IBM Rational Developer for System z v7.6 Host Configuration Release Notes

This Tech note covers release-specific Host Configuration information (such as problems and limitations) that did not become available until after the product documentation and product readme were finalized. In this tech note, you can find additional information on the following topics:

The following annotations must be made to the host configuration documentation. Abbreviated names are used for these publications for easy reference. Use the following table to match an abbreviated name to a publication:

Publication title	Order number	Abbreviation
Rational Developer for System z Host Configuration Quick Start	GI11-9201	Host Configuration Quick Start
Rational Developer for System z Host Configuration Guide	SC23-7658	Host Configuration Guide

WLM information for Developer for System z

As documented in “RSE as a Java application” (Chapter 11. Understanding Developer for System z) on page 165 of the Host Configuration Guide, RSE daemon (started task) spawns a child process for each Thread Pool server (which supports a variable number of clients).

This child process is active in a separate address space, and needs proper classification in Workload Manager (WLM) to ensure it gets the resources it needs to support the clients it serves. .

Note that because these are spawned processes, they are classified via the WLM OMVS classification rules, not the STC classification rules.

All Developer for System z servers, RSE daemon, RSE thread pool, Lock daemon and JES Job Monitor, are servicing real-time client requests, and need to be assigned to a high-performance service class.

_BPXK_SETIBMOPT_TRANSPORT

The description of the `_BPXK_SETIBMOPT_TRANSPORT` directive states that the default is TCPIP. This description can be misinterpreted. Developer for System z does not bind to stack TCPIP by default. Developer for System z binds to each available stack by default. The following note should be added to these sections:

- Section “FEJJCNFG, JES Job Monitor configuration file” section (Chapter 2. Basic customization) on page 25 of the Host Configuration Guide
- Section “rsed.envvars, RSE configuration file” (Chapter 2. Basic customization) on page 33 of the Host Configuration Guide
- Section “FEJJCNFG, JES Job Monitor configuration file” section (Chapter 2. Basic customization) on page 11 of the Host Configuration Quick Start
- Section “rsed.envvars, RSE configuration file” (Chapter 2. Basic customization) on page 15 of the Host Configuration Quick Start

Note:
When this directive is not active, Developer for System z binds to every available stack on the system (BIND INADDRANY).

FEJJCNFG, JES Job Monitor configuration file

The description of the `GEN_CONSOLE_NAME` directive references `&USERID` as value for the `CONSOLE_NAME` directive. This description should reference `&SYSUID`. The “FEJJCNFG, JES Job Monitor configuration file - GEN_CONSOLE_NAME” section (Chapter 2. Basic customization) on page 26 of the Host Configuration Guide should be as documented below.

GEN_CONSOLE_NAME

Enables or disables automatic generating of alternative console names. The default is OFF. Uncomment and change to ON to enable alternative console names.

This directive is only used when CONSOLE_NAME equals &SYSUID and the user ID is not available as console name.

rsed.envvars, RSE configuration file

Customizable directives

enable.automount

The enable.automount directive is new in rsed.envvars. The following information should be added to section "Defining extra Java startup parameters with _RSE_JAVAOPTS" (Chapter 2. Basic customization) on page 36 of the Host Configuration Guide

#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Denable.automount=true"

Support home directories created by z/OS UNIX automount. The default is false. Uncomment and specify true to ensure that z/OS UNIX automount uses the client user ID as owner of the directory.

Note:
z/OS UNIX automount uses the user ID of the process that invoked the service when creating a file system. If this option is disabled, this process is the RSE thread pool server (user ID STCRSE). If this option is enabled, a new, temporary, process is created using the client user ID before accessing the directory.

_BPXK_SETIBMOPT_TRANSPORT (2)

The _BPXK_SETIBMOPT_TRANSPORT directive is erroneously referenced as BPXK_SETIBMOPT_TRANSPORT (missing leading underscore (_) character) in section "rsed.envvars, RSE configuration file" (Chapter 2. Basic customization) on page 33 of the Host Configuration Guide.

Non-customizable directives

The following non-customizable directives have changed or are new in rsed.envvars. The following information updates the sample rsed.envvars in section "rsed.envvars, RSE configuration file" (Chapter 2. Basic customization) on page 29 of the Host Configuration Guide.

```
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDSTORE_KEEPALIVE_RESPONSE_TIMEOUT=30000"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDSTORE_IO_SOCKET_READ_TIMEOUT=90000"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DRSECOMM_LOGFILE_MAX=0"
```

Using REXEC (or SSH)

Using REXEC (or SSH) as an alternate RSE connection method has been discouraged since Developer for System z Version 7.5. To emphasize this, the `server.zseries` invocation script is no longer copied automatically to `/etc/rdz`. The following note should be added to section "Using REXEC (or SSH) - Alternate RSE connection method" (Chapter 6. Other customization tasks) on page 86 of the Host Configuration Guide.

