

Provider Appeals

Oracle System Documentation

For internal use by the Department of Medical Assistance Services (DMAS)

July 2000

Table of Contents

1. MANUAL INTRODUCTION	1
1.1 INTENDED USE	1
1.2 ASSUMPTIONS ABOUT THE READER	1
1.3 PROVIDER APPEALS OVERVIEW	1
1.3.1 <i>Application Functionality</i>	1
1.3.2 <i>System Design</i>	2
1.3.3 <i>Relationship of Screen Areas to Blocks and Base Tables</i>	2
2. DIRECTORIES	2
3. CUSTOM MENU	3
4. PROGRAM UNITS.....	3
4.1 PAAPPL.FMB PROGRAM UNITS	3
4.1.1 <i>ALL_REQUIRED_ITEMS_PRESENT</i>	3
4.1.1 <i>CALCULATE_DUE_DATE</i>	3
4.1.2 <i>CHECK_PACKAGE_FAILURE</i>	5
4.1.3 <i>CLEAR_ALL_MASTER_DETAILS</i>	5
4.1.4 <i>DEL_REC</i>	5
4.1.5 <i>FORMAT_DEFORMAT</i>	5
4.1.6 <i>HOW_MANY_APPEALS</i>	6
4.1.7 <i>HOW_MANY_REFERENCES</i>	6
4.1.8 <i>POPULATE_PROVIDER_DATA</i>	6
4.1.9 <i>PROVIDER_SPECIAL_QUERY</i>	6
4.1.10 <i>PROV_ALL_CONTACT_NAMES_PRESENT</i>	6
4.1.11 <i>PROV_ALL_PERSON_NAMES_PRESENT</i>	7
4.1.12 <i>PROV_ALL_REQUIRED_PRESENT</i>	7
4.1.13 <i>QUERY_MASTER_DETAILS</i>	7
4.2 ADHOC.FMB PROGRAM UNITS	8
4.2.1 <i>CHECK_PROV_NO</i>	8
4.2.2 <i>CONVERT_TO_DISPLAY</i>	8
4.2.3 <i>SHOW_NUMERIC</i>	8
5. TRIGGERS AND FLOWCHARTS.....	8
5.1 FORM-LEVEL TRIGGERS	8
5.2 BLOCK-LEVEL TRIGGERS	9
5.3 ITEM-LEVEL TRIGGERS.....	10
6. REPORTS	14
7. DATABASE TABLES AND PARENT/CHILD RELATIONSHIPS.....	15
7.1 <i>APPEAL_CROSS_REFERENCE</i>	15
7.2 <i>HEARING_OFFICER</i>	15
7.3 <i>INFORMAL_APPEALS_AGENT</i>	16
7.4 <i>PROVIDER</i>	16
7.5 <i>PROVIDER_APPEAL</i>	17
8. ENTITY RELATIONSHIP DIAGRAM	20

1. Manual Introduction

This manual explains the Oracle system processes associated with the functions of the Provider Appeals system. Appendixes of flowcharts provide visual representations of narrative explanations within the manual.

Read the subsections below to learn whom this manual is intended for and what the Provider Appeals system does.

1.1 Intended Use

This manual is intended for use by Oracle software developers or database administrators who need to make changes or enhancements to the Provider Appeals application but lack experience in working with it.

This manual should be read online. Hyperlinks connect the main document (i.e., this MS Word document) to Visio flowcharts that represent system processes.

1.2 Assumptions About the Reader

This manual's content and approach rest on two major assumptions:

- that the reader is an experienced Oracle software developer or individual with similar skills;
- and that the reader has no familiarity with the Provider Appeals interface or the corresponding Oracle files.

In accordance with the latter assumption, an informational table has been provided (see Table 1.3.3-1 in subsection 1.3.3) to “bridge the gap” between the main interface screen of Provider Appeals and the Oracle Forms components that comprise it. This integrates for the reader the process of becoming familiar with the application interface and the files that make it up.

1.3 Provider Appeals Overview

In comparison to many of the other DMAS Oracle applications, the Provider Appeals system is relatively small.

The following subsections describe the functionality and design of the Provider Appeals application.

1.3.1 Application Functionality

The Provider Appeals system tracks Medicaid Providers' appeals of DMAS decisions. Provider information is stored in a table that is accessible from the Maintenance menu; the main interface displays Appeal information associated with different Providers. Main interface fields pertain to Informal Appeals, Formal Appeals, court proceedings, Provider representatives (such as attorneys, accountants, etc.), CSAs, and related Medicaid Recipients.

The system allows users to cross-reference Appeals and/or Providers that share some piece(s) of information. In this way, multiple Appeals for a single Provider can be managed effectively.

A flexible Reports module allows users to hand-pick report columns and data parameters for every report. Since users customize the content of reports, they also name them. In these ways, the Provider Appeals Reports module is dynamic and highly flexible.

The Provider Appeals application does not automate any functions or processes, per se; it tracks information about Providers' Appeals, allows users to update Appeals and other information, and allows users to decide what information they want to see in the reports they generate.

1.3.2 System Design

The Provider Appeals system uses two Forms, multiple Blocks, numerous Canvas-Views, and a custom menu. It also uses several alerts, many Property Classes, multiple Record Groups, several LOVs, and a handful of Visual Attributes. A Reports module handles reporting functionality. (Note that a self-contained section in this document addresses reports programming separately.)

1.3.3 Relationship of Screen Areas to Blocks and Base Tables

The following table indicates which areas of the application correspond to which Blocks, and the database tables that correspond to those Blocks (where applicable). The purpose of this table is to show you how the user interface relates to the Oracle Forms components that make it up.

For additional information on database tables, see Section 7, "Database Tables and Parent/Child Relationships".

Area of Application	Block	Base Table
Main interface	PROVIDER_APPEAL	PROVIDER_APPEAL
Provider information form (Provider option under Maintenance menu)	PROVIDER	PROVIDER
Appeal cross-reference form (Cross-ref button on main interface)	APPEAL_CROSS_REFERENCE	APPEAL_CROSS_REFERENCE
Informal Appeals Agent information form (Informal Agent option under Maintenance menu)	INFORMAL_APPEALS_AGENT	INFORMAL_APPEALS_AGENT
Hearing Officer information form (Hearing Officer option under Maintenance menu)	HEARING_OFFICER	HEARING_OFFICER

Table 1.3.3-1. Application Areas, Blocks, and Corresponding Base Tables.

