Synthesys™ &
Noetica Voice Platform
Databases and Core Tables

Noetica.

Synthesys™ Databases

- There are 6 Databases that are part of the Synthesys™ System:
  - Phoenix – This is the Live Database that holds all data for dialler
  - Phoneyx – This is a legacy Database and is redundant in later Synthesys™ versions
  - Synthesys™_General - This is the main Script database that holds all Scripting Data
  - Synthesys™_General_Admin – This is the main Script Engine Database that holds all User and Scripted App information
  - Synthesys™_General_Reporting – This is the main Reporting Database that holds Views of the tables in the other Synthesys™_General_databases and also is used to house custom reporting tables
  - Synthesys™_Main – This holds infrastructure information about the Synthesys™ environment

Noetica.

The Sequence_ID

- The Sequence ID is the unique identifier for a script submitted through the Synthesys™ application
- It is in the format of a GUID
- It is used to track a call and any activity that relates to a script throughout the system

Noetica.

The Customer_ID

- The Customer ID is the unique identifier for Customer throughout the Synthesys™ application
- If the CustomerID is not supplied from a third party source (CSV, Database, Webservice etc.) then the format is the CRMPrefix_Sequential Number e.g. ELECT_1
- It is used to track a Customer and any activity that relates to a customer throughout the system

Noetica.

The SwitchCallID

- The SwitchCallID is the unique identifier for a telephony call submitted through the Noetica Voice Platform
- It is in the format of a GUID
- It is used to track a telephony call and any subsequent telephony actions made from the call

Noetica.

Core Tables: Phoenix_Statistics

- This is the most important table in the database for reporting
- It contains all script and dialler events
- The table also holds some key information about the call such as
  - The Call Result or/and the Dialler Result
  - The Operator who took/initiated the call
  - The Scripted App the Call relates to
  - The Outbound List the call relates to (If Outbound)
  - The Duration of the Script
  - The Date Time the Call/Script Started

Noetica.

Core Tables: Phoenix_Statistics

The EventID holds information to the type of activity that the record relates to and is enumerated as follows:

Text	Type	Description
I/B Call	Script	Inbound Call that was completed
I/B Call Failed	Script	Inbound Call that was aborted
O/B Call	Script	Outbound Call that was completed
O/B Call Failed	Script	Outbound Call that was aborted
Predictive Dialler	Dialler	An outbound call that was dialler by the predictive dialler
Runner Dial	Dialler	An outbound call that was initiated by the agent in Preview
Manual Call	Script	Manual Call that was completed
Manual Call Failed	Script	Manual Call that was aborted

See attached SQL for a function to derive the Type from the EventID.

EventIDTypes.sql

Noetica.

Core Tables: Phoenix_Statistics

- The Contact_ID is a key that links the script event in Phoenix_Statistics to the dialler events within Phoenix_Statistics.
- The Contact ID can also be used to join to Phoenix_recycling to obtain the recycling result of the call so you can see the new Queue state.

Noetica.

Core Tables: Phoenix_CallTimes

- The table is Keyed on the Sequence_ID field and can be joined to script events in Phoenix_Statistics
- The table contains the following Fields
  - Idle_Time – The amount of time the agent was idle since the last call
  - Preview_Time – The amount of time the script is open before dialling
  - Talk_Time – The amount of time the agent is talking to the customer
  - Wrap_Time – The amount of time the agent has the script open after the call has ended
- The timings in this tables are in seconds to 3 decimal places.

```
SELECT *
  FROM
       Phoenix_Statistics LEFT OUTER JOIN Phoenix_CallTimes
       ON phoenix_calltimes.sequence_id
          =
          phoenix_statistics.sequence_id;
```

Noetica.

