IBM z/OS Dynamic Test Runner

Host Configuration Guide

IBM

Note

Before using this information, be sure to read the general information under "Notices" on page 19.

Fourth edition (June 2021)

Contents

Tables

About this document

You can use Host Configuration Assistant for Z Development (HCA) to generate customized checklists for installing and configuring IBM Developer for z/OS, IBM Explorer for z/OS, and their components and companion products, such as z/OS Source Code Analysis. You can also find the configuration information for other host components such as z/OS Explorer, z/OS Debugger, and Dependency Based Build. After generating a configuration checklist, refer to this guide and the other host configuration guides for instructions.

This document discusses the configuration of IBM z/OS Dynamic Test Runner on your z/OS host system.

The following names are used in this document:

- *IBM z/OS Dynamic Test Runner* is called *Dynamic Test Runner*.
- *IBM z/OS Debugger* is called *z/OS Debugger*.
- *IBM z/OS Automated Unit Testing Framework* is called *ZUnit*.
- *Customer Information Control System Transaction Server* is called *CICSTS*, abbreviated to *CICS*®.

For the most up-to-date versions of this document, see the Developer for z/OS Knowledge Center available at https://www.ibm.com/support/knowledgecenter/SSQ2R2/rdz_welcome.html.

For the most up-to-date versions of the complete documentation, including installation instructions, white papers, podcasts, and tutorials, see the library page of the IBM Developer for z/OS website (http://www.ibm.com/support/docview.wss?uid=swg27048563).

Who should use this document

This document is intended for system programmers who are installing and configuring IBM z/OS Dynamic Test Runner.

This document lists in detail the steps that are needed to do a full setup of the product. To use this document, you must be familiar with the MVS™ host systems.

Part 1. Host Configuration Guide

Chapter 1. Planning

You can use Host Configuration Assistant for Z Development (HCA) to generate customized checklists for installing and configuring IBM Developer for z/OS, IBM Explorer for z/OS, and their components and companion products, such as z/OS Source Code Analysis. You can also find the configuration information for other host components such as z/OS Explorer, z/OS Debugger, and Dependency Based Build. After generating a configuration checklist, refer to this guide and the other host configuration guides for instructions.

Use the information in this chapter to plan the installation and deployment of Dynamic Test Runner.

For a complete list of the Dynamic Test Runner hardware and software requirements, including prerequisites and corequisites, see the Software Product Compatibility Reports (SPCR) tool (https://www.ibm.com/software/reports/compatibility/clarity/index.html).

Planning considerations

Product overview

IBM z/OS Dynamic Test Runner can intercept calls made by user application programs to various subsystems or other programs and record details about those calls. Once the data is recorded, those same user application programs can be rerun in batch, without the need for the original environment, and be fed back the values from the recorded data. Additionally, the user application programs can be modified with required changes and then rerun to help ensure they perform without adverse effects.

The benefit of this is to allow the user to automate the testing process of online transactions and batch programs, whether they are testing a single program or thousands of transactions making up an entire system test.

The z/OS Automated Unit Testing Framework (ZUnit) component of IBM Developer for z/OS provides an Eclipse-based client that can drive the services provided by Dynamic Test Runner.

To learn more about the functionality that is offered by Developer for z/OS, see the Developer for z/OS IBM Documentation at https://www.ibm.com/docs/en/adfz/developer-for-zos, or your local IBM representative.

Skill requirements

SMP/E skills are needed for a Dynamic Test Runner host installation.

The configuration of Dynamic Test Runner requires more than the typical system programming permissions and expertise, so assistance from others might be needed. Table 3 on page 4 and Table 4 on page 4 list the administrators who are needed for the required and optional customization tasks.

Time requirements

The amount of time that is required to install and configure the Dynamic Test Runner host system components depends on various factors such as these:

- The current subsystem setup
- The availability of a user, who has successfully installed IBM Developer for z/OS, to test the installation and report any problems that might occur

Experience has shown that the installation and configuration process for the Dynamic Test Runner host system requires less than one day to complete. This time requirement is for a clean installation performed by an experienced system programmer. If problems are encountered, or if the required skills are not available, the setup will take longer.