2 NOTE: Blocks that do not have Base tables are not recognized in the table above.

2. Directories

Files for the Provider Appeals system are stored as follows:

S:\DMAS\Testapp\Provappl\Bin – Contains programs and files for testing; can be modified as needed.

S:\DMAS\Source\Provappl – Developer source directory; contains .rdf, .pll, .fmb, .fmx, and .mmmb files.

3. Custom Menu

Provider Appeals uses two custom menu modules called `pamenu.mmb` and `pamenu2.mmb`. The former is the menu that appears when a user is on the main interface screen or on the Formal Reports screen; the latter corresponds to the Informal Appeals screen.

4. Program Units

This section describes the functionality of the procedures under the Program Units nodes of `paappl.fmb` and `adhoc.fmb` and names the Triggers that call each of them.

4.1 Paappl.fmb Program Units

The following Program Units belong to `paappl.fmb`.

4.1.1 ALL_REQUIRED_ITEMS_PRESENT

2 NOTE: This Program Unit does not execute. Instead, the When-Validate-Item Trigger fires on the REPRESENT1_FIRST_NAME Item on the PROVIDER_APPEAL Block.

By checking for a value in the `PROVIDER_ID` Item on the `PROVIDER_APPEAL` Block, this Program Unit ensures that the user does not add an Appeal that does not have an associated Provider. If a Provider has not been specified for the Appeal the user is trying to add, the following message appears: "A provider must be specified for this appeal."

Called By: Pre-Insert Trigger on PROVIDER_APPEAL Block (Block-level).

4.1.1 CALCULATE_DUE_DATE

This Program Unit ensures that programmed due dates do not fall on Saturdays, Sundays, or holidays recognized by DMAS. Table 4.1.2-1 shows the recognized legal holidays and when they occur.

Holiday	When
Lee-Jackson Day	Second Friday in January
Martin Luther King, Jr. Day	Third Monday in January
George Washington Day	Third Monday in February
Labor Day	First Monday in September
Columbus Day	Second Monday in October
Memorial Day	Last Monday in May
Thanksgiving Day	Fourth Thursday in November
Friday after Thanksgiving Day	Day after Thanksgiving
New Year's Day	January 1*
Independence Day	July 4*
Veterans' Day	November 11*
Christmas Day	December 25*

Table 4.1.2-1. DMAS Legal Holidays.

*If any of these dates falls on a Saturday, the holiday is observed the day before, on Friday. If any of these dates falls on a Sunday, the holiday is observed the next day, on Monday.

Table 4.1.2-2 below shows the programmed date fields, corresponding Item names, and date calculation formulas.

Programmed Date Fields	Item Names	Date Calculation Formulas
<i>Informal Appeal Section:</i>		
"Written Submission Due"	WRITTEN_ SUBMISSIONS_ DUE	"Request Received" date + 90 days
"Case Summary Due"	CASE_SUMMARY_ DUE	"Request Received" date + 30 days
"IFFC Due"	IFFC_DUE_DATE	"Request Received" date + 90 days
"Additional Docu Due"	ADDITIONAL_DOCU_ DUE	"IFFC Held Date" date + 30 days
"Informal Disposition Due"	INFORMAL_ DECISION_DUE	"Request Received" date + 180 days
<i>Formal Appeal Section:</i>		
"Request Due"	FORMAL_ REQUEST_DUE	"Informal Disposition Made" + 30 days . NOTE: The "Informal Disposition Made" field is in the Informal Appeal section of the screen.
"Documentary Evidence Due"	DOCUMENTARY_ EVIDENCE_DUE	"Request Received" date + 21 days
"Objections to Docu Due"	OBJECTIONS_TO_D OCU_DUE	"Documentary Evidence Filed" date + 7 days
"HO Rule Objections Due"	HO_RULE_ OBJECTIONS_ DUE	"Objections to Docu Filed" + 7 days
"Hearing Due"	FORMAL_ HEARING_DUE	"Request Received" date + 45 days
"Opening Briefs Due"	OPENING_ BRIEFS_DUE	"Hearing Held" date + 30 days
"Reply Briefs Due (P)"	REPLY_BRIEFS_ DUE	"Opening Briefs Filed (P)" date + 10 days
"Reply Briefs Due (D)"	DEPT_REPLY_ BRIEFS_DUE	"Opening Briefs Filed (D)" date + 10 days
"HO Recommendation Due"	HO_ RECOMMENDATIONS_ DUE	"Request Received" date + 120 days
"Exceptions to HO Due"	EXCEPTIONS_TO_ HO_DUE	"HO Recommendation Received" date + 30 days
"Formal Disposition Due"	DIRECTOR_ DECISION_DUE	"HO Recommendation Received" date + 60 days

Table 4.1.2-2. Programmed Date Fields and Corresponding Date Calculations.

Called By: Pre-Text-Item Trigger on CASE_SUMMARY_DUE Item on PROVIDER_APPEAL Block; Pre-Text-Item Trigger on IFFC_DUE_DATE Item on PROVIDER_APPEAL Block; Pre-Text-Item Trigger on ADDITIONAL_DOCU_DUE Item on PROVIDER_APPEAL Block; Pre-Text-Item Trigger on WRITTEN_SUBMISSIONS_DUE Item on PROVIDER_APPEAL Block; Pre-Text-Item Trigger on INFORMAL_DECISION_DUE Item on PROVIDER_APPEAL Block; Pre-Text-Item Trigger on FORMAL_REQUEST_DUE Item on PROVIDER_APPEAL Block; Pre-Text-Item Trigger on DOCUMENTARY_EVIDENCE_DUE Item on PROVIDER_APPEAL Block; Pre-Text-Item Trigger on OBJECTIONS_TO_DOCU_DUE Item on PROVIDER_APPEAL Block; Pre-Text-Item Trigger on HO_RULE_OBJECTIONS_DUE Item on PROVIDER_APPEAL Block; Pre-Text-Item Trigger on FORMAL_HEARING_DUE Item on PROVIDER_APPEAL Block; Pre-Text-Item Trigger on OPENING_BRIEFS_DUE Item on PROVIDER_APPEAL Block; Pre-Text-Item Trigger on REPLY_BRIEFS_DUE Item on PROVIDER_APPEAL Block; Pre-Text-Item Trigger on HO_RECOMMENDATION_DUE

*Item on PROVIDER_APPEAL Block; Pre-Text-Item Trigger on EXCEPTIONS_TO_HO_DUE
Item on PROVIDER_APPEAL Block; Pre-Text-Item Trigger on DIRECTOR_DECISION_DUE
Item on PROVIDER_APPEAL Block; Pre-Text-Item Trigger on DEPT_REPLY_BRIEFS_DUE
on PROVIDER_APPEAL Block; When-Validate-Item Trigger on INFORMAL_REQUEST_RECEIVED
Item on PROVIDER_APPEAL Block.*

4.1.2 CHECK_PACKAGE_FAILURE

This Program Unit checks to see that the most recently performed action resulted in success. If it did not, the Trigger fails.