Note:

> To discourage the usage of REXEC (or SSH) as alternate logon method, `server.zseries` is no longer copied automatically to `/etc/rdz`. If you want to use this function, you must manually copy it from `/usr/lpp/rdz/bin`, as shown in the following sample command:

```
cp /usr/lpp/rdz/bin/server.zseries /etc/rdz
```

Operator commands

The output of `MODIFY rsed,APPL=DISPLAY PROCESS` has been enhanced with the `Order()` field. The "Modify (F) - RSE daemon" section (Chapter 8. Operator commands) on page 110 of the Host Configuration Guide should now be as documented below.

DISPLAY PROCESS[,CLEANUP]

Display the RSE thread pool processes. There can be multiple processes, which are used for load balancing the connected users.

```
ProcessId(<processid>) Memory Usage(<java heap usage>%)
  Clients(<number of clients>) Order(<startup order>) <error status>
```

Note:

- `<processid>` can be used in process specific z/OS UNIX operator commands.
- Each process has its own Java™ heap, whose size can be set in `rsed.envvars`.
- `<startup order>` is a sequential number that indicates the order that the thread pools were started. The number corresponds to the number used in the filename of the `stderr.*.log` and `stdout.*.log` files.

z/OS UNIX permission bits

Program Control authorization

ICU version 4.0.1 is now being used instead of ICU 3.4.0, and a new program, fekfrivp, has been added to Developer for System z. The list of z/OS UNIX files that require being program controlled in section "Program Control authorization" (Chapter 9. Troubleshooting) on page 127 of the Host Configuration Guide should now be as documented below.

The following Developer for System z files must be program controlled:

- /usr/lpp/rdz/bin/
    - fekfdivp
    - fekfrivp
- /usr/lpp/rdz/lib/
    - fekfdir.dll
    - libfekdcore.so
    - libfekfmain.so
- /usr/lpp/rdz/lib/icuc/
    - libicudata.dll
    - libicudata40.1.dll
    - libicudata40.dll
    - libicuuc.dll
    - libicuuc40.1.dll
    - libicuuc40.dll

APF authorization

A new program, fekfrivp, has been added to Developer for System z. This is the first z/OS UNIX file that requires being APF authorized. The following sub-section should be added to section “z/OS UNIX permission bits” (Chapter 9. Troubleshooting) on page 127 of the Host Configuration Guide.

APF authorization

The PassTicket IVP (Installation Verification Program), which is part of the RSE started task IVP, must be APF authorized in order to do additional security checks.

The z/OS UNIX APF bit is set during SMP/E install where needed. This permission bit might get lost if you did not preserve it during a manual copy of the Developer for System z directories.

The following Developer for System z files must be APF authorized:

- /usr/lpp/rdz/bin/
    - fekfrivp

Use z/OS UNIX command **ls -E** to list the extended attributes, in which the APF bit is marked with the letter a, as shown in the following sample ($ is the z/OS UNIX prompt):

```
$ cd /usr/lpp/rdz
$ ls -E bin/fekfrivp
-rwxr-xr-x  aps-  2 user     group     114688 Sep 17 06:41 bin/fekfrivp
```

Use z/OS UNIX command **extattr +a** to set the APF bit manually, as shown in the following sample ($ and # are the z/OS UNIX prompt):

```
$ cd /usr/lpp/rdz
$ su
# extattr +a bin/fekfrivp
# exit
```

```
$ ls -E bin/fekfrivp
-rwxr-xr-x  aps-  2 user     group     114688 Sep 17 06:41 bin/fekfrivp
```

Note:
To be able to use the **extattr +a** command, you must have at least READ access to the `BPX.FILEATTR.APF` profile in the `FACILITY` class of your security software, or be a superuser (UID 0) if this profile is not defined. For more information, refer to *UNIX System Services Planning* (GA22-7800).