Core Tables: Synthesys™_General_Admin.dbo.UserSessions

- This stores the date time for when certain applications were opened and what time they were closed.
- The Logon Type is enumerated as 1 = Interaction Studio, 4 = Synthesys™ Management, 3 = Agent Portal
- The Primary Key is SessionID and phoenix_statistics contains this as a Foreign Key
- If the session is ongoing then the end time will equal the login time.

```
SELECT *
  FROM
       phoenix_statistics
       LEFT OUTER JOIN Synthesys™_General_admin.dbo.usersessions us
       ON us.SessionID
          =
          phoenix_statistics.SessionID;
```

Noetica.

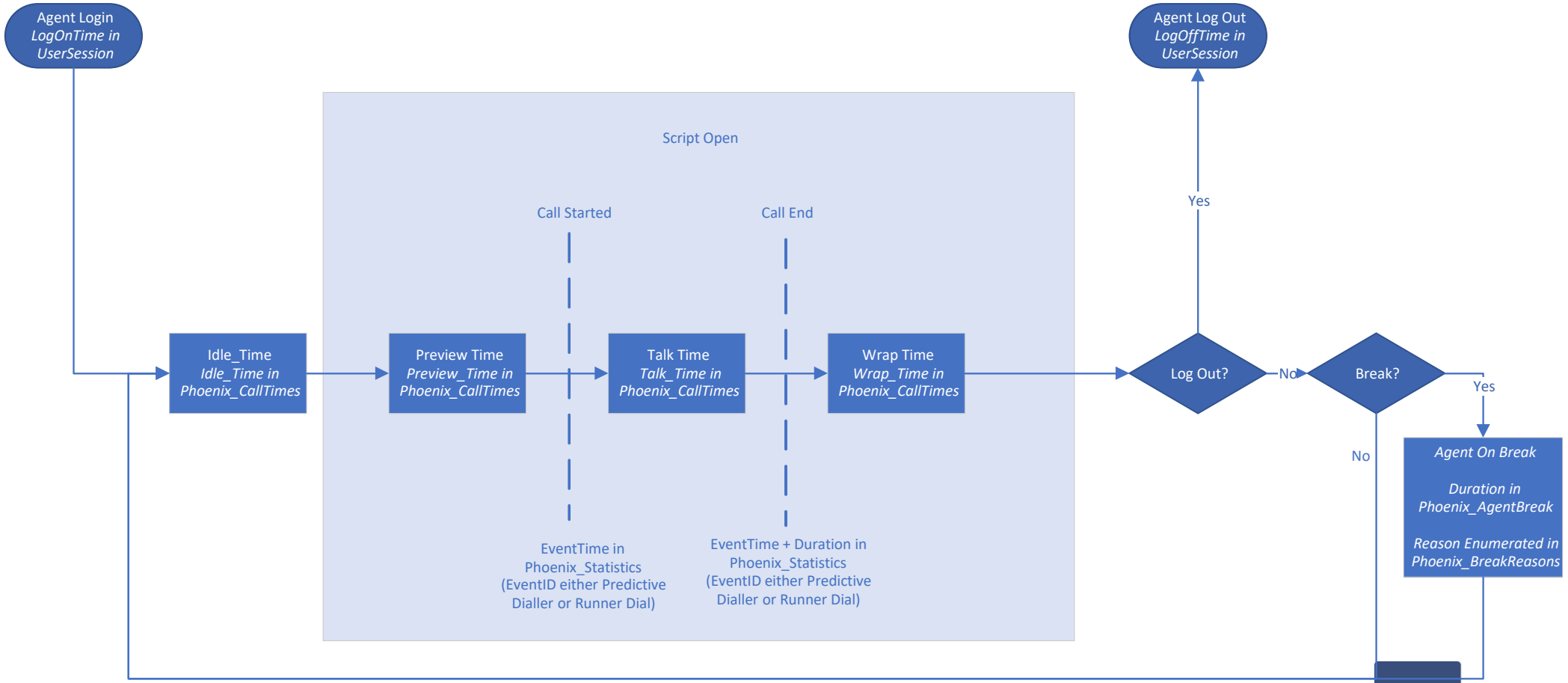
Core Tables: Phoenix_AgentBreaks

- This table is populated when ever the agent selects the break button from the Synthesys™ Application
- Its Primary Key is the ID but there is a foreign key to the Agents SessionID from Phoenix Session
- The Duration is in seconds
- If enabled the BreakReasonCode is populated and can be enumerated in Phoenix_BreakReasons joining on the BreakReasonCode field

```
SELECT *
  FROM
       Phoenix_Session LEFT OUTER JOIN Phoenix_AgentBreak
       ON Phoenix_AgentBreak.SessionID
          =
          Phoenix_Session.Session_ID
                       LEFT OUTER JOIN Phoenix_BreakReasons
       ON Phoenix_AgentBreak.BreakReasonCode
          =
          Phoenix_BreakReasons.BreakReasonCode;
```

Noetica.

Time Line



Noetica.

Core Tables: Synthesys™_General_Admin.dbo.Users

- This table contains information about all users using Synthesys™
- The Primary Key is UserID
- The field Name contains the users Username and the Users Information
- The field Active shows if the agent is active(1) or deactivated(0)

Noetica.

Core Tables: Phoenix_Queue

- This is the table that contains all records that the dialler is going to dial when Outbound Calling
- This table contains the CustomerID and the Telephone number for the record the dialler is going to initiate
- The Phoenix_Queue table is visible through Campaign Manager
- The table contains a state for the record that relates to what the dialler has done if the lead has been closed or the next action if the lead is still active.

Noetica.

Core Tables: Phoenix_Queue

- The State column contains the current state of the record and is enumerated as follows:

ID	Text	Can be Dialled?	ID	Text	Can be Dialled?
