User Guide

Interaction Data Server

AVAYA

Contact Center Express

Release 2.0 - Issue 0
Notice

Every effort was made to ensure that the information in this manual was complete and accurate at the time of printing. However, information is subject to change.

Your Responsibility for Your System's Security

Toll fraud is the unauthorized use of your telecommunications system by an unauthorized party, for example, persons other than your company's employees, agents, subcontractors, or persons working on your company's behalf. Note that there may be a risk of toll fraud associated with your telecommunications system and, if toll fraud occurs, it can result in substantial additional charges for your telecommunications services.

You and your system manager are responsible for the security of your system, such as programming and configuring your equipment to prevent unauthorized use. The system manager is also responsible for reading all installation, instruction, and system administration documents provided with this product in order to fully understand the features that can introduce risk of toll fraud and the steps that can be taken to reduce that risk. Avaya Inc. does not warrant that this product is immune from or will prevent unauthorized use of common-carrier telecommunication services or facilities accessed through or connected to it. Avaya Inc. will not be responsible for any charges that result from such unauthorized use.

Avaya Fraud Intervention

If you suspect that you are being victimized by toll fraud and you need technical support or assistance, call the Technical Service Center Toll Fraud Intervention Hot-line at 1-800-643-2353.

Trademarks

- Avaya Computer Telephony is a registered trademark of Avaya Inc.
- Avaya CallMaster is a registered trademark of Avaya Inc.
- Definity is a registered trademark of Avaya Inc.
- MultiVantage is a registered trademark of Avaya Inc.
- INTEL and Pentium are registered trademarks of Intel Corporation.
- Microsoft, MS, MS-DOS, and Windows are registered trademarks of Microsoft Corp.

All other product names mentioned herein are the trademarks of their respective owners.

Avaya National Customer Care Center

Avaya provides a telephone number for you to use to report problems or to ask questions about your contact center. The support telephone number is 1-800-242-2121. For technical support, customers outside the United States should call their Avaya representative or distributor.

European Union Declaration of Conformity

Avaya Inc. Business Communications Systems declares that the equipment specified in this document conforms to the referenced European Union (EU) Directives and Harmonized Standards listed below:

- EMC Directive 89/336/EEC
- Low Voltage Directive 73/23/EEC

The CE” mark affixed to the equipment means that it conforms to the above Directives.

Website

For more information on all Avaya Contact Center Express products, refer to the company website (http://www.AvayaContactCenterExpress.com).
Software License Agreement

Definitions

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avaya</td>
<td>Avaya Inc.</td>
</tr>
<tr>
<td>You, your or licensee</td>
<td>The person or business entity who purchased this license to use this client software or for whom such license was purchased.</td>
</tr>
<tr>
<td>Client software</td>
<td>A software application that operates on a computer system.</td>
</tr>
<tr>
<td>Documentation</td>
<td>The manual and any other printed material provided by Avaya for the client software.</td>
</tr>
<tr>
<td>License</td>
<td>The license purchased and granted pursuant to this agreement.</td>
</tr>
</tbody>
</table>

License and Protection

License Grant. Avaya grants to you, subject to the following terms and conditions, a nonexclusive, nontransferable right to use the client software on one or more single-user devices. The maximum simultaneous users of the client software being limited to the number of single-user licenses purchased and owned by you. Avaya reserves all rights not expressly granted to you.

Protection of Software. You agree to take all reasonable steps to protect the client software and documentation from unauthorized copy or use. The client software source code represents and embodies trade secrets of Avaya and/or its licensors. The source code and embodied trade secrets are not licensed to you and any modification, addition, or deletion is strictly prohibited. You agree not to disassemble, decompile, or otherwise reverse engineer the client software in order to discover the source code and/or the trade secrets contained in the source code or for any other reason. To the extent that the client software is located in a Member State of the European Community and you need information about the client software in order to achieve interoperability of an independently created software program with the client software, you shall first request such information from Avaya. Unless Avaya refuses to make such information available, you shall not take any steps, such as reverse assembly or reverse compilation, to derive a source code equivalent to the client software. Avaya may charge you a reasonable fee for the provision of such information.

Copies. You may make multiple copies of the client software for your own use with Avaya contact center agent digital voice terminals, provided you do not violate the License Grant in paragraph 1, and you do not receive any payment, commercial benefit, or other consideration for reproduction or use. You may not copy documentation unless it carries a statement that copying is permitted. All proprietary rights notices must be faithfully reproduced and included on all copies.

Ownership. Ownership of, and title to, the client software and documentation (including any adaptations or copies) remains with Avaya and/or its licensors.

Restrictions. You agree not to rent, lease, sublicense, modify or time share the client software or documentation.

Termination. This agreement shall automatically terminate if you breach any of the terms or conditions of this agreement. You agree to destroy the original and all copies of the client software and documentation, or to return them to Avaya, upon termination of this license.
Limited Warranty and Limited Liability

Compatibility. The client software is only compatible with certain computers and operating systems. The software is not warranted for noncompatible systems.

Software. Avaya warrants that if the client software fails to substantially conform to the specifications in the documentation and if the client software is returned to the place from which it was purchased within one (1) year from the date purchased, then Avaya will either replace the client software or offer to refund the license fee to you upon return of all copies of the client software and documentation to Avaya. In the event of a refund, the license shall terminate.

Disclaimer of Warranties. Avaya makes no warranty, representation or promise not expressly set forth in this agreement. Avaya disclaims and excludes any and all implied warranties of merchantability or fitness for a particular purpose. Avaya does not warrant that the client software or documentation will satisfy your requirements or that the client software or documentation are without defect or error or that the operation of the software will be uninterrupted. Some states or countries do not allow the exclusion of implied warranties or limitations on how long an implied warranty lasts, so the above limitation may not apply to you. This warranty gives you specific legal rights which vary from state to state.