*Called By: QUERY_MASTER_DETAILS Program Unit; CLEAR_ALL_MASTER_DETAILS
Program Unit; On-Populate-Details Trigger on PROVIDER_APPEAL Block (Block-level).*

4.1.3 CLEAR_ALL_MASTER_DETAILS

First, this Program Unit gets the name of the detail Block that is part of the relation involving the current master block – but only if the detail Block has changes. If the detail block does not have changes, the procedure looks for a detail Block that does have changes. It then looks for additional relations and changed Blocks.

Next, the Program Unit looks for the coordination-causing event that occurred on the master Block. If it is anything other than CLEAR_RECORD or SYNCHRONIZE_BLOCKS, then the procedure continues to look. If the driving master Block is the trigger Block, and if the Form is changed, then the procedure finds the first changed detail Block. If a changed detail Block is found, the procedure moves the cursor to that Block and commits the changes.

Then, the procedure clears all detail Blocks for the current master without any further asking to commit. In the end it puts the cursor back where it started.

Called By: On-Clear-Details Trigger (Form-level)

4.1.4 DEL_REC

If the current record's status is 'NEW' or 'INSERT', then the procedure only clears the record (since it is not yet in the database). If the current record has already been added to the database, the procedure displays `Delete_Record_Alert` ('YOU ARE DELETING AN ENTIRE RECORD. DO YOU WISH TO PROCEED??? Note: 'Yes' will save all pending changes.) If the user says "yes", then the record is deleted. If the delete operation is successful, then the following message appears: "This record has been successfully deleted". If the delete operation fails, then the following message appears: "The attempt to delete this record has failed because of an internal error. Please contact technical support".

Called By: N/A

4.1.5 FORMAT_DEFORMAT

This Program Unit removes format masks from queryable fields so that the user can perform queries on those fields without using formatted data. This Program Unit replaces format masks on queryable fields after a query so that data is presented in the proper format.

*Called By: Post-Query Trigger on PROVIDER_APPEAL Block; Pre-Query Trigger on
PROVIDER Block; Pre-Query Trigger on PROVIDER_APPEAL Block; Post-Query Trigger
on APPEAL_CROSS_REFERENCE Block; Post-Query Trigger on PROVIDER Block; Pre-
Query Trigger on APPEAL_CROSS_REFERENCE Block.*

4.1.6 HOW_MANY_APPEALS

This Program Unit queries the PROVIDER_APPEAL table for the number of PROVIDER_IDS that are the same as the ID currently in the PROVIDER_ID field. (This number represents the total number of Appeals that the current Provider has. The number of Appeals that a Provider has is indicated in the “# Appeals” field on the main interface screen.)

Called By: Post-Query Trigger on PROVIDER_APPEAL Block; Post-Insert Trigger on PROVIDER_APPEAL Block; Post-Update Trigger on PROVIDER_APPEAL Block.

4.1.7 HOW_MANY_REFERENCES

This Program Unit queries the APPEAL_CROSS_REFERENCE table for the number of APPELLANT_ID1s that are the same as the value currently in the APPELLANT_ID Item of the PROVIDER_APPEAL Block. This number represents the number of cross-references the current Appeal has. (The APPEAL_CROSS_REFERENCE table is a child of the PROVIDER_APPEAL table; the primary key is APPELLANT_ID; the foreign keys are and APPELLANT_ID1 and APPELLANT_ID2.)

Called By: Post-Query Trigger on PROVIDER_APPEAL Block; Post-Insert Trigger on PROVIDER_APPEAL Block; Post-Update Trigger on PROVIDER_APPEAL Block; Post-Insert Trigger on APPEAL_CROSS_REFERENCE Block; Post-Update Trigger on APPEAL_CROSS_REFERENCE Block; Post-Delete Trigger on APPEAL_CROSS_REFERENCE Block.

4.1.8 POPULATE_PROVIDER_DATA

This Program Unit pulls information for the current Provider from the PROVIDER table and uses it to populate the Provider information fields on the PROVIDER_APPEAL Block. (The procedure determines the current Provider by looking at the PROVIDER_ID Item on the PROVIDER_APPEAL Block and looking for the same number in the PROVIDER_ID column of the PROVIDER table.)

Called By: Post-Query Trigger on PROVIDER_APPEAL Block; When-Validate-Item Trigger on PROVIDER_ID Item on PROVIDER_APPEAL Block.

4.1.9 PROVIDER_SPECIAL_QUERY

This Program Unit builds a query “Where” clause based on what a user enters in a Provider Information field at the top of the main interface. Without this Program Unit it would not be possible to query PROVIDER table fields from the main interface screen, which has as its base table the PROVIDER_APPEAL table. (The purpose of querying for a Provider name from the main interface is to bring up the Appeal associated with that Provider.)

Called By: Pre-Query Trigger on PROVIDER_APPEAL Block.

4.1.10 PROV_ALL_CONTACT_NAMES_PRESENT

This Program Unit ensures that users enter required Contact name information on the *Providers of the DMAS Provider Appeal* form, which is accessible by the Provider option under the Maintenance menu.

First, this procedure checks to see if the user has entered a Contact’s middle initial but left out a first or last name. If so, the procedure displays the following message: “You must specify a contact’s first and last name if you have specified a contact middle initial”. Then the Trigger fails.

Next, the procedure looks to see if the user has entered a last name but not a first name for the Contact. If so, the following message displays: "You must specify a contact's first name if you have specified a contact last name". Then the Trigger fails. (Contacts are Provider employees who communicate directly with DMAS by phone, email, fax, etc.)

Called By: Pre-Insert Trigger on PROVIDER Block; Pre-Update Trigger on PROVIDER Block.

4.1.11 PROV_ALL_PERSON_NAMES_PRESENT

This Program Unit ensures that users enter required Provider name information on the *Providers of the DMAS Provider Appeal* form, which is accessible by the Provider option under the Maintenance menu.

