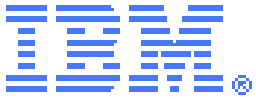

SupportPac CA1T –

CICS Transaction Gateway – High Availability User Exits

Installation and User's Guide
Version 3.0 – March 2013

Authors

Phil Wakelin, Rob Jones, Andrew Smithson
IBM United Kingdom Limited.
Hursley Park, Winchester, SO21 2JN. UK.

Chris Skaife
IBM CICS Technical support - Kaiser Account

Note! Before using this information and the product it supports, be sure to read the general information under “Notices” on page 20.

Third edition (March 2013)

This edition applies to Version 3.0 of SupportPac CA1T – High availability exits for use with CICS TG and to all subsequent versions, releases, and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

1. Contents

1. Contents 3

2. Introduction 4

3. SupportPac files 5

4. Overview 6

Workload balancing 6

Request validation 7

5. Preconditions and notes 8

Release history 8

6. Installing the SupportPac 9

Preparation 9

Setup 9

Environment variables: 10

JVM properties: 11

Sample Gateway daemon configuration 12

Logging destinations 12

HA configuration 12

Command functions 15

7. Further information 17

8. Troubleshooting 18

9. Notices 20

10. Trademarks 21

2. Introduction

SupportPac CA1T provides sample CICS request exits to be used with IBM CICS Transaction Gateway products to create high availability and request validation rules.

These exits enable a sysplex-wide high availability infrastructure to be created quickly, using rules defined in a simple text based configuration file. The rules define a set of policies to validate and workload balance ECI requests across defined CICS servers, providing increased availability to CICS applications accessed from remote clients using the facilities of the CICS Transaction Gateway.

Support Pac CA1T V3.0 can be used with:
CICS Transaction Gateway for z/OS V8.0, or later
CICS Transaction Gateway for Multiplatforms V8.1, or later
CICS Transaction Gateway Desktop Edition V8.1, or later

This document focuses on configuring and using SupportPac CA1T with the CICS Transaction Gateway for z/OS. Configuration and usage with the Multiplatforms and Desktop Edition products is not explicitly shown – but the same environment variables, java properties and CICS Request Exit commands are used on non-z/OS platforms, specified in the usual way for the platform in question.

3. SupportPac files

This SupportPac contains the following artefacts:

- License Directory - Directory of license information.
- ca1t_v3.pdf - This document
- ca1t_v3.jar - JAR file for use with CICS TG V8.0, or later
- ca1t_v3.ini - Sample exit configuration file
- ca1t_v3.stdenv - Sample environment variables

4. Overview

Version 2.1 of the SupportPac provides a combined workload balancing and request validation infrastructure to be used with CICS Transaction Gateway for z/OS V8.0. Version 3.0 allows validation rules to be applied on an inclusive basis, as per Version 2.0/Version 2.1, or on an exclusive basis. When combined with a policy re-load operation (CA1T_RELOAD command), the validation rules can be used in exclusive mode to pro-actively reject requests for specific transactions, programs or user IDs.

Configuration is based on a set of rules defined in HFS or MVS configuration files. Rules for workload balancing can be generic per Gateway, based on server aliases, or on the payload type (channel or COMMAREA). Rules for request validation can be specified based on user IDs, transaction IDs or program names, and a configurable retry interval is provided to improve the efficiency of request distribution when using workload balancing

A configuration file failover mechanism is provided to enable failover between primary and secondary groups of CICS regions. All rules can be dynamically updated whilst the Gateway daemon is running. Workload balancing or request validation polices can be defined as follows:

Workload balancing

Two different workload balancing polices (round robin or fail over) are supplied both of which use server rules from the HA configuration file to remap and retry failed requests. For example the following server rule defines the server alias SERVER1 as mapping to 3 actual CICS servers CICSA, CICSB and CICSC.

```
SERVER1=CICSA,CICSB,CICSC
```

The *round robin* policy will distribute requests in a round robin fashion, so that requests are evenly distributed across servers CICSA, CICSB and CICSC. By contrast, the *fail over* policy will distribute requests in a hierarchical order so that all requests are sent to CICSA, if this is unavailable requests will be sent to CICSB, and if CICSB is unavailable requests will be sent to CICSC. Note for either policy, once the exit determines a prior request has failed with a retryable error, the server is bypassed until the retry interval has elapsed.