Exclusive Remedy. Except for bodily injury caused by Avaya's negligence, Avaya's entire liability arising from or relating to this agreement or the client software or documentation and your exclusive remedy is limited to direct damages in an amount not to exceed $10,000. Avaya shall not in any case be liable for any special incidental, consequential, indirect or punitive damages even if Avaya has been advised of the possibility of such damages. Avaya is not responsible for lost profits or revenue, loss of use of the client software, loss of data, costs of recreating lost data, the cost of any substitute equipment or program, or claims by any party other than you. Some states or countries do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

General Conditions

Governing Law. This agreement shall be governed by, and interpreted in accordance with, the substantive laws of the State of New Jersey of the United States of America.

Entire Agreement. This agreement sets forth the entire understanding and agreement between you and Avaya and may be amended only in a writing or writings signed by you and Avaya. No vendor, distributor, dealer, retailer, sales person or other person is authorized to modify this agreement or to make any warranty, representation or promise which is different than, or in addition to, the representations or promises of this agreement about the software.

Export. Licensee hereby agrees that it will not knowingly, directly or indirectly, without prior written consent, if required, of the Office of Export Licensing of the U.S. Department of Commerce, Washington D.C. 20230, export or transmit any of the Products to any group Q, S, W, Y, or Z country specified in the Export Administration Regulations issued by the U.S. Department of Commerce or to any country which such transmission is restricted by applicable regulations or statues.

U.S. Government Restricted Rights. Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in FAR 52.227-14 (June 1987) Alternate III (g)(3) (June 1987), FAR 52.227-19 (June 1987), or DFARS 52.227-7013 (c)(1)(ii) (June 1988), as applicable Contractor/Manufacturer is Avaya Inc. 11900 North Pecos Street, Westminster, Colorado 80234.

Assignment. Avaya may without your consent or notice to you, assign this agreement to an entity to which it transfers ownership of the client software. Upon the effective date of such assignment, you agree that Avaya shall be released and discharged from all obligations and liabilities under this agreement.
### Contents

**Software License Agreement** 3

**Chapter 1  Preface** 7
- Document Conventions................................................................. 8
- Related Documents ........................................................................ 9
  - Related Documents .................................................................... 9
- Product Name Change .................................................................. 10
- Knowledge Base ......................................................................... 10

**Chapter 2  Introduction** 11
- What is the Interaction Data Server? ........................................... 12
- Configuration ................................................................................ 15
- Call Event Logging ....................................................................... 15
- Error Logging ............................................................................... 16

**Chapter 3  Installation** 17
- Install Application ........................................................................ 18

**Chapter 4  Configuration** 19
- Create/Update Database ............................................................. 20
- Configurable Parameters ............................................................ 23
- Configure Interaction Data Server via Ini File .............................. 27
- Start Interaction Data Server ...................................................... 28

**Database** 30
- Overview ...................................................................................... 31
- System Data .................................................................................. 31
- Event Data ..................................................................................... 31
- Call Summary Data .................................................................... 32
- Statistics Data ............................................................................. 32
- User Data ..................................................................................... 32
- Configuration Issues .................................................................. 33
  - Definity/MultiVantage/Avaya CM server and Telephony Server .... 33
  - Interaction Data Server ............................................................ 33
  - Key Configuration Points ......................................................... 33
- Database Diagram ....................................................................... 35
- Tables ........................................................................................... 36
  - System Tables ........................................................................... 36
  - Event Tables ............................................................................. 38
- Call Summary Table ................................................................... 40
  - Statistics Table ........................................................................ 40
  - User Table ................................................................................ 40
- Stored Procedures ....................................................................... 42
  - Stored Procedures used by Interaction Data Server .................. 42
This chapter provides information that will help you use this document.

**In This Chapter**

Document Conventions .................................................. 8  
Related Documents.......................................................... 9  
Product Name Change...................................................... 10  
Knowledge Base............................................................... 10
# Document Conventions

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Capital Letters</td>
<td>Names of windows and dialog boxes. For example, the Add VDN dialog box appears.</td>
</tr>
<tr>
<td>[key] or [button]</td>
<td>The name of a button or keyboard key. For example, click the [Blind] button or press the [F5] key.</td>
</tr>
<tr>
<td>Key+key</td>
<td>Hot key combinations you press down simultaneously to make the computer perform a function. For example, the Ctrl+S hot key combination saves your document.</td>
</tr>
<tr>
<td><em>Italicized text</em></td>
<td>Reference documents.</td>
</tr>
<tr>
<td>Click and double-click</td>
<td>The action of pressing the left or right mouse button once or twice. Always click the left button unless the right button is specified.</td>
</tr>
</tbody>
</table>
Related Documents

There are many documents that are helpful when setting up a CTI environment. The following are included on the Avaya Computer Telephony CD-ROM:

- Telephony Services and CallVisor PC Installation (Install.pdf)
- Telephony Services Administration and Maintenance (Netmangd.pdf)
- Administration and Maintenance for Definity Enterprise Communications Server and CallVisor PC (Defnetm.pdf)
- Programmer's Guide for Definity Enterprise Communications Server (Defprog.pdf)
- Telephony Services Simulator User's Guide for Definity Enterprise Communications Server (Simguide.pdf)
- Telephony Services Application Programming Interface Programmer's Reference (Tsapi.pdf)
- Java Telephony API (JTAPI) Programmer's Reference (JTAPI v1.2 Specification) (Jtapi.pdf)
- Java Telephony API (JTAPI) for Definity Enterprise Communications Server Programmer's Reference (Jcli.pdf)
- Telephony Services CSTA Services PBX Driver Interface Specification (Csdi.pdf)
- Telephony Services PBX Driver Interface Specification (Tsdi.pdf)
- CallVisor PC Programmer's Reference for Definity ECS (Callvisr.pdf)
- Definity Enterprise Communications Server CallVisor ASAI Technical Reference (Asaitech.pdf)
- Definity Enterprise Communications Server CallVisor ASAI Protocol Reference (Asaiprot.pdf)

For more documentation on the Definity Enterprise Communications Server, refer to Avaya's Support Center website (see Product Documentation - http://support.avaya.com).

Related Documents

For information on how to develop client applications using Interaction Data Client, refer to the Interaction Data Client Developer Guide.
Product Name Change

Avaya Contact Center Express

With release 5.0, the Avaya Contact Center Express suite of CTI applications and development tools was known as Avaya Active Telephony.

