engenic

"Automating Telecommunications"

TeleFlow Relay

Database Dictionary

Revision: 1

Last Edited: 2010-04-16

TeleFlow Relay Version History
Chris Gosbee, V1.0, 2009/10/30
Chris Gosbee, V1.1, 2010/03/12

Revision History – Before TFRelay
Chris Gosbee 2005/06/01
Tim Forner 2007/01/16
Chris Gosbee 2007/01/24
Tim Forner 2007/02/05
Tim Forner 2007/04/30
Tim Forner 2007/06/19
Chris Gosbee 2008/09/09

Suite 618 - 1111 Melville Street
Vancouver, B.C.   V6E 3V6
604-639-6391 - 888-engenic - 888-364-3642
www.engenic.com

TABLE OF CONTENTS

Design Rules 3
TeleFlow Relay – Brief Database Overview 3
DB Dictionary – Table Description / Layout 3
app_sub_billing_defaults 4
app_subscription_billing (client) 5
app_type_task_fail_settings 6
application_type 7
area_exchange 9
call_log_master 10
call_log (client) 12
callout_application_defaults 14
callout_schedule (client) 16
callout_settings (client) 19
callout_task 22
client 25
client_application_subscription 27
client_ip_access 29
client_details (client) 30
custom_message (client) 33
dnis_service 35
event_log (client) 37
event_log_master 39
file_application_transfer (client) 41
guest_active (client) 42
ivrconfig 43
media_file (client) 47
media_file_transfer (client) 51
port_task 52
registered_sip_uri 55
sales_affiliate 56
swipool 57
tfr_server 59
timezone 60
user_access_log 61
user_active (client) 61
user_banned (client) 62
user_details (client) 63
user_master 66

Design Rules

See the “TeleFlowRelay_DatabaseDesign_Rules” document.

TeleFlow Relay – Brief Database Overview

A TeleFlow Relay “site” will consist of one or more TeleFlow Relay servers, each with its own TeleFlow Relay functions. A TeleFlow Relay database server will house a TeleFlow Relay master database, in addition to one or more client databases. The master database contains data for managing a site, while the client databases all have client-specific data only.

Special: In environments where only a single client database is required, the TeleFlow Relay master and client databases can be a single database, although the applications in the site will treat them as master and client.

DB Dictionary – Table Description / Layout

The Database Dictionary is sorted alphabetically by table name.

Generally, the table description format/layout is as follows:

Table name: The name of the table. For client-level database tables, this is followed by (client) to make the table of contents easier to read.

Database: Indicates whether the table can be found in the top-level management TeleFlow Relay database, or in TFRelay client databases. Can be one of: “TeleFlow Relay” or “Client”.

Primary Key: The table's primary key field.

Foreign Keys: List of foreign keys in the table. Following each foreign key field name, the table the field can be found in is defined in (parentheses).

Indexes: Lists all indexes on the table. (Not including the assumed PK clustered index)

Relationships: Describes the relationships between this table and any others. Notes the tables that it has a relationship with in **bold**, followed by the name of the foreign key(s) used to make the link in the relationship in (parentheses).

Description: A description of the table precedes the table contents.

Table/field contents: A table describing all the fields in the table, their data types, sizes/lengths, and any special information about the data they contain.

SPECIAL: Yellow highlights are fields that need the logic they suggest built up around them.

app_sub_billing_defaults

Database: TeleFlow Relay
Primary Key: appsub_billing_defaults_id
Foreign Keys: app_type_name (application_type)
Indexes: app_type_name
Relationships: One-to-one with application_type.

The default billing settings for a client application subscription.

Field Name	Type	Size	Description
appsub_billing_defaults_id	int	10	Primary Key (Auto-Incremented, Unique).
app_type_name	varchar	30	Foreign key: Links to application_type.app_type_name. Indicates which application the billing defaults apply to.
appsub_billing_package	varchar	30	A name/code describing a billing package. (A billing package would essentially be a named code representing what values the remaining fields in this table will have.)
appsub_billing_min_charge	decimal	10,2	A minimum monthly charge for the service.
appsub_billing_max_charge	decimal	10,2	A maximum monthly charge for the service.
appsub_transaction_charge	decimal	10,2	Amount to charge per transaction for the service.
appsub_transaction_unit	varchar	10	Code indicating what constitutes a charge for this service. Possible values are: CALL MINUTE
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.

app_subscription_billing (client)

Database: Client
Primary Key: appsub_billing_id
Foreign Keys: app_type_name (application_type)
Indexes: app_type_name
Relationships: One-to-one with **application_type** (app_type_name).
One-to-one with **user_details** (u_id).

The billing settings for a client application subscription.

Field Name	Type	Size	Description
appsub_billing_id	int	10	Primary Key (Auto-Incremented, Unique).
app_type_name	varchar	30	Foreign key: Links to application_type.app_type_name. Indicates which application the billing defaults apply to.
u_id	int	10	Foreign key: Links to user_details.u_id. Indicates the user to contact for billing inquiries.
appsub_billing_package	varchar	30	A name/code describing a billing package. (A billing package would essentially be a named code representing what values the remaining fields in this table will have.)
appsub_billing_min_charge	decimal	10,2	A minimum monthly charge for the service.
appsub_billing_max_charge	decimal	10,2	A maximum monthly charge for the service.
appsub_transaction_charge	decimal	10,2	Amount to charge per transaction for the service.
appsub_transaction_unit	varchar	10	Code indicating what constitutes a charge for this service. Possible values are: CALL MINUTE
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.
create_dtm_client	datetime	-	Row create date/time, in client's local time.
update_dtm_client	datetime	-	Last date/time the row was updated, in client's local time.

app_type_task_fail_settings

Database: TeleFlow Relay
Primary Key: app_task_fail_settings_id
Foreign Keys: app_type_name (application_type)
Indexes: app_type_name
Relationships: One-to-one with **application_type** (app_type_name).

The settings to use to indicate an application task (Eg. Reaching a remote party in a call-out application to deliver a message, such as an appointment reminder) ended in a failure (I.e. Based on system rules, no more calls will be placed to complete message delivery). These settings are primarily intended to allow a generic application (currently, the TFRelay callout application) to make determinations based on call rules to update a client-level application table to indicate when a task could not be completed successfully.

Field Name	**Type**	**Size**	**Description**
app_task_fail_settings_id	int	10	Primary Key (Auto-Incremented, Unique).
app_type_name	varchar	30	Foreign key: Links to application_type.app_type_name. Indicates which application the task failure settings apply to.
fail_update_state_table	varchar	30	The table to reflect a task failure in.
fail_update_app_id_field	varchar	30	The primary key to use when updating a record with a task failure.
fail_update_state_field_1	varchar	30	The first field to receive the fail state update.
fail_update_state_update_1	varchar	30	The value to update fail_update_state_field_1 with, to indicate an application task failure.
fail_update_state_field_2	varchar	30	The second field to receive the fail state update.
fail_update_state_update_2	varchar	30	The value to update fail_update_state_field_2 with, to indicate an application task failure.
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.

application_type

Database:	TeleFlow Relay
Primary Key:	application_type_id
Foreign Keys:	None
Indexes:	app_type_name app_type_status
Relationships:	One-to-one with **callout_application_default** (app_type_name). One-to-one with **callout_settings** (app_type_name). (In Client DB).

Describes the applications available on the TeleFlow Relay server(s) and indicates application files and their locations.

Field Name	**Type**	**Size**	**Description**
application_type_id	int	10	Primary Key (Auto-Incremented, Unique). This key IS NOT TO BE USED IN JOINS. Application types are related by what are effectively "defines" (which match the app_type_name field) used throughout TeleFlow Relay applications, be they IVR or Web.
app_type_name	varchar	30	A code or name that is used by applications to gather application type data.
app_type_status	varchar	10	A status code indicating the present state of the application. Valid values: ACTIVE: Available to be assigned as a service for clients. DELETED: Application's supporting files are not available.
app_type_group_codes	varchar	100	Application grouping codes under which to group the application. This is primarily used for displaying the application along with others. An example would be: "MR" for app_type_name APPOINTMENT_REMINDER indicating that the appointment reminder application is to be grouped with other Medical Relay applications.
app_type_group_menu_order	int	10	The order in which this application should be displayed along with other applications in the same group.
app_type_billing_code	varchar	10	A code indicating how billings are done for the application.
app_type_display	varchar	30	The display name for the application. This is used in configuration and display screens to represent the application type.
app_type_desc	varchar	200	A description of the application and its purpose.
app_type_tf_app	varchar	200	The path to and name of the application's primary TeleFlow Application (TAP) file.
app_type_web_app	varchar	200	The relative path (from the website root) to and name of the application's primary web application (.html/.php/etc) file.

app_type_callout_fail_tf_app	varchar	200	The path to and name of the TeleFlow Application (TAP) file to run if this application type requires callout, and if/when such a callout fails. (This allows a given application to perform database updates or other logic based on a failed callout.) Presently, most such updates are being performed using the app_type_task_fail_settings system, but this placeholder exists in case it needs to be put to use.
app_tel_connect_delay	int	4	Default setting for dnis_service.tel_connect_delay. A deliberate pause after answering an inbound call, before proceeding to run the appropriate application, expressed in milliseconds. This pause exists to account for the Telco's delay completing the connection from caller to IVR.
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.

area_exchange

Database:	TeleFlow Relay
Primary Key:	area_exchange_id
Foreign Keys:	None
Indexes:	area_exchange (compound: server_area_code, server_exchange, remote_area_code, remote_exchange)
Relationships:	None

Contains cached results from HTTP lookup for local/long-distance phone number resolution.

Field Name	Type	Size	Description
area_exchange_id	int	10	Primary Key (Auto-Incremented, Unique)
server_area_code	int	3	Area code the server is calling from
server_exchange	int	3	Exchange number (the first 3 digits of a phone number) the server is calling from.
remote_area_code	int	3	Area code the server is calling to
remote_exchange	int	3	Exchange number (the first 3 digits of a phone number) the server is calling to.
local_yn	char	1	Y/N: Combination of phone is a local call?
area_exchange_status	varchar	10	Status of this Area/Exchange row. Possible values are: GOOD Exchange has generated at least one successfully answered call, so the local_yn flag must be good/valid. MANUAL local_yn flag has been update manually. UNKNOWN Exchange has not yet been fully resolved via a successful answered call.
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.

call_log_master

Database: TeleFlow Relay
Primary Key: call_log_id
Foreign Keys: app_type_name (application_type)
client_id (client)
callout_task_id (callout_task)
callout_app_id (application_type-specific table)
Indexes: None
Relationships: Many-to-One with **application_type** (app_type_name).
Optional Many-to-One with **callout_task** (callout_task_id). (Outbound calls only).
Optional Many-to-One with **client** (client_id).
Optional Many-to-One with **[application_type-specific table]** (callout_app_id).
(Outbound calls only: callout_app_id is a “generic” foreign key in this table, coming from a table that is specific to the application type. Eg. For app_type_name “APPOINTMENT_REMINDER” callout_app_id would be a foreign key from the appointment table in the client's database.)