If using primary and secondary configuration files, the rules from the secondary file will be activated whenever a request is to be retried but all servers for the rule are unavailable. The rules from the primary configuration file will be automatically reactivated after the duration set by the `CTG_HAAUTOSWAP` variable has passed or when a SWAP command is issued.

Request validation

Request validation is supported for non-XA requests, using the TRANSID, PROGRAM and USERID rules to validate or reject each request. Each request is validated against the rules defined in the HA configuration file in the Gateway daemon. Rejected requests will not be retried and the error ECI_ERR_INVALID_CALL_TYPE (-14) is returned. For example the following rule will cause the Gateway to reject any requests that did not specify the mirror transaction CSMI, that specified a userid, or that did not specify PROG1 or PROG2 as the program.

```
TRANSID=CSMI
USERID=NULL
PROGRAM=PROG1,PROG2
```

5. Preconditions and notes

The following is assumed in order to use this SupportPac:

- CICS Transaction Gateway for z/OS – V8.0, or later
- CICS Transaction Gateway for Multiplatforms – V8.1, or later
- CICS Transaction Gateway Desktop Edition – V8.1, or later
- Supported configuration – Any remote clients issuing ECI requests using the Gateway daemon on z/OS in remote mode

Release history

Version 1.0:
The original version of this SupportPac for CICS TG V7.2:

Version 1.1:
- Version numbers at start-up
- Comment lines in configuration file

Version 2.0:
- CICS TG for z/OS V8 support
- Retry intervals
- MVS configuration files
- Command support
- Integrated request validation and high availability policies

Version 2.0.1
- Prevent request validation for XA call types
- Fixed bug in payload rules
- Calltype in log messages

Version 2.1
- Failover using two configuration files
- SWAP command
- Improved logging performance

Version 3.0
- Added validation styles INCLUSIVE or EXCLUSIVE
- Bug fix for missing configuration file

6. Installing the SupportPac

Preparation

To prepare for installation you will need to download and unpack the SupportPac as follows:

1. Copy the file ca1tv3.zip to a temporary directory on your chosen system and uncompress using unzip.

2. Transfer the ca1t_v3.ini, ca1t_v3.stdenv and ca1t_v3.jar files to a HFS directory (for instance /u/cicstg/tmp) on your z/OS system, or to your Multiplatforms machine where CICS TG is installed. If using ftp to transfer the support pac materials to z/OS, use *binary* mode for ca1t_v3.jar and *ascii* mode for ca1t_v3.ini and ca1t_v3.stdenv to ensure correct data conversion.

Setup

1. Add the location of the ca1t_v3.jar to the CLASSPATH used to start your Gateway. For example, on z/OS update your CLASSPATH statement in your STDENV as follows:

```
CLASSPATH=/u/cicstg/tmp/ca1t_v3.jar:/usr/lpp/cicstg800/
classes/ctgsamples.jar
```

2. Create a configuration file in your HFS, MVS dataset or PDS member (z/OS) or local file system (Multiplatforms). Define the location to your Gateway using the variable CTG_HACONFIG. For example to use the supplied sample file ca1t_v3.ini in the HFS directory /u/cicstg/tmp, set the following variable in your Gateway STDENV definition.

```
CTG_HACONFIG=/u/cicstg/tmp/ca1t_v3.ini
```

To enable failover between a primary (main) and secondary (alternate) configuration policy, define the secondary configuration file as follows:

```
CTG_HACONFIG=/u/cicstg/tmp/ca1t_v3.ini;/u/cicstg/tmp/ca1t_v3bk.ini
```

For more details on how to create configuration files refer to the HA configuration file section.