Prior to release 5.0, Avaya Contact Center Express was known as Avaya Active Enterprise.

Avaya Computer Telephony

The Avaya Computer Telephony (Avaya CT) software referred to in this document was previously known as CentreVu Computer Telephony (CentreVu CT). Contact Center Express runs with CentreVu CT, Release 3.3, 9.1, 9.5 and 10.1, as well as Avaya CT, Release 11.1 (with Service Pack 3) and Avaya CT 1.2.

Knowledge Base

For information on any errors and updates relating to this document, visit the Avaya Contact Center Express Knowledge Base via the website (http://www.AvayaContactCenterExpress.com).
CHAPTER 2

Introduction

This chapter explains the function of the Interaction Data Server.

In This Chapter

What is the Interaction Data Server? ........................................ 12
Configuration.............................................................................. 15
Call Event Logging ................................................................. 15
Error Logging ........................................................................... 16
What is the Interaction Data Server?

The Interaction Data Server (AIDServer.exe) is an application that runs as a service on a Windows NT or 2000 server. It monitors VDNs, splits/skills, trunk groups and agent extensions to gather detailed statistical information about all facets of a call. The server receives real-time information and, based on the regularity you specify, calculates statistics such as talk and wait-time averages. The information is managed locally in memory and recorded to a SQL database.

The Interaction Data Client control, AIDSLibX.ocx, is distributed as part of the Developer toolkit. It allows developers to build applications that access real-time statistical data stored in the Interaction Data Server and display it on individual client PCs.

For information on the methods and events exposed by AIDSLibX.ocx, refer to Interaction Data Client Developer Guide.
Simple Architecture

At its simplest configuration, the Interaction Data Server receives information about a single Definity/MultiVantage/Avaya CM server. Clients connect to it and request data about devices. In this example, client A requires real-time information about station 8000 on Definity 1. The client makes that request to Interaction Data Server 1, which constructs the reply and forwards it back to the client.

![Diagram of simple architecture]

Complex Architecture

In a more complex configuration, a Interaction Data Server acts as part of a network of Interaction Data Servers receiving information from multiple Definity/MultiVantage/Avaya CM servers. Clients are attached to any of the servers in the network. Requests received from a client are forwarded to the correct Interaction Data Server, which processes the request and passes the response back to the original server.
In this example, client A is connected to Interaction Data Server 2 and requires real-time information about station 8000 on Definity 1. The client makes that request to Interaction Data Server 2. Interaction Data Server 2 has been programmed with the location of the Interaction Data Server managing the link to Definity 1, in this case Interaction Data Server 1. Interaction Data Server 2 forwards the request to Interaction Data Server 1, which constructs the correct reply and returns it to Interaction Data Server 2. Interaction Data Server 2 passes the information to client A.
Configuration

On startup, the Interaction Data Server extracts all the information it needs to run via the Configuration Client. Configuration information can be sourced from a networked Configuration Server or the local AIDServer.ini file.

Configuration information includes the names of VDNs, splits/skills and station numbers to be monitored, as well as the interval for statistical calculations. It also includes information about other instances of the Interaction Data Server running on other machines.

Once it has received its configuration data, the server opens a connection to the Telephony Server and starts issuing the necessary monitor requests to gather information. It also starts listening for Interaction Data Server clients and opens a connection to any other Interaction Data Servers.

Call Event Logging

The Interaction Data Server logs all call event data received from the Telephony Server to a SQL database for future reporting. Logging may be turned on and off through initial configuration parameters or dynamically while the Interaction Data Server is running. The server uses the Definity/MultiVantage/Avaya CM server-generated Universal Call ID and the switch identifier to store each event received from the Telephony Server. Either all or no events are logged for a switch. The Interaction Data Server also stores agent login and logout events to the database.

The Interaction Data Server also receives event information from clients and stores it in the database as a client event. Each client event is in the form of a name/value data pair.

Logging of call-related events to the Interaction Data Server database is independent of any other switch-related activity.
Error Logging

The Interaction Data Server logs error information relating to its own operation to a series of log files.

A new log file is created for each day of the week. Each error log is overwritten on a weekly cycle. The name of the error log file records the day of week and clearly identifies the file, for example, AIDSMon.log.

The type of errors logged by error log are determined by the logging level retrieved from its configuration data. Levels of error logging are:

- ERROR_LEVEL_NONE. No error logging takes place.
- ERROR_LEVEL_INFORMATION. Logs fatal, major, minor and trace information.
- ERROR_LEVEL_MINOR. Logs fatal, major and minor errors.
- ERROR_LEVEL_MAJOR. Logs fatal and major errors.
- ERROR_LEVEL_FATAL. Logs fatal errors only.

Each file records the selected logging type as well as the date, time and description of every error that occurs.
CHAPTER 3

Installation

In This Chapter

Install Application ......................................................... 18
Install Application

For full instructions on how to install this application, refer to the Contact Center Express Installation Guide (Contact Center Express Installation Guide.pdf).

In addition to the standard installation steps, this document explains:

- how you can use command line parameters during installation to dictate where configuration information is retrieved.
- how you can change the configuration source from the local .ini file to the Configuration Server if an application has already been installed.
- how to automatically (silently) install an application following a set of pre-defined selection options.

The Contact Center Express Installation Guide is on the Contact Center Express CD (Overview and Miscellaneous folder) or can be downloaded from the Avaya Contact Center Express website (http://www.AvayaContactCenterExpress.com).
CHAPTER 4

Configuration

This chapter shows you how to register, configure and start the Interaction Data Server.

In This Chapter

Create/Update Database .................................................. 20
Configurable Parameters .................................................. 23
Configure Interaction Data Server via Ini File ....................... 27
Start Interaction Data Server ............................................ 28
Create/Update Database

Contact Center Express provides a set of database script files (.sql extension) and batch files (.bat extension) for the creation and maintenance of all databases used by its applications:

- Interaction Data Server
- Contact Database
- Configuration Server
- Preview Contact Media Store
- Email Media Store
- Simple Messaging Media Store

You will find these scripts in a folder named SQL Script under the individual application folder structure. The folders reside on the distribution CD and are also installed when you install these applications.