First, this procedure checks to see if the user has entered a Provider's middle initial but left out a first or last name. If so, the procedure displays the following message: "You must specify a Provider's first and last name if you have specified a Provider's middle initial". Then the Trigger fails. Next, the procedure looks to see if the user has entered a first name but not a last name for the Provider. If so, the following message displays: "You must specify a Provider's last name if you have specified a Provider first name". Then the Trigger fails. Finally, the procedure checks to see if the user has entered a last name for the Provider but not a first name. If so, the following message displays: "You must specify a Provider's first name if you have specified a Provider's last name". Then the Trigger fails.

Called By: Pre-Insert Trigger on PROVIDER Block; Pre-Update Trigger on PROVIDER Block.

4.1.12 PROV_ALL_REQUIRED_PRESENT

When the user tries to save a new Provider or update an old one, this Program Unit checks to see that there are values in the PROVIDER_NAME and LAST_NAME Items on the PROVIDER Block. If both Items are NULL, the system displays the following message: "You did not provide the required data for the Provider (either Provider name or person's name). You must enter them before trying to save". Then, the Trigger fails.

Called By: Pre-Insert Trigger on PROVIDER Block; Pre-Update Trigger on PROVIDER Block.

4.1.13 QUERY_MASTER_DETAILS

First, this Program Unit checks to see if the DEFERRED_COORDINATION Relation_Property is 'FALSE'. If it is, the procedure executes a GO_BLOCK (detail). If the DEFERRED_COORDINATION Relation_Property is 'TRUE', then the procedure marks the detail Block's COORDINATION_STATUS property as 'NON-COORDINATED'.

Called By: On-Populate-Details Trigger on PROVIDER_APPEAL Block.

4.2 Adhoc.fmb Program Units

Adhoc.fmb is the Form associated with Provider Appeal reports. For information on reports, please, turn to Section 6, "Reports".

The following Program Units belong to adhoc.fmb.

4.2.1 CHECK_PROV_NO

When the user has chosen the Informal Reports option from the Reports menu and has typed a Provider Number in the "Provider No" field, this Program Unit verifies that the user has entered a seven-digit Provider Number. If the user has entered something other than numbers in the "Provider No" field, this Program Unit calls the SHOW_NUMERIC Program Unit, which displays the message, 'Provider Number must be numeric.'

Called By: When-Validate-Item Trigger on PROVIDER_NO1 Item on CRTK1 Block.

4.2.2 CONVERT_TO_DISPLAY

This Program Unit organizes the user's "sort" selections on the Informal Report screen. (Towards the bottom of that screen, the user can choose a primary "sort" field by which items on the report will be sorted. Any secondary sort fields apply only to groupings of information yielded by the primary sort.) Originally there were 8 sort fields; currently there are 4.

Called By: When-Button-Pressed Trigger on RUN_REPT Button on CRTK3 Block.

4.2.3 SHOW_NUMERIC

This Program Unit displays the message, 'Provider Number must be numeric' when the user has entered something other than numbers in the "Provider No" field on the Informal Report screen.

Called By: CHECK_PROV_NO Program Unit. (The When-Validate-Item Trigger on the PROVIDER_NO1 Item on the CRTK1 Block calls this Program Unit.)

5. Triggers and Flowcharts

Most of the system's logic resides in PL/SQL procedures that are executed by Form, Block, and Item-level Triggers. Tables in this section provide links to corresponding flowcharts for each category of Trigger (e.g., Form-, Block-, and Item-level). The flowcharts depict each Trigger's functionality.

Note that "standard" Triggers, such as Key-Up, Key-Down, and Key-Nxtrec, are not represented by flowcharts. [These Triggers have "(not included)" in the "Function of PL/SQL Code" column of the following tables.]

5.1 Form-level Triggers

The Provider Appeals system has two Forms: paappl.fmb and adhoc.fmb. The former includes all parts and functions of the application and the Formal Report screen and functionality. The latter includes only the Informal Appeal report screen and functionality.

Table 5.1-1 below lists all Form-level Triggers for `paappl.fmb`. To view flowcharts depicting Trigger functionality, click the hyperlink in the “Function of PL/SQL Code” column of the table.

Form	Trigger	Function of PL/SQL Code
Paappl.fmb	Key-Nxtblk	(not included)
	Key-Prvblk	(not included)
	On-Clear-Details	Appendix A (Form-level Trigger Flowcharts).vsd
	When-New-Form-Instance	Appendix A (Form-level Trigger Flowcharts).vsd
	When-Mouse-Enter	(not included)
	When-Timer-Expired	(not included)

Table 5.1-1. Paappl.fmb Form-level Triggers.

Table 5.1-2 below names the only Form-level Trigger for `adhoc.fmb`.

Form	Trigger	Function of PL/SQL Code
Adhoc.fmb	When-New-Form-Instance	(not included)

Table 5.1-2. Adhoc.fmb Form-level Trigger.

5.2 Block-level Triggers

Paappl.fmb has numerous Blocks. Table 5.2-1 below lists the Blocks and the corresponding Triggers. To view flowcharts depicting Trigger functionality, click the hyperlink in the “Function of PL/SQL Code” column of the table.

Block	Trigger	Function of PL/SQL Code
PROVIDER_APPEAL	When-New-Record-Instance	Appendix B (Block-level Trigger Flowcharts).vsd
	When-Validate-Record	Appendix B (Block-level Trigger Flowcharts).vsd
	Key-Crerec	Appendix B (Block-level Trigger Flowcharts).vsd
	On-Check-Delete-Master	Appendix B (Block-level Trigger Flowcharts).vsd
	On-Populate-Details	Appendix B (Block-level Trigger Flowcharts).vsd
	Post-Insert	Appendix B (Block-level Trigger Flowcharts).vsd
	Post-Update	Appendix B (Block-level Trigger Flowcharts).vsd
	Pre-Query	Appendix B (Block-level Trigger Flowcharts).vsd
	Pre-Insert	Appendix B (Block-level Trigger Flowcharts).vsd
	Pre-Update	Appendix B (Block-level Trigger Flowcharts).vsd
	Post-Query	Appendix B (Block-level Trigger Flowcharts).vsd
PROVIDER	Pre-Insert	Appendix B (Block-level Trigger Flowcharts).vsd
	Pre-Update	Appendix B (Block-level Trigger Flowcharts).vsd
	When-New-Block-Instance	Appendix B (Block-level Trigger Flowcharts).vsd
	Pre-Query	Appendix B (Block-level Trigger Flowcharts).vsd
	Post-Query	Appendix B (Block-level Trigger Flowcharts).vsd
APPEAL_CROSS_REFERENCE	Post-Delete	Appendix B (Block-level Trigger Flowcharts).vsd
	Post-Insert	Appendix B (Block-level Trigger Flowcharts).vsd
	Post-Update	Appendix B (Block-level Trigger Flowcharts).vsd
	Post-Query	Appendix B (Block-level Trigger Flowcharts).vsd
	Pre-Query	Appendix B (Block-level Trigger Flowcharts).vsd
INFORMAL_APPEALS_AGENT	Pre-Insert	Appendix B (Block-level Trigger Flowcharts).vsd
	Pre-Update	Appendix B (Block-level Trigger Flowcharts).vsd