3. Ensure that your Gateway has read access to the configuration file. For example to give the file owner rw access and group r access issue the following USS (z/OS) or Unix/Linux command:

```
> chmod  640 /u/cicstg/tmp/ca1t_v3.ini
```

4. Update your Gateway daemon configuration file (ctg.ini) to specify either a round robin or failover workload management policy. For example to enable use of the round robin policy specify:

```
cicsrequestexit=com.ibm.ctg.samples.ha.RoundRobinExit
```

Or to enable use of the fail over policy specify:

```
cicsrequestexit=com.ibm.ctg.samples.ha.FailOverExit
```

5. Start the Gateway daemon using your normal operating procedures

Environment variables:

The following environment variables are used by the SupportPac.

- CTG_HACONFIG (required) - Location of the HA configuration file. Failure to set this variable will throw an exception that will cause the Gateway to fail to start. This must be set to the location of an HFS file, an MVS dataset or PDS member. MVS datasets or PDS members should be specified using the JZOS format: `//'HLQ.QUAL.PDS(MEMBER)'`
- CTG_HAINTERVAL (optional) - Retry interval in seconds used to retry requests to failed servers, defaults to 60 seconds
- CTG_HAAUTOSWAP – Auto swap interval in seconds used to swap back to the primary configuration file. If CTG_HAAUTOSWAP is zero then the autoswap function is disabled, defaults to 60 seconds.
- CTG_HAVALIDATION (optional) - Validation style can be either "INCLUSIVE" (default) or "EXCLUSIVE". In INCLUSIVE mode, validation rules define those requests which will be processed. In EXCLUSIVE mode, validation rules define those requests which will be rejected.
- CTG_HACOUNT (optional) – Maximum request count when retrying failed requests. Defaults to the length of the longest server alias rule in the HA configuration file. Minimum value is 1.
- _BPX_SHAREAS (z/OS only) - USS address space sharing, this must be set to YES when using JES logging

On z/OS, environment variables are typically specified using the STDENV DD card of the CTGBATCH program. On Multiplatforms, specify them as System level environment variables and re-boot the machine to ensure that the IBM CICS Transaction Gateway service picks them up. Unix/Linux users typically specify environment variables through their CTGD service configuration, `ctgd.conf`.

Note: CTG_HACOUNT can be used to explicitly restrict the maximum number of retries. By default, it is set to the length of the longest server alias rule in the HA configuration file, and this is recommended for most cases. For instance, if the longest rule maps to four CICS servers…

```
PAYROLL=CICS1,CICS2,CICS3,CICS4
```

…then CA1T will retry the request up to three times. This allows each of the CICS servers in rule to tried at least once in the event failure.

JVM properties:

The following JVM properties are used by the SupportPac and can be set as startup overrides using the CTGSTART_OPTS environment variable.

- com.ibm.ctg.samples.haexit.out
  - This sets the name of the output log. This can be an HFS file, an MVS dataset or a DD (z/OS) or a file on the local file system (Multi-platforms). The default stdout, which is not appropriate on Multiplat-forms, but maps to the STDOUT DD card under CTGBATCH on z/OS.
- com.ibm.ctg.samples.haexit.ca1t
  - This sets the logging level, valid values are 0, 1, 2 or 3, the default is 1

Sample Gateway daemon configuration

The following sample environment variables are supplied in the file ca1t_v3.stdenv. These should be added to existing environment variables for your Gateway daemon.

```
CTG_HAINTERVAL=60
CTG_HACOUNT=3
CTG_HACONFIG=/u/cicstg/tmp/ca1t_v3.ini
CTG_HAVALIDATION=INCLUSIVE
CTGSTART_OPTS=-j-Dcom.ibm.ctg.samples.haexit.out=//DD:HALOG
-j-Dcom.ibm.ctg.samples.haexit.ca1t=1
_BPX_SHAREAS=YES
```

Logging destinations

The exit writes log messages to a destination controlled by the Java property "com.ibm.ctg.samples.haexit.out". On Multiplatforms or Desktop Edition, this property must specify a path/filename, otherwise CA1T log messages will be written to the default stdout destination, which is likely to be discarded on Windows, or on Unix/Linux under CTGD.