Run the scripts to either create or maintain the database by double clicking the Run AS Maintain Database.bat batch file. This records information about your current folder and then runs a batch file (AS Maintain Database.bat) which in turn runs a sql script (AS Maintain Database.sql) - both of which reside in the Utilities folder. This suite of batch and script files reads a control file (AS Script Master.txt) that directs, based on the current state of the database, the specific script files that will be run for the database.

With each release of Contact Center Express you must run Run AS Maintain Database.bat for the application databases you are already using to update the database to the current version. You must also run this batch file for databases you would like to install for the first time (this will create the database).
After you run Run AS Maintain Database.bat, you can view the message log (ASDatabaseLog_YYYYMMDD_HHmmSS.txt) which summarizes the processes that have taken place. Note that YYYYMMDD and HHmmSS will be replaced by the date and time you ran the job. This log file will appear in the folder from which you ran the batch file; or, if you ran it from the CD, in the Temp folder of the Windows directory of your database server. The message shown by Run AS Maintain Database.bat shows the exact location of the log file.

Please note that the log file should end with the message “**Job SUCCEEDED to completely maintain ....... database ....” If it ends with “**Job FAILED to completely maintain ....”, read the detail of the log file, make necessary adjustments and run Run AS Maintain Database.bat again. Running it additional times will not harm the database - it will do nothing if there is nothing to do.

The suite of scripts creates a table tblASSystem that holds the current database version and logs all attempts (successful or otherwise) to modify the database with the scripts.
Note: Prior to Contact Center Express Release 2.0, it was necessary to run individual scripts from a SQL Server database administrator tool for each database update. Now you need only run the one batch file, Run AS Maintain Database.bat.
**Configurable Parameters**

On startup, the Interaction Data Server extracts all the information it needs to run via the Configuration Client. This configuration information can be sourced from a networked Configuration Server or the AIDServer.ini file.

The configuration parameters are grouped in sections:

**[Settings]**

ServerID. A unique number that identifies the Interaction Data Server.

ServerName: A user-specified, user-friendly name for the Interaction Data Server. **Note:** Do not include spaces.

ServerPort. The IP port number the Interaction Data Server uses to accept connections from clients and other servers. The default is 29090.

ErrorLogLevel. The value that determines what level of error detail will be saved in the error log. 0=None, 1=Information, 2=Minor, 4=Major, 8=Fatal.

StatDataCacheInterval. The regularity, in seconds, in which the server internally caches statistical data. **Note:** The suggested value is 1 or 2. Do not exceed 5.

**[ASMClientToBeLoaded]**

Reserved for future use.

**[Switchx]**

SwitchID. The ID number of the switch the Interaction Data Server is connected to. The number should be unique. The switch ID is particularly important in complex configurations, where there is a network of Interaction Data Servers receiving information from multiple Definity/MultiVantage/Avaya CM servers.

SwitchName. The telephony link (T-Link) name of the Avaya CT Server (Telephony Server) this server or application will connect to for information.

LoginName. A valid user name on the Avaya CT Server (as entered in the Avaya CT security database).

Password. The password associated with above user name. By default, the Contact Center Express application will encrypt this data. For more information, see the *Contact Center Express Installation Guide* (Configuration Commands).

DatabaseServer. The name of the SQL server where events and statistical data will be stored to.

DatabaseName. The database name in the SQL server. The database name is: 'ActiveInteractionData' and it is automatically created when the database script is run.
**DatabaseUserName.** The name of the user who will access the database. The database user name is 'ActiveInteractionData' and it is automatically created when the database script is run.

**DatabasePassword.** The password associated with the above user name. By default, the Contact Center Express application will encrypt this data. For more information, see the *Contact Center Express Installation Guide* (Configuration Commands).

**LogEvent.** The value that determines if event logging is activated or not. 0=Off, 1=On.

**LogCallSummary.** The value that determines if call summary logging is activated or not. 0=Off, 1=On. Note: If event logging (**LogEvent**) is activated, call summary logging must also be activated.

**LogStatisticsData.** The value that determines if statistical data logging is activated or not. 0=Off, 1=On. Note: The setting for statistical data logging is not dependent on event logging (**LogEvent**).

**LogUserData.** The value that determines if user data logging is activated or not. 0=Off, 1=On. Note: The setting for statistical data logging is not dependent on event logging (**LogEvent**).

**ServiceLevelTime.** The amount of time, in seconds, logged-in agents are expected to take to answer calls. **ServiceLevelTime** is a switch-wide service level setting for VDNs or split/skills that do not have their own specific service levels. For example: ServiceLevelTime=15

**VDN.** The VDNs to be monitored by the Interaction Data Server. **Note:** Separate VDNs with commas. Do not include spaces. Use a hyphen to specify a range. For example: VDN=4481-4485,4542,4819,4830

To assign a specific service level to a VDN, separate the VDN number and the service level by a full colon. For example: **VDN=4481-4485,4542:20,4819:30,4830**

In this example, VDNs 4481-4485 and 4830 use the service level value set for the switch (see **ServiceLevelTime**). VDN 4542 has a service level of 20 seconds and 4819 has a service level of 30 seconds.

**Station.** The station extensions to be monitored by the Interaction Data Server. **Note:** Separate extensions with commas. Do not include spaces. Use a hyphen to specify a range. For example: **Station=4401-4424,4637**

**ACDSplit.** The splits/skills to be monitored by the Interaction Data Server. **Note:** Separate splits/skills with commas. Do not include spaces. Use a hyphen to specify a range. To assign a specific service level to a VDN, separate the VDN number and the service level by a full colon.

For example: **ACDSplit=8001,8002,8003:25,8004:10,8005**

In this example, splits/skills 8001, 8002 and 8005 use the service level value set for the switch (see **ServiceLevelTime**). Split/skill 8003 has a service level of 25 seconds and 8004 has a service level of 10 seconds.
**TrunkGroup.** The trunk groups to be monitored by the Interaction Data Server.  
**Note:** Separate groups with commas. Do not include spaces. Use a hyphen to specify a range.