Block	Trigger	Function of PL/SQL Code
	When-New-Block-Instance	Appendix B (Block-level Trigger Flowcharts).vsd
HEARING_OFFICER	Pre-Insert	Appendix B (Block-level Trigger Flowcharts).vsd
	Pre-Update	Appendix B (Block-level Trigger Flowcharts).vsd
	When-New-Block-Instance	Appendix B (Block-level Trigger Flowcharts).vsd
HORIZ_TOOL_BLK	When-Mouse-Leave	(not included)
LITTLE_HORIZ_TOOL_BLK	When-Mouse-Leave	(not included)
CONTROL_BLOCK	N/A	N/A

Table 5.2-1. Paapp1.fmb Block-level Triggers.

Table 5.2-2 below shows that the Blocks belonging to adhoc.fmb do not have any Triggers.

Block	Trigger	Function of PL/SQL Code
CRTK1	N/A	N/A
CRTK2	N/A	N/A
CRTK3	N/A	N/A

Table 5.2-2. Adhoc.fmb Block-level Triggers.

5.3 Item-level Triggers

Table 5.3-1 below lists paapp1.fmb Items and their corresponding Blocks and Triggers. To view flowcharts depicting Trigger functionality, click the hyperlink in the “Function of PL/SQL Code” column of the table.

Block	Item	Trigger	Function of PL/SQL Code
PROVIDER_APPEAL	PROVIDER_ID	When-Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	REPRESENT1_FIRST_NAME	When-Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	REPRESENT1_LAST_NAME	When-Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	REPRESENT1_CITY	When-Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	REPRESENT2_FIRST_NAME	When-Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	REPRESENT2_LAST_NAME	When-Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	REPRESENT2_CITY	When-Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	RECIPIENT_FIRST_NAME	When-Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	RECIPIENT_LAST_NAME	When-Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	CASE_SUMMARY_DUE	Pre-Text-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	IFFC_DUE_DATE	Pre-Text-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	ADDITIONAL_DOCU_DUE	Pre-Text-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	WRITTEN_SUBMISSIONS_DUE	Pre-Text-Item	Appendix C (Item-level Trigger Flowcharts).vsd

(Table continues on next page.)

Block	Item	Trigger	Function of PL/SQL Code
	INFORMAL_ DECISION_DUE	Pre-Text-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	FORMAL_ REQUEST_DUE	Pre-Text-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	DOCUMENTARY_ EVIDENCE_DUE	Pre-Text-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	OBJECTIONS_TO _DOCU_DUE	Pre-Text-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	HO_RULE_ OBJECTIONS_ DUE	Pre-Text-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	FORMAL_ HEARING_DUE	Pre-Text-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	OPENING_ BRIEFS_DUE	Pre-Text-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	REPLY_BRIEFS_ DUE	Pre-Text-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	DEPT_REPLY_ BRIEFS_DUE	Pre-Text-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	HO_ RECOMMENDATIONS_ DUE	Pre-Text-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	EXCEPTIONS_TO _HO_DUE	Pre-Text-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	DIRECTOR_ DECISION_DUE	Pre-Text-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	INFORMAL_ REQUEST_ RECEIVED	When- Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	IFFC_HELD_ DATE	When- Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	INFORMAL_ DECISION_DATE	When- Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	FORMAL_ REQUEST_ RECEIVED	When- Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	DOCUMENTARY_ EVIDENCE_ FILED	When- Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	OBJECTIONS_TO _DOCU_FILED	When- Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	FORMAL_ HEARING_HELD	When- Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	PROV_OPENING_ BRIEFS_FILED	When- Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	DEPT_OPENING_ BRIEFS_FILED	When- Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	HO_ RECOMMENDA- TION_RECEIVED	When- Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	FORMAL_AGENCY_ COUNSEL	When-New- Item-Instance	Appendix C (Item-level Trigger Flowcharts).vsd
PROVIDER	FIRST_NAME	When- Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd

(Table continues on next page.)

Block	Item	Trigger	Function of PL/SQL Code
	LAST_NAME	When-Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	CITY	When-Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	CONTACT_FIRST_NAME	When-Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	CONTACT_LAST_NAME	When-Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	OK_BTN	When-Button-Pressed	Appendix C (Item-level Trigger Flowcharts).vsd
	CANCEL_BTN	When-Button-Pressed	Appendix C (Item-level Trigger Flowcharts).vsd
APPEAL_CROSS_REFERENCE	OK_BTN	When-Button-Pressed	Appendix C (Item-level Trigger Flowcharts).vsd
	CANCEL_BTN	When-Button-Pressed	Appendix C (Item-level Trigger Flowcharts).vsd
INFORMAL_APPEALS_AGENT	FIRST_NAME	When-Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	LAST_NAME	When-Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	OK_BTN	When-Button-Pressed	Appendix C (Item-level Trigger Flowcharts).vsd
	CANCEL_BTN	When-Button-Pressed	Appendix C (Item-level Trigger Flowcharts).vsd
HEARING_OFFICER	FIRST_NAME	When-Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	LAST_NAME	When-Validate-Item	Appendix C (Item-level Trigger Flowcharts).vsd
	OK_BTN	When-Button-Pressed	Appendix C (Item-level Trigger Flowcharts).vsd
	CANCEL_BTN	When-Button-Pressed	Appendix C (Item-level Trigger Flowcharts).vsd
HORIZ_TOOL_BLK	QUERY_H	When-Mouse-Click	(not included)
	COMMIT_H	When-Mouse-Click	(not included)
	LIST_H	When-Mouse-Click	(not included)
	INSERT_H	When-Mouse-Click	(not included)
	DELETE_H	When-Mouse-Click	(not included)
	EDIT_H	When-Mouse-Click	(not included)
	DOWN_H	When-Mouse-Click	(not included)
	UP_H	When-Mouse-Click	(not included)
	LEFT_H	When-Mouse-Click	(not included)
	RIGHT_H	When-Mouse-Click	(not included)
CLEAR_H	When-Mouse-Click	(not included)	

(Table continues on next page.)