A log of each call performed by or made to the system. NOTE: CallCapture calls are not considered calls in the traditional sense, and as such, are not reflected in the call_log tables, but rather, are stored as media_file entries.

Field Name	**Type**	**Size**	**Description**
call_log_id	int	10	Primary Key (Auto-Incremented, Unique)
call_direction	varchar	3	IN/OUT: Indicates whether the call logged was an inbound or outbound call.
app_type_name	varchar	30	Foreign key. Links to application_type.app_type_name. Indicates the TeleFlow Application (.TAP) run when the call was answered. (based on the entry in the dnis_service table at the time)
client_id	int	10	Foreign key: Links to client.client_id. Indicates the client the call was on behalf of.
call_start_dtm_utc	datetime	-	The date/time the call started, in UTC.
call_end_dtm_utc	datetime	-	The date/time the call completed, in UTC.
call_start_dtm_ivr	datetime	-	The date/time the call started, expressed in the local time of the server the call was handled on.
call_end_dtm_ivr	datetime	-	The date/time the call completed, expressed in the local time of the server the call was handled on.
call_duration	int	6	The amount of time, in seconds, the call lasted from ANSWER to HANGUP. This will be 0 in calls that were busy, failed, rang without answering.
call_attempt_duration	int	6	The amount of time, in seconds, the call lasted from the start of placing the call to determining the call was over. (So, in the case of a call that was answered, this is call_duration plus the time it took for the caller to answer) For inbound calls, call_attempt_duration and call_duration are the same.

ivr_computer_name	varchar	50	IVR computer name of the IVR server the call took place on.
ivr_port	int	3	The IVR port the call took place on.
called_number	varchar	20	The phone number called to initiate this call. For inbound calls, this indicates the DNIS, or the number dialed to reach the IVR. For outbound calls, this indicates the number the IVR called to reach an contact someone externally/an external system.
calling_number	varchar	20	The phone number of the initiating party of this call. For inbound calls, this is the caller's caller id. For outbound calls, this is the number the system attempts to send as its caller id.
callout_task_id	int	10	Foreign key: callout_task. callout_task_id.
callout_app_id	int	10	Foreign key: Links to the ID or primary key field value for the table used for the specific callout type. (In the case of an appointment call-out system, the appointment table’s appointment_id.)
initial_voice_length	int	10	The amount of time, in milliseconds, of the initial voice length when the call was placed. (When a call is placed and answered, the length of the answer is determined as the amount of time the answering party continues to talk.) Continuous talk is terminated by a call rule determined silence timeout.)
callout_result_code	varchar	20	A ‘code’ indicating the call’s level of success. Possible codes are: ANSWER The call was answered BUSY The call was not successful because of a busy signal. TIMEOUT The call was not successful because it wasn’t answered within a certain ring count. HANGUP The call was answered, but the call recipient hung up the phone without hearing enough information. (hung up within a short period of time) FAIL The call failed. This usually indicates a problem with the telephone system. ERROR The call was not successful because an application error occurred.

call_log (client)

Database: Client
Primary Key: call_log_id
Foreign Keys: app_type_name (teleflow_relay.application_type)
callout_schedule_id (callout_schedule)
callout_app_id (application_type-specific table)
Indexes: None
Relationships: Many-to-One with **application_type** (app_type_name).
Optional Many-to-One with **callout_schedule** (callout_schedule_id). (Outbound calls only).
Optional Many-to-One with **[application_type-specific table]** (callout_app_id). (Outbound calls only: callout_app_id is a "generic" foreign key in this table, coming from a table that is specific to the application type. Eg. For app_type_name "APPOINTMENT_REMINDER" callout_app_id would be a foreign key from the appointment table in the client's database.)

A log of each call performed by or made to the system for the client.

Field Name	Type	Size	Description
call_log_id	int	10	Primary Key (Auto-Incremented, Unique)
call_direction	varchar	3	IN/OUT: Indicates whether or not the call was inbound/received or outbound/placed by the system.
app_type_name	varchar	30	Foreign key. Links to application_type.app_type_name. Indicates the TeleFlow Application (.TAP) to run when the call is answered.
call_start_dtm_utc	datetime	-	The date/time the call started, in UTC.
call_end_dtm_utc	datetime	-	The date/time the call completed, in UTC.
call_start_dtm_client	datetime	-	The date/time the call started, expressed in the client's local time.
call_end_dtm_client	datetime	-	The date/time the call completed, expressed in the client's local time.
call_duration	int	6	The amount of time, in seconds, the call lasted from ANSWER to HANGUP. This will be 0 in calls that were busy, failed, rang without answering.
call_attempt_duration	int	6	The amount of time, in seconds, the call lasted from the start of placing the call to determining the call was over. (So, in the case of a call that was answered, this is call_duration plus the time it took for the caller to answer) For inbound calls, call_attempt_duration and call_duration are the same.
ivr_computer_name	varchar	50	IVR Server name of the server the call took place on.
ivr_port	int	3	The IVR port the call took place on.
called_number	varchar	20	The phone number called to initiate this call. For inbound calls, this indicates the DNIS, or the number dialed to reach the IVR. For outbound calls, this indicates the number the

			IVR called to reach an contact someone externally/an external system.
calling_number	varchar	20	The phone number of the initiating party of this call. For inbound calls, this is the caller's caller id. For outbound calls, this is the number the system attempts to send as its caller id.
callout_schedule_id	int	10	Foreign key: callout_schedule. callout_schedule_id.
callout_app_id	int	10	Foreign key: Links to the ID or primary key field value for the table used for the specific callout type. (In the case of an appointment call-out system, the appointment table's appointment_id.)
initial_voice_length	int	10	The amount of time, in milliseconds, of the initial voice length when the call was placed. (When a call is placed and answered, the length of the answer is determined as the amount of time the answering party continues to talk.) Continuous talk is terminated by a call rule determined silence timeout.)
callout_result_code	char	20	A 'code' indicating the call's level of success. Possible codes are: ANSWER The call was answered BUSY The call was not successful because of a busy signal. TIMEOUT The call was not successful because it wasn't answered within a certain ring count. HANGUP The call was answered, but the call recipient hung up the phone without hearing enough information. (hung up within a short period of time) FAIL The call failed. This usually indicates a problem with the telephone system. ERROR The call was not successful because an application error occurred.

callout_application_defaults

Database: TeleFlow Relay
Primary Key: application_default_id
Foreign Keys: app_type_name (application_type)
Indexes: app_type_name
callout_max_no_answer
callout_max_fail
Relationships: One-to-one with app_type_name in **application_type** table.

Contains the default settings for a particular type of callout application (Eg: Appointment Reminder). These settings are used as the default when setting up a client for a new callout application type. (To help the client with population of the **callout_settings** table in the Client DB)

Field Name	Type	Size	Description
application_default_id	int	10	Primary Key (Auto-Incremented, Unique)
app_type_name	varchar	30	Foreign key. Links to application_type. app_type_name
callout_max_fail	int	3	Maximum number of times to allow a CALL_FAIL on a particular callout task before the task is considered a failure.
callout_max_no_answer	int	3	Maximum number of times to allow a a particular callout task to not get answered (busy, or no-answer/timeout, short hangup – if short hangups are being used to force a retry - result) before the task is considered a failure.
early_hangup_check_yn	char	1	Y/N: Check whether or not the call recipient hung up quickly? See also early_hangup_max_seconds and early_hangup_retry_call.
early_hangup_max_seconds	int	5	If early_hangup_check_yn = 'Y', this value is the maximum number of seconds to consider a quick hangup.
early_hangup_retry_call	char	1	If early_hangup_check_yn = 'Y', and the hang up was less than or equal to early_hangup_max_seconds, this field determines whether or not an early hangup results in a retry of the call-out. Whether retry is on or off, checking for early hangups marks all early hangups with HANGUP as the call result to indicate an early hangup was encountered.
machine_check_yn	char	1	Y/N: Check to see if the call is answered by an automated system or answering machine?
machine_min_length	int	5	The minimum voice length when a call is answered that is to be considered an automated system.
machine_run_app_yn	char	1	Y/N: Run the primary application when the call recipient has been determined to be an automated system?

callout_start_tm_weekdays	time	-	The daily start time for weekday calls, expressed in local time. SPECIAL: In this case, “local time” refers to the local time anywhere, because this is a default setting for a client setting. (EG. We might suggest that a client start appointment reminder calls at 6pm each day. That would be 6pm local for them, regardless of their timezone.)
callout_end_tm_weekdays	time	-	The daily end time for weekday calls, expressed in local time(See “SPECIAL” note in callout_start_tm for more information).
callout_start_tm_sat	time	-	The daily start time for Saturday calls, expressed in local time(See “SPECIAL” note in callout_start_tm for more information).
callout_end_tm_sat	time	-	The daily end time for Saturday calls, expressed in local time(See “SPECIAL” note in callout_start_tm for more information).
callout_start_tm_sun	time	-	The daily start time for Sunday calls, expressed in local time(See “SPECIAL” note in callout_start_tm for more information).
callout_end_tm_sun	time	-	The daily end time for Sunday calls, expressed in local time(See “SPECIAL” note in callout_start_tm for more information).
callout_start_num_days_ahead	int	3	The number of days before an application-specific date to start a given callout task. (Eg: For appointment reminder, the callout could start at least two days before the appointment so office staff can fill canceled slots.) This field and the next one assume that the DOW doesn't matter (I.e. They don't make special considerations for what the day of the week is when determining when to call).
callout_end_num_days_ahead	int	3	The number of days before an application-specific key date to have all callout tasks for the application type for the client completed. (Eg. Appointment reminder calls should be completed 1 day before the appointment at the latest.)
callout_retry_seconds	int	6	The number of seconds between attempts to retry an incomplete callout task.
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.

callout_schedule (client)

Database: Client
Primary Key: callout_schedule_id
Foreign Keys: app_type_name (teleflow_relay.application_type)
callout_app_id (application_type-specific table)
Indexes: app_type_name
callout_status
call_date_range (compound: callout_start_dt_client, callout_end_dt_client)
Relationships: One-to-Many with **call_log** (callout_schedule_id).

Callout applications are scheduled and processed through this table.

Field Name	Type	Size	Description
callout_schedule_id	int	10	Primary Key (Auto-Incremented, Unique)
app_type_name	varchar	30	Foreign key. Links to application_type.app_type_name. Indicates the TeleFlow Application (.TAP) to run when the call is answered.
callout_app_id	int	10	Foreign key: Links to the ID or primary key field value for the table used for the specific callout type. (In the case of an appointment call-out system, the appointment table's appointment_id.)
callout_phone	varchar	20	The phone number to call out to.
send_ani	varchar	20	The ANI to send with the call. This field is optional. If it is left empty, the generic dialer will lookup the ANI to send in the client table.
callout_start_dt_client	date or varchar	- or 10	The earliest date calls for this callout schedule can start, expressed in the client's local time.
callout_end_dt_client	date or varchar	- or 10	The latest date calls for this callout task can be placed, expressed in the client's local time.
last_call_dtm_client	datetime	-	The date/time on which the most recent call for this task was placed, expressed in the client's local time.
last_call_dtm_utc	datetime	-	The date/time on which the most recent call for this task was placed, expressed in UTC.
callout_status	varchar	20	A 'code' indicating how much progress has been made on this scheduled callout. The following codes apply to the standard automated process: READY This callout has not yet been exported to/created in the TeleFlow Relay callout_task table for the current calling period. WORKING This callout has been exported to the TeleFlow Relay callout_task table, and is in progress.

