

Oracle® Communications Diameter Signaling Router

Zero Balance Application User Guide



Release 9.1.0.0.0
G16506-01
November 2024

ORACLE®

Copyright © 2018, 2024, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

1	Introduction	
1.1	Audience	1-1
1.2	References	1-1
1.3	Acronyms	1-2
1.4	My Oracle Support	1-2
2	Understanding ZBA Functionality and ZBA Logic	
2.1	ZBA Functionality	2-2
2.2	ZBA Logic Process	2-2
3	Configuring the ZBA Application	
3.1	Verifying the ZBA Activation	3-1
3.2	Importing the ZBA JSON File	3-1
3.3	Configuring ZBA Config_Data, Egress_StatusCode, and Ingress_StatusCode Tables	3-1
3.4	Configure Active UDR NOAM for ZBA application	3-2
3.5	Disabling ZBA	3-3
4	Upgrading the ZBA	
5	Customize the ZBA Behavior	
6	ZBA Functions	
6.1	Custom MEALS	6-1
6.2	General Options	6-1
6.3	Trial MPs Assignment	6-1
6.4	Application Control	6-1
6.5	System Options (SO Only)	6-2

What's New in This Guide

This section introduces the documentation updates for release 9.1.0.0.0.

Release 9.1.0.0.0 - G16506-01 , November 2024

Added [Upgrading the ZBA](#) procedure to provide information about upgrading to ZBA.

1

Introduction

Zero Balance Application (ZBA) allows network operators to reduce the load (offload) sent to the Online Charging Servers (OCS) by blocking those subscribers whose accounts have exceeded their credit limit, that is, Zero Balance. To achieve this, the ZBA detects the Credit Control Request (CCR) and Credit Control Answer (CCA) exchanged with the OCS and prevents subscribers whose account balance have gone down to zero from repeatedly querying the OCS. ZBA maintains records of subscribers, identified by their Mobile Station International Subscriber Directory Number (MSISDN), whose account balance have gone down to zero.

This document provides information about ZBA and its functionalities. The ZBA menu options allow you to work with:

- Custom MEALS
- General Options
- Trial MPs Assignment
- Application Control
- System Options (SO only)

ZBA is available as a Diameter Custom Application (DCA) framework application. Like other DCA framework applications, you can use ZBA to work with the DCA framework functions. If ZBA is in the DCA framework GUI menu, the application has already been activated.

The document provides the following types of information:

- ZBA tasks
- ZBA application logic
- Procedures to configure and manage ZBA components including configuration parameters and ZBA profile tables
- ZBA components and GUI elements

1.1 Audience

This content is intended for personnel using the ZBA within the DCA framework. The content includes procedures for performing tasks using the product GUI. However, it does not describe how to install or replace software or hardware.

The ZBA software interacts with UDR. For this reason, this content includes references to the shared applications, and might describe GUI options that are not visible or applicable to ZBA.

1.2 References

The following references capture the source material used to create this document. These documents are included in the Oracle Communications Diameter Signaling Router (DSR) documentation set.

- DCA Programmer's Guide
- DCA Provisioning Interface

- DCA Feature Activation
- Diameter User's Guide

1.3 Acronyms

An alphabetized list of acronyms or terms used in this document.

Table 1-1 Acronyms/Terms

Acronym/Term	Definition
API	Application Program Interface
ART	Application Routing Table
CCA	Credit Control Answer
CCA-I/E	Credit Control Answer (Initial/Event)
CCR	Credit Control Request
CCR-I/E	Credit Control Request (Initial/Event)
CTF	Charging Triggering Function
DCA	Diameter Custom Application
DET	Zero Balance Detection Interval (configured)
DRA	Diameter Routing Agent
DSR	Diameter Signaling Router
ECC	Extended Command Code
GUI	Graphic User Interface
MEALS	Measurements, Events, Alarms, and Logs
MMI	Machine to Machine Interface
MP	Measurement Platform
OC-DRA	Online Charging DRA
OCS	Online Charging Server
PCEF	Policy and Charging Enforcement Function
SO	System Options
UDR	Unified Data Repository
ZBA	Zero Balance Application
ZBC	Zero Balance Counter
ZB-DB	Zero Balance Application Database
ZBT	Zero Balance Blocking Timer
ZDT	Zero Balance Detection Timer

1.4 My Oracle Support

My Oracle Support (<https://support.oracle.com>) is your initial point of contact for all product support and training needs. A representative at Customer Access Support can assist you with My Oracle Support registration.

