of the buffer used to receive blocks of data from the network. A failure will occur if a client application sends a buffer larger than the maximum size. To allow larger blocks of data to be sent to and from the client, increase the value. 256K (default) or required size</td>
</tr>
<tr>
<td>TRACEFULLZEDC</td>
<td>Traces the entire buffer, not just the first few bytes. If a minimal trace is not enough, specify YES. YES (default) Do not enable full zEDC tracing.</td>
</tr>
<tr>
<td>ZEDCCOMPRESSION</td>
<td>Enables or disables the use of the zEDC compression hardware device. If you have the zEDC compression hardware and wish to use it, specify YES. YES (default) Do not enable zEDC compression.</td>
</tr>
</tbody>
</table>
Table 15. Parameters for enabling zEDC (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Valid values</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZEDCMINDATASIZE</td>
<td>Sets the minimum amount of data that the server will compress with the zEDC hardware. Before changing the value, test with a minimum size of 8K.</td>
</tr>
</tbody>
</table>

3. Optional: If you need to trace ZEDC calls for diagnostic purposes and you enabled zEDC tracing with TRACEZEDC_COMPRESSION, to verify that zEDC is in use, look for ZED events in the Server Trace.