			INCOMPLETE The callout was not completed during the most recent callout period, and will be attempted again during the next callout period. DONE The callout completed successfully. FAILED The callout completed unsuccessfully (Either the callout could not complete within the schedule, or reached the maximum number of failed tries) The remaining codes cover special cases outside the standard automated processes: REMOVED The call is removed from calling out and will never complete. ERROR An error occurred the last time this task was attempted.
task_fail_count	int	2	A count of the total number of times this callout task has been tried and failed. [default = 0]
task_noanswer_count	int	2	A count of the total number of times this callout task has been tried and no one answered, including failures. [default = 0]
task_error_count	int	2	A count of the total number of times this callout task has been tried and resulted in an application error. [default = 0]
task_max_fail	int	2	The maximum number of time calls can fail before the system gives up on this task.
task_max_noanswer	int	2	The maximum number of time calls aren't answered (EG. BUSY, NO ANSWER, early HANGUP) before the system gives up on this task.
callout_task_priority	int	10	A priority number from 0-99. Callout tasks are assigned to be done based on business rules, but when there are many tasks currently to be completed, they are sorted descending on this field prior to any other field. (The higher this number, the higher a priority the callout task receives) [default = 1]
callout_retry_seconds	int	6	The number of seconds between retry attempts for the callout task. [default = 1800 (30 minutes)]
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.

create_dtm_client	datetime	-	Row create date/time, in client's local time.
update_dtm_client	datetime	-	Last date/time the row was updated, in client's local time.

callout_settings (client)

Database:	Client
Primary Key:	callout_settings_id
Foreign Keys:	app_type_name (teleflow_relay.application_type)
Indexes:	app_type_name
Relationships:	One-to-one with **application_type** (app_type_name). (In TFRelay DB)

The client's call-out application settings for a specific application. (Eg: Appointment Reminder for XYZ Medical Clinic)

Field Name	**Type**	**Size**	**Description**
callout_settings_id	int	10	Primary Key (Auto-Incremented, Unique)
app_type_name	varchar	30	Foreign key. Links to application_type.app_type_name. Indicates the application these call-out settings apply to.
send_caller_id	varchar	20	Caller ID to send with each call placed.
call_transfer_status	varchar	10	ACTIVE/DISABLED: Indicates whether or not the client is willing to receive call transfers from this application.
call_transfer_phone	varchar	20	Phone number the application should use when transferring calls to the client.
call_transfer_ext	varchar	10	Extension number (if any) to dial when transferring to call_transfer_phone when the remote party answers.
call_transfer_ext_pause	int	10	The wait in milliseconds between the remote party answer and dialing the call_transfer_ext
callout_max_fail	int	3	Maximum number of times to allow a CALL_FAIL on a particular callout task before the task is considered a failure.
callout_max_no_answer	int	3	Maximum number of times to allow a a particular callout task to not get answered (busy, or no-answer/timeout, short hangup – if short hangups are being used to force a retry - result) before the task is considered a failure.
early_hangup_check_yn	char	1	Y/N: Check whether or not the call recipient hung up quickly? See also early_hangup_max_seconds and early_hangup_retry_call.
early_hangup_max_seconds	int	5	If early_hangup_check_yn = 'Y', this value is the maximum number of seconds to consider a quick hangup.
early_hangup_retry_call	char	1	If early_hangup_check_yn = 'Y', and the hang up was less than or equal to early_hangup_max_seconds, this field determines whether or not an early hangup results in a retry of the call-out. Whether retry is on or off, checking for

			early hangups marks all early hangups with HANGUP as the call result to indicate an early hangup was encountered.
machine_check_yn	char	1	Y/N: Check to see if the call is answered by an automated system or answering machine?
machine_min_length	int	5	The minimum voice length when a call is answered that is to be considered an automated system.
machine_run_app_yn	char	1	Y/N: Run the primary application when the call recipient has been determined to be an automated system?
callout_start_tm_weekdays	time or varchar	- or 8	The daily start time for weekday calls, expressed in local time. SPECIAL: In this case, “local time” refers to the local time anywhere, because this is a default setting for a client setting. (EG. We might suggest that a client start appointment reminder calls at 6pm each day. That would be 6pm local for them, regardless of their timezone.)
callout_end_tm_weekdays	time or varchar	- or 8	The daily end time for weekday calls, expressed in local time(See “SPECIAL” note in callout_start_tm for more information).
callout_start_tm_sat	time or varchar	- or 8	The daily start time for Saturday calls, expressed in local time(See “SPECIAL” note in callout_start_tm for more information).
callout_end_tm_sat	time or varchar	- or 8	The daily end time for Saturday calls, expressed in local time(See “SPECIAL” note in callout_start_tm for more information).
callout_start_tm_sun	time or varchar	- or 8	The daily start time for Sunday calls, expressed in local time(See “SPECIAL” note in callout_start_tm for more information).
callout_end_tm_sun	time or varchar	- or 8	The daily end time for Sunday calls, expressed in local time(See “SPECIAL” note in callout_start_tm for more information).
callout_start_num_days_ahead	int	3	The number of days before an application-specific date to start a given callout task. (Eg: For appointment reminder, the callout could start at least two days before the appointment so office staff can fill canceled slots.) This field and the next one assume that the DOW doesn't matter (I.e. They don't make special considerations for what the day of the week is when determining when to call).
callout_end_num_days_ahead	int	3	The number of days before an application-specific date to ensure a given callout task is completed. No calls will be done after that point. (Eg: For appointment reminder, the callout should be done the day before the appointment takes place.)

			[default = 1]
callout_retry_seconds	int	6	The number of seconds between attempts to retry an incomplete callout task. [default = 1800]
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.
create_dtm_client	datetime	-	Row create date/time, in client's local time.
update_dtm_client	datetime	-	Last date/time the row was updated, in client's local time.

callout_task

Database:	TeleFlow Relay
Primary Key:	callout_task_id
Foreign Keys:	app_type_name (application_type) client_id (client) callout_schedule_id (client.callout_schedule) callout_app_id (client.application_type-specific table)
Indexes:	app_type_name task_status date_range_status (compound: callout_start_dtm_utc, callout_end_dtm_utc, task_status)
Relationships:	One-to-Many with **call_log_master**. Many-to-One with **client** (client_id). One-to-One with **callout_schedule** (callout_schedule_id). One-to-One with **application_type-specific table** (callout_app_id).

Call-out tasks are assigned and processed through this table.

Field Name	Type	Size	Description
callout_task_id	int	10	Primary Key (Auto-Incremented, Unique)
app_type_name	varchar	30	Foreign key. Links to application_type.app_type_name. Indicates the TeleFlow Application (.TAP) to run when the call is answered.
client_id	int	10	Foreign key: Links to client.client_id. Indicates the client the call-out task was created for.
callout_schedule_id	int	10	Foreign key: Links to callout_schedule.callout_schedule_id. Indicates the client-level db callout_schedule this task was created from.
callout_app_id	int	10	Application_type-specific Foreign key: Links to the ID or primary key field value for the table used for the specific callout type. (In the case of an appointment call-out system, the appointment table's appointment_id.)
send_ani	varchar	20	The ANI to send with the call. This field is optional. If it is left empty, the generic dialer will lookup the ANI to send in the client table.
callout_phone	varchar	20	The phone number to call out to.
callout_start_dtm_utc	datetime	-	The date/time calls for this callout task will start.
callout_end_dtm_utc	datetime	-	The latest date/time calls for this callout task can be placed.
last_call_dtm_utc	datetime	-	The date/time on which the last call for this task was placed.
last_call_dtm_ivr	datetime	-	The date/time on which the last call for this task was placed.
last_call_ivr_computer_name	char	50	The IVR COMPUTERNAME of the IVR server that made

			the most recent call for this task.
task_status	char	20	A ‘code’ indicating how much progress has been made on this callout task. The possible codes are: NEW No calls have been placed in an effort to complete this task. WORKING One or more calls have been attempted to complete this task, but it remains incomplete. (probably a busy or no answer result) ASSIGNED The task is currently assigned to a port to be called. (I.e. The call is either in progress, or about to start) DONE Whether it is finished because the task could not be completed (because of repeated unanswered/failed/errored calls), or because it was successful, this indicates the task is completed(I.e. The task will not be processed further). REMOVED The call is removed from calling out and will never complete. ERROR An error occurred the last time this task was attempted.
task_fail_count	int	2	A count of the total number of times this callout task has been tried and failed.
task_noanswer_count	int	2	A count of the total number of times this callout task has been tried and no one answered, including failures. (This also includes early hangups)
task_error_count	int	2	A count of the total number of times this callout task has been tried and resulted in an application error.
task_max_fail	int	2	The maximum number of time calls can fail before the system gives up on this task.
task_max_noanswer	int	2	The maximum number of time calls aren't answered (EG. BUSY, NO ANSWER, early HANGUP) before the system gives up on this task.
early_hangup_check_yn	char	1	Y/N: Check whether or not the call recipient hung up quickly? See also early_hangup_max_seconds and early_hangup_retry_call.
early_hangup_max_seconds	int	5	If early_hangup_check_yn = ‘Y’, this value is the maximum number of seconds to consider a quick hangup.
early_hangup_retry_call	char	1	If early_hangup_check_yn = ‘Y’, and the hang up was less than or equal to early_hangup_max_seconds, this field determines whether or not an early hangup results in a retry of the call-out. Whether retry is on or off, checking for early hangups marks all early hangups with HANGUP as the call result to indicate an early hangup was encountered.

machine_check_yn	char	1	Y/N: Check to see if the call is answered by an automated system or answering machine?
machine_min_length	int	5	The minimum voice length when a call is answered that is to be considered an automated system.
machine_run_app_yn	char	1	Y/N: Run the primary application when the call recipient has been determined to be an automated system?
callout_task_priority	int	10	A priority number from 0-99. Callout tasks are assigned to be done based on business rules, but when there are many tasks currently to be completed, they are sorted descending on this field prior to any other field. (The higher this number, the higher a priority the callout task receives) [default = 1]
callout_retry_seconds	int	6	The number of seconds between retry attempts for the callout task. [default = 1800 (30 minutes)]

client

Database:	TeleFlow Relay
Primary Key:	client_id
Foreign Keys:	None
Indexes:	client_status
Relationships:	One-to-Many with **callout_task** One-to-Many with **call_log_master** One-to-Many with **client_application_subscription** One-to-Many with **client_ip_access** One-to-Many with **dnis_service** One-to-Many with **event_log_master** One-to-Many with **user_access_log** One-to-Many with **user_master**

Special: Every client has a client database (defined in this table), in which all client-specific tables exist. Client therefore has a variety of additional relationships, which are not specifically defined as such, but are apparent within the context of the client database.