Corequisites

The following corequisites should be added to section "z/OS host corequisites" (Appendix E. Requisites) on page 283 of the Host Configuration Guide.

z/OS

Program Number	**Product Name**	**PTFs or Service Levels Required**
5694-A01	z/OS v 1.10	SCLM: • APAR OA28866 (ACCTINFO return code 12) PTF UA47306

Debug Tool for z/OS

Program Number	**Product Name**	**Programming Language**	**APARs, PTFs, or Service Levels Required**
	IBM Debug Tool for z/OS V10.1	COBOL, PL/I, C/C++, assembler, and additional features	No PTF or Service Level required

Fault Analyzer

Program Number	**Product Name**	**PTFs or Service Levels Required**
	IBM Fault Analyzer v 10.1	No PTF or Service Level required

Endevor

CA Endevor® Software Change Manager Release 12 must be installed to use the Developer for System z Interface for CA Endevor® SCM.

CA Endevor® SCM is a product from Computer Associates. The related product website is:

http://www.ca.com/us/products/product.aspx?ID=259

Activating the CA Endevor® SCM Integration

Extra functionality has been added to the CA Endevor® Software Configuration Manager (SCM) Integration, which requires additional configuration work compared to what is documented in the Host Configuration Guide. The new setup method provides pre-configured samples which can be tailored to your needs. The following section should replace the "(Optional) Activating the CA Endevor® SCM RAM" (Chapter 3. CARMA) section on page 54 of the Host Configuration Guide.

(Optional) Activating the CA Endevor® SCM RAM

The IBM® Rational® Developer for System z Interface for CA Endevor® Software Configuration Manager gives Developer for System z clients direct access to CA Endevor® SCM. From here on, IBM® Rational® Developer for System z Interface for CA Endevor® SCM is abbreviated to CA Endevor® SCM RAM (Repository Access Manager).

In contradiction with the sample RAMs documented in this publication, CA Endevor® SCM RAM is a production type RAM. You should not activate both types of RAM in the same setup.

Attention: The provided setup jobs for CA Endevor® SCM RAM replace the active CARMA setup with one that holds only the CA Endevor® SCM RAM.

Note:
The TSO/ISPF Client Gateway startup method can not be used together with the CA Endevor® SCM RAM.

Requirements and checklist

You will need the assistance of a security administrator and a TCP/IP administrator to complete this customization task, which requires the following resources or special customization tasks:

- TCP/IP port range for internal communication
- (Optional) Security rule to allow users to submit CRA* jobs
- (Optional) LPA update

In order to start using the CA Endevor® SCM RAM at your site, you must perform the following tasks. Unless otherwise indicated, all tasks are mandatory.

1. Allocate and prime VSAM data sets that define the CA Endevor® SCM RAM to CARMA. For details, see Define the CA Endevor® SCM RAM.
2. Choose your preferred startup method, batch submit or CRASTART, and do the required customization of the related configuration files. For details see:

   - CA Endevor® SCM RAM startup using batch submit
   - CA Endevor® SCM RAM startup using CRASTART
3. Optionally customize the allocation exec used for dynamic allocation of user-specific data sets. For details, see (Optional) Customize CRANDVRA.
4. Optionally customize the CA Endevor® SCM RAM specific configuration files. For details, see (Optional) Customize the CA Endevor® SCM RAM.

Define the CA Endevor® SCM RAM

The following CARMA components must be customized, regardless of the chosen startup method. The sample members referenced below are located in `FEK.#CUST.JCL`, unless you specified a different location when you customized and submitted job `FEK.SFEKSAMP(FEKSETUP)`. See Customization setup for more details.

1. Customize and submit the `FEK.#CUST.JCL(CRA#VCAD)` JCL. Refer to the documentation within `CRA$VDEF` for customization instructions. `CRA#VCAD` creates and primes the CARMA configuration VSAM data set, `CRADEF`.
2. Customize and submit the `FEK.#CUST.JCL(CRA$VMSG)` JCL. Refer to the documentation within `CRA$VMSG` for customization instructions. `CRA$VMSG` creates and primes the CARMA message VSAM data set, `CRAMSG`.

   Note:
   This is the same job as for the sample RAMs.

3. Customize and submit the `FEK.#CUST.JCL(CRA#VCAS)` JCL. Refer to the documentation within `CRA$VSTR` for customization instructions. `CRA#VCAS` creates and primes the CARMA custom information VSAM data set, `CRASTRS`.