**ShiftStartTimes.** The start times for shifts throughout the day. (Note: A shift can be any value between 1 and 24 hours.) At the end of each shift, statistical data will be written to the database and then reset for the next shift calculation. The times will be used in the order they are entered. If shift times are missing, a default of one shift will be assumed that covers a single day (i.e. that starts at 00:00 and ends at 23:59).  
**Note:** Separate start times with commas.

For example: **ShiftStartTimes** = 08:00, 14:00, 22:30, 03:00

This example will split the day into four shifts. Shift 1 will start at 08:00 and run until the start of shift 2 at 14:00. Shift 2 will run until the start of shift 3 at 22:30. Shift 3 will run until the start of shift 4 at 03:00 the next day. Shift 4 will run until shift 1 starts again at 08:00. Every day will have the same shift times.

**StatisticInterval.** The amount of time, in minutes, between statistic calculations. The interval can be any value between 1 and 60 minutes. Values outside this range will not be accepted. At the end of each interval, statistical data will be written to the database and then reset for the next interval calculation.

Where possible, select an interval that is a divisor of 60 minutes. Where a value is chosen that is not a divisor of 60 minutes, the result will be a smaller final period in the hour. For example, if 13 minutes is chosen, this will result in four intervals of 13 minutes and one final interval of 8 minutes.

If the value is missing or invalid, the interval is set to 5 minutes.

**QueryAgentInterval.** The amount of time, in seconds, between agent state queries.  
**Note:** To turn this feature off, use 0.

**QueryTrunkGroupInterval.** The amount of time, in seconds, between trunk group queries.  
**Note:** To turn this feature off, use 0.

**ForceAgentPolling.** If enabled (set to 1), this parameter forces the application to poll earlier versions of Definity (prior to release 10) for agent state information. If an older version of Definity mistakenly informs the application that it has Agent States functionality (only present with Release 10 and above), this parameter prevents the application from receiving no agent event information at all. (The Agent States feature allows Contact Center Express 2.0 applications to automatically receive agent event information for all agents it is monitoring.)  
**Note:** To turn this feature off, use 0.

**[Other AIDServer x]**

**ServerID.** A unique number that identifies the Interaction Data Server.

**ServerName.** A user-specified, user-friendly name for the Interaction Data Server.  
**Note:** Do not include spaces.

**ServerIP.** The IP address of the Interaction Data Server.

**ServerPort.** The IP port number the Interaction Data Server uses to accept connections from clients and other Interaction Data Servers.
SwitchIDs. The ID numbers of the switches the Interaction Data Server is connected to.

*Note:* If you want the Interaction Data Server to monitor information for more than one switch, you need to copy and paste the [Switchx] section and modify it accordingly. If you want the Interaction Data Server to accept connections from more than one other Interaction Data Server, you need to copy and paste the [Other AIDServer x] section and modify it accordingly.

Never delete a default configuration parameter. If you don't need to enter a value, leave the line blank, for example, VDN=
Configure Interaction Data Server via Ini File

1. Click the [Start] button on the Windows Taskbar and select Programs > Avaya Contact Center Express > Server > Interaction Data Server > Edit AIDServer.ini from the pop-up menu.

2. Add configuration information as necessary. For detailed parameter information, refer to Configurable Parameters (on page 23).
Start Interaction Data Server

1 To start the Interaction Data Server, click the [Start] button on the Windows Taskbar and select Programs > Avaya Contact Center Express > Server > Interaction Data Server > Interaction Data Server Manager from the pop-up menu.

2 If you want the Interaction Data Server to retrieve its configuration data from:

- the .ini file in the default AIDServer.ini file folder, leave the Command line blank.
- an .ini file stored on a shared network or location other than the default install folder, type /f newfilepath\AIDServer.ini /z AIDServer in the Command line text box.
- the Configuration Server based on the local machine name, type /s serverIPaddress /a M=%%M /z Application Name and click the [OK] button.
  (Where M=%%M is the machine name and Application Name is the name given to the application in Configuration Server.)
- the Configuration Server based on the user network login name, type /s serverIPaddress /a U=%%U /z Application Name and click the [OK] button.
  (Where U=%%U is the user network login name and Application Name is the name given to the application in Configuration Server.)

3 To start the server, click the [Start] button.
4. To view error log information, click the [View Error Log] button.

<table>
<thead>
<tr>
<th>Type</th>
<th>Date</th>
<th>Time</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>08/08/02</td>
<td>03:30:01</td>
<td>Statistics cycle of switch AVAYA#AVAYA_ECS_...</td>
</tr>
<tr>
<td></td>
<td>08/08/02</td>
<td>03:23:09</td>
<td>Database connection established in statistics database</td>
</tr>
<tr>
<td></td>
<td>08/08/02</td>
<td>03:23:09</td>
<td>Database connection established in CVCT event</td>
</tr>
<tr>
<td></td>
<td>08/08/02</td>
<td>03:23:05</td>
<td>Statistics data logging thread started</td>
</tr>
<tr>
<td></td>
<td>08/08/02</td>
<td>03:23:05</td>
<td>Trunk group state updating feature of Switch AVAYA#AVAYA_ECS_...</td>
</tr>
<tr>
<td></td>
<td>08/08/02</td>
<td>03:23:05</td>
<td>Agent state updating feature of Switch AVAYA#AVAYA_ECS_...</td>
</tr>
<tr>
<td></td>
<td>08/08/02</td>
<td>03:23:05</td>
<td>Statistics data caching thread started</td>
</tr>
<tr>
<td></td>
<td>08/08/02</td>
<td>03:23:05</td>
<td>CVCT event logging thread started</td>
</tr>
<tr>
<td></td>
<td>08/08/02</td>
<td>03:23:03</td>
<td>Active Interaction Data Server started</td>
</tr>
<tr>
<td></td>
<td>08/08/02</td>
<td>03:23:03</td>
<td>Statistics thread started</td>
</tr>
<tr>
<td></td>
<td>08/08/02</td>
<td>03:23:03</td>
<td>Communication manager thread started</td>
</tr>
</tbody>
</table>
CHAPTER 5

Database

In This Chapter

Overview .......................................................... 31
Configuration Issues ......................................... 33
Database Diagram .............................................. 35
Tables .............................................................. 36
Stored Procedures ............................................ 42
Scripts ............................................................. 45
Overview

Interaction Data Server does not need a database for its own correct operation and by default will not use it.