Represents a TeleFlow Relay client. Primarily, this table provides the means to access a client-specific database, wherein client data (including details about the client) are found.

Field Name	Type	Size	Description
client_id	int	10	Primary Key (Auto-Incremented, Unique)
client_name	varchar	50	The client name (Probably a client company name, like “Engenic”)
client_status	varchar	10	Indicates the client's activation status. ACTIVE: This is an active client. NOTE: Other codes not yet determined.
client_permissions_level	int	10	A number indicating what permissions the client has overall on the TeleFlow Relay host. NOTE: The levels and what they allow/disallow haven't been defined yet.
client_db_driver	varchar	50	The database driver to use when connecting to the client's database. NOTE: This is actually used in ODBC connection strings for the “DRIVER=” portion, and as such, must be perfect in order to work. Known “DRIVER=” values that work and the databases they connect to: SQL Native Client: SQL Server (including SQL Server Express) MySQL ODBC 3.51 Driver: MySQL
client_db_type	varchar	20	This field should be used to express a “type” of database connection, when it is important to know for what special connection rules. Currently, only “SQLEXPRESS” is required, because the ODBC connection string for SQL Express has a

			special condition that no other db connection has in order to connect. (That is that you have to specify hostname\SQLEXPRESS when specifying the database server ip/hostname)
client_db_host	varchar	50	Database server host address/ip address.
client_db_host_remote	varchar	50	For services remote to the database server (I.e. Not within the same IP network), this is the host address/ip address to use to reach the database server.
client_db_name	varchar	50	Client's database name.
client_db_user	varchar	20	Client's database user/user id.
client_db_password	varchar	20	Client's database password.
client_db_connect_options	varchar	50	Additional ODBC connect string options besides those covered by the previous fields. For connection options, you must provide the option/parameter and its value, and they must be strung together with semicolons as a separator. Some known/used connect options are: SQL Server specific options "MARS_Connection=yes": Stands for "multiple active results sets", indicating that you can have more than one database handle open at one time (I.e. You can use one database connection to run multiple nested queries.). "Trusted_Connection=yes": Using a trusted connection indicates the connection uses Windows Authentication. In order to take advantage of this, the program/service must be logged in as a windows user with the appropriate rights to do what the application needs to do with the database.
next_dst_scan_dtm_utc	datetime	-	Date/time (in UTC) the next daylight savings adjustment scan should be performed on the client. (Management applications update a client's current dst adjusted GMT offset at spring forward / fall back)
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.

client_application_subscription

Database:	TeleFlow Relay
Primary Key:	client_application_subscription_id
Foreign Keys:	client_id (client) app_type_name (application_type)
Indexes:	client_id
Relationships:	Many-to-one with **client** (client_id) Many-to-one with **application_type** (app_type_name)

Tracks client's subscriptions to applications/services, including when key data for a given application was last loaded, scanned, or when it is due to be checked again.

Field Name	**Type**	**Size**	**Description**
client_application_subscription_id	int	10	Primary Key (Auto-Incremented, Unique)
client_id	int	10	Foreign key: Links to client.client_id. Indicates the client subscribing to the service (the service is defined by app_type_name)
app_type_name	varchar	30	Foreign key: Links to application_type.app_type_name. Indicates the service subscribed to.
app_type_group_menu_order	int	10	The order this item falls within its application type group. This is used when displaying (primarily) applications a client has subscribed to on web menu screen. It allows all applications in one group (EG: MedicalRelay, CallCapture) to be displayed together, and in a specific order. This value is pulled from application_type by default, but can be ordered to suit the client in this table.
subscription_management_file	varchar	100	A file used to manage aspects of the subscription, such as a .def file that defines how dataloads are done as they relate to the client's service.
subscription_status	varchar	20	Code indicating the current status of this subscription. Valid codes are: ACTIVE: The subscription is currently active/ongoing. NOTE: Other status codes are TBD.
last_data_change_dtm_utc	datetime	-	The date/time of the last data change for the client/application that is cause for a scan to see if any call jobs need to be scheduled.
next_data_scan_dtm_utc	datetime	-	The next date/time to perform a client/application scan to see if any call jobs need to be scheduled. (When a job is initially loaded, checking for rows in this tables where last_data_scan_dtm_utc is less than last_data_change_dtm_utc is enough to indicate a scan is needed. Since some jobs can be scheduled across time periods where calling is/n't allowed, this field allows scheduler applications to set a next scan time for the

			next possible calling period rather than scanning constantly during blackout periods.
last_data_scan_dtm_utc	datetime	-	The date/time of the last scan of client/application to see if any call jobs need to be scheduled.
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.

client_ip_access

Database: TeleFlow Relay
Primary Key: cia_id
Foreign Keys: client_id
Indexes: cia_ip_range (compound: cia_ip_address_range_start, cia_ip_address_range_end)
Relationships: Many-to-one with **client** (client_id)

Provides automatic client link (I.e Users are assumed to belong to a certain TFRelay client) for users reaching TFRelay website(s) from defined IP ranges. In some cases(determined on a case-by-case or app-by-app basis), this might also be used as a means to lock down who can access web services at all.

Field Name	**Type**	**Size**	**Description**
cia_id	int	3	Primary Key (Auto-Incremented, Unique)
client_id	int	10	Foreign key: Links to client.client_id. Indicates which client to assume is logging in when the remote ip address is in the range.
cia_status	varchar	10	Status code indicating current ip access availability for the range/client. Valid codes are: ACTIVE NOTE: Other codes are TBD.
cia_ip_address_range_start	varchar	20	Start of range for the IP address(es). Users coming from IP addresses in the range are assumed to be the client associated with client_id, and are allowed access. For a specific/individual IP address, set the range_start and range_end fields to the same value.
cia_ip_address_range_end	varchar	20	End of range for the IP address. For a specific/individual IP address, set the range_start and range_end fields to the same value.
cia_ip_address_description	varchar	200	Simple description of what the access is for/what it covers.
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.

client_details (client)

Database: Client
Primary Key: client_id
Foreign Keys: None
Indexes: None
Relationships: None (This is an oversimplification of sorts; In the client-level database, where this table is found, all tables are assumed to belong to the client. The client_id in this table matches that at the TeleFlow Relay top-level database table “client”. All relationships between the client and other top-level entities essentially also exist with this table.)

One row table with details about / settings for the client.

Field Name	Type	Size	Description
client_id	int	10	Unique PK, copied from teleflow_relay master database client.client_id field when the client was created. This table is a one-row table in the client database. Technically, this PK isn't necessary, but it is used as a safety feature.
client_name	char	50	The client name (Probably a client company name, like “Engenic”)
primary_contact_u_id	int	10	Foreign key: Links to user_details.u_id. Indicates the user who is the primary contact for the client.
client_utc_offset	decimal	4,1	The client’s current UTC offset, used to determine the local time at the client’s location. NOTE: This value is adjusted for DST if client_observes_dst_yn = 'Y'.
client_timezone	varchar	4	The client's timezone, expressed as an abbreviation, per the teleflow_relay top-level table, “timezone”. (Field: timezone_abbreviation) This is used for display purposes, and for setting client_utc_offset.
client_observes_dst_yn	char	1	Indicates whether or not daylight savings time is observed for this client. This is used along with client_utc_offset when calculating local client time for the client.
client_send_callerid	varchar	20	Caller ID to send with each call placed on behalf of this client.
client_transfer_status	varchar	10	ACTIVE/DISABLED: Indicates whether or not the client is willing to receive call transfers from any applications on the system (individual applications can be configured as well).
client_transfer_phone	varchar	20	Default phone number applications should use when transferring calls to this client. (this default is used to populate application-specific settings tables)
client_transfer_ext	varchar	10	The default extension number (if any) to dial when transferring to client_transfer_phone when the remote party answers.
client_transfer_ext_pause	int	10	The default wait in milliseconds between the remote

			party answer and dialing the client_transfer_ext
client_phone	varchar	20	Client's primary contact phone number.
client_fax	varchar	20	Client's primary contact fax number.
client_email_address	varchar	50	Client's primary contact email address.
client_website	varchar	255	Client's email address.
client_address	varchar	40	Client's business/company street address.
client_city	varchar	20	City client resides in.
client_state	varchar	20	State client resides in.
client_zip	varchar	15	Zip code for client's primary address.
client_country	varchar	15	Country client resides in.
client_type	varchar	30	A code indicating the type of client, which may affect how certain application functions work. Valid codes are: DEMO: This client is used for demonstration purposes only. LIVE: A real client, with applications in production. NOTE: Other codes are TBD.
client_package	varchar	15	Package the Client is subscribed to, valid values are “free”, “trial” and “full”. If empty, this indicates the Client has not filled in enough information to select a package yet.
client_promo_code	varchar	20	Promotional code the client signed up with.
client_cc_name	varchar	40	Client's name as it appears on the client's credit card.
client_bill_address1	varchar	255	Client's billing address.
client_bill_address2	varchar	255	Additional info field for client's billing address.
client_bill_city	varchar	30	City for client's billing address.
client_bill_state	varchar	30	State for client's billing address.
client_bill_country	varchar	30	Country for client's billing address.
client_bill_zip	varchar	10	Zip code for client's billing address.
client_bill_phone	varchar	20	Phone number to contact client at for billing inquiries.
client_cc_type	varchar	20	Credit card type for the credit card to bill for client services. Eg. VISA.
client_cc_verify_code	int	10	Credit card verification code.

client_cc_num	varchar	100	Credit card number. (encrypted)
client_cc_expiry_month	int	10	Credit card expiry month, expressed as a number. (I.e 1=January; 2=February; etc)
client_cc_expiry_year	int	10	Credit card expiry year.
client_last_billing	varchar	20	
client_referrer	varchar	30	Contains codes to answer the question: “How did you hear about [MedicalRelay]?”
client_enews	varchar	10	Is client is subscribed to e-news letter? (Y/N)
client_startdate	varchar	10	Date that the clients “free” or “trial” account started (see client_package field). Note that this is not the date the client signed up. This field is not populated until the client makes at least one call or test call.
triallength	varchar	10	Number of days in trial period.
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.
create_dtm_client	datetime	-	Row create date/time, in client's local time.
update_dtm_client	datetime	-	Last date/time the row was updated, in client's local time.

custom_message (client)

Database: Client
Primary Key: custom_msg_id
Foreign Keys: app_type_name (teleflow_relay.application_type)
Indexes: app_type (compound: app_type_name, app_custom_type)
Relationships: Many-to-One with application_type (app_type_name).
Many-to-One for an application_type specific table.
(app_custom_type is a link to a field in another table. This table varies based on the application, but the field's purpose remains as follows: To identify the application's custom message type)

Contains custom message details. Custom messages are those played based on a combination of business rules and data from external sources imported into this system.

CustomMessages.doc is an additional document detailing the possible codes in this table.