On z/OS, the exit supports a choice of logging destinations. The default of stdout is acceptable on z/OS under CTGBATCH, however JES, HFS or MVS datasets can also be used by setting the com.ibm.ctg.samples.haexit.out JVM property. For instance, to log to a specific JES target, add the following DD statement to the JCL used to start the Gateway.

```
//HALOG   DD   SYSOUT=*
```

and set the following JVM property in the CTGSTART_OPTS variable:

```
CTGSTART_OPTS=-j-Dcom.ibm.ctg.samples.haexit.out=//DD:HALOG
```

Alternatively to log to the MVS dataset HLQ.QUAL.DS set the CTGSTART_OPTS variable in your STDENV as follows

```
CTGSTART_OPTS=-j-Dcom.ibm.ctg.samples.haexit.out=//'HLQ.QUAL.DS'
```

Note: The CTGSTART_OPTS environment variable is only applicable on z/OS. For Multiplatforms or Desktop Edition, please use the appropriate method for the operating system to specify JVM properties for the Gateway daemon.

HA configuration

The HA configuration file defines a policy for both workload balancing and request validation. Optionally, a secondary (or alternate) policy can be defined. HA configuration files can be created as MVS datasets, PDS members or HFS files.

Policy rules are defined by key/value statements with all lines in the form KEY=VALUE1,VALUE2,VALUEn.

The following example rules are supplied in the sample file ca1t_v3.ini

```
SERVER1=CICSA,CICSB,CICSC
SERVER2=CICSA,CICSB
NULL=CICSC
COMMAREA=CICSA
CHANNEL=CICSB
#*= CICSA,CICSB,CICSC
TRANSID=NULL,CSMI,CPMI
PROGRAM=PROGRAM1,PROGRAM2/COMMAREA,PROGRAM3/CHANNEL
USERID=BOB,ALICE,PHIL,NULL
```

Note: V3 of the SupportPac is not compatible with configuration files previously deployed with V1 of the SupportPac

The sample configuration file will perform the following functions if unmodified.

- Requests to SERVER1 will match the SERVER1 rule and be distributed across actual servers CICSA, CICSB and CICSC.
- Requests to SERVER2 will match the SERVER2 rule and be distributed across actual servers CICSA and CICSB
- Requests that do not specify a server name will match the NULL rule and be sent to actual server CICSC
- Request that do not match a server rule and have a COMMAREA payload will be sent to actual server CICSA
- Request that do not match a server rule and have a CHANNEL payload will be sent to actual server CICSB
- All failed retryable requests will be tried up to 3 times (based on the length of the SERVER1 rule) using either a round robin or fail over policy as specified in the `cicsrequestexit` parameter in ctg.ini.

Note: Instead of using specific server rules a generic rule can be set as shown in the example *=CICSA,CICSB,CICS. This will cause all requests to be distributed across the specified actual servers no matter which server was specified in the initial request.

- In addition requests will be validated to ensure that
    1. The mirror transaction is either unspecified (null), or is CSMI or CPMI
    2. The CICS program name is specified as PROGRAM1, PROGRAM2 or PROGRAM3
    3. The program payload type is as follows
        - Requests to PROGRAM2 must pass a COMMAREA
        - Requests to PROGRAM3 must pass a CHANNEL
    4. The user ID is either not set (null), or is specified as BOB or ALICE.

  Any requests that fail validation will return the error *ECI_INVALID_CALL_TYPE* to the client application.