Note:

- The CA Endevor® SCM RAM uses RAM ID 0 by default.
- Refer to sample job `FEK.#CUST.JCL(CRA#UADD)` if you need to merge the definitions for a (custom) RAM into an existing VSAM configuration. This job must be customized and submitted for each CARMA VSAM that changes. Refer to the *Rational Developer for System z Common Access Repository Manager Developer's Guide* (SC23-7660) for more information on the record structure used by the different CARMA VSAMs.
- Use sample job `FEK.#CUST.JCL(CRA#UQRY)` to extract the active definitions from a VSAM to a sequential data set.

CA Endevor® SCM RAM startup using batch submit

Do not execute this step if you use the CRASTART method to start the CARMA server with the CA Endevor® SCM RAM.

Developer for System z can use the batch submit CARMA server startup method to start the CA Endevor® SCM RAM. The method submits the CARMA server as a long-running batch job in your JES.

Refer to CARMA server startup using batch submit for more information on the batch submit startup method.

Adjust CRASRV.properties

RSE server uses the settings in `/etc/rdz/CRASRV.properties` to start and connect to a CARMA server, as documented in RSE interface to CARMA. You can edit the file with the TSO **OEDIT** command. Note that RSE must be restarted for the changes to take effect.

Change the value of the `clist.dsname` directive to the data set and member name of the `CRASUBCA` CARMA server startup CLIST, as shown in the following example. Refer to RSE interface to CARMA for more information on the different directives.

Figure x1. CRASRV.properties - CA Endevor® SCM RAM startup using batch submit

```
port.start=5227
port.range=100
startup.script.name=/usr/lpp/rdz/bin/carma.startup.rex
clist.dsname='FEK.#CUST.CNTL(CRASUBCA)'
```

Adjust CRASUBCA

Customize the `CRASUBCA` CLIST, as shown in the following code sample. Refer to the documentation within `CRASUBCA` for customization instructions. The `CRASUBCA` CLIST submits a CARMA server for CA Endevor® SCM.

`CRASUBCA` is located in `FEK.#CUST.CNTL`, unless you specified a different location when you customized and submitted job `FEK.SFEKSAMP(FEKSETUP)`. See Customization setup for more details.

Figure x2. CRASUBCA - CA Endevor® SCM RAM startup using batch submit

```
PROC 1 PORT TIMEOUT(420)
SUBMIT * END($$)
//CRA&PORT JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//RUN      EXEC PGM=IKJEFT01,DYNAMNBR=125,REGION=0M,TIME=NOLIMIT,
//            PARM='%CRANDVRA NDVRC1 PGM(CRASERV) PARM(&PORT &TIMEOUT)'
//STEPLIB  DD DISP=SHR,DSN=FEK.SFEKLOAD
//         DD DISP=SHR,DSN=CA.NDVR.AUTHLIB
//         DD DISP=SHR,DSN=CA.NDVRU.AUTHLIB
//CRADEF   DD DISP=SHR,DSN=FEK.#CUST.CRADEF
//CRAMSG   DD DISP=SHR,DSN=FEK.#CUST.CRAMSG
//CRASTRS  DD DISP=SHR,DSN=FEK.#CUST.CRASTRS
//*
//SYSPROC  DD DISP=SHR,DSN=ISP.SISPCLIB
//         DD DISP=SHR,DSN=FEK.SFEKPROC
//ISPEXEC  DD DISP=SHR,DSN=ISP.SISPEXEC
```

```
//ISPMLIB  DD DISP=SHR,DSN=ISP.SISPMENU
//ISPPLIB  DD DISP=SHR,DSN=ISP.SISPPENU
//ISPSLIB  DD DISP=SHR,DSN=ISP.SISPSENU
//ISPTLIB  DD DISP=SHR,DSN=ISP.SISPTENU
//ISPCTL0  DD DISP=(NEW,DELETE,DELETE),DSN=&&CTL0,UNIT=SYSALLDA,
//            SPACE=(TRK,(1,1)),LRECL=80,RECFM=FB
//ISPCTL1  DD DISP=(NEW,DELETE,DELETE),DSN=&&CTL1,UNIT=SYSALLDA,
//            SPACE=(TRK,(1,1)),LRECL=80,RECFM=FB
//ISPPROF  DD DISP=(NEW,DELETE,DELETE),DSN=&&PROF,UNIT=SYSALLDA,
//            SPACE=(TRK,(1,1,5)),LRECL=80,RECFM=FB
//*
//CARMALOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD DUMMY
//*
//CONLIB   DD DISP=SHR,DSN=CA.NDVR.CONLIB
//JCLOUT   DD SYSOUT=(A,INTRDR),DCB=(LRECL=80,RECFM=F,BLKSIZE=80)
//EXT1ELM  DD DSN=&&EXT1ELM,DISP=(NEW,DELETE),UNIT=SYSALLDA,
//            RECFM=VB,LRECL=4096,BLKSIZE=27998,SPACE=(TRK,(5,5))
//EXT1DEP  DD DSN=&&EXT1DEP,DISP=(NEW,DELETE),UNIT=SYSALLDA,
//            RECFM=VB,LRECL=4096,BLKSIZE=27998,SPACE=(TRK,(5,5))
//MSG3FILE DD DSN=&&MSG3FILE,DISP=(NEW,DELETE),UNIT=SYSALLDA,
//            RECFM=FB,LRECL=133,BLKSIZE=27930,SPACE=(TRK,(5,5))
//C1MSGS1  DD DSN=&&C1MSGS1,DISP=(NEW,DELETE),UNIT=SYSALLDA,
//            RECFM=FB,LRECL=133,BLKSIZE=27930,SPACE=(TRK,(5,5))
//C1EXMSGS DD DSN=&&C1MSGS1,DISP=(NEW,DELETE),UNIT=SYSALLDA,
//            RECFM=FB,LRECL=133,BLKSIZE=27930,SPACE=(TRK,(5,5))
//TYPEMAP  DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRATMAP)
//SHOWVIEW DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRASHOW)
$$
EXIT CODE(0)
```