Field Name	**Type**	**Size**	**Description**
custom_msg_id	int	10	Primary Key (Auto-Incremented, Unique)
app_type_name	varchar	30	Foreign key: Links to application_type.app_type_name (in tfrelay top-level database). Indicates the application the custom message is for.
custom_msg_status	varchar	10	Status code indicating whether or not the custom message part is currently available for use. ACTIVE: Available custom message code. DELETED: Custom message – or at least this segment of it – has been deleted, and should not be offered to users.
app_custom_type	varchar	100	The application's custom message type. Likely relates to a specific field in the application's primary table. (The applications dependent on these tables will have a proper link between these fields)
cust_type_short_display	varchar	30	Display for custom message: Used when displaying the message type for selection.
cust_type_description	varchar	100	A description of the custom message.
custom_msg_app_num	int	10	Indicates which custom message number this is for the application it is used for.
custom_msg_menu_order	int	10	When the custom message is available for selection by a user, this number is used to prioritize the order in which custom message options are offered to the user.
custom_msg_order_num	int	10	Indicates which part of the custom message number this is. (If playing custom_msg_app_num 1, you might play custom_msg_order_num 1, 2, and then 3 for it to complete the entire custom message)
custom_msg_code	varchar	50	A "code" indicating what the application playing the custom message should do. See CustomMessages.doc for a complete list of valid codes.
custom_msg_data	varchar	50	Any data the application playing the custom message might

			need in order to complete the message. The data relates to the nature of the custom_msg_code.
custom_msg_script	varchar	255	The script for the piece of the custom message. If one of the custom message pieces is missing its data, such as a file that couldn't be played, this script can be played using text-to-speech.
multi_play_yn	char	1	Y/N: If "N", don't play this message when multiple messages are played in the same call.
multi_play_alternate_id	int	10	An alternate custom_msg_id for a different message to play when playing the second or later messages within a single call.
play_once_only	char	1	
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.
create_dtm_client	datetime	-	Row create date/time, in client's local time.
update_dtm_client	datetime	-	Last date/time the row was updated, in client's local time.

dnis_service

Database:	TeleFlow Relay
Primary Key:	dnis_service_id
Foreign Keys:	app_type_name (application_type) client_id (client)
Indexes:	service_dnis client_id
Relationships:	Many-to-One with **application_type** (app_type_name) Optional Many-to-One with **client** (client_id)

The service (application) to provide (run) for a DNIS (and optionally on behalf of which client) when a call is answered, and some rules/settings for how that service is provided.

Field Name	**Type**	**Size**	**Description**
dnis_service_id	int	10	Primary Key (Auto-Incremented, Unique)
app_type_name	varchar	30	Foreign key: Links to application_type.app_type_name. Indicates the application to run when the dnis is called. Special: In this table (only), app_type_name can be set to "AVAILABLE", indicating that the service_dnis is available for use by an application. (I.e. The service_dnis can be assigned to a service)
client_id	int	10	Foreign key: Indicates the client the application is run on behalf of (if any).
service_dnis	varchar	30	The DNIS the circuit provides for calls to the service_phone_number(s).
internal_service_dnis	varchar	30	As service_dnis, but for special "shortcut" dnis for internal systems. (When using SIP, for example, you can set up a main 10-digit DNIS for external calls coming in through the PSTN, but callers on internal SIP extensions can call in with a 4-digit DNIS for convenience)
service_phone_number	varchar	30	The phone number callers dial to reach the service. (For an 800 number, the service_phone_number and the service_dnis are usually different, as a local number is typically substituted when the 800 number is called).
service_notes	varchar	200	A simple free-form notes field for the dnis-service. (Eg. To note things like "this is used for annual service X, but can be used for testing the rest of the year")
home_ivr_computer_name	varchar	50	The name of the computer where the dnis normally terminates, and where it terminates first. (I.e. Indicates the computer hosting the circuit that is the first circuit that calls with the DNIS go to, prior to rollover)
circuit_id	varchar	30	The circuit id of the circuit where the dnis normally terminates, and where it terminates first. (see home_ivr_computer_name)
dnis_status	varchar	10	A status code indicating whether or not the service is currently available. Valid codes are:

			ACTIVE: The service is active/available. INACTIVE: The service is not currently available.
tel_connect_delay	int	4	A deliberate pause after answering an inbound call, before proceeding to run the appropriate application, expressed in milliseconds. This pause exists to account for the Telco's delay completing the connection from caller to IVR.
db_pre_connect_yn	char	1	Y/N: Connect to the client's database prior to running the application associated with the DNIS?
db_pre_connect_handle	varchar	50	When dp_pre_connect_yn = 'Y', this is the database handle name TeleFlowRelay uses when connecting to the client database. (If left blank, the handle to use in the client application will be blank, as well)
max_service_instances	int	10	The maximum number of simultaneous calls the system should permit for this service. NOTE: This field is not presently supported. [default: -1]
srtt_type	varchar	5	Code indicating whether touch tone interaction or speech recognition interaction (or both) is required in the application to run. When speech recognition is required, the top-level TFRelay application can set up barge-in appropriately before starting the call (I.e before running the application that processes the call). Valid codes are: TT: Touch-tone only. SR: Speech recognition only. SRTT: Mixed touch-tone and speech recognition mode.
srtt_engine	varchar	30	The speech recognition engine to use when use SR or SRTT modes. Valid codes are: LUMENVOX NUANCE This setting really only does some of the most basic setup for an SR call. Installation, configuration, grammars, etc must all be set up for SR to work.
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/ server time.

event_log (client)

Database: Client
Primary Key: none
Foreign Keys: logging_app (application_type.app_type_name)
Indexes: evt_date
evt_time
Relationships: Many-to-One with **application_type** (logging_app)

Logs application events or section accesses, such as when a call is answered, or a user logs in. NOTE: This table is only currently in use for one client, at the client-level only. If we put this table to further use, I would prefer the client version of the table copy the field formats of the TeleFlow Relay top-level database version (see below, event_log_master), instead of this "inferior" version.

Field Name	Type	Size	Description
event_log_id	int	10	Primary Key (Auto-Incremented, Unique)
evt_date	varchar	10	Date the event occurred, expressed in local client time.
evt_time	varchar	8	Time on evt_date the event occurred, expressed in local client time.
evt_computer_name	varchar	50	The name of the computer the event occurred on.
evt_port	varchar	3	The port number (for IVR applications) the event occurred on.
log_format_version	varchar	3	TeleFlow log format version.
evt_code	varchar	30	An application-specific event code indicating what the event was.
evt_start	int	10	For IVR applications: Indicates the time the call the event is from started, expressed as seconds since midnight. (This value can be used to "group" one call's events together)
call_duration	int	10	For IVR applications: This indicates how far into the call (expressed as the number of seconds since the call started) the event occurred.
logging_app	varchar	30	Foreign Key: Links to application_type.app_type_name. Indicates the application that generated the event.
site_code	varchar	30	A code indicating the site the event was for. (A site in this context could be any location, and is really based on the application/client involved)
region_code	varchar	30	A code indicating the region the event was for. (A region in this context could be any location, and is really based on the application/client involved)
evt_type	varchar	30	A code indicating the type of event. Event types are specific to the application and client logging the event.
evt_data1	varchar	30	Free-form field #1 for some extra/special data. This data must usually be put into some form of context, often requiring knowledge of the evt_code, the client, and the application involved.

evt_data2	varchar	30	Free-form field #2 for some extra/special data. This data must usually be put into some form of context, often requiring knowledge of the evt_code, the client, and the application involved.
evt_data3	varchar	30	Free-form field #3 for some extra/special data. This data must usually be put into some form of context, often requiring knowledge of the evt_code, the client, and the application involved.
evt_timestamp	int	10	A Unix timestamp at the time the event occurred.

event_log_master

Database: TeleFlow Relay
Primary Key: event_log_id
Foreign Keys: client_id (client)
logging_app (application_type.app_type_name)
Indexes: evt_dtm_host
Relationships: Many-to-One with **client** (client_id)
Many-to-One with application_type (logging_app)

Logs application events or section accesses, such as when a call is answered, or a user logs in. NOTE: This table is only currently in use for one client, at the client-level only. This table has been reserved for future use. If it is put to use, I would prefer the client version of the table copy the field formats of this one, instead of the “inferior” version currently in place, above.

Field Name	**Type**	**Size**	**Description**
event_log_id	int	10	Primary Key (Auto-Incremented, Unique)
client_id	int	10	Foreign Key: Links to client.client_id. Indicates the client the event relates to.
evt_dtm_utc	datetime	-	Date/time the event occurred, expressed in UTC.
evt_dtm_host	datetime	-	Date/time the event occurred, expressed in local database host/server time.
evt_computer_name	varchar	50	The name of the computer the event occurred on.
evt_port	varchar	3	The port number (for IVR applications) the event occurred on.
log_format_version	varchar	3	TeleFlow log format version.
evt_code	varchar	30	An application-specific event code indicating what the event was.
evt_start	int	10	For IVR applications: Indicates the time the call the event is from started, expressed as seconds since midnight. (This value can be used to “group” one call's events together)
call_duration	int	10	For IVR applications: This indicates how far into the call (expressed as the number of seconds since the call started) the event occurred.
logging_app	varchar	30	Foreign Key: Links to application_type.app_type_name. Indicates the application that generated the event.
site_code	varchar	30	A code indicating the site the event was for. (A site in this context could be any location, and is really based on the application/client involved)
region_code	varchar	30	A code indicating the region the event was for. (A region in this context could be any location, and is really based on the application/client involved)
evt_type	varchar	30	A code indicating the type of event. Event types are specific to the application and client logging the event.

evt_data1	varchar	30	Free-form field #1 for some extra/special data. This data must usually be put into some form of context, often requiring knowledge of the evt_code, the client, and the application involved.
evt_data2	varchar	30	Free-form field #2 for some extra/special data. This data must usually be put into some form of context, often requiring knowledge of the evt_code, the client, and the application involved.
evt_data3	varchar	30	Free-form field #3 for some extra/special data. This data must usually be put into some form of context, often requiring knowledge of the evt_code, the client, and the application involved.
evt_timestamp	int	10	A Unix timestamp at the time the event occurred.

file_application_transfer (client)

Database: Client
Primary Key: fat_id
Foreign Keys: fat_app_type_name
Indexes: none
Relationships: One-to-one with **media_file_transfer**
Many-to-One with **application_type** (fat_app_type_name)

File application conversion and transfer processing.

Field Name	Type	Size	Description
fat_id	int	10	Primary Key (Auto-Incremented, Unique)
fat_app_type_name	varchar	50	Foreign Key: Links to teleflow_relay.application_type.app_type_name. Indicates the application the file application transfer is for.
fat_status	varchar	10	A status code indicating whether or not this application's file transfers are active or not. Valid codes are: ACTIVE: Process file transfers for the application. INACTIVE: Processing disabled.
fat_wav_type	varchar	5	YES/NO: Create/convert files to WAV prior to file transfer?
fat_gsm_type	varchar	5	YES/NO: Create/convert files to GSM prior to file transfer?
fat_mp3_type	varchar	5	YES/NO: Create/convert files to mp3 prior to file transfer?
fat_dest_type	varchar	30	Type of system to transfer files to. Valid codes are presently: TELEFLOWRELAY: A TeleFlowRelay host system. TRANSCRIPTION: A transcription service.
fat_dest_host_name	varchar	100	The destination system ip address/URL/name to connect to when transferring files. Eg. ftp.engenictfrelay.com.
fat_dest_ssl_type	varchar	5	YES/NO: Use SSL when connecting to / sending files to the destination?
fat_dest_user_name	varchar	30	Username to use when connecting to the destination system.
fat_dest_user_pass	varchar	20	Password to use when connecting to the destination system.
fat_dest_folder	varchar	250	Folder/directory to place transferred files on the destination system.
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_client	datetime	-	Row create date/time, in client's local time.
update_dtm_client	datetime	-	Last date/time the row was updated, in client's local time.

guest_active (client)

Database:	Client
Primary Key:	none
Foreign Keys:	none
Indexes:	none
Relationships:	none

Used for guest connections to web services.