The syntax rules for the HA configuration file are as follows:

- All lines must define either a server remapping rule or a request validation rule
- Server rules can specify either a specific alias or the keywords NULL, COMMAREA, CHANNEL or *
- "*" specifies a generic server rule for any server not named in a specific rule or the NULL rule
- Values for server rules must use actual server names defined either as CICS APPLIDs for EXCI connections or IPICSERVERs names (from ctg.ini)
- COMMAREA is a special server that is used for requests with a COMMAREA, and can not be used for XA requests.
- CHANNEL is a special server rule that is used for requests with a CHANNEL and should specify the name of an IPICSERVER definition, and can not be used for XA requests
- TRANSID, PROGRAM, USERID, are reserved key words for request validation, and can not be used as server aliases.
- Payload can be specified in a PROGRAM validation rule and is specified using a "/" after the program name, it must be specified as either COMMAREA or CHANNEL
- NULL is valid as a server rule alias, a transaction ID, or a user ID and signifies no value (i.e. binary zeros)
- Comment lines start with '#' and are ignored
- All rules are case insensitive
- All rules are optional

Request validation is performed before server remapping and occurs in the following order:

1. Transaction IDs
2. Programs and payload
3. User IDs

When Validation style (CTG_HAVALIDATION) is set to INCLUSIVE, requests which match one of the validation rules will be processed as normal, and all other requests are rejected. The default for validation style is INCLUSIVE.

When Validation style (CTG_HAVALIDATION) is set to EXCLUSIVE, requests which match one of the validation rules will be rejected, and all other requests are processed as normal.

Server alias matching occurs in the following order:

1. Specific server aliases (such as SERVER1, SERVER2 or NULL in the example)
2. The CHANNEL server rule (when not using XA)
3. The COMMAREA server rule (when not using XA)
4. The generic rule (*) for any server

Command functions

The following command functions are supported using the CREXIT command. For illustration, the following examples use the z/OS modify command. However, the CA1T CREXIT commands can equally be used with CICS TG for Multiplatforms, through the ctgadmin command line interface.

1. LIST - This lists the current state of the defined rules and the server status

```
z/OS: /F <JOB>,APPL=CREXIT,CMD=CA1T_LIST
MP/DE: ctgadmin -a crexit -cmd=CA1T_LIST
```

2. RESET - This resets the state of all servers as active and performs a list command

```
z/OS: /F <JOB>,APPL=CREXIT,CMD=CA1T_RESET
MP/DE: ctgadmin -a crexit -cmd=CA1T_RESET
```

3. RELOAD - This reloads the configuration file (or files), and performs a reset command

```
z/OS: /F <JOB>,APPL=CREXIT,CMD=CA1T_RELOAD
MP/DE: ctgadmin -a crexit -cmd=CA1T_RELOAD
```

4. FILE – This refreshes the active configuration using the file (or files) specified, and implicitly performs a RESET. It supports only upper case file names as either HFS files or MVS datasets:

```
z/OS:
/F <JOB>,APPL=CREXIT,CMD=CA1T_FILE=/U/CICSTS/TMP/CTGHAV3.INI
/F <JOB>,APPL=CREXIT,CMD=CA1T_FILE=//'HLQ.QUAL.PDS(MEMBER)'
/F <JOB>,APPL=CREXIT,CMD=CA1T_FILE=//'HLQ.QUAL.PDS(POLICY1)';
//'HLQ.QUAL.PDS(POLICY2)'
MP/DE:
ctgadmin -a crexit -cmd=CA1T_FILE=C:\ca1t\ctghav3.ini
```

Note: The maximum number of request retries can only be set at Gateway startup. Thus if a new HA configuration file is loaded and this contains a server rule that is longer than any previous rule, the additional servers may not be selected for round robin distribution until the Gateway is restarted.

5. SWAP - This manually swaps the configuration file to the alternate configuration file when using primary and secondary files

```
z/OS: /F <JOB>,APPL=CREXIT,CMD=CA1T_SWAP
MP/DE: ctgadmin -a crexit -cmd=CA1T_SWAP
```

6. DISABLE - This disables the CICS request exit. When the exit is disabled no server name remapping, request retry or validation functions will be performed.

```
z/OS: /F <JOB>,APPL=CREXIT,CMD=CA1T_DISABLE
MP/DE: ctgadmin -a crexit -cmd=CA1T_DISABLE
```

Note: Once the exit is disabled any requests that fail will be returned to the application with ECI error *ECI_ERR_INVALID_CALL_TYPE* despite the cause of the underlying error.