0	Done	No	8	Outbound Delete	No
1	Queued	Yes	9	Wrong Number	No
2	Rescheduled	Yes	10	Unobtainable	No
3	Attention	No	11	PD Call Running	No
4	Never Call	No	12	Recycling Complete	No
5	Running	No	13	N/A	No
6	Sleeping (From Queued)	Yes	14	Instant Callback	Yes
7	Sleeping (From Scheduled)	Yes	15	Moved	No*

* This QueueID will be closed but a new QueueID is created in a new List

See attached SQL for a function to enumerate the Queue States

QueueStates.sql

Noetica.

Core Tables: Phoenix_Queue

- If the record is in the Rescheduled State or the Sleeping States then there will be the date time the lead will become active in the Retry_Time column.
- There will also be a date in the Latest_Time column that relates to the amount of time the dialler has to make the dial attempt (Dialling Window)
- E.g. if the Maximum Lateness is set to 30 minutes and the record is rescheduled for 09:00am then the Retry_Time will be 09:00am and the Latest_Time will be 09:30am

Warning: Be very careful when reporting on Phoenix_Queue as any locks that are created on this table can have a detrimental affect on the performance of the dialler. We advise you avoid reporting on this table and if you do query this table use the with (nolock) hint http://msdn.microsoft.com/en-gb/library/ms187373.aspx

Noetica.

Core Tables: Phoenix_Recycling

- This is the table that contains all the outcomes of the Recycling Rules process
- The table contains details of the call that triggered the recycling
- It also contains what the next action will be for the call
- You can join this table to Phoenix_Statistics on the ContactID

Noetica.

Core Tables: CRM Tables

- Every CRM that is release has 2 tables created
  - CS_<PREFIX>_Customer
  - CS_<PREFIX>_History
- There are also 2 views created to make reporting easier
  - CS_<PREFIX>_Customer_View
  - CS_<PREFIX>_History_View
- When querying customer data it is recommended to use the CS_<PREFIX>_Customer_View and CS_<PREFIX>_History_View as these make the fields simplier to query.
- The primary key of the CS_<PREFIX>_Customer_View is the Customer_ID but it will be named differently dependent on what the agent calls the field in the CRM Designer, in CS_<PREFIX>_Customer it is known as p001
- The CS_<PREFIX>_History contains the Customer_ID as a foreign key and can return all events for the customer

Noetica.