Note:

- You can add your own DD statements and data set concatenations to customize the CARMA TSO environment, thus mimicking a TSO logon procedure.
- You can optionally change CARMA's timeout value by modifying the `PROC 1 PORT TIMEOUT(420)` line in `FEK.#CUST.CNTL(CRASUBMT)` CLIST. The timeout value is the number of seconds CARMA will wait for the next command from the client. Setting a value of 0 results in the default timeout value, currently 420 seconds (7 minutes).
- Details of the CARMA startup process are shown in `rsecomm.log`. Refer to (Optional) RSE tracing for more information on setting the detail level of `rsecomm.log`.
- Changes are in effect for all CARMA servers started after the update.

CA Endevor® SCM RAM startup using CRASTART

Do not execute this step if you use the batch submit method to start the CARMA server with the CA Endevor® RAM.

Developer for System z can use the CRASTART CARMA server startup method to start the CA Endevor® SCM RAM. The method uses CRASTART to start the CARMA server as a subtask within RSE.

Refer to (Optional) Alternative CARMA server startup using CRASTART for more information on the CRASTART startup method.

Note:
Details of the CARMA startup process are shown in `rsecomm.log`. Refer to (Optional) RSE tracing for more information on setting the detail level of `rsecomm.log`.

Adjust CRASRV.properties

RSE server uses the settings in `/etc/rdz/CRASRV.properties` to start and connect to a CARMA server, as documented in RSE interface to CARMA. You can edit the file with the TSO **OEDIT** command. Note that RSE must be restarted for the changes to take effect.

Change the value of the `clist.dsname` directive to `*CRASTART` and provide the correct values for the `crastart.*` directives, as shown in the following example. Refer to RSE interface to CARMA for more information on the different directives.

Figure x3. CRASRV.properties - CA Endevor® SCM RAM startup using CRASTART

```
port.start=5227
port.range=100
startup.script.name=/usr/lpp/rdz/bin/carma.startup.rex
clist.dsname=*CRASTART
crastart.stub=/usr/lpp/rdz/bin/CRASTART
crastart.configuration.file=/etc/rdz/crastart.endevor.conf
crastart.syslog=Partial
crastart.timeout=420
#crastart.steplib=FEK.SFEKLPA
#crastart.tasklib=TASKLIB
```

Note:
System abend 522 for module `CRASERV` will occur if the `JWT` parameter in the `SMFPRMxx` parmlib member is set to a value lower than the time out value in `CRASRV.properties`. This does not impact CARMA operations, as the server is restarted automatically if needed.

Adjust crastart.endevor.conf

CRASTART uses the definitions in `crastart.endevor.conf` to create a valid (TSO/ISPF) environment to invoke CA Endevor® SCM. Developer for System z uses this environment to run the CA Endevor ® SCM RAM.

`crastart.endevor.conf` is located in `/etc/rdz/`, unless you specified a different location when you customized and submitted job `FEK.SFEKSAMP(FEKSETUP)`. See Customization setup for more details. You can edit the file with the TSO **OEDIT** command.

Note:
Changes are in effect for all CARMA servers started after the update.