Block	Item	Trigger	Function of PL/SQL Code
	PRINT_H	When-Mouse-Click	(not included)
	EXIT_H	When-Mouse-Click	(not included)
	HINT_H	When-Mouse-Click	(not included)
LITTLE_HORIZ_TOOL_BLK	EDIT_H	When-Mouse-Click	(not included)
	COMMIT_H	When-Mouse-Click	(not included)
	LIST_H	When-Mouse-Click	(not included)
	LEFT_H	When-Mouse-Click	(not included)
	RIGHT_H	When-Mouse-Click	(not included)
	CLEAR_H	When-Mouse-Click	(not included)
	PRINT_H	When-Mouse-Click	(not included)
	EXIT_H	When-Mouse-Click	(not included)
CONTROL_BLK	LOOKUP_BTN	When-Button-Pressed	Appendix C (Item-level Trigger Flowcharts).vsd
	REF_BTN	When-Button-Pressed	Appendix C (Item-level Trigger Flowcharts).vsd

Table 5.3-1. Paappl . fmb Item-level Triggers

Table 5.3-2 below lists adhoc . fmb Items and their corresponding Blocks and Triggers. To view flowcharts depicting Trigger functionality, click the hyperlink in the “Function of PL/SQL Code” column of the table.

Block	Item	Trigger	Function of PL/SQL Code
CRTK1	PROVIDER_NO1	When-Validate-Item	(not included)
		Key-Prev-Item	(not included)
	AGENT_ID	When-Validate-Item	(not included)
CRTK2	N/A	N/A	N/A
CRTK3	RUN_REPT	When-Button-Pressed	Appendix C (Item-level Trigger Flowcharts).vsd
	EXIT_BUT	When-Button-Pressed	Appendix C (Item-level Trigger Flowcharts).vsd

Table 5.3-2. Adhoc . fmb Item-level Triggers.

6. Reports

The Provider Appeals Source and Production directories store the `rdf` folders that contain the application's report files. (See Section 2, "Directories", for information on locating the `rdf` folders.)

`Adhoc.fmb` is the Provider Appeals Form that contains the Reports Triggers and Program Units. For information on `adhoc.fmb` Triggers and Program Units, see the "Triggers and Flowcharts" and "Program Units" and sections of this guide.

Table 6-1 below shows the Reports menu options and the corresponding report files. Note that these report files (e.g., `adhoc`, `adhoc2`, `adhoc3`, and `adhoc4`) contain straightforward Program Units that are not included in this manual in the form of narratives or flowcharts.

For ad-hoc report query code, see Q1 under the Data Model / Queries nodes in the ad-hoc report files.

Reports Menu Option	Corresponding <code>rdf</code> File	<code>Rdf</code> File Explanation	Comment
Informal Appeals	ADHOC	Standard dynamic report that allows the inclusion of up to 6 field selections	For ad-hoc report query code, see Q1 under the Data Model / Queries nodes in the specified <code>.rdf</code> file.
	ADHOC2	Extended dynamic report that allows the inclusion of 7-12 field selections	For ad-hoc report query code, see Q1 under the Data Model / Queries nodes in the specified <code>.rdf</code> file.
	ADHOC3	Standard dynamic report that includes information in the "Informal Comments" field on the main interface screen and allows the inclusion of up to 6 field selections	For ad-hoc report query code, see Q1 under the Data Model / Queries nodes in the specified <code>.rdf</code> file.
Formal Appeals	ADHOC4	Sole report file for Formal Appeal reports	For ad-hoc report query code, see Q1 under the Data Model / Queries nodes in the specified <code>.rdf</code> file.

Table 6-1 Reports Menu Options and Corresponding Report Files.

/NOTES:

- The application does not currently use `Provapp1.rdf` and `Provapp2.rdf`.

- For ad-hoc report query code, see Q1 under the Data Model / Queries nodes of the specified `.rdf` files.

7. Database Tables and Parent/Child Relationships

This section describes each table and names its fields, attributes, constraints, and indexed columns (where applicable). Note that indexed columns are indicated with a “ζ” at the right.

Also, corresponding Blocks are named in cases where the table is a base table. Any parent/child tables, primary keys, and foreign keys are indicated as well.

7.1 APPEAL_CROSS_REFERENCE

Stores Appellant ID information about Appeals/Providers that have been cross-referenced. The Appellant ID in the APPELLANT_ID1 column represents the ID of the Appeal that was used (i.e., the Appeal that was on the main interface screen) to create the cross-reference association.

Name	Null?	Type
APPELLANT_ID1	NOT NULL	NUMBER(12)ζ
APPELLANT_ID2	NOT NULL	NUMBER(12)ζ

The APPEAL_CROSS_REFERENCE table is the base table for the APPEAL_CROSS_REFERENCE Block.

Parent Table	Primary Key	Foreign Key
PROVIDER_APPEAL	APPELLANT_ID	APPELLANT_ID1
		APPELLANT_ID2

Table 7.1-1. Parent Table of APPEAL_CROSS_REFERENCE Table.

7.2 HEARING_OFFICER

Stores names of Hearing Officers and information indicating when and by whom a Hearing Officer record was most recently changed.

Name	Null?	Type
HEARING_OFFICER_ID	NOT NULL	VARCHAR2(10)ζ
LAST_NAME		VARCHAR2(30)
FIRST_NAME		VARCHAR2(30)
MIDDLE_INITIAL		CHAR(1)
LAST_MODIFIED_BY		VARCHAR2(10)
LAST_MODIFIED		DATE

The HEARING_OFFICER table is the base table for the HEARING_OFFICER Block.

Child Table	Primary Key	Foreign Key
PROVIDER_APPEAL	HEARING_OFFICER_ID	HEARING_OFFICER_ID

Table 7.2-1. Child Table of HEARING_OFFICER Table.

7.3 INFORMAL_APPEALS_AGENT

Stores names of Informal Appeals Agents and information indicating when and by whom an Informal Appeals Agent record was most recently changed.

Name	Null?	Type
INFORMAL_AGENT_ID	NOT NULL	VARCHAR2(10)ζ
LAST_NAME		VARCHAR2(30)
FIRST_NAME		VARCHAR2(30)
MIDDLE_INITIAL		CHAR(1)
LAST_MODIFIED_BY		VARCHAR2(10)
LAST_MODIFIED		DATE

The INFORMAL_APPEALS_AGENT table is the base table for the INFORMAL_APPEALS_AGENT Block.