Preinstallation considerations

For detailed instructions on the SMP/E installation of the product, see *Program Directory for IBM z/OS Dynamic Test Runner (GI13-4928)*.

The IBM z/OS Dynamic Test Runner FMID is shared by multiple IBM products, so it is possible you already have installed the FMID. In this case you can reuse the existing installation and go directly to the configuration tasks.

Required resources

Dynamic Test Runner requires the allocation of the systems resources listed in Table 1 on page 4. The resources listed in Table 2 on page 4 are required for optional services. Plan to have these resources available because, depending on the policies at your site, it might take some time to get them.

Table 1. Required resources		
Resource	**Default value**	**Information**
Product registration	none	"Product enablement in IFAPRDxx" on page 7
APF authorized data set	BZU.SBZULINK	"APF authorizations in PROGxx" on page 9
LINKLIST data set	BZU.SBZULINK	"LINKLIST definitions in PROGxx" on page 10
PROCLIB update	BZUPPLAY	"PROCLIB changes" on page 10

Table 2. Optional resources		
Resource	**Default value**	**Information**
PROCLIB update	BZUP*	"PROCLIB changes" on page 10
CICS TS update	none	"Support for CICS applications" on page 13

The configuration of Dynamic Test Runner requires more than the typical system programming permissions and expertise; therefore, assistance from others might be needed. Table 3 on page 4 and Table 4 on page 4 list the administrators who are needed for the required and optional customization tasks.

Table 3. Administrators needed for required tasks		
Administrator	**Task**	**Information**
System	Typical system programmer actions are required for all customization tasks	N/A

Table 4. Administrators needed for optional tasks		
Administrator	**Task**	**Information**
Security	Define data set profiles	"Define the data set profiles" on page 17

Table 4. Administrators needed for optional tasks (continued)		
Administrator	**Task**	**Information**
CICS TS	• Update CICS region JCL • Update CICS region CSD • Update CICS region SIP • Define CICS resources	"Support for CICS applications" on page 13

Preconfiguration considerations

Product selection

The host components of Dynamic Test Runner are used by different products. When in doubt, contact the person responsible for the purchase of Dynamic Test Runner, or possibly your local IBM representative, to learn which product was purchased. With this information, you can select the correct product registration method, as pricing and available features differ for each of the following products, and the host components will register only as one of the possible products.

IBM Developer for z/OS (program number 5724-T07)
The Developer for z/OS client enhances the functionality provided by Z Open Unit Test. Activation of client features is managed by the client.

IBM Developer for z/OS Enterprise Edition (program number 5755-AC5)
The Developer for z/OS Enterprise Edition provides the same client functionality as Developer for z/OS. However, activation of client features is managed by the host, and this product provides additional debug related capabilities, such as a 3270 interface.

IBM Application Delivery Foundation for z/OS (program number 5655-AC6)
Application Delivery Foundation for z/OS provides Developer for z/OS Enterprise Edition, which is described earlier, combined with other products useful for z/OS application development activities.

IBM Z Virtual Test Platform 1.x (program number 5697-VTP)
Activates non-client features.

IBM Z Virtual Test Platform 2.x (program number 5696-VTP)
Activates non-client features.

Predeployment considerations

Dynamic Test Runner supports the cloning of an installation to a different system, thus avoiding the need for an SMP/E installation on each system.

The following data sets are mandatory for deployment to other systems. If you copied a data set to a different location, this data set must replace its counterpart in the following lists.

• BZU.SBZULINK(*)
• BZU.SBZULOAD(*)
• BZU.SBZURESPL(*)
• BZU.#CUST.CONFIG(*)
• BZU.#CUST.JCL(*)
• BZU.#CUST.PROCLIB(*)
• BZU.#CUST.XML(*)
• definitions and data sets resulting from customization jobs in BZU.#CUST.JCL

Note: BZU is the high-level qualifier used during the installation of the z/OS Dynamic Test Runner. BZU.#CUST is the default location used during the customization of the product.