Typically, you will want to turn on database logging only if you have the capability to read and analyse the database. Examples of tools that may help in this are Microsoft SQL Server Enterprise Manager, Microsoft Access, Microsoft Excel or Crystal Reports.

The Interaction Data Server database contains the following broad categories of data:

- System
- Event
- Call Summary
- Statistics
- User

The data the Interaction Data Server writes to the database depends on the configuration parameters used to start the server. Although some configuration parameters are mentioned in this chapter, please refer to the Contact Center Express Installation Guide for information on how to install the database, and the Interaction Data Server User Guide for a fuller explanation of configuration parameters.

System Data

System files hold data that the Interaction Data Server uses to manage the database - such as the switches that generate the data, look-up tables and database error logs. It is largely static data and does not generally need maintenance.

Event Data

Event data is the largest by volume. If the LogEvent configuration parameter is enabled, each event generated by the Telephony Server is passed to the Interaction Data Server, which in turn stores it in a series of tables.

The design objective for event data was to provide a complete cradle to grave record of a call. This data can provide an accurate and entire picture of a switch’s telephony transactions throughout any historical period.

The database stores event data in a series of building block tables. A Definity/MultiVantage/Avaya CM server event always generates multiple database records. This approach (rather than storing one large discrete record for each event) was taken to:

- Better normalize the database
- Avoid having similar (but difficult to relate) data elements in different records
- Easily associate similar data types from different events
- Enable more flexible querying of stored data

However, to avoid multiple round trips to SQL Server, the Interaction Data Server calls only one stored procedure to store one Telephony Server event.

---

**Database**

---

Call Summary Data

Enable **LogCallSummary** to summarise a complete call (and its transferred or conferenced parties) in one record. This level of recording can often provide adequate data, without the overhead of the detail contained in event data.

---

**Note:** Call summary data is automatically stored if you choose to store event data. This allows you to retain a historical archive in summary format if you elect to retain event data for only a limited period.

---

Statistics Data

Enable **LogStatisticsData** to collect data statistics that give an overview of switch usage and contact center performance.

---

User Data

A table that allows custom user software to store application-dependent data that relates to telephony events - for example, client-specific data collected via screen pops that needs to follow and perhaps even outlive a call. The **LogUserData** configuration parameter must be enabled to store this data.
Configuration Issues

Definity/MultiVantage/Avaya CM server and Telephony Server

The Interaction Data Server database depends on the Universal Call ID (UCID) to relate different tables in the database.

To have the database work correctly, you must configure each Definity/MultiVantage/Avaya CM server to have 'Create UCID' turned on, and also add a valid UCID network node ID.

Additionally, you need to configure the switch to pass the UCID to the Telephony Server ('Send UCID to ASAI?' is yes).

For more information, refer to the Definity production documentation (see "Related Documents" on page 9) (Installation/Administration).

Interaction Data Server

Before the Interaction Data Server can perform any database operations, you must follow the instructions in the Interaction Data Server User Guide to create the database.

Besides the Interaction Data Server configuration parameters mentioned above to log the individual data types, you also need to provide values for SwitchId, DatabaseServer, DatabaseName, DatabaseUserName and DatabasePassword for each switch the Interaction Data Server handles. See the Interaction Data Server User Guide for detail.

The SwitchId is a key parameter. A corresponding record, with the same SwitchId value, must exist in the tblSwitch table (see below).

A single Interaction Data Server handling multiple switches may, depending on its configuration, store records for its switches in the same or in different databases. Additionally, one database may be shared by multiple Interaction Data Server's. The key issue is that each switch has its own unique SwitchId over the organisation’s entire telephony site or sites.

Note: If all the logging data configuration parameters (LogEvent, LogCallSummary, LogStatisticsData, LogUserData) are switched off, the Interaction Data Server does not connect to a database. In this case, you do not need to provide the SwitchId, DatabaseServer, DatabaseName, DatabaseUserName and DatabasePassword set of parameters.

Key Configuration Points

Before the Interaction Data Server will store data in its database you must:

- Configure your Definity/MultiVantage/Avaya CM server to add UCID to its event records and pass on the UCID to the Telephony Server.
- Create the database using the AIDCreateDB.sql script.
- Tell the Interaction Data Server (via configuration parameters) which database you want it to use for each Definity/MultiVantage/Avaya CM server.
- Configure Interaction Data Server to store those categories of data that you want.
Tables

The following characteristics apply to all tables:

- The size and type of field is governed by the maximum allowable length and type of data that the Avaya Telephony Server generates.
- Double-byte (unicode) data is catered for, except where MultiVantage/Telephony Server design prohibits it (e.g., DeviceId).
- All records have a unique primary key, based primarily on identities that are automatically generated by SQL Server.

System Tables

tblSwitch

tblSwitch contains an ID number and name for each switch that may write data to the database.

The database create script (AIDCreateDB.sql) populates this table with two records – one with an ID of 0 for ‘Unknown Switch’ and another default switch record with an ID of 1 and name of ‘First switch’. You may change this switch name to something more meaningful to your organisation. Add additional switches as required to this table. However the switch ID’s used must coincide with those read in as configuration parameters.

tblEventType

Defines the EventTypeId for the different Telephony Server events, along with a true / false (bit) that indicates which event records are stored for the particular event. Each core tblEvent record (see below) includes one of these valid EventTypeId’s.

The database create script (AIDCreateDB.sql) populates this table with 19 records – one for each of the 18 Telephony Server events recorded, plus one for the additional Interaction Data Server user’s LogData event.

tblCallFunction

Defines the different types of call record that the database stores. Each tblCall record must use a valid CallFunctionId from this table.

The database create script (AIDCreateDB.sql) populates this table with four records - one for each of the valid call functions:

<table>
<thead>
<tr>
<th>CallFunctionId</th>
<th>CallFunction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Connection</td>
</tr>
<tr>
<td>2</td>
<td>PrimaryOldCall</td>
</tr>
<tr>
<td>3</td>
<td>SecondaryOldCall</td>
</tr>
<tr>
<td>4</td>
<td>Party</td>
</tr>
</tbody>
</table>
**tblDeviceFunction**

Defines the different types of device record that the database stores. Each tblDevice record must use a valid DeviceFunctionId from this table.