Field Name	Type	Size	Description
g_id	varchar	15	
g_timestamp	bigint		

ivrconfig

Database:	TeleFlow Relay
Primary Key:	ivrconfig_id
Foreign Keys:	None
Indexes:	None
Relationships:	None

Configuration table for the IVR, containing all soft settings of the system.

Field Name	Type	Size	Description
ivrconfig_id	int	10	Primary Key (Auto-Incremented, Unique)
ivr_computer_name	varchar	50	IVR computer name the soft-settings apply to.
callout_mgr_computer_name	varchar	50	Computer name of the computer that will manage this IVR's callouts. Only one manager per TFRelay database at one time should be allowed to run. (to avoid having multiple managers attempt to handle a given task)
cpu_wait	int	6	Number of milliseconds to wait between series of queries for applications that poll database(s) for jobs to do.
local_utc_offset	int	3	The current, DST adjusted UTC offset for the IVR system/server.
local_timezone	varchar	4	The IVR's local timezone, expressed as an abbreviation, per the “timezone” table. (Field: timezone_abbreviation) This is used for display purposes, and for setting local_utc_offset.
location_observes_dst_yn	char	1	Y/N: Does this location observe daylight savings time?
tts_server_ip_primary	varchar	100	IP Address/hostname of the primary text-to-speech server the IVR is to use.
tts_ip_port_primary	varchar	10	IP port of the primary text-to-speech server the IVR is to use.
tts_server_ip_secondary	varchar	100	IP Address/hostname of the secondary/backup text-to-speech server the IVR is to use. (When the IVR tries to use the primary TTS server, it then tries the same utterance again using the backup/secondary server)
tts_ip_port_secondary	varchar	10	IP port of the secondary/backup text-to-speech server the IVR is to use.
voip_media_ip_address	varchar	100	For VoIP/SIP communication: The IP address VoIP media transmits across.
voip_media_ip_min_port	varchar	10	For VoIP/SIP communication:The base IP port for VoIP media transmission over IP. Each TeleFlow application instance uses this value + @PORT to set the ip port to use for media transmission.
email_alert_list	varchar	100	List of email addresses to send alarms/alerts to if/when the IVR recognizes an important system error that requires immediate intervention.

email_warn_list	varchar	100	List of email addresses to send alarms/alerts to if/when the IVR recognizes a system error that requires intervention either in the near future, or that could escalate to an issue requiring immediate intervention in a short while.
default_transfer_type	varchar	15	A system-wide default to use when performing call transfers on the IVR. This is used only if the call transfer application is run without specifying the transfer type to use. Valid values are: SUPER_SWITCH: Common transfer type for digital and SIP extensions. One application with a caller on the line requests that an outbound line call a remote party. When the remote party answers, the two are switched together, and remain on the two lines. If the remote doesn't answer, the requesting application can put the original caller back in the application, or provide feedback to the caller. BLIND_SWITCH: Requires testing. May not be properly supported. PBX_BLIND: Performs a flash-hook, dials the digits to reach the remote party, and hangs up. (The original caller should then hear all the call activity, such as rings, busy, or a remote party answering.) PBX_SUPER: Uses a supervised flash-hook transfer, wherein a failed call brings the original caller back on the line and can continue with IVR functionality. 2B_TRANSFER: Only available on ISDN, with NI2 protocol variant. One application with a caller on the line requests that an outbound line call a remote party. When the remote party answers, the two are switched together, and the IVR lines are freed up. If the remote doesn't answer, the requesting application can put the original caller back in the application, or provide feedback to the caller.
local_area_codes	varchar	20	This is a list of area codes that could be local area codes from the IVR. All area codes outside of this list are automatically assumed to be long distance. Dialer applications automatically check to see if a number dialed is local or long distance(using the area_exchange table and web service lookup), and add the appropriate digits to the beginning of local or long distance dial strings.
local_caller_id	varchar	20	The IVR server's primary trunk's caller id. Used as part of local/long distance lookup. (see “local_area_codes” field description, above)
pre_local	varchar	5	Digits to dial preceding a local call, if any. Eg. “9,1”. Although this is a text field it is to only store digits, commas, periods and the minus sign.
pre_long	varchar	5	Digits to dial preceding a long distance call, if any.

			Eg: “9,1”. Although this is a text field it is to only store digits, commas, periods and the minus sign.
ring_count	int	2	Number of rings to use for Place Call steps before timing out.
isdn_calling_number_plan	varchar	20	When performing outbound calls on an ISDN circuit, this is the ISDN calling number plan to use. In many locations, DEFAULT works fine, but where they are required, only setting this according to the circuit/telco setting will work. Valid values are: DEFAULT ISDN TELEPHONY PRIVATE UNKNOWN
isdn_calling_number_type	varchar	20	When performing outbound calls on an ISDN circuit, this is the ISDN calling number type to use. In many locations, DEFAULT works fine, but where they are required, only setting this according to the circuit/telco setting will work. Valid values are: DEFAULT NATIONAL LOCAL INTERNATIONAL
isdn_called_number_plan	varchar	20	When performing outbound calls on an ISDN circuit, this is the ISDN called number plan to use. In many locations, DEFAULT works fine, but where they are required, only setting this according to the circuit/telco setting will work. Valid values are: DEFAULT ISDN TELEPHONY PRIVATE UNKNOWN
isdn_called_number_type	varchar	20	When performing outbound calls on an ISDN circuit, this is the ISDN called number type to use. In many locations, DEFAULT works fine, but where they are required, only setting this according to the circuit/telco setting will work. Valid values are: DEFAULT NATIONAL LOCAL INTERNATIONAL
callout_silence_timeout	int	4	**To be removed (deprecated):** Should be set at the application/client level now. The silence timeout (ms) to use for the Place Call step for this application; required because some applications are more or less sensitive to how long it takes to recognize we have someone on the phone.

max_callout_errors	int	2	**To be removed (deprecated):** Should be set at the application/client level now. The maximum number of errors on a callout before the callout task is given up on.
call_wait	int	3	Number of seconds to wait after a HANGUP event before attempting the next PLACE CALL.
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.

media_file (client)

Database: Client
Primary Key: media_file_id
Foreign Keys: media_owner_u_id (user_details.u_id)
app_type_name (teleflow_relay.application_type)
Indexes: file_transfer_status
file_app_status
media_file_name
Relationships: Many-to-one with **user_details** (media_owner_u_id)
Many-to-one with **application_type** (app_type_name)

Any media created on, by, or for TeleFlow Relay applications is stored and processed through this table.

NOTE: Any table with a one-to-one relationship with this table (which is almost expected, given that each application has its own application-specific data it attaches to a file) will have media_file_id as a foreign key.

Field Name	**Type**	**Size**	**Description**
media_file_id	int	10	Primary Key (Auto-Incremented, Unique)
media_file_directory	varchar	255	The volume or date subdirectory the media file resides in. (NOTE: The subdirectory is a subdirectory to the standard TFRelay media root directory for the client)
media_file_name	varchar	255	Media file's filename, including the extension.
media_owner_u_id	int	10	Foreign key: Links to user_details.u_id. Indicates the user to which the file belongs, or the user that created it (or both).
file_reference_id	int	10	A unique confirmation or reference id/number that can be provided to users/callers, which they can use to locate the recording.
app_type_name	varchar	30	Foreign key. Links to application_type.app_type_name. Indicates the application that created/generated the media file.
media_file_type	varchar	30	Indicates the nature of the media (I.e. Describes what created the media or where it came from). Valid types include: RECORDING: A “typical” voice audio recording, wherein a caller was prompted to record a voice message. Types from CallCapture recordings: CALL: A call recording. CONTINUOUS: A continuous recording is one that is from a line that records continuously, only stopping to periodically end one recording before starting a new one. RADIO: A “radio call” recording. Radio calls start when activity (I.e. noise) is detected on the line, and end when enough contiguous silence is detected on the line. FAX: A fax recording.

ivr_computer_name	varchar	50	IVR computer name on which the source/original media file was created.
ivr_port	int	3	IVR/telephony port on ivr_computer_name on which the the source/original media file was created.
file_create_start_dtm_utc	datetime	-	The datetime the file started being written, in UTC.
file_create_end_dtm_utc	datetime	-	The datetime the file finished being written, in UTC.
file_create_start_dtm_client	datetime	-	The datetime the file started being written, in client's local time.
file_create_end_dtm_client	datetime	-	The datetime the file finished being written, in client's local time.
file_duration	int	5	Approximate duration of the file (if audio/video), expressed in seconds. Note that silence-trimmed recordings may differ from this value by several seconds.
file_size	int	10	Size of file on disk, expressed in bytes.
file_app_status	varchar	15	A code for the media file's application status. Indicates where in the application's processes the media file falls (Eg. The file is NEW, or the file has been SAVED by the user, etc) Valid codes/status values are: CREATED: Media entry created, in advance of file creation. NEW: File available to users. SAVED: File saved by user. DELETED: File is marked for deletion. (user delete) ABANDONED: File creation abandoned. (Eg. In a dictation system, this would indicate the caller hung up without saving the file.) PURGED: File has been purged/deleted from disk. REPLACED: Media transfer was run a second time.
app_status_update_dtm_utc	datetime	-	The last datetime file_app_status changed, in UTC.
app_status_update_dtm_client	datetime	-	The last datetime file_app_status changed, in the client's local time.
file_transfer_status	varchar	15	The file's status in the transfer process. Valid codes/status values are: NEW: File has not been transferred anywhere. PROCESSING: File is currently in the process of being transferred to one or more destination locations. COMPLETED: All file transfers required for the file have been completed.
file_note	varchar	255	Notes on/about the file. Typically, these notes are made by the/a user.

<table>
<tr><th></th><th></th><th></th><th></th></tr>
<tr><td>call_direction</td><td>varchar</td><td>10</td><td>IN/OUT/NONE/UNKNOWN: Indicates whether the call (if any) in which the media file was created was an inbound or outbound call.</td></tr>
<tr><td>called_number</td><td>varchar</td><td>20</td><td>The phone number called to initiate the call in which the media file was created. For inbound calls, this indicates the DNIS, or the number dialed to reach the IVR. For outbound calls, this indicates the number the IVR called to reach an contact someone externally/an external system.</td></tr>
<tr><td>calling_number</td><td>varchar</td><td>20</td><td>The phone number of the initiating party of the call in which the media file was created. For inbound calls, this is the caller's caller id. For outbound calls, this is the number the system attempts to send as its caller id.</td></tr>
<tr><td>recording_start_trigger</td><td>varchar</td><td>40</td><td>For recordings, this indicates what event (if any is being recorded) that triggered the recording start.