Call the Customer Access Support main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at <http://www.oracle.com/us/support/contact/index.html>. When calling, make the selections in the sequence shown below on the Support telephone menu:

1. Select **2** for New Service Request.

2. Select **3** for Hardware, Networking and Solaris Operating System Support.
3. Select one of the following options:
 - For Technical issues such as creating a new Service Request (SR), select **1**.
 - For Non-technical issues such as registration or assistance with My Oracle Support, select **2**.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

2

Understanding ZBA Functionality and ZBA Logic

Zero Balance Application (ZBA) is a business logic application that functions within the DCA framework.

The DCA frame must first be active before activating the ZBA to access the ZBA GUI menu functionality.



Note:

The DCA framework is a set of APIs and services made available to DCA developers (internal or external) who need to development application that run in the Diameter application layer of DSR.

When a mobile data service subscriber initiates a data session, a network Element (PCEF or CTF) might attempt to set up a data session for the subscriber and query the OCS for service units. In case the subscriber has no credit in their account, the OCS rejects the request, usually with an indication that the subscriber has no credit. However, in most cases the PCEF, or other Charging Trigger Functions (CTFs), continues to attempt to query the OCS. This can create an overload on the OCS to handle requests for subscribers that are always denied during the period in which the subscriber has no credit.

The purpose of the ZBA is to detect when a subscriber, identified by a MSISDN, has a zero balance in the account, the ZBA under certain configurable conditions, might respond on behalf of the OCS and reduce the number of queries the OCS must handle for subscribers who temporarily have no credit.

ZBA is configured as the owner of a UDR database. To avoid overloading of ZBA, the Application Routing Table (ART) is configured to route only CCR- I/E type messages to ZBA.

ZBA can be enabled and disabled as a DCA framework. Disabling ZBA on a specific site is possible only if ZBA has been disabled on all the DA-MPs on that specific site.

If ZBA is in the DCA framework GUI menu, this indicates that the application is already enabled. However, it does not guarantee that it is provisioned. After ZBA is installed, a manual enablement is necessary on at least one DA-MP server.



Note:

You can enable or disable ZBA from **Diameter**, and then **Maintenance**, and then **Applications** screens on the SOAM GUI. When the application is initially installed, it is disabled and it must be manually enabled.

The ZBA functionality varies between the SOAM and NOAM. The following menu options are available on the NOAM:

- The **Custom MEALs** menu provisions the measurements linked to the ZBA application.

- The **General Options** sets the options of the application.
- The **Trial MP Assignment** chooses the MP to run the application.
- The **Application Control** menu creates a new development environment and modifies them.

The SOAM has the same menu options in read only and an additional menu option called **System Options** where the behavior of the application can be configured in case of errors.

2.1 ZBA Functionality

ZBA can inspect ingress Gy/Ro Request and Answer messages to verify if subscribers who might have zero balance accounts must be blocked (based on answer messages). It uses MSISDN to check the subscriber's status in UDR. It forwards the ingress Gy/Ro Request messages to the OCSs if the subscriber's status is not in the "blocking" state.

Subscribers might have zero balance, but their messages might be routed to OCS if the number of Zero Balance events detected is less than a configured threshold during a configured detection interval.

ZBA blocks subscribers whose number of Zero Balance events has exceeded the configured threshold within the configured detection interval and generates a response message (CCA-I/E) on behalf of the OCS when a subscriber is blocked.

2.2 ZBA Logic Process

ZBA logic follows this process:

1. The ZBA logic is triggered with the reception of a CCR-I/E message.
2. The ZB reads the MSISDN from the Subscription-ID Grouped or the User-Name AVPs in the CCR-I/E message.
3. ZBA checks if the MSISDN extracted from the AVP already has a record entry in the UDR.
4. If no entry exists for that user, the application forwards the CCR-I/E to OCS using DRL and creates a new entry in the UDR. If a qualified ZB event has been received in the CCA-I/E, that subscriber is then monitored. If the OCS answer contains a result code, which does indicate zero balance, then the Zero Balance counter is incremented in the UDR and detection state is set to ON. ZBA forwards the CCA-I/E received from the OCS using DRL.
5. If an entry for that user exists, the application reads the status of the blocking and detection timers from the UDR to determine whether the subscriber is blocked or being monitored.
6. If the subscriber is not blocked, or blocked with expired blocking timer, ZBA forwards the CCR-I/E to OCS using DRL and resets the record to its initial state, for example, not monitored, not blocked.
7. If a request is received when the subscriber is blocked and not expired (blocking timer is ON, but not expired), the request is not forwarded to OCS, and instead the ZBA sends an answer message with the configurable egress blocking result code to CTF using DRL (the OCS is not involved).
8. ZBA continues to monitor the subscriber and answer on behalf of the OCS until the blocking timer expires.
9. ZBA does not process the Diameter messages in the following cases:
 - Missing MSISDN in the CCR-I/E

- ZBA UDR Database is unavailable
- Stack event errors have been generated during processing

3

Configuring the ZBA Application

Activate the DCA framework and ZBA.



