

Dentrix® API Exceptions Reference

Version 2.1

727 E. Utah Valley Drive
American Fork, Utah 84043
800.734.5561
www.dentrix.com/ddp
ddp@dentrix.com



Publication Date

October 2014

Copyright

1987-2014 Henry Schein, Inc. All rights reserved.

Software License Notice

Your license agreement with Dentrix Dental Systems, Inc., which is included with the product, specifies the prohibited uses of the product. Any unauthorized duplication or use of Dentrix in whole or in part, or in any storage and retrieval system is forbidden.

Licenses and Trademarks

Dentrix, Henry Schein, and the 'S' logo are registered trademarks of Henry Schein, Inc.; Microsoft, Windows, Windows XP Professional, Windows, Vista, Excel, and Word are trademarks of Microsoft Corporation; c-treeACE and FairCom are registered trademarks of the FairCom Corporation; all ADA CDT codes are protected by U.S. and International copyright laws. All rights reserved by the American Dental Association.

Table of Contents

Publication Date	1
Copyright	1
Software License Notice	1
Licenses and Trademarks	1
Table of Contents	2
Exception Handling	3
Null Checking	3
Exception Codes and Descriptions	4

Exception Handling

The Dentrix API uses exception handling as its primary method of error handling. The section for each Stored Procedure function detailed in this document lists exceptions, if any, that may be thrown at runtime. These exceptions are identified as “Throws”. The table on page 31 provides the associated error code and description for each exception that may be thrown. Please note that error codes in the 50000 – 59999 range have been reserved for Dentrix API-specific exceptions. Error codes outside of this range correspond to Faircom® c-tree generated errors/exceptions. See page 223 of the [Faircom c-treeACE Reference Guide](#) online for details regarding c-tree generated errors.

Thrown exceptions can be caught using standard “try, catch, and finally (optional)” blocks in your exception handler code. In order to catch and handle exceptions that may be thrown when running the Dentrix API Stored Procedures, you will need to add “try catch” code to catch/trap for the DhSQLException. The DhSQLException.getDiagnostics() function can then be used to retrieve details for the DhSQLException. More information regarding the DhSQLException class can be found on the [FairCom website](#).

The following websites provide examples for implementing the “try, catch and finally” exception handling method in your code.

- **Java:** <http://download.oracle.com/javase/tutorial/essential/exceptions/handling.html>
- **C#:** [http://msdn.microsoft.com/en-us/library/dszsf989\(v=VS.90\).aspx](http://msdn.microsoft.com/en-us/library/dszsf989(v=VS.90).aspx)
- **C++:** [http://msdn.microsoft.com/en-us/library/9xtt5hxz\(v=VS.90\).aspx](http://msdn.microsoft.com/en-us/library/9xtt5hxz(v=VS.90).aspx)
- **.NET:** <http://www.codeproject.com/KB/dotnet/UsingTryCatchFinally.aspx>
- **Visual Basic:** [http://msdn.microsoft.com/en-us/library/fk6t46tz\(v=VS.90\).aspx](http://msdn.microsoft.com/en-us/library/fk6t46tz(v=VS.90).aspx)

Null Checking

It is recommended that you include “null checking” logic in your application’s code to handle database pointers that may be NULL / empty. See the sample code for using the Reader-Safe or Faircom® Reader-Safe' [data retrieving methods](#) in this document as well as in the ‘Dentrix SQL Browser Source’ folder in the DentrixSDK_2.1.zip file accessible [online](#).

Exception Codes and Descriptions

Error Code(s)	Description
51001, 52005	Patient not found
51002	Patient status archived. Requested function cannot be performed.
51003	Referring patient status archived. Requested function cannot be performed.
51101	Provider not found
51102	Provider status inactive
51103	Secondary provider not allowed
51104	Operatory not found
51201	Note not found
51301	Practice Prescription definition not found
51302	Patient prescription not found
51401	Referral source not found. The Referral ID passed in for the procedure does not match an existing Referral Source in the database OR the Referral ID passed in in for the procedure does not match the Referral Type (Referred By / Referred To) required for the procedure.
51402	Referral specialty definition ID not found
51403	Referral not found. The Referral Acts ID passed in for the procedure does not match an existing Patient Referral in the database.
51404	Referral last name required
51405	Invalid non-person flag
51501	Invalid contact type. Valid contact types for this stored procedure are 3, 4, 5 or 20.
51502	Office Journal entry not found
51601	Appointment not found
51602	Appointment status definition not found
51603	Appointment note not found. The Appointment ID passed in for the procedure does not match an existing Appointment Note in the database.
51604	Appointment Date Note not found

51701	Database locked and cannot be updated
51801	Claim not found
52201	Outstanding insurance claims. The requested function cannot be performed as the patient currently has outstanding claims.
52202	Insured record cannot be found
52203	Invalid claim status. The status specified for the claim is not valid according to the available claim statuses.
52204	Cannot apply sent status unless the claim is SENT
52205	Invalid Insurance Table Name definition id
52300	Appointment length exceeds 24 time unit maximum
52301	Resource does not match class
52302	Resource does not match active/inactive state
52303	Operatory is already in use at requested time
52304	Provider is over booked at requested time. Another scheduling is not permitted.
52305	Appointment ID must be a valid scheduled appointment.
52306	Invalid Appointment Flag
52307	Invalid Initial Reason
52308	Invalid Appointment Type
52309	Invalid Appointment Follow-up
52310	Procedure not associated with Appointment Patient.
52311	Too many patient identifiers provided
52312	Appointment Patient cannot be changed
52313	Insufficient Patient identifiers provided
52314	Insufficient reason info provided
52315	Invalid Appointment Reason Type Code
52316	Invalid Appointment Reason Code
52317	Appointment Patient Name Missing
52318	Appointment Phone Number Invalid

52319	Appointment Time Hour Invalid
52320	Appointment Time Minute Invalid
52321	Auto UpdateCC Invalid
52322	Invalid Appointment Pattern
52323	Invalid Referral Type
52324	Invalid Amount Format
52325	Invalid Amount
52326	Invalid Appointment Broken Class
52327	Invalid Pinboard Flag
52328	Existing Procedures cannot be attached more than once to an appointment
52329	Appointment Patient Name Missing
52330	Invalid Appointment Set Amount
52331	Error occurred attempting to parse Appointment XML
52332	Operatory cannot be cleared when Appointment Date is NULL
52400	LabCase not found
52401	LabCase link type not valid for this operation
52402	Attaching Lab Patient must match the Appointment Patient
52450	Invalid Payment Type definition specified
52451	Invalid ChartStatus definition specified
52452	Invalid Procedure Alternate Code specified
52453	Invalid Adjustment Type specified
52454	Invalid Procedure Code ID specified
52455	Invalid Tax Adj/Discount Code specified
52456	Invalid Diagnostic Code Category specified
52457	Invalid Procedure Code Type
52600	Invalid Provider Specialty ID
52900	Unable to parse DateTime string
52901	Invalid Definition

52902	Invalid Fee Schedule Name ID
52903	Invalid Address ID
52904	Malicious String
52905	Invalid State
52906	Invalid Email Address Length
52907	Invalid Email Address Pattern
52908	Invalid Zip Code
52909	Could not create Address record
52910	Could not create Note record