User checklist

Users of Dynamic Test Runner must know the result of certain host system customizations, such as the location of the PROCLIB members. Use these checklists to gather the information needed.

Table 5. User checklist: Mandatory parts	
Customization	**Value**
Location of the BZUP* procedures if they are not in a system procedure library. The default is BZU.#CUST.PROCLIB. See the note on JCLLIB in "PROCLIB changes" on page 10.	

Table 6. User checklist: Optional parts	
Customization	**Value**
Location of the sample BZUALLOC JCL if they are not using the ZUnit component of IBM Developer for z/OS to allocate the playback data set. The default is BZU.#CUST.JCL. See "Support for CICS applications" on page 13.	

Chapter 2. Basic customization

The following customization steps are common for the various Dynamic Test Runner services. See the chapters about the specific components for their customization requirements.

Customization setup

Dynamic Test Runner contains several sample configuration files and sample JCL. To avoid overwriting your customizations when applying maintenance, copy all of these members to a different location, and customize the copy.

To create customizable copies of configuration files and configuration JCL, customize and submit the sample BZUSETUP member in the BZU.SBZUSAMP data set. The required customization steps are described within the member.

This job performs the following tasks:

- Create BZU.#CUST.CONFIG and populate it with sample configuration files.
- Create BZU.#CUST.PROCLIB and populate it with sample SYS1.PROCLIB members.
- Create BZU.#CUST.JCL and populate it with sample setup JCL.
- Create BZU.#CUST.XML and populate it with sample XML.

Notes:

- The configuration steps in this publication use the member locations created by the BZUSETUP job, unless noted otherwise. The original samples, which should not be updated, are in BZU.SBZUSAMP.
- For more details on which sample members are copied to which data set, see the comments in BZU.SBZUSAMP(BZUSETUP).
- To aid in migrating an existing setup, the comments in BZU.SBZUSAMP(BZUSETUP) also document the changes between different versions of z/OS Dynamic Test Runner.

PARMLIB changes

The following PARMLIB changes are documented in this section:

- "Product enablement in IFAPRDxx" on page 7
- "APF authorizations in PROGxx" on page 9
- "LINKLIST definitions in PROGxx" on page 10

For more information about the PARMLIB definitions listed in the next sections, see *MVS Initialization and Tuning Reference* (SA22-7592). For more information about the sample console commands, see *MVS System Commands* (SA22-7627).

Product enablement in IFAPRDxx

The host components of Dynamic Test Runner are used by different products. When in doubt, contact the person responsible for the purchase of Dynamic Test Runner, or possibly your local IBM representative, to learn which product was purchased. With this information, you can select the correct product registration method, as pricing and available features differ for each of the following products, and the host components will register only as one of the possible products.

IBM Developer for z/OS (program number 5724-T07)
  The Developer for z/OS client enhances the functionality provided by Z Open Unit Test. Activation of client features is managed by the client.

IBM Developer for z/OS Enterprise Edition (program number 5755-AC5)
The Developer for z/OS Enterprise Edition provides the same client functionality as Developer for z/OS. However, activation of client features is managed by the host, and this product provides additional debug related capabilities, such as a 3270 interface.

IBM Application Delivery Foundation for z/OS (program number 5655-AC6)
Application Delivery Foundation for z/OS provides Developer for z/OS Enterprise Edition, which is described earlier, combined with other products useful for z/OS application development activities.

IBM Z Virtual Test Platform 1.x (program number 5697-VTP)
Activates non-client features.

IBM Z Virtual Test Platform 2.x (program number 5696-VTP)
Activates non-client features.