7. ENABLE - This enables the CICS request exit, using the rules previously loaded from the configuration file.

```
z/OS: /F <JOB>,APPL=CREXIT,CMD=CA1T_ENABLE
MP/DE: ctgadmin -a crexit -cmd=CA1T_ENABLE
```

8. Logging control – This controls the output of log messages from the exit as follows:

Command	Level	Action
CA1T_0	No logging	Log messages on startup only, no messages during operation
CA1T_1	Level 1	Information messages about request validation failures (default)
CA1T_2	Level 2	Summary messages for each request
CA1T_3	Level 3	Full debug trace

For example to enable level 2 logging issue the following command

```
/F <JOB>,APPL=CREXIT,CMD=CA1T_2
```

Note: The default logging level can also be modified using the JVM property com.ibm.ctg.samples.haexit.ca1t, for example to decrease the logging level to 0 by default set the following in STDENV

```
CTGSTART_OPTS=-j-Dcom.ibm.ctg.samples.haexit.ca1t=0
```

7. Further information

The supplied ca1t_v3.jar contains both the compiled byte code and the Java source enabling further customisation if required. To modify and compile the code you will need the following:

- IBM SDK V5 (on any supported platform)
- ibmjzos.jar - Available in the Java 5 IBM SDK for z/OS in the lib/ext directory.
- ctgclient.jar – Available with the CICS TG V8 in the classes directory

For further information on implementing high availability scenarios using the CICS request exit refer to the CICS TG information centers.

z/OS:
http://pic.dhe.ibm.com/infocenter/cicstgzo/v9r0/index.jsp?topic=/com.ibm.cics.tg.zos.doc/ctgzos/c0100140.html

Multiplatforms/Desktop Edition:

http://pic.dhe.ibm.com/infocenter/cicstgmp/v9r0/index.jsp?topic=%2Fcom.ibm.cics.tg.doc%2Fctgunx%2Fc0100140.html

For further information on developing a CICS request exit refer to the com.ibm.ctg.ha package in the CICS TG javadoc:
http://pic.dhe.ibm.com/infocenter/cicstgzo/v9r0/index.jsp?topic=%2Fcom.ibm.cics.tg.zos.doc%2Fhajavadoc%2Findex.html

For further information on the ibmjzos package used in this SupportPac refer to:
http://www.ibm.com/developerworks/java/zos/javadoc/jzos/overview-summary.html

8. Troubleshooting

If the exit is correctly configured then at start-up you will see the following messages:

a. At Gateway startup if the round robin is successfully loaded:

```
05/05/11 15:20:58:416 [0] CTG8447I CICS Request exit
com.ibm.ctg.samples.ha.RoundRobinExit installed successfully
```

b. At exit initialisation when the sample configuration file is loaded (time stamps removed):

```
CA1T CICS request exit RoundRobinExit V2.1.0-20110505
CA1T Logging destination: default (stdout)
CA1T Loading main policy file /u/rcjones/ca1t/policy1.ini
CA1T Policy(Main): Server rule CHANNEL using actual servers [CICSB]
CA1T Policy(Main): Server rule COMMAREA using actual servers [CICSA]
CA1T Policy(Main): Server rule NULL using actual servers [CICSC]
CA1T Policy(Main): Server rule SERVER2 using actual servers [CICSA, CICSB]
CA1T Policy(Main): Server rule SERVER1 using actual servers [CICSA, CICSB, CICSC]
CA1T Policy(Main): PROGRAM validation rules set for [PROGRAM1, PROGRAM2/COMMAREA,
PROGRAM3/CHANNEL]
CA1T Policy Policy(Main): TRANSID validation rules set for [NULL, CSMI, CPMI]
CA1T Policy Policy(Main): USERID validation rules set for [BOB, ALICE, PHIL, NULL]
```

If an alternate policy is defined, it is also loaded and its rules are logged.