Figure x4. crastart.conf - CA Endevor® SCM RAM startup using CRASTART

```
TASKLIB = FEK.SFEKLOAD,CA.NDVR.AUTHLIB,CA.NDVRU.AUTHLIB
CRADEF  = FEK.#CUST.CRADEF
CRAMSG  = FEK.#CUST.CRAMSG
CRASTRS = FEK.#CUST.CRASTRS

SYSPROC = ISP.SISPCLIB,FEK.SFEKPROC
SYSEXEC = ISP.SISPEXEC
ISPMLIB = ISP.SISPMENU
ISPPLIB = ISP.SISPPENU
ISPSLIB = ISP.SISPSENU
-COMMAND=ALLOC FI(ISPCTL0)  NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
BLKSIZE(6160) SPACE(2,2) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPCTL1)  NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
BLKSIZE(6160) SPACE(2,2) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPPROF)  NEW DELETE DSORG(PO) DIR(5) RECFM(F,B)
LRECL(80) BLKSIZE(6160) SPACE(2,2) TRACKS UNIT(SYSALLDA)
ISPTLIB = -ISPPROF,ISP.SISPTENU
ISPTABL = -ISPPROF

CARMALOG= SYSOUT(H)
SYSPRINT= SYSOUT(H)
SYSTSPRT= SYSOUT(H)
SYSTSIN = DUMMY

TYPEMAP = FEK.#CUST.PARMLIB(CRATMAP)
SHOWVIEW= FEK.#CUST.PARMLIB(CRASHOW)
CONLIB  = CA.NDVR.CONLIB
-COMMAND=ALLOC FI(JCLOUT)   SYSOUT(A) WRITER(INTRDR) RECFM(F) LRECL(80)
BLKSIZE(80)
-COMMAND=ALLOC FI(EXT1ELM)  NEW DELETE DSORG(PS) RECFM(V,B) LRECL(4096)
BLKSIZE(27998) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(EXT1DEP)  NEW DELETE DSORG(PS) RECFM(V,B) LRECL(4096)
BLKSIZE(27998) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(MSG3FILE) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(133)
BLKSIZE(27930) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(C1EXMSGS) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(133)
BLKSIZE(27930) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(C1MSGS1)  NEW DELETE DSORG(PS) RECFM(F,B) LRECL(133)
BLKSIZE(27930) SPACE(5,5) TRACKS UNIT(SYSALLDA)

PROGRAM=IKJEFT01 %CRANDVRA NDVRC1 PGM(CRASERV) PARM(&CRAPRM1. &CRAPRM2.)
```

(Optional) Customize CRANDVRA

Both the batch submit and the CRASTART startup method invoke REXX exec CRANDVRA to allocate user-specific data sets used by CA Endevor® SCM RAM.

DD	**Data set name**	**Type**
DEPEND	&SYSPREF..&SYSUID..&SYSNAME..CRA$NDVR.DEPEND	Permanent
BROWSE	&SYSPREF..&SYSUID..&SYSNAME..CRA$NDVR.BROWSE	Temporary
C1PRINT	&SYSPREF..&SYSUID..&SYSNAME..CRA$NDVR.LISTING	Temporary

You can customize a copy of this allocation REXX exec if certain defaults, like the data set name, don't match your site standards. `CRANDVRA` is located in `FEK.SFEKPROC`, unless you used a different high level qualifier during the SMP/E install of Developer for System z.

Refer to the documentation within `CRANDVRA` for customization instructions.

Note:
It is recommended that you copy the sample allocation REXX to a new data set and customize this copy to avoid overwriting it when applying maintenance. When doing so, you must update the reference to `SFEKPROC` in the SYSEXEC DD of your chosen CARMA startup method to match your new data set name.

(Optional) Customize the CA Endevor® SCM RAM

The following CARMA components can be customized, regardless of the chosen startup method. The sample members referenced below are located in `FEK.#CUST.PARMLIB`, unless you specified a different location when you customized and submitted job `FEK.SFEKSAMP(FEKSETUP)`. See Customization setup for more details.

1. (Optional) Customize `FEK.#CUST.PARMLIB(CRASHOW)`. Refer to the documentation within `CRASHOW` for customization instructions. `CRASHOW` defines default filters for CA Endevor® SCM environments, systems, etc.
2. (Optional) Customize `FEK.#CUST.PARMLIB(CRATMAP)`. Refer to the documentation within `CRATMAP` for customization instructions. `CRATMAP` overrides CA Endevor® SCM type to file extension mappings.