Products to be enabled on z/OS are defined in SYS1.PARMLIB(IFAPRDxx). Define PROD=xx in the IEASYSxx parmlib member to specify which IFAPRDxx parmlib member should be used during IPL.

The z/OS Dynamic Test Runner host components will attempt to register using the following order of product definitions. The process stops on the first successful registration.

Specify the following in IFAPRDxx to define IBM Z Virtual Test Platform 2.x (product code 5696-VTP):

```
PRODUCT OWNER('IBM CORP')
        NAME('VIRT TEST PLTFM')
        ID(5696-VTP)
        VERSION(*) RELEASE(*) MOD(*)
        FEATURENAME(*)
        STATE(ENABLED)
```

Specify the following in IFAPRDxx to define IBM Z Virtual Test Platform 1.x (product code 5697-VTP):

```
PRODUCT OWNER('IBM CORP')
        NAME('VIRT TEST PLTFM')
        ID(5697-VTP)
        VERSION(*) RELEASE(*) MOD(*)
        FEATURENAME(*)
        STATE(ENABLED)
```

Specify the following in IFAPRDxx to define IBM Application Delivery Foundation for z Systems® (product code 5655-AC6):

```
PRODUCT OWNER('IBM CORP')
        NAME('IBM APP DLIV FND')
        ID(5655-AC6)
        VERSION(*) RELEASE(*) MOD(*)
        FEATURENAME(*)
        STATE(ENABLED)
```

Specify the following in IFAPRDxx to define IBM Developer for z/OS Enterprise Edition (product code 5755-AC5):

```
PRODUCT OWNER('IBM CORP')
        NAME('IBM IDz EE')
        ID(5755-AC5)
        VERSION(*) RELEASE(*) MOD(*)
        FEATURENAME(*)
        STATE(ENABLED)
```

Specify the following in IFAPRDxx to define IBM Developer for z/OS (product code 5724-T07):

```
PRODUCT OWNER('IBM CORP')
        NAME('IBM IDz')
        ID(5724-T07)
        VERSION(*) RELEASE(*) MOD(*)
        FEATURENAME(*)
        STATE(ENABLED)
```

After the `IFAPRDxx` parmlib member is updated, it can be activated dynamically (until the next IPL) with the following console command:

```
SET PROD=xx
```

Note: All product flavors of the host components register the following feature:

- DYNAMIC TEST RUN (for z/OS Dynamic Test Runner, FMID HAL6xxx)

A product that is not defined in `IFAPRDxx`, or defined with `STATE(DISABLED)` or `STATE(NOTDEFINED)` will not be selected for registration. If none of the documented products is defined in `IFAPRDxx`, z/OS Dynamic Test Runner will halt.

If you change how a product is purchased, for example, you upgrade from using a stand-alone version to the IBM Application Delivery Foundation for z Systems product bundle, you must explicitly remove the existing product definition from the in-storage tables kept by z/OS when activating the new definition. Follow this scenario to do this dynamically (without IPL):

1. In `IFAPRDxx`, define the new product as described before, and update the old product with `STATE(DISABLED)`.
2. Activate the update with operator command `SET  PROD=xx`.
3. You can now safely remove the old product definition from `IFAPRDxx`.

IBM advises against defining `IFAPRDxx` entries that have `NAME(*)` or `ID(*)` fields, because this will result in ALL z/OS applications that utilize product registration to find a match on the first test, and adhere to the related `STATE()` definition. For z/OS Dynamic Test Runner with `STATE(ENABLED)`, this means that the application will register as *IBM Z Virtual Test Platform 2.x* (product code 5696-VTP).

APF authorizations in PROGxx

In order for Dynamic Test Runner to work, BZU.SBZULINK must be made APF authorized. Additional Dynamic Test Runner libraries will require to be made APF authorized when they are added to STEPLIB DDs that already hold APF authorized data sets:

- BZU.SBZURESL must be made APF authorized for Dynamic Test Runner to be active in Db2 or IMS.
- BZU.SBZULLEP must be made APF authorized when using Dynamic Test Runner's extended Language Environment (LE) features in Db2 or IMS.