The database create script (AIDCreateDB.sql) populates this table with the following valid device functions:

<table>
<thead>
<tr>
<th>DeviceFunctionId</th>
<th>DeviceFunction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ConfController</td>
</tr>
<tr>
<td>2</td>
<td>AddedParty</td>
</tr>
<tr>
<td>3</td>
<td>Static</td>
</tr>
<tr>
<td>4</td>
<td>TransferringDevice</td>
</tr>
<tr>
<td>5</td>
<td>TransferredDevice</td>
</tr>
<tr>
<td>6</td>
<td>Alerting</td>
</tr>
<tr>
<td>7</td>
<td>Answering</td>
</tr>
<tr>
<td>8</td>
<td>Calling</td>
</tr>
<tr>
<td>9</td>
<td>Called</td>
</tr>
<tr>
<td>10</td>
<td>Releasing</td>
</tr>
<tr>
<td>11</td>
<td>Diverting</td>
</tr>
<tr>
<td>12</td>
<td>Failing</td>
</tr>
<tr>
<td>13</td>
<td>Holding</td>
</tr>
<tr>
<td>14</td>
<td>Agent</td>
</tr>
<tr>
<td>15</td>
<td>TrunkUsed</td>
</tr>
<tr>
<td>16</td>
<td>Queue</td>
</tr>
<tr>
<td>17</td>
<td>Retrieving</td>
</tr>
<tr>
<td>18</td>
<td>LastRedirection</td>
</tr>
<tr>
<td>19</td>
<td>LocalConnectionInfo</td>
</tr>
<tr>
<td>20</td>
<td>Distributing</td>
</tr>
<tr>
<td>21</td>
<td>OriginalCallInfoCalling</td>
</tr>
<tr>
<td>22</td>
<td>OriginalCallInfoCalled</td>
</tr>
</tbody>
</table>

**tblErrorLog**

If the Interaction Data Server has any problems in storing records, it writes error information to this log table. Data logged includes:

- SQL Server’s error number
- The type of database operation that caused the error condition
- The parameters that were passed to the stored procedure in which the error occurred
- The EventId of the parent tblEvent record (if applicable to the database operation)
## Event Tables

The following table shows how event tables are stored for each Telephony Server event that the Interaction Data Server processes.

<table>
<thead>
<tr>
<th>Telephony Server event</th>
<th>Event TypeId</th>
<th>Number of records stored in tables</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Event</td>
<td>Call</td>
</tr>
<tr>
<td>Delivered</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Established</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Conferenced</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>plus 1</td>
<td>1</td>
</tr>
<tr>
<td>Transferred</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>plus 1</td>
<td>1</td>
</tr>
<tr>
<td>CallCleared</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>ChargeAdvice</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>ConnectionCleared</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>Diverted</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>EnteredDigits</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>Failed</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>Held</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>LoggedOff</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>LoggedOn</td>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>NetworkReached</td>
<td>14</td>
<td>1</td>
</tr>
<tr>
<td>Originated</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>Queued</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>Retrieved</td>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>Telephony Server event</td>
<td>Event Typeld</td>
<td>Number of records stored in tables</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>ServiceInitiated</td>
<td>18</td>
<td>1</td>
</tr>
<tr>
<td>LogData</td>
<td>19</td>
<td>1</td>
</tr>
</tbody>
</table>

**tblEvent.** This is the core stub that is stored for all events. It contains fields that occur in all Telephony Server events such as SwitchId, EventTypeld, UCID, EventFiredDateTime, and MonitorDeviceId. Its primary key, EventId, becomes the foreign key for all other event category records.

**tblCall.** One tblCall record is stored for each call type - and there may be multiples within a single Telephony Server event. The CallFunctionId field corresponds to a value found in tblCallFunction.

**tblDevice.** One tblDevice record is stored for each device used - and there are most often multiple tblDevice records for a single Telephony Server event. The DeviceFunctionId field corresponds to a value found in tblDeviceFunction.

**tblLookAheadInfo.** A little-used record which is stored only for Delivered and Established events.

**tblOriginalCallInfo.** This table retains data of an original call in circumstances which may generate a subsequent call. Used by the Delivered, Established, Conferenced and Transferred events.

**tblUser.** Stores user-to-user information (UUI) in its Info fields and collected digits in its EnteredCode fields. tblUser occurs only for Delivered, Established, ConnectionCleared and Originated events.

**tblMiscellaneous.** This record contains miscellaneous fields that simply don’t fit into any other more logical record categories. It is stored for over half of all Telephony Server events.
Call Summary Table

**tblCallSummary**

As a call progresses, the Interaction Data Server accumulates pertinent data (based on the call’s unique UCID) from many events and writes a tblCallSummary record at the end of the call.

Detailed event data for the call can be found in the event tables (keyed on the same UCID) if event logging is also on.

The table stores significant milestone times (call start, answer and end times), numbers called to and from, devices used along with agents, stations and call types for up to five transfers or conferences.

The CallDirection field has a value I for Inbound or O for Outbound. CalInType has a value C for Conferenced or T for Transferred.

The EnteredCodeData stores collected digits, while InfoValue records the user-to-user information (UUI). Both these fields store values as at the end of the call - that is after modification, if any, by transferred or conferenced calls.

Statistics Table

The statistics tables (tblAgentStats, tblSplitSkillStats, tblStationStats, tblTrunkGroupStats and tblVDNStats) operate similarly.

They count pertinent statistics over the interval (in minutes) given by the configuration parameter `StatisticInterval`. If the value of `StatisticInterval` is missing or invalid, the interval is set to one hour.

All records carry a ShiftNumber field. A value of zero means the statistic record was accumulated over the above `StatisticInterval`. A non zero value (eg 2) indicates the data applies to Shift 2, as defined in the configuration parameter `ShiftStartTimes`.

The statistics tables carry information to identify the object for which statistics are gathered, and the beginning and end times of the interval.