Core View: Phoenix_Workspace_View

- This View contains information about all Workspaces created using Interaction Studio
- The WorkspaceID is the Primary Key

Noetica.

Core View: Phoenix_ScriptedApp_View

- This table contains information about all ScriptedApps created using Interaction Studio.
- The ScriptedApp_ID is the Primary Key
- The table contains the Workspace_ID as a foreign key and can be used to find the Workspace_ID the ScriptedApp relates to
- Phoenix_Statistics contains Campaign_ID as a foreign key and can be used to identify what ScriptedApp the call relates to

Noetica.

Core View: Phoenix_Campaign_View

- This View contains information about all Campaigns created using the Campaign Manager
- The Campaign_ID is the Primary Key

Noetica.

Core View: Phoenix_List_View

- This view contains information about all Lists created using Campaign Manager
- The List_ID is the Primary Key
- The table contains the Campaign_ID as a foreign key and can be used to find the Campaign the Outbound Campaign relates to
- Phoenix_Statistics contains OBCampaign_ID as a foreign key and can be used to find the List the call relates to
- Phoenix_Queue contains OBCampaign_ID as a foreign key and can be used to find the List the queued record relates to

Noetica.

Synthesys™_General Overview

- When ever a Scripted App is released it creates a suite of database tables in the Synthesys™_General database.
- The names of these tables are prefixed with the Account_Prefix and the Scripted App Internal Name e.g. TBC_SalesScript
- The data from the script that has been selected to be written to the database tables are written into these tables

Noetica.

Script Tables: The Main Table

- Every Scripted App release has a Main table and it is named <Account Prefix>_<Webflow Internal Name> e.g. TBC_SalesScript
- The Table has a primary key of Sequence_ID
- The field names have a naming convention of SectionName _ ControlName _ PropertyName e.g. Introduction_Memo_Value
- The Main table holds the data for all Sections that are on the Default Branch of a script and the default branches of any Branching Controls on the Default Branch

Noetica.

Script Tables: The Branch Table

- Every Branch created on the Scripted App has a Branch table and it is named <Account Prefix>_<Webflow Internal Name>_<BranchingControlName>_<BranchName>

  e.g. TBC_TestScript_BranchHasCar_AnsweredYes
- The Table has a primary key of Sequence_ID
- The field names have a naming convention of SectionName _ ControlName _ PropertyName e.g. Introduction_Memo_Value
- The Branch table holds the data for all Sections that are on the Branch of the script.

Noetica.

Noetica Voice Platform: Phoenix_Switch_Billing

- This table is used to record every call event that has been handled by the NVP™.
- The Primary Key is the SwitchCallID
- The field CallType is used to classify the type of call. E.g. Inbound, Outbound, Transfer etc.
- The outcome field is used to classify the outcome of the call E.g. Normal (For a connection), No Answer, Unobtainable
- The table has 4 date time values
  - TimeRequested - The time the call was Requested
  - TimeRinging - The time the call started Ringing
  - TimeConnection - The time the call was connected to an Agent
  - TimeDisconnected - The time the Agent disconnected the call

Noetica.

Noetica Voice Platform: Phoenix_Switch_Billing

- The Origination column holds the CLI we present on Outbound calls, on Inbound calls it will be the customers CLI
- The Destination column holds the number we called on Outbound calls, on Inbound calls it will be the DDI the customer called.
- The Extension column holds the Extension of the agent we transferred the call to
- The SequenceID is written with the script that was open whilst the call was in progress.

VoicePlatform: Phoenix_Switch_Recording

- This table is used to write an audit of the call recordings for each call.
- The table is has SwitchCallID as a foreign key
- The table holds the number dialled and the date of the call.
- The table records multiple channels for each call and the RecordingType is used to categorise the call recording
- The WavFile field holds the UNC path to the recording.

Noetica.