APF authorizations are defined in `SYS1.PARMLIB(PROGxx)` by default. Define `PROG=xx` in the `IEASYSxx` parmlib member to specify which PROGxx parmlib member should be used during IPL.

APF authorizations can be set dynamically (until the next IPL) with one of the following console commands, where *volser* is the volume on which the data set resides if it is not SMS-managed:

- `SETPROG APF,ADD,DSN=BZU.SBZULINK,SMS`
- `SETPROG APF,ADD,DSN=BZU.SBZULINK,VOL=volser`

Notes:

- All APF-authorized libraries allow access to privileged system routines. Ensure you have proper security controls in place for these libraries.
- Libraries are automatically APF-authorized when located in LPA, and thus do not require explicit definitions.
- If LNKAUTH=LNKLST is specified in `IEASYSxx` then all data sets in the LNKLST concatenation are treated as APF-authorized.
- If one STEPLIB library is APF-authorized, then all the other libraries in that STEPLIB must be authorized as well. Libraries lose their APF authorization when they are mixed with non-authorized libraries in STEPLIB.

LINKLIST definitions in PROGxx

In order for z/OS Dynamic Test Runner to work, BZU.SBZULINK must be made available through either STEPLIB, LINKLST, or LPA. LINKLIST is the recommended method.

The code in BZU.SBZULINK is not expected to be called frequently enough to warrant using limited LPA resources.

LINKLIST data sets are defined in SYS1.PARMLIB(PROGxx), if your site followed IBM recommendations. Define PROG=xx in the IEASYSxx parmlib member to specify which PROGxx parmlib member should be used during IPL.

The required definitions will look like the following, where listname is the name of the LINKLIST set that will be activated, and volser is the volume on which the data set resides if it is not cataloged in the master catalog:

- LNKLST ADD NAME(listname) DSNAME(BZU.SBZULINK) VOLUME(volser)
- LNKLST ADD NAME(listname) DSNAME(BZU.SBZULINK)

LINKLIST definitions can be created dynamically (until the next IPL) with the following actions:

1. Create a new PARMLIB member named PROGxx, for example PROGLL.
2. Place the following commands in your new PROGxx member, where *volser* is the volume on which the data set resides if it is not cataloged in the master catalog:

   - LNKLST DEFINE,NAME=LLTMP,COPYFROM=CURRENT
   - LNKLST ADD NAME=LLTMP,DSN=BZU.SBZULINK,VOL=volser
   - LNKLST ACTIVATE,NAME=LLTMP

3. Activate the new definitions by issuing the **SET PROG=xx** console command, where *xx* matches the last 2 characters of your new PROGxx member name.

PROCLIB changes

The JCL procedures listed here must reside in a system procedure library defined to your JES subsystem. In the instructions in the following sections, the IBM default procedure library, SYS1.PROCLIB, is used.

z/OS Dynamic Test Runner provides sample JCL procedures that can be used to activate services of the product. These procedures allow installations to apply their own standards, and ensure that developers use the same procedures with the same default options.

The sample procedures and their function are listed in .

Table 7. Sample BZUP* procedures

Member	Purpose
BZUINCL	Include member that specifies variables used by the procedures.
BZUPBCP	Record batch processing
BZUPBMP	Record IMS batch processing (IMS/DB using BMP)
BZUPDB2	Record Db2 batch processing
BZUPDCP	Create a formatted dump of a recorded playback file
BZUPDLI	Record IMS batch processing (IMS/DB using DL/I)
BZUPECP	Extract data from a recorded playback file
BZUPMCP	Create the COBOL source for a master testcase callback program
BZUPPLAY	Replay a recorded program

All procedures rely on the BZUINCL include member for the definition of common high-level qualifiers. Customize the sample include member, BZU.#CUST.PROCLIB(BZUINCL) as described within the member. You can use Table 8 on page 11 to assist with this customization.