Note:

Following activation, individual applications are in the disabled state. When disabled, no Diameter traffic is delivered to an application.

- To enable ZBA as an independent application, enable ZBA from the **Diameter**, and then **Maintenance**, and then **Applications** screen on the SO on the desired DA-MPs.

For more information, refer to the *Diameter User's Guide*.

3.1 Verifying the ZBA Activation

1. To verify whether ZBA is activated, navigate to the DCA Framework menu.
Zero Balance Application must be displayed on the screen.
2. To enable business logic and provision configuration data, by accessing the **DCA Framework**, and then **Zero Balance Application**, and then **Application Control** screen.

3.2 Importing the ZBA JSON File

Perform the following procedure to import the business logic, configuration tables, and MEALS from a previously exported json file.

Ensure that ZBA is activated.

1. From the NOAM, navigate to **DCA Framework**, and then **Zero Balance Application**, and then **Application Control**.
2. Click **Import Business Logic** to import the json file.
3. Select the file.
4. Click **Import**.

This creates the configuration data, the custom MEAL templates.

When imported, the application must be syntax checked and compiled before it can go to the production cycle.

3.3 Configuring ZBA Config_Data, Egress_StatusCode, and Ingress_StatusCode Tables

ZBA uses the following tables to collect configuration values:

- Config_Data

- Egress_ResultCode
- Ingress_ResultCode

For information about ZBA tables, refer to [Configured ZBA Database Tables](#).

This procedure configures Config_Data, Egress_ResultCode, and Ingress_ResultCode tables. The Config_Data table contains the configurable parameters that determine the detection period, the Blocking period, and the Max Zero Balance allowable Events. The Egress_ResultCode table contains the configurable result code used when ZBA generates an answer during a blocking period. The Ingress_ResultCode table contains the configurable result codes used to determine a Zero Balance event when an answer message contains one of the configured result codes.



Note:

The available GUI choices differ between the SO and the NO. For information about GUI details and functionality, refer to the *DCA Programmer's Guide*.

The fields are described in the associated database tables elements.

1. On the SOAM, navigate to **DCA Framework**, and then **Zero Balance Application**, and then **Application Control**.
2. Select a version name.
3. Click **Config Data**.
4. Select a table name.
5. Select an action, for example, View to view data, Provision Table to populate it.
6. Enter the fields to define or edit the selected table.

3.4 Configure Active UDR NOAM for ZBA application

1. From the main Menu, Navigate to **UDR**, then select **Subscriber Entity Configuration**, then select **Transparent Entity**, select **Base Field Set**, and click **Edit**.
2. Under **SprProfileBFS** add these fields for ZBA Application.

Figure 3-1 SprProfileBFS screen

ACLE User Data Repository 12.6.1.0.0-18.7.0

Pause Updates | Help | Logged in Account: guidadmin | Log Out

Main Menu: UDR -> Subscriber Entity Configuration -> Transparent Entity -> Base Field Set -> [Edit]

Warning

Base Field Set Name * SprProfileBFS
Name of this Base Field Set Definition.
[Default = n/a; Range = A 64-character string. Valid characters are alphanumeric and underscore. Must contain at least one alpha and must not start with a digit.] [A value is required.]

Element String * subscriber
XML element name under which defined fields reside.
[Default = n/a; Range = A 64-character string. Valid characters are alphanumeric and underscore. Must contain at least one alpha and must not start with a digit.] [A value is required.]

Version Identifier Element
Check box if multiple versions of this entity are allowed. [Default = unchecked]

Allow Versions ☐ Version Value
XML element name of the version identifier.
[Range = A 64-character string. Valid characters are alphanumeric and underscore. Must contain at least one alpha and must not start with a digit.]

Version value associated with this BaseFieldSet. [Range = Must be an integer]

XML Storage Format Field Name Based
Select if entity version is element based or field name based.
[Default = Element Based; Select value from list]

Element String	Type	Field Set Name	Min Value	Max Value	Updatable	Resettable	Default Value	Min Occur	Max Occur	Special

 Note:

If DSA application is activated on that setup, the following fields are present.