c. The exit then shows the active policy and global settings (time stamps removed):

```
CA1T CICS request exit RoundRobinExit V2.1.0 -20110505
CA1T Loading alternate policy file /u/rcjones/ca1t/policy2.ini
CA1T Active policy(Main): Server rule CHANNEL using actual servers [CICSB]
CA1T Active policy(Main): Server rule COMMAREA using actual servers [CICSA]
CA1T Active policy(Main): Server rule NULL using actual servers [CICSC]
CA1T Active policy(Main): Server rule SERVER2 using actual servers [CICSA, CICSB]
CA1T Active policy(Main): Server rule SERVER1 using actual servers [CICSA, CICSB,
CICSC]
CA1T Active policy(Main): PROGRAM validation rules set for [PROGRAM1, PRO-
GRAM2/COMMAREA, PROGRAM3/CHANNEL]
CA1T Policy Active policy(Main): TRANSID validation rules set for [NULL, CSMI,
CPMI]
CA1T Policy Active policy(Main): USERID validation rules set for [BOB, ALICE,
PHIL, NULL]
CA1T Configuration parameters: Intervals(retry=15s autoswap=60s) Logging level(1)
CA1T Set maximum retry count (4)
```

d. When the Gateway receives a command function to change the logging level the following message will be logged:

```
05/05/11 15:23:31:654 CA1T Command [CA1T_2] received
05/05/11 15:23:31:655 CA1T Level 2 tracing enabled
```

e. When the Gateway receives a swap command the following message will be logged, assuming an alternative policy is defined (time stamps removed):

```
CA1T Command [CA1T_SWAP] received
CA1T Swapping configuration files
CA1T Swapping to alternative policy: /u/rcjones/ca1t/policy2.ini
CA1T Active policy(Alternative): Server rule CHANNEL using actual servers
[CICSY]
CA1T Active policy(Alternative): Server rule COMMAREA using actual servers
[CICSX]
CA1T Active policy(Alternative): Server rule NULL using actual servers
[CICSZ]
CA1T Active policy(Alternative): Server rule SERVER2 using actual servers
[CICSX, CICSY]
CA1T Active policy(Alternative): Server rule SERVER1 using actual servers
[CICSX, CICSY, CICSZ]
CA1T Active policy(Alternative): PROGRAM validation rules set for [PROGRAM1,
PROGRAM2/COMMAREA, PROGRAM3/CHANNEL]
CA1T Policy Active policy(Alternative): TRANSID validation rules set for
[NULL, CSMI, CPMI]
CA1T Policy Active policy(Alternative): USERID validation rules set for [BOB,
ALICE, PHIL, NULL]
```

f. At exit termination when the shutdown event is fired the following message will be logged.

```
05/05/11 15:18:07:318 CA1T Shutdown initiated in RoundRobinExit
```

9. Notices

The provisions set out in the following two paragraphs do not apply in the United Kingdom or any other country where such provisions are inconsistent with local law:

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 states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

Information contained and techniques described in this publication have not been submitted to any formal IBM test and are distributed on an "AS IS" basis.

The use or implementation of any information contained and/or of any technique described in this document is the user's responsibility and depends on the user's ability to evaluate and integrate the information and/or technique into the user's operational environment. While IBM has reviewed each item for accuracy in a specific situation, IBM offers no guarantee or warranty that the same or similar results will be obtained elsewhere. Users attempting to adapt any technique described in this document to their own environments do so at their own risk.

The information contained in this publication could include technical inaccuracies or typographical errors.

Changes are periodically made to the information contained 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 reference in this publication to an IBM licensed program or another IBM product is not intended to state or imply that only IBM's program or other product may be used. Any functionally equivalent program that does not infringe applicable intellectual property rights may be used instead of the referenced IBM licensed program or other IBM product.

The user is responsible for evaluating and verifying the operation of the material supplied in conjunction with this publication in conjunction with other products, except those expressly designated by IBM.

International Business Machines Corporation may have patents or pending patent applications covering subject-matter described in this document. The furnishing of this document does not give you any license to any such patent. You can send license inquiries, in writing, to:

The IBM Director of Licensing
International Business Machines Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

10. Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

CICSPlex®, CICS®, CICS Explorer

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a trademark of The Open Group in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.