Review the sample build procedure members, BZU.#CUST.PROCLIB(BZUP*), and customize them if required.

Table 8. BZUINCL high-level qualifier checklist		
Product	**Default HLQ**	**Value**
Dynamic Test Runner (customized)	BZU.#CUST	
Dynamic Test Runner	BZU	
Debugger	EQAW	
Language Environment (LE)	CEE	
Db2	DSN	
IMS (product)	DFS	
IMS (customized)	DFS	

Once the BZUP* members are customized, copy them to SYS1.PROCLIB. Ensure that the BZUINCL include member remains with the BZUP* procedures. If the members cannot be copied into a system procedure library, ask the Dynamic Test Runner users to add a JCLLIB card (right after the JOB card) to the job properties on the client.

```
//MYJOB    JOB <job parameters>
//PROCS    JCLLIB ORDER=(BZU.#CUST.PROCLIB)
```

Security definitions

To create the security definitions for Dynamic Test Runner, customize and submit the sample BZURACF member. The user submitting this job must have security administrator privileges, such as being RACF® SPECIAL.

BZURACF is located in BZU.#CUST.JCL, unless you specified a different location when you customized and submitted the BZU.SBZUSAMP(BZUSETUP) job. For more details, see "Customization setup" on page 7.

The following list of security-related definitions for Dynamic Test Runner are discussed in detail.

- Define data set profiles
- Verify the security settings

Specific components

Dynamic Test Runner related customizations to other products:

- "Support for CICS applications" on page 13
- "Support for Db2 applications" on page 13
- "Support for IMS applications" on page 14

Chapter 3. Other customization tasks

This section has various optional customization tasks related to other products. To configure the required service, follow the instructions in the appropriate section.

Support for CICS applications

You need the assistance of a CICS administrator to complete this customization task, which requires specific resources or special customization tasks.

- Update CICS system initialization parameters (SIP)
- Update CICS region JCL
- Define resources to CICS

This customization task extends CICS to support testing CICS applications through recording of the parameters used in EXEC CICS calls.

To record a CICS application, z/OS Dynamic Test Runner requires the following CICS updates:

- CICS system initialization parameter (SIP) updates

  - Specify RENTPGM=NOPROTECT, as documented in the BZU.SBZUSAMP(BZUSIP) sample SIP update member.

    Note: This definition is not required if you only want to collect events that can be captured using the standard CICS interfaces.

- CICS JCL updates:

  - Specify REGION=0M on the EXEC statement to provide sufficient room to buffer the recording.
  - Define the BZU.SBZULOAD load library in the region's DFHRPL DD statement, where BZU is the high-level qualifier used during the SMP/E install of z/OS Dynamic Test Runner.

- CICS CSD updates:

  Define z/OS Dynamic Test Runner to your CICS region, as documented in the BZUCSD sample CSD update job. BZUCSD is located in BZU.#CUST.JCL, unless you specified a different location when you customized and submitted job BZU.SBZUSAMP(BZUSETUP). For more information. see .

- Allocate work data sets:

  - Define a temporary holding area for recorded data, as documented in the BZUVSAM sample allocation job. BZUVSAM is located in BZU.#CUST.JCL, unless you specified a different location when you customized and submitted job BZU.SBZUSAMP(BZUSETUP). For more information. see . The allocated data set is referenced in the CICS CSD updates.
  - When users extract their data from the temporary holding area it is placed in a playback data set. While it is likely that your users will allocate this data set themselves, it can also be allocated for them using the BZUALLOC sample allocation job. BZUALLOC is located in BZU.#CUST.JCL, unless you specified a different location when you customized and submitted job BZU.SBZUSAMP(BZUSETUP). For more information. see . The allocated data set is referenced in the CICS CSD updates.

Support for Db2 applications

Note: This customization task requires the following customization:

- APF authorize SBZURESL

This customization task extends Db2 to support testing Db2 applications through recording of the parameters used in Db2 calls.