Child Table	Primary Key	Foreign Key
PROVIDER_APPEAL	INFORMAL_AGENT_ID	INFORMAL_AGENT_ID

Table 7.3-1. Child Table of INFORMAL_APPEALS_AGENT Table.

7.4 PROVIDER

Stores information that pertains to the different means of contacting a Provider (e.g., phone, fax, email, address, etc.).

Name	Null?	Type
PROVIDER_ID	NOT NULL	NUMBER(12) ζ
LAST_NAME		VARCHAR2(30) ζ
FIRST_NAME		VARCHAR2(30) ζ
MIDDLE_INITIAL		CHAR(1)
PROVIDER_NAME		VARCHAR2(80) ζ
MEDICAID_NUMBER		VARCHAR2(12) ζ
ADDRESS_LINE_1		VARCHAR2(30)
ADDRESS_LINE_2		VARCHAR2(30)
CITY		VARCHAR2(25)
STATE		VARCHAR2(2)
ZIP_CODE		VARCHAR2(9)
CONTACT_LAST_NAME		VARCHAR2(30)
CONTACT_FIRST_NAME		VARCHAR2(30)
CONTACT_MIDDLE_INITIAL		CHAR(1)
PHONE		VARCHAR2(10)
FAX		VARCHAR2(10)
EMAIL		VARCHAR2(50)

The PROVIDER table is the base table for the PROVIDER Block.

Child Table	Primary Key	Foreign Key
PROVIDER_APPEAL	PROVIDER_ID	PROVIDER_ID

Table 7.4-1. Child Table of PROVIDER Table.

7.5 PROVIDER_APEAL

Stores the information in the main interface fields. Includes Representative, Recipient, CSA, Informal Appeal, Formal Appeal, and Court information.

Name	Null?	Type
APPELLANT_ID	NOT NULL	NUMBER(12)
PROVIDER_ID		NUMBER(12)
REPRESENT1_LAST_NAME		VARCHAR2(30)
REPRESENT1_FIRST_NAME		VARCHAR2(30)
REPRESENT1_MIDDLE_INITIAL		CHAR(1)
REPRESENT1_FIRM_NAME		VARCHAR2(80)
REPRESENT1_ADDRESS_LINE_1		VARCHAR2(30)
REPRESENT1_ADDRESS_LINE_2		VARCHAR2(30)
REPRESENT1_CITY		VARCHAR2(25)
REPRESENT1_STATE		VARCHAR2(2)
REPRESENT1_ZIP_CODE		VARCHAR2(9)
REPRESENT2_LAST_NAME		VARCHAR2(30)
REPRESENT2_FIRST_NAME		VARCHAR2(30)
REPRESENT2_MIDDLE_INITIAL		CHAR(1)
REPRESENT2_FIRM_NAME		VARCHAR2(80)
REPRESENT2_ADDRESS_LINE_1		VARCHAR2(30)
REPRESENT2_ADDRESS_LINE_2		VARCHAR2(30)
REPRESENT2_CITY		VARCHAR2(25)
REPRESENT2_STATE		VARCHAR2(2)
REPRESENT2_ZIP_CODE		VARCHAR2(9)
TYPE_OF_APEAL		VARCHAR2(30)
RECIPIENT_LAST_NAME		VARCHAR2(30)
RECIPIENT_FIRST_NAME		VARCHAR2(30)
RECIPIENT_MIDDLE_INITIAL		CHAR(1)
CSA_PROVIDER_CHAIN_NAME		VARCHAR2(80)
CSA_CR_PERIOD_ENDED1		DATE
CSA_CR_PERIOD_ENDED2		DATE
CSA_CR_PERIOD_ENDED3		DATE
DEPT_ACTION_APEALED_FROM		DATE
INFORMAL_REQUEST_RECEIVED		DATE
INFORMAL_AGENT_ID		VARCHAR2(10)
CASE_SUMMARY_DUE		DATE
CASE_SUMMARY_RECEIVED		DATE
IFFC_TYPE		VARCHAR2(3)
IFFC_DUE_DATE		DATE
IFFC_SCHEDULED_DATE		DATE
IFFC_HELD_DATE		DATE
IFFC_DEC_DRAFT_BEGIN_DATE		DATE
IFFC_DEC_DRAFT_END_DATE		DATE
ADDITIONAL_DOCU_DUE		DATE
ADDITIONAL_DOCU_RECEIVED		DATE
WRITTEN_SUBMISSIONS_DUE		DATE
WRITTEN_SUBMISSIONS_RECEIVED		DATE
INFORMAL_DECISION_DUE		DATE
INFORMAL_DECISION_DATE		DATE
INFORMAL_DISPOSITION		VARCHAR2(3)
INFORMAL_DECISION_RESULT		VARCHAR2(3)
INFORMAL_COMMENTS		VARCHAR2(800)
FORMAL_REQUEST_DUE		DATE

FORMAL_REQUEST_RECEIVED	DATE
HEARING_OFFICER_ID	VARCHAR2(10)
HO_APPOINTED_DATE	DATE
DOCUMENTARY_EVIDENCE_DUE	DATE
DOCUMENTARY_EVIDENCE_FILED	DATE
OBJECTIONS_TO_DOCU_DUE	DATE
OBJECTIONS_TO_DOCU_FILED	DATE
HO_RULE_OBJECTIONS_DUE	DATE
HO_RULE_OBJECTIONS_DATE	DATE
FORMAL_HEARING_DUE	DATE
FORMAL_HEARING_SCHEDULED	DATE
FORMAL_HEARING_HELD	DATE
OPENING_BRIEFS_DUE	DATE
PROV_OPENING_BRIEFS_FILED	DATE
DEPT_OPENING_BRIEFS_FILED	DATE
REPLY_BRIEFS_DUE	DATE
PROV_REPLY_BRIEFS_FILED	DATE
DEPT_REPLY_BRIEFS_FILED	DATE
HO_RECOMMENDATION_DUE	DATE
HO_RECOMMENDATION_RECEIVED	DATE
HO_DECISION_RESULT	VARCHAR2(3)
EXCEPTIONS_TO_HO_DUE	DATE
PROV_EXCEPTIONS_TO_HO_FILED	DATE
DEPT_EXCEPTIONS_TO_HO_FILED	DATE
DIRECTOR_DECISION_DUE	DATE
DIRECTOR_DECISION_DATE	DATE
FORMAL_DISPOSITION	VARCHAR2(3)
DIRECTOR_DECISION_RESULT	VARCHAR2(3)
FORMAL_COMMENTS	VARCHAR2(800)
COURT_NOTICE_RECEIVED	DATE
COURT_PETITION_RECEIVED	DATE
COURT_RECORD_SUBMITTED	DATE
COURT_HEARING_SCHEDULED	DATE
COURT_REPORTER_OBTAINED	DATE
COURT_HEARING_HELD	DATE
COURT_ORDER_ENTERED	DATE
COURT_DISPOSITION	VARCHAR2(3)
COURT_DECISION_RESULT	VARCHAR2(3)
COURT_COMMENTS	VARCHAR2(800)
ENTERED_BY	VARCHAR2(10)
DATE_ENTERED	DATE
LAST_MODIFIED_BY	VARCHAR2(10)
LAST_MODIFIED	DATE
REPRESENT1_PHONE	VARCHAR2(10)
REPRESENT1_FAX	VARCHAR2(10)
REPRESENT1_EMAIL	VARCHAR2(50)
REPRESENT2_PHONE	VARCHAR2(10)
REPRESENT2_FAX	VARCHAR2(10)
REPRESENT2_EMAIL	VARCHAR2(50)
DEPT_REPLY_BRIEFS_DUE	DATE
INFORMAL_CLOSE	CHAR(1)
FORMAL_CLOSE	CHAR(1)
COURT_CLOSE	CHAR(1)