Figure 3-2 SprProfileBFS field screen

Main Menu: UDR -> Subscriber Entity Configuration -> Transparent Entity -> Base Field Set -> [Edit] Tue Aug 24 02:24:52 2021

Warning

XML Storage Format: Field Name Based [Default = Element Based; Select value from list]

Element String	Type	Field Set Name	Min Value	Max Value	Updatable	Resettable	Reset Value	Defaultable	Default Value	Min Occur	Max Occur	Special Format
01 MSISDN	RegEx		0	0	☒	☐		☐		0	3	^d{8,15}\$
02 IMSI	RegEx		0	0	☒	☐		☐		0	3	^d{10,15}\$
03 NAI	RegEx		0	0	☒	☐		☐		0	3	^[x20-x7e]\$
04 AccountId	RegEx		0	0	☒	☐		☐		0	1	^[x20-x7e]\$
05 IMEI	RegEx		0	0	☒	☐		☐		0	2	^d{8,14}\$
06 lastUN	RegEx		0	0	☒	☐		☐		0	1	^[x20-x7e]\$
07 VLR	RegEx		0	0	☒	☐		☐		0	1	^[x20-x7e]\$
08 VPLMN	RegEx		0	0	☒	☐		☐		0	1	^[x20-x7e]\$
09 MCC	RegEx		0	0	☒	☐		☐		0	1	^[x20-x7e]\$
10 MMER	RegEx		0	0	☒	☐		☐		0	1	^[x20-x7e]\$
11 MMEH	RegEx		0	0	☒	☐		☐		0	1	^[x20-x7e]\$

3. Configure the following fields.

ZBA Specific Fields :
MSISDN - key

ZbEventCount -- value contains digit : Reg Exp : ^d+\$
BlockingStatus : value contains string "unblock/block" :
^[x20-x7e]{0,255}\$
BlockTime: value contains digit Reg Exp: ^d+\$
DetectionState : value contains string "ON/OFF" Reg Exp :
^[x20-x7e]{0,255}\$
DetectionTime : value contains digit : Reg Exp : ^d+\$

Figure 3-3 Configuration fields

20 ZbEventCount	RegEx				☒	☐		☐		0	1	^d+\$
21 BlockingStatus	RegEx				☒	☐		☐		0	1	^[x20-x7e]{0,255}\$
22 BlockTime	RegEx				☒	☐		☐		0	1	^d+\$
23 DetectionState	RegEx				☒	☐		☐		0	1	^[x20-x7e]{0,255}\$
24 DetectionTime	RegEx				☒	☐		☐		0	1	^d+\$

3.5 Disabling ZBA

Perform the following procedure to disable ZBA from the SO.

 Note:

The available GUI choices differ between the SO and the NO. For information about GUI details and functionality, refer to the *DCA Programmer's Guide*.

1. Navigate to **Diameter**, and then **Maintenance**, and then **Applications**.
2. Select the DCA_ZBA on the MP where you want to disable the application.
3. Click **Disable**.

4

Upgrading the ZBA

The upgrade from Zero Balance Application (ZBA) with Universal Session Binding Repository U-SBR to ZBA with UDR DSR release is not supported.

The upgrade from ZBA with UDR to ZBA with UDR DSR Release is supported.

1. Export the SOAM level configuration data to the previous release setup.
 - Log in to the SOAM GUI.
 - From the **Main Menu**, navigate to **DCA Framework, Zero Balancing Application**, and then **Application Control**. Ensure that an entry is added in **DCA application** version details table.
 - Select the required **Zero Balance Application** version name.
 - Click **B Level Config Data** in the **Export** section of the **Application Control** page.

Figure 4-1 Application Control Page

Main Menu: DCA Framework -> Zero Balancing Application -> Application Control Thu Jun 20 09:57:15 2024 EDT

Version Name	Status	Comments	Creation Time	Production Time	Flowchart Checksum
Version1	Development	DCA Based Zero Balance Application Version 1	2024-Jun-20 05:53:29 EDT		1e3ec4bf5b142c8b0d9dee15b039a051

Filter*

Config Data Development Environment

Import: **B Level Config Data**

Export: **B Level Config Data**

2. To import the configuration data to a new release setup, the ZBA should be fresh installed on a new release. The application should be deactivated, then activated, and then NOAM json file of the upgraded release should be imported. For more information, see [Importing the ZBA JSON File](#).
3. Perform the following procedure to import the SOAM level json:
 - a. Log in to the active SOAM GUI.
 - b. From the SOAM GUI **Main Menu**, navigate to **DCA Framework, Zero Balancing Application**, and then **Application Control**.
 - c. Click **B Level Config Data** in the import section of the Application Control page , select the **file**, and then click **Import** to import the SOAM configuration data.