This primarily exists for CallCapture, although other applications could use it to indicate why/how the recording was started.

For CallCapture: Event (from telephony card) that triggered the start of the recording</td></tr>
<tr><td>recording_stop_trigger</td><td>varchar</td><td>40</td><td>For recordings, this indicates what event (if any is being recorded) that triggered the end of the recording.

This primarily exists for CallCapture, although other applications could use it to indicate why/how the recording completed (Eg. HANGUP).

For CallCapture: Event (from telephony card) that triggered the end of the recording</td></tr>
<tr><td>recording_protocol</td><td>varchar</td><td>10</td><td>Indicates the “protocol“ of the channel that received the call.

This primarily exists for CallCapture(which has its own definition of “protocol” in this respect), but it could theoretically be used by any application to indicate the channel’s telephony protocol, such as “T1” or “Wink”.

The possible values for CallCapture are:

<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>Loop</td><td>Loop start.</td></tr>
<tr><td>ISDN</td><td>ISDN.</td></tr>
<tr><td>Wink</td><td>Wink start.</td></tr>
<tr><td>Radio</td><td>Radio recording: Could be recording ISDN/Loop, etc, but records based on activity/silence detected on the line.</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td><table><tr><td>Continuous</td><td>Could be recording ISDN/Loop, etc, but records continuously, regardless of activity on the line/channel.</td></tr><tr><td>NGX</td><td>NGX digital station set recording.</td></tr></table></td></tr>
<tr><td>cc_agent_id</td><td>int</td><td>10</td><td>CallCapture-specific field: Agent that took the call, foreign key: links to Agent.agent_id.
(Not currently supported.)</td></tr>
<tr><td>cc_extension</td><td>varchar</td><td>10</td><td>CallCapture-specific field: The physical station/phone extension number associated with the port. (at the date/ time of the call)</td></tr>
<tr><td>error_text</td><td>varchar</td><td>150</td><td>Textual description of an error, if file_transfer_status = ERROR.</td></tr>
<tr><td>create_dtm_utc</td><td>datetime</td><td>-</td><td>Row create date/time in UTC.</td></tr>
<tr><td>update_dtm_utc</td><td>datetime</td><td>-</td><td>Last date/time the row was updated in UTC.</td></tr>
<tr><td>create_dtm_client</td><td>datetime</td><td>-</td><td>Row create date/time, in client's local time.</td></tr>
<tr><td>update_dtm_client</td><td>datetime</td><td>-</td><td>Last date/time the row was updated, in client's local time.</td></tr>
</table>

media_file_transfer (client)

Database: Client
Primary Key: mft_id
Foreign Keys: fat_id (file_application_transfer)
media_file_id (media_file)
Indexes: mft_status
Relationships: One-to-one with **file_application_transfer** (fat_id)
Many-to-one with **media_file** (fat_id)

Media file transfers are processed through this table.

Field Name	Type	Size	Description
mft_id	int	10	Primary Key (Auto-Incremented, Unique)
fat_id	int	10	Foreign key: Links to file_application_transfer.fat_id. Indicates the type of file application transfer performed (or to perform) for the media file.
media_file_id	int	10	Foreign key: Links to media_file.media_file_id. Indicates the media file the transfer record is for.
mft_status	varchar	15	Transfer status, indicates where in the process of moving media from one location to another this media transfer is.
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_client	datetime	-	Row create date/time, in client's local time.
update_dtm_client	datetime	-	Last date/time the row was updated, in client's local time.

port_task

Database: TeleFlow Relay
Primary Key: port_task_id
Foreign Keys: callout_task_id (callout_task)
Indexes: callout_task_id
port_number
task_status
Relationships: Optional One-to-One with **callout_task**. (callout_task_id)

This table indicates what each port/channel in the IVR server is currently doing, and is used to assign tasks to currently idle ports that are waiting for a task. New callout tasks are assigned to ports through this table.

Field Name	**Type**	**Size**	**Description**
port_task_id	int	10	Primary Key (Auto-Incremented, Unique)
callout_task_id	int	10	Foreign key: callout_task.callout_task_id.
ivr_computer_name	varchar	50	IVR computer name the port_task entry is for.
port_number	int	5	The physical/IVR port number to assign a task to.
port_phone_number	varchar	20	This indicates what phone number belongs to this physical port (typically, this applies only on analog).
port_extension_number	varchar	10	This indicates what extension number belongs to this physical port (typically, this applies only on analog).
port_direction	varchar	5	IN/OUT/NA: Indicates whether the port is being used for inbound or outbound call duty. (Primarily, this exists so that if a TAL file is modified, the inbound ports can take themselves out of outbound duty, so they aren't assigned tasks they will never complete.) This will always be “NA” (Not Applicable) for CallCapture, as the direction of the call may vary for the port, from one call to the next.

<table>
<tr><td>task_status</td><td>varchar</td><td>15</td><td>The port’s task status, indicating whether or not the port can receive a new task. Possible values include:
<table>
<tr><th>Code</th><th>Description</th></tr>
<tr><td>INACTIVE</td><td>The port has been de-activated, and should not be used.</td></tr>
<tr><td>READY</td><td>Port is ready to receive a new callout task</td></tr>
<tr><td>ASSIGNED</td><td>The port has been assigned the callout task indicated by callout_task_id</td></tr>
<tr><td>IN_PROGRESS</td><td>The port is working on the callout_task that was ASSIGNED</td></tr>
<tr><td>IDLE</td><td>The port is an inbound port, and is currently idle (I.e. It is waiting for a call to come in).
CallCapture code: There is currently no call in progress.</td></tr>
<tr><td>IN_CALL</td><td>The port is an inbound port, and is currently in a call.</td></tr>
<tr><td>LIVE</td><td>CallCapture code: An active call is being tracked on the line right now. (But no recording is taking place)</td></tr>
<tr><td>RECORDING</td><td>CallCapture code: An active call is being recorded on the line right now.</td></tr>
<tr><td>INBOUND</td><td>CallCapture code: A call is coming in on the port.</td></tr>
<tr><td>OUTBOUND</td><td>CallCapture code: A call is being placed on the port.</td></tr>
</table>
</td></tr>
<tr><td>cc_recording_protocol</td><td>varchar</td><td>10</td><td>CallCapture-specific port setting: The protocol that determines the behavior of CallCapture for the port. (Determines what events to watch for to determine when and why to record)
<table>
<tr><td>Valid Protocol</td><td>Description</td></tr>
<tr><td>ISDN</td><td>ISDN, T1/E1 protocol</td></tr>
<tr><td>Loop</td><td>Analog loop start protocol</td></tr>
<tr><td>Radio</td><td>Radio recording: records on activity/silence events.</td></tr>
<tr><td>Continuous</td><td>Records continuously, regardless of events. Essentially, this records to a single file for a set amount of time (Eg. An hour), then starts a new recording immediately.</td></tr>
<tr><td>Wink</td><td>Wink (CAS), T1 protocol</td></tr>
<tr><td>Test</td><td>Test for all possible event, regardless of the actual telephony protocol in place.</td></tr>
<tr><td>NGX</td><td>Digital station side recording. When the cc_recording_protocol = NGX, cc_recording_custom_tam will be set to a</td></tr>
</table>
</td></tr>
</table>

			custom .TAM file that works with the particular location's PBX/station set configuration.
cc_recording_custom_tam	varchar	100	CallCapture-specific port setting: Indicates a custom .TAM file to use for recording. Used when cc_recording_protocol = NGX.
cc_recording_rule_code	varchar	10	CallCapture-specific port setting: Determines if recording should currently be on for the port. Valid values: ON: Record all calls. OFF: No recording on this port at all. SCHEDULE: Record according to a schedule. (see cc_schedule_id, below) NOTE: Scheduled recording is not currently supported.
cc_schedule_id	int	10	Foreign Key: Links to cc_schedule.cc_schedule_id. CallCapture-specific port setting: Indicates the CallCapture recording schedule to use on the port, if any. Not currently supported.
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/ server time.

registered_sip_uri

Database:	TeleFlow Relay
Primary Key:	registered_sip_uri_id
Foreign Keys:	none
Indexes:	none
Relationships:	none

Used in conjunction with TeleFlow's “Register SIP URI” step to register SIP URI's with the SIP Registrar.

About SIP/SIP Registrars: A SIP registrar is a server in an Session Initiation Protocol (SIP) network that accepts and processes SIP REGISTER requests. The SIP registrar provides a location service which registers one or more IP addresses to a certain SIP URI, indicated by the sip: scheme.

NOTE: While this table might seem application-specific (and therefore belonging at the client level), it controls how calls are processed in a SIP network, and as such, affects an entire TeleFlow Relay server or servers that use/depend on SIP for communications. As such, all registrations must be handled at the TFRelay top-level.

Field Name	**Type**	**Size**	**Description**
registered_sip_uri_id	int	10	Primary Key (Auto-Incremented, Unique)
sip_registrar	varchar	50	The “HOST:IP Port” of the SIP Registrar to register a URI with.
uri_register_to	varchar	50	
uri_register_from	varchar	50	
contact_uri	varchar	50	
uri_expiry_seconds	int	10	The number of seconds the registrar should keep a SIP URI registered. When this time expires, the registration expires, invalidating the URI.

sales_affiliate

Database: TeleFlow Relay
Primary Key: sales_id
Foreign Keys: none
Indexes: none
Relationships: none

Sales affiliates in the affiliate network.

Field Name	Type	Size	Description
sales_id	int	10	Primary Key (Auto-Incremented, Unique)
sales_login	varchar	30	Affiliate login id.
sales_password	varchar	10	Affiliate login password.
sales_name	varchar	50	Company affiliate name.
sales_contact	varchar	30	Name of individual contact for the affiliate.
sales_address	varchar	30	Affiliate street address.
sales_city	varchar	30	Affiliate location, city of residence.
sales_state	varchar	30	Affiliate location, state or province of residence.
sales_zip	varchar	10	Affiliate location, zip code of residence.
sales_country	varchar	15	Affiliate location, country of residence.
sales_phone	varchar	20	Affiliate contact's phone number.
sales_email	varchar	30	Affiliate contact's email address.
sales_company_email	varchar	30	Affiliate company email address.
sales_discounttype	varchar	10	
sales_code	varchar	20	
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.

swipool

Database:	TeleFlow Relay (LOCAL)
Primary Key:	swipool_id
Foreign Keys:	none
Indexes:	port_status (compound: port, status)
Relationships:	none

Switch pool. Call transfers are processed through this table.