User Table

**tblClientLog**

This record exists entirely for the benefit of users. Through the Interaction Data Client, a user application may write and retrieve up to 8064 bytes of data in one tblClientLog record. Because tblClientLog records have a parent tblEvent record, the call to which the data belongs (through the UCID) and its sequence (through EventFiredDateTime) within the call are preserved.
The ClientApplication field differentiates between applications when more than one exists on a site. Within applications, data is logged in key/value pairs. For a given ClientKey, you may store up to 64 single byte characters in ClientValue. Use ClientLargeValue for more demanding applications - you may store up to 8000 bytes of binary data.
Stored Procedures

Stored Procedures used by Interaction Data Server

Stored procedures named procInsertTableName are the foundation procedures for writing records to the database. There is one of these insert type stored procedures for each table. They write to tblErrorLog if there is an error on the record insert, and return the error to the calling routine. A zero return indicates a successful insert. Typically, other stored procedures use procInsertTableName procedures.

procStoreEventNameEvent stored procedures use procInsertTableName procedures to insert all records required to completely represent the event - as shown in the table above. They also write to tblErrorLog if there is an error in their stored procedure calls, and return the error to the calling routine. A zero return indicates a successful event store. Typically, the Interaction Data Server uses procStoreEventNameEvent procedures directly.

procGetClientLogEvent

Input parameters:

@ UCID
@ ClientApplication
@ ClientKey

This procedure reads back user data (the ClientValue, the length of ClientLargeValue and ClientLargeValue) requested by customer application programs through the Interaction Data Client; but users may also want to execute it directly.

If @UCID is not passed, then data for all UCID’s is returned; if @ClientApplication is not passed, then data for all ClientApplications is returned; and if @ClientKey is not passed, then data for all ClientKeys is returned. Combinations of specific values for the parameters with no values will return varying amounts of data - the most specific request yielding the ClientValue, the length of ClientLargeValue and ClientLargeValue for the given ClientApplication, ClientKey and UCID input parameters.

Utility Stored Procedures

Users may directly execute stored procedures for deleting various record types from the database.

The following stored procedures all delete categories of records. All can delete records older than a given number of minutes (@OlderThanMinutes) or between specified dates and times.

Probably it is easier to routinely use @OlderThanMinutes and as a guideline for setting this parameter:
There are presently no procedures to delete tblCallSummary records or tblErrorLog records.

**procDeleteEventDetail**

Input parameters:

- `@OlderThanMinutes`
- `@EarlyDateTime`
- `@LateDateTime`

`procDeleteEventDetail` deletes all tblEvent and child event data that is older than `@OlderThanMinutes` or between `@EarlyDateTime` and `@LateDateTime`. If `@OlderThanMinutes` is passed, then values provided for `@EarlyDateTime` and `@LateDateTime` are ignored. If `@EarlyDateTime` is not passed, it defaults to the beginning of the 20th century, effectively deleting all event records up to `@LateDateTime`. This delete routine ensures that all event records for a particular call (based on UCID) are deleted, leaving no orphaned event records.

**procDeleteClientLog**

Input parameters:

- `@ClientApplication`
- `@ClientKey`
- `@OlderThanMinutes`
- `@EarlyDateTime`
- `@LateDateTime`

`procDeleteClientLog` deletes all User (tblClientLog) data for the given `@ClientApplication` and `@ClientKey` combination that is older than `@OlderThanMinutes` or between `@EarlyDateTime` and `@LateDateTime`. If `@ClientKey` is not passed, then all records for the nominated `@ClientApplication` are deleted; otherwise only those records with the passed `@ClientApplication` and `@ClientKey` are deleted. If `@OlderThanMinutes` is passed, then values provided for `@EarlyDateTime` and `@LateDateTime` are ignored. If `@EarlyDateTime` is not passed, it defaults to beginning of the 20th century, effectively deleting all tblClientLog records up to `@LateDateTime`.

**procDeleteStats**

Input parameters:

- `@OlderThanMinutes`
proDeleteStats deletes all statistics data that is older than @OlderThanMinutes or between @EarlyDateTime and @LateDateTime.

If @OlderThanMinutes is passed, then values provided for @EarlyDateTime and @LateDateTime are ignored. If @EarlyDateTime is not passed, it defaults to the beginning of the 20th century, effectively deleting all statistics records up to @LateDateTime.

**procGetClientLogEvent**

See discussion above under Stored Procedures used by Interaction Data Server.

**procGetCallSummary**

Input parameters:

- @CallFromNumber
- @EnteredCodeData
- @StartDateTime
- @EndDateTime

Returns all data from tblCallSummary with the given CallFromNumber (defaults to any), the given EnterCodeData (defaults to any) and between StartDateTime (defaults to 1970/01/01) and @EndDateTime (defaults to present date and time).
Scripts

AIDCreateDB.sql

Drops an existing ActiveInteractionData database and creates a new one, including all tables, relationships and stored procedures needed by the Interaction Data Server to correctly store data in the database.

This script also populates the system tables with data needed by the Interaction Data Server.
Index

C
Call Event Logging • 15
Call Summary Data • 32
Call Summary Table • 40
Configurable Parameters • 23
Configuration • 15, 19
Configuration Issues • 33
Configure Interaction Data Server via Ini File • 27
Create/Update Database • 20

D
Database • 30
Database Diagram • 35
Definity/MultiVantage/Avaya CM server and Telephony Server • 33
Document Conventions • 8

E
Error Logging • 16
Event Data • 31
Event Tables • 38

I
Install Application • 18
Installation • 17
Interaction Data Server • 33
Introduction • 11

K
Key Configuration Points • 33
Knowledge Base • 10

O
Overview • 31

P
Preface • 7
Product Name Change • 10

R
Related Documents • 9

S
Scripts • 45
Software License Agreement • 3
Start Interaction Data Server • 28
Statistics Data • 32
Statistics Table • 40
Stored Procedures • 42
Stored Procedures used by Interaction Data Server • 42
System Data • 31
System Tables • 36

T
Tables • 36

U
User Data • 32
User Table • 40
Utility Stored Procedures • 42

W
What is the Interaction Data Server? • 12