5

Customize the ZBA Behavior

ZBA can be customize using the **DCA Framework**, and then **Zero Balance Application** screen or as an independent application on the following DSR menus:

- Custom MEALS
- General Options
- Trial MPs Assignment
- Application Control
- System Options (SO only)



Note:

For specific information about configuration activities, refer to [Configured ZBA Database Tables](#). Also, refer to the *DCA Programmer's Guide* and *DCA Provisioning Interface*.

The ZBA configuration database schema uses the following options to customize the application behavior:

- **Detection Interval:** The time interval during which Zero Balance events are monitored. Range: 1-30. 30. Default is 5 (values are in seconds).
- **Blocking Interval:** The time duration during which a subscriber is blocked after exceeding the number of allowed zero balance events. Range: 60-1800. Default is 60 (values in seconds).
- **Maximum ZB Events:** The number of allowed zero events during a detection interval before blocking a subscriber. Range: 1-5. Default is 1 (value is integer).
- **Egress Result Code:** The configurable egress result code used when ZBA generates an answer during a blocking period (value is string).
- **Ingress Result Code:** One to four ingress result codes can be configured to determine a Zero Balance event when an answer message contains one of these result codes (values are string).

6

ZBA Functions

Like other DCA framework applications, you can use ZBA to work with the DCA framework functions. If ZBA is on the DCA framework screen, the application is already activated. However, it does not guarantee that it is provisioned. After ZBA is installed, enable it manually and complete the configuration items, such as DB association, Config_Data parameters, Diameter connections, ART, and so on.

For information about specific configuration activities, refer to [Configured ZBA Database Tables](#).

6.1 Custom MEALS

A list of any previously configured ZBA custom defined measurements and events (NO view GUI page). Use this screen to insert, edit, and delete custom MEALS.

6.2 General Options

Available from the NO and SO (view only), General options allows you to configure the ZBA functionality.

From the NO, this screen enables specifying the Perl Subroutine for Diameter Request and Answer.

From the SO, this screen allows you to view the Perl Subroutine specified for Diameter Request and Answer in the read-only mode.

6.3 Trial MPs Assignment

From the NO, this screen allows specifying which MPs run the trial version of an application. If there is a production version, all MPs enabled for the ZBA application run the application regardless of which is the trial MP. If there is a trial application version configured in the system, but no trial MPs are specified, a warning message is generated.

From the SO, this screen allows you to view which MPs run the trial version of an application. The page is accessible in the read-only mode.

6.4 Application Control

From the NO, this page allows:

- Listing all application versions configured in the system
- Inserting a new application version
- Copying and modifying an existing application version
- Exporting an application version entirely (business logic + provisioned data from the NO)
- Exporting only the NO provisioned data of an application version
- Importing a previously exported application version (business logic + NO provisioned data)

- Importing only the NO provisioned data to an existing application version
- Accessing the application version configuration tables
- Accessing a flowchart of an application version
- Deleting an existing application version
- Changing the status of an application version (Development, Trial, Production, Archived)

From the SO, this page allows:

- Listing all application versions configured in the system
- Exporting only the SO provisioned data of an application version
- Importing only the SO provisioned data to an existing application version
- Accessing the application version configuration tables
- Accessing a flowchart of an application version (read-only)

You can control a variety of ZBA functions from the Application Control screen, such as creating new application versions, copying and editing existing versions, importing and exporting business logic, and making trials. You can also copy an entire application with the business logic and the NO provisioned data and assign a new version name to it.

6.5 System Options (SO Only)

Use this task to work with ZBA Systems Options, which includes the following categories:

- **Run-time error configuration:** Behavior of the DSR application when a Run-time error occurs
- **Resource exhaustion configuration:** Behavior of the DSR application when internal resources are being exhausted
- **Application unavailable configuration:** Action to be taken when the application is unavailable to process messages
- **Configuration for the DCA generated Answer:** To specify a realm and FQDN different from the default one

7

Configured ZBA Database Tables

This section describes ZBA application database table content and functionality. Navigate to **DCA Framework**, and then **Zero Balance Application**, and then **Application Control**, and select the current version of the ZBA, and then click **Config Data** to access the configuration table page.

Table 7-1 ZBA Database Table Fields

Field	Description	Single Row Setting	Level
Config_Data	Contains configuration parameters for the ZBA	YES	SO
Egress_ResultCode	Contains the configurable result code used when ZBA generates an answer during a blocking period	YES	SO
Ingress_ResultCode	Contains one to four ingress result codes that can be configured to determine a Zero Balance event when an answer message contains one of these result codes	YES	SO