Record Db2 batch

To record a batch application accessing the Db2 Databases, Dynamic Test Runner requires no updates to your Db2 environment, but does require a relink of select Db2 load modules.

Customize and submit sample JCL BZULDB2 to relink some Db2 load modules to SBZURESL so that Dynamic Test Runner can intercept Db2 calls. BZULDB2 is located in BZU.#CUST.JCL, unless you specified a different location when you customized and submitted job BZU.SBZUSAMP(BZUSETUP). For more information, see .

Note: This relink must be done each time the related Db2 or Dynamic Test Runner load modules are updated.

The users need to use the BZUPDB2 JCL procedure to run the batch application. BZUPDB2 is located in BZU.#CUST.PROCLIB, unless you specified a different location when you customized and submitted job BZU.SBZUSAMP(BZUSETUP). For more information, see .

Note:

- If you customize an existing JCL procedure to add support for Dynamic Test Runner, ensure that Dynamic Test Runner's SBZURESL library is placed ahead of Db2's SDSNLOAD library. This applies to classic Db2 batch jobs like BZUPDB2, as well as references to SDSNLOAD in other subsystems like IMS.
- If one STEPLIB library is APF-authorized, like DB2's SDSNLOAD, then all the other libraries in that STEPLIB, like Dynamic Test Runner's SBZURESL, must be authorized as well. Libraries lose their APF authorization when they are mixed with non-authorized libraries in STEPLIB.

Support for IMS applications

Note: You need the assistance of an IMS administrator to complete this customization task, which requires specific resources or special customization tasks.

- Define the recording service to your IMS control region
- APF authorize SBZURESL
- Update IMS control region JCL
- Update IMS batch JCL

This customization task extends IMS to support testing IMS/BMP applications through recording of the parameters used in IMS/BMP calls, as well as interactions with other subsystems such as Db2.

When recording interactions with other subsystems, or when utilizing extended Language Environment (LE) functions in a recording, Dynamic Test Runner requires that the configuration tasks for those other subsystems or LE are also completed.

To record a batch IMS/BMP application, Dynamic Test Runner requires completion of the following tasks:

- Customize and submit sample JCL BZUALLOC to allocate a sequential 'playback' data set that will hold the recorded data. BZUALLOC is located in BZU.#CUST.JCL, unless you specified a different location when you customized and submitted job BZU.SBZUSAMP(BZUSETUP). For more information, see .
- Customize and submit sample JCL BZULIMS to relink some IMS load modules to the SBZURESL load library so that Dynamic Test Runner can intercept IMS calls. BZULIMS is located in BZU.#CUST.JCL, unless you specified a different location when you customized and submitted job BZU.SBZUSAMP(BZUSETUP). For more information, see .

  Note: This relink must be done each time the related IMS or Dynamic Test Runner load modules are updated.
- Customize and submit sample JCL BZUIMSTR to define the recording service to your IMS control region. BZUIMSTR is located in BZU.#CUST.JCL, unless you specified a different location when you customized

and submitted job BZU.SBZUSAMP(BZUSETUP). For more information, see .

- Add load library SBZURESL to DD STEPLIB of the IMS control region, ahead of the IMS SDFSRESL library. This implies that SBZURESL must be APF authorized to avoid that other STEPLIB libraries lose their APF authorization.
- Restart the IMS control region to activate the changes.
- Customize sample JCL procedure BZUPBMP to work with your existing IMSBATCH procedure and instruct your users to use BZUPBMP when recording a batch IMS/BMP application. DD BZUPLAY references the playback data set that was created in an earlier task using sample JCL BZUALLOC. BZUPBMP is located in BZU.#CUST.PROCLIB, unless you specified a different location when you customized and submitted job BZU.SBZUSAMP(BZUSETUP). For more information, see .