The PROVIDER_APPEAL table is the base table for the PROVIDER_APPEAL Block.

Parent Tables	Foreign Key	Primary Key
PROVIDER	PROVIDER_ID	PROVIDER_ID
INFORMAL_APPEALS_AGENT	INFORMAL_AGENT_ID	INFORMAL_AGENT_ID
HEARING_OFFICER	HEARING_OFFICER_ID	HEARING_OFFICER_ID

Table 7.5-1. Parent Tables of PROVIDER_APPEAL Table.

Child Table	Foreign Key	Primary Key
APPEAL_CROSS_REFERENCE	APPELLANT_ID1 APPELLANT_ID2	APPELLANT_ID

Table 7.5-2. Child Table of the PROVIDER_APPEAL Table.

(See Section 8, "Entity Relationship Diagram", on next page.)

8. Entity Relationship Diagram

In the Provider Appeals System, an Appeal must have one and only one Provider record associated with it; it may or may not have one or more Cross-reference records associated with it; it may or may not have one and only one Hearing Officer record associated with it, and it may or may not have one and only one Informal Appeals Agent record associated with it.

A Provider record may have one or more Appeal records associated with it.

A Cross-reference record must have two Appeals records associated with it.

A Hearing Officer record may have one or more Appeal records associated with it.

An Informal Appeals Agent record may have one or more Appeal records associated with it.

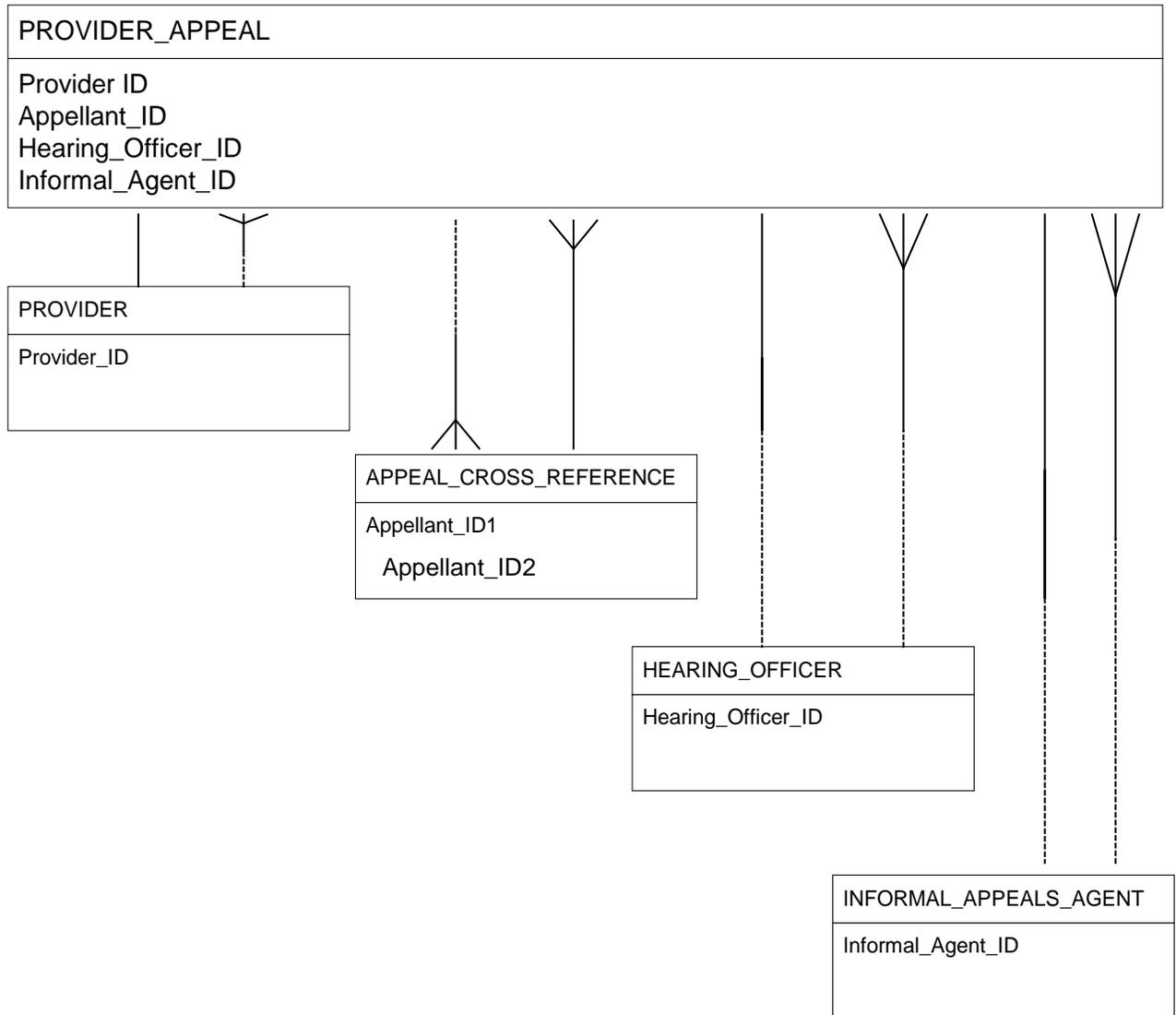


Figure 8-1. Entity Relationship Diagram.