Field Name	Type	Size	Description
swipool_id	int	10	Primary Key (Auto-Incremented, Unique)
port	int	5	Telephony port available to place, placing, or in a call for a call transfer request.
ivr_computer_name	varchar	50	The IVR computer name of the server/computer the telephony port resides on.
status	varchar	10	The port's current status. Possible values are: READY: Outbound port in call transfer pool is waiting for an outbound call request. OUT: Indicates the original party's port has requested the outbound port to place a call on its behalf. CALLING: Outbound port received a call request and is about to (or is) calling the remote party. CONNECT: Indicates original party and remote party are connected. NOSPEECH: Indicates a hang up condition was reached before getting someone on the phone. HANGUP: Indicates a hang up condition was reached after getting the remote side on the phone. TIMEOUT: Outbound call failed because the remote party didn't answer. BUSY: Outbound call failed due to busy signal. FAILURE: Outbound call failed for a reason of FAIL, usually indicating some form of dial error, such as no dial tone. (also used when a fatal error occurs but there was no active call on the calling port) NA: Fatal error occurred on the calling port after the remote party was reached.
reqport	int	5	The telephony port requesting an outbound call be placed on its behalf to complete a call transfer. (Will be on the same ivr_computer_name as the port placing the call)
req_dtm_utc	datetime	-	The UTC date/time the outbound call request was made.
req_dtm_ivr	datetime	-	The local date/time on the IVR when the outbound call request was made.
reqcallref	varchar	20	For release link (2B) transfers, the call reference number from the inbound channel. The call reference number is required to complete a 2B transfer.

req_client_id	int	10	Foreign Key: Links to client.client_id in the TeleFlow Relay top-level database. Indicates the client the transfer request is on behalf of.
req_app_type_name	varchar	30	Foreign Key: Links to application_type.app_type_name. Indicates the application that initiated the transfer request.
phone	varchar	30	The phone number to place an outbound call to.
extension	varchar	10	An extension number to dial when the remote side answers the outbound call.
extension_xfer_pause	int	10	The amount of time to wait (in ms) between reaching the remote side and dialing the extension. (if an extension to dial is even defined)
xfer_type	varchar	20	The type of call transfer to perform. Valid values are: SUPER_SWITCH: Common transfer type for digital and SIP extensions. One application with a caller on the line requests that an outbound line call a remote party. When the remote party answers, the two are switched together, and remain on the two lines. If the remote doesn't answer, the requesting application can put the original caller back in the application, or provide feedback to the caller. BLIND_SWITCH: Requires testing. May not be properly supported. PBX_BLIND: Performs a flash-hook, dials the digits to reach the remote party, and hangs up. (The original caller should then hear all the call activity, such as rings, busy, or a remote party answering.) PBX_SUPER: Uses a supervised flash-hook transfer, wherein a failed call brings the original caller back on the line and can continue with IVR functionality. 2B_TRANSFER: Only available on ISDN, with NI2 protocol variant. One application with a caller on the line requests that an outbound line call a remote party. When the remote party answers, the two are switched together, and the IVR lines are freed up. If the remote doesn't answer, the requesting application can put the original caller back in the application, or provide feedback to the caller.
xfer_network_type	varchar	10	Indicates the type of network the remote side is on, and therefore how the call transfer is performed. PSTN: Traditional PSTN transfer. VOIP: SIP/VoIP network transfer.
ringcount	int	2	The number of rings to wait before giving up on an outbound call attempt. (Resulting in a Place Call TIMEOUT)
send_ani	varchar	20	The ANI to send when placing the outbound call. (NOTE: This is only supported on ISDN and SIP, and is not available on all circuits.)

tfr_server

Database:	TeleFlow Relay
Primary Key:	tfr_server_id
Foreign Keys:	none
Indexes:	none
Relationships:	none

Tracks TeleFlow Relay servers/host systems and their functions.

Field Name	Type	Size	Description
tfr_server_id	int	10	Primary Key (Auto-Incremented, Unique)
server_name	varchar	50	Server's computer name.
server_status	varchar	10	Server's status. Valid codes are: ACTIVE NOTE: Other codes are TBD.
server_ip_address	varchar	20	Server's IP address.
web_server_yn	char	1	Y/N: Is this server a TFR web server?
db_server_yn	char	1	Y/N: Is this server a TFR database server?
tf_server_yn	char	1	Y/N: Is this server a TFR TeleFlow server?
tf_monitor_ip_port	int	10	Server's TeleFlow Monitor ip port, for servers where tf_server_yn=Y only.
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.

timezone

Database:	TeleFlow Relay
Primary Key:	none
Foreign Keys:	none
Indexes:	none
Relationships:	none

North American timezones and their daylight savings time utc offset adjustments. Typically, this table would be populated with the next 15-20 years of timezones, providing a system everything it needs to ensure that all DST adjustments can be accounted for.

Field Name	**Type**	**Size**	**Description**
timezone_id	int	10	Primary Key (Auto-Incremented, Unique)
timezone_abbreviation	varchar	4	An abbreviation used to express the timezone, and generally used as the lookup to locate a timezone's data.
timezone_description	varchar	50	Full timezone description.
timezone_standard_utc_offset	decimal	4,1	The unadjusted utc offset for the timezone (without daylight savings factored in).
timezone_dst_start_dtm_utc	datetime	-	Date/time the timezone's daylight savings adjustment starts, in UTC.
timezone_dst_end_dtm_utc	datetime	-	Date/time the timezone's daylight savings adjustment ends, in UTC.
timezone_dst_adjust_minutes	int	10	When adjusting the UTC offset for daylight savings time, this is the number of minutes to adjust it by.
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.

user_access_log

Database:	TeleFlow Relay
Primary Key:	none
Foreign Keys:	none
Indexes:	none
Relationships:	none

Logs all user access attempts (and successful accesses) on TeleFlow Relay websites.

Field Name	Type	Size	Description
user_access_log_id	int	10	Primary Key (Auto-Incremented, Unique)
ip_from	varchar	23	The IP address the access was from.
session_id	varchar	60	The session id for the web session the access was in.
access_type	varchar	20	A code indicating the nature of the access logged. Valid codes are: Login Logout LoginAttempt LoginFail AccessLockout AccessUnlock SetNewPass
user_id	int	10	Foreign Key: Links to user_master.user_master_id. Indicates the user accessing the system.
client_id	int	10	Foreign Key: Links to client.client_id. Indicates the client the access was for/from, or the client the user accessing the system belonged to.
page_access	varchar	50	Name of the web page accessed.
optional_command	varchar	50	Additional data to log in the context of the access_type. In this case, a command/function run on the page accessed.
optional_detail	varchar	50	Additional data to log in the context of the access_type. In this case, additional details, such as the email address used to log-in.
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.

user_active (client)

Database:	Client
Primary Key:	none

Foreign Keys: none
Indexes: none
Relationships: none

Used for connections to web services.

Field Name	Type	Size	Description
u_username	varchar	30	
u_timestamp	int	11	

user_banned (client)

Database: Client
Primary Key: none
Foreign Keys: none
Indexes: none
Relationships: none

Used for connections to web services.

Field Name	Type	Size	Description
u_username	varchar	30	
u_timestamp	int	11	
client_id	int	11	

user_details (client)

Database: Client
Primary Key: u_id
Foreign Keys: client_id
Indexes: user_login_name
user_phone_login
u_status
Relationships: Many-to-one with **client_details** (client_id)
Many-to-one with **client** (client_id)
One-to-one with **user_master** (u_id >> user_master_id)

Field Name	Type	Size	Description
u_id	int	10	Primary Key (Unique. Copied from teleflow_relay.user_master.user_master_id)
client_id	int	10	Foreign key: Links to client_details.client_id. Indicates what client the user “belongs to”, although technically, this is obvious because the user_details table is in the client database. (This key remains in case db merging is ever required for anything)
user_login_name	varchar	100	The user's global TeleFlow Relay log-in for logging into visual interfaces. (This will be an email address)
u_userid	varchar	32	Encrypted login user id used by security system for log-ins.
u_password	varchar	32	Encrypted login password used by security system for log-ins.
user_phone_login	varchar	20	The user's global TeleFlow Relay log-in id for logging into phone interfaces. (This will be all numeric)
user_phone_text	varchar	30	
user_phone_pass	int	10	The user's global TeleFlow Relay log-in password for logging into phone interfaces. (This will be all numeric)
u_oauth_access_key	varchar	100	User's access key for oauth transactions.
u_oauth_secret_access_key	varchar	100	User's secret access key for oauth transactions.
u_first	varchar	20	User's first name.
u_last	varchar	20	User's last name.
u_full_name	varchar	42	User's full name, expressed as “last, first”
u_email	varchar	50	User's email address.
u_telephone	varchar	20	User's phone number.
u_telephone_type	varchar	10	Code indicating the type of phone the user's primary telephone number is for. Eg. CELL.

			Not currently in use.
tt_first	varchar	20	User's first name, as spelled using touch-tones.
tt_last	varchar	20	User's last name, as spelled using touch-tones.
email_broadcast_groups	varchar	255	A comma-delimited list of email groups to which this user belongs. These are essentially arbitrary, but indicate under what circumstances the user will be emailed. They also indicate in which part of the email list (TO/CC/BCC) the user is emailed. Eg: CLIENT_CC: Indicates this user is CC'd on all messages broadcasted for the client (regardless of application).
email_broadcast_setting	varchar	10	Status field indicating whether or not the user should currently be emailed if a broadcast email goes out to one of the groups (see email_broadcast_groups) the user belongs to. ACTIVE INACTIVE
u_status	varchar	20	User status code, indicating whether or not the user has access to any systems. Supported values: ACTIVE: User is active in the system and can log-in/access applications. DELETED:User's access has been removed permanently.
u_permissions_level	int	10	Indicates what applications, services, functions the user can use or perform in TeleFlow Relay. What a user can do at each level is based both on this value, and on what each service/application permits users to do. Current supported values: 1: Basic user access. 9: Administrator. (client-level only) 99: Super Admin: User can access anything in TFRelay, and can access of client's applications as though he was a user in that client as well.
u_timestamp	int	10	Unix timestamp indicating when the user was created.
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.
create_dtm_client	datetime	-	Row create date/time, in client's local time.
update_dtm_client	datetime	-	Last date/time the row was updated, in client's local time.

user_master

Database: TeleFlow Relay
Primary Key: user_master_id
Foreign Keys: client_id
Indexes: user_login_name
client_id
Relationships: Many-to-One with client table.

Master user records. Because TeleFlowRelay allows many client users to access it, a top-level master reference for a user is necessary. This provides a means to link a user by their log-in name to the client to which the user belongs. All user detail information is stored at the client level.

Field Name	**Type**	**Size**	**Description**
user_master_id	int	10	Primary Key (Auto-Incremented, Unique)
user_login_name	varchar	100	The user's log-in name. When a user attempts to log-in, user_master and client are joined on client.client_id to determine the appropriate client-level database for the user. The client-level db has additional information for the user in user_details, including a password for log-in.
client_id	int	10	Foreign key: client.client_id.
create_dtm_utc	datetime	-	Row create date/time in UTC.
update_dtm_utc	datetime	-	Last date/time the row was updated in UTC.
create_dtm_host	datetime	-	Row create date/time, in local database host/server time.
update_dtm_host	datetime	-	Last date/time the row was updated, in local database host/server time.