Note: If you opt to customize your existing IMSBATCH procedure to match BZUPBMP, know that DD BZUCFG is only required when program-to-program interceptions in batch IMS/BMP applications is desired. DD BZUCFG references a program-to-program interception configuration file. A sample configuration file is provided in SBZUSAMP(BZUSCFG).

Extending Language Environment

This customization task does not require assistance, special resources, or special customization tasks.

This customization task extends Language Environment (LE) to allow for actions such as replaying with the recorded timestamp, or recording and replaying file I/O.

Extend LE

A relink of select LE load modules allows Dynamic Test Runner to:

- Replay with the recorded timestamp
- Record file I/O to QSAM or VSAM (KSDS, ESDS or RRDS)
- Replay file I/O from QSAM or VSAM (KSDS, ESDS or RRDS)

Customize and submit sample JCL BZULLE to relink some LE load modules to SBZULLEP so that Dynamic Test Runner can extend LE. BZULLE is located in BZU.#CUST.JCL, unless you specified a different location when you customized and submitted job BZU.SBZUSAMP(BZUSETUP). For more information, see .

Note: This relink must be done each time the related LE or Dynamic Test Runner load modules are updated.

The users need to place the relinked load modules (located in SBZULLEP) in STEPLIB, before the normal LE load libraries (SCEERUN and SCEERUN2).

Chapter 4. Security definitions

Customize and submit the sample BZURACF job, which has sample RACF commands to create the basic security definitions for Dynamic Test Runner.

BZURACF is located in BZU.#CUST.JCL, unless you specified a different location when you customized and submitted the BZU.SBZUSAMP(BZUSETUP) job.

See the *RACF Command Language Reference (SA22–7687)* for more information about RACF commands.

Requirements and checklist

To complete the security setup, the security administrator must know the values that are listed in Table 9 on page 17. These values were defined during previous steps of the installation and customization of z/OS Dynamic Test Runner.

Table 9. Security setup variables		
Description	• **Default value** • **Where to find the answer**	**Value**
Dynamic Test Runner product high-level qualifier	• BZU • SMP/E installation	
Dynamic Test Runner customization high-level qualifier	• BZU.#CUST • BZU.SBZUSAMP(BZUSETUP), as described in "Customization setup" on page 7.	

The following list is an overview of the actions that are required to complete the basic security setup of Dynamic Test Runner. As documented in the following sections, different methods can be used to fulfill these requirements, depending on the required security level.

- "Define the data set profiles" on page 17
- "Verify the security settings" on page 18

Define the data set profiles

READ access for users and ALTER for system programmers is sufficient for most Dynamic Test Runner data sets. Replace the #sysprog placeholder with valid user IDs or RACF group names. Also, ask the system programmer who installed and configured the product for the correct data set names. BZU is the default high-level qualifier used during installation and BZU.#CUST is the default high-level qualifier for data sets created during the customization process.

-
```
ADDGROUP (BZU) OWNER(IBMUSER) SUPGROUP(SYS1)
DATA('z/OS Dynamic Test Runner - HLQ STUB')
```

-
```
ADDSD  'BZU.*.**' UACC(READ)
DATA('z/OS Dynamic Test Runner')
```

-
```
PERMIT 'BZU.*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
```

-
```
SETROPTS GENERIC(DATASET) REFRESH
```

Notes:

- Protect BZU.SBZULINK against updates because this data set is APF authorized.
- The sample commands in this publication and in the BZURACF job assume that Enhanced Generic Naming (EGN) is active. When EGN is active, the ** qualifier can be used to represent any number of qualifiers in the DATASET class. Substitute ** with * if EGN is not active on your system. For more information about EGN, see *Security Server RACF Security Administrator's Guide* (SA22-7683).

Verify the security settings

Use the following sample commands to display the results of your security-related customizations.

- Data set profiles
  - LISTGRP BZU
  - LISTDSD PREFIX(BZU) ALL

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-IBMproducts. 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.

Programming interface information

Trademarks

IBM, the IBM logo, and 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 www.ibm.com/legal/copytrade.shtml.

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.

IBM®