



CHOUETTE

Maintenance, accompagnement et recette de logiciels pour les échanges de données multimodales

Mapping CHOUETTE - GTFS

Correspondance entre éléments de données GTFS et objets de la base Chouette

Auteurs :	Michel ETIENNE, Zakaria BOUZIANE (CityWay)
Relecteurs	Patrick GENDRE (CEREMA), Philippe CARON (KEOLIS), Laaroussi Amziane (SOLUCOM), Andrew Byrd (Conveyal)

Résumé :

CHOUETTE est un logiciel libre développé à l'initiative du ministère français chargé des transports (et du développement durable) dans le but de faciliter l'échange de données d'offre (théorique) de transport collectif (TC), en s'appuyant pour cela sur la norme NFP 99506, dite Neptune, qui spécifie un profil d'échange XML.

Les utilisateurs visés sont les collectivités locales Autorités Organisatrices de Transport (AOT), les exploitants des réseaux TC, ainsi que leurs prestataires (bureaux d'étude ou société de services) et d'autres acteurs tels que services de l'état, éditeurs de logiciels, opérateurs de services d'information, chercheurs...


L'application CHOUETTE est disponible sous la forme d'une application WEB destinée à être déployée sur un serveur WEB, et d'une application en ligne de commande.

Ce document établit la correspondance entre les attributs des objets de la Base Chouette et les champs des fichiers GTFS.

Agence française pour l'information multimodale et la billettique



**Informations sur ce document :**

Organisme commanditaire : AFIMB			
Titre : Mapping GTFS - CHOUETTE			
Sous-titre :			
Organismes auteurs CITYWAY	Rédacteurs Zakaria BOUZIANE Michel ETIENNE	Participants Patrick Gendre, CEREMA Jean Seng, AFIMB Andrew Byrd, Conveyal Philippe Caron, Keolis Laaroussi Amziane, Solucom	
Maitre d'ouvrage AFIMB			
Mots clés : profil d'échange Neptune, information multi-modale, application Chouette, GTFS		Diffusion :  publique (licence Creative Commons CC-by-nd)	
Nombre de pages : 19 pages	Date Novembre 2014	Confidentialité : Non	Bibliographie : Oui

Historique des versions / révisions :

Version	Date d'application	Description des changements	Auteur
1.0		22/04/2013	M. ETIENNE Z. BOUZIANE
1.1		28/06/2013	M. ETIENNE
2.0		07/11/2014	M. ETIENNE



Table des matières

1	Introduction.....	4
2	Modèles de donnés.....	4
2.1.	Modèle GTFS.....	4
2.2.	Modèle Chouette.....	5
2.3.	Table de correspondance GTFS – Chouette.....	5
3	Règles de conversion.....	7
3.1.	Généralités.....	7
3.2.	Réseau.....	8
3.3.	Transporteur.....	8
3.4.	Arrêts.....	9
3.5.	Correspondances.....	11
3.6.	Calendriers.....	13
3.7.	Lignes, séquences d'arrêts, missions et courses.....	15
3.8.	Périodes de fréquence.....	21



1 INTRODUCTION

Ce document établit la correspondance entre les attributs des objets chouette et les champs des fichiers GTFS

seuls les fichiers suivants sont exploités :

- agency.txt
- stops.txt
- routes.txt
- trips.txt
- stop_times.txt
- calendar.txt
- calendar_dates.txt
- frequencies.txt
- transfers.txt

Il est prévu dans une évolution prochaine de prendre en compte :

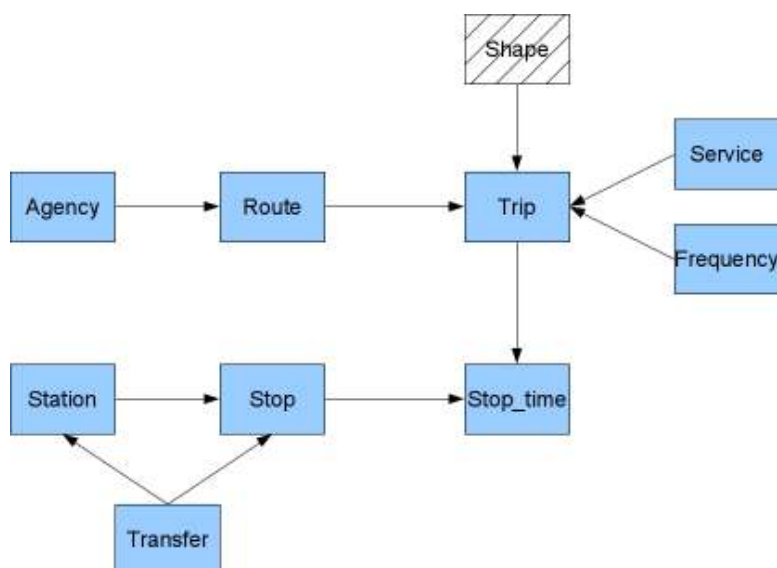
- shapes.txt
- feed_info.txt

ces fichiers sont spécifiés par Google sur <https://developers.google.com/transit/gtfs/reference?hl=fr>

2 MODÈLES DE DONNÉS

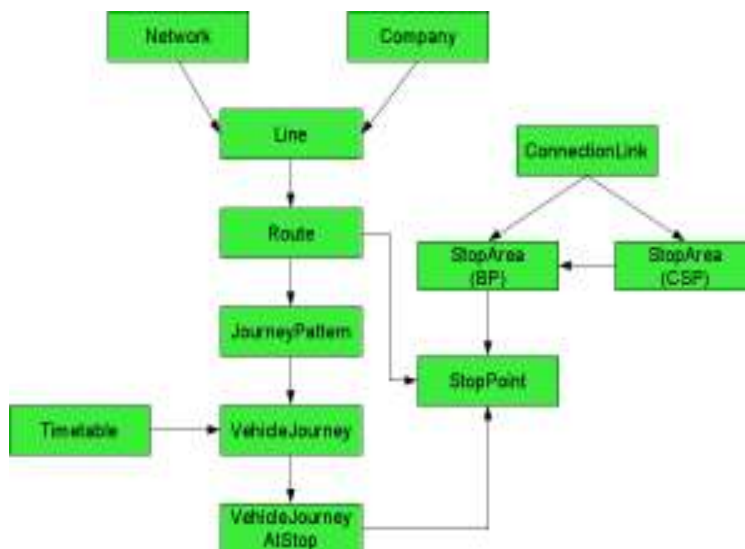
2.1. Modèle GTFS

Le schéma se limite aux objets échangés avec Chouette



2.2. Modèle Chouette

Le schéma se limite aux objets échangés en GTFS



2.3. Table de correspondance GTFS - Chouette

GTFS	Sens	Chouette
		Network



Agency	↔	Company
Route	↔	Line
Trip	←	VehicleJourney
	→	Route, JourneyPattern, VehicleJourney
Stop	↔	StopArea type BoardingPosition, Quay
Station	↔	StopArea type CommercialStopPoint
StopTime	↔	StopPoint, VehicleJourneyAtStop
Frequency	→	VehicleJourney, VehicleJourneyAtStop (clonage)
Service (Calendar.txt Calendar_dates.txt)	↔	Timetable
Transfer	↔	ConectionLink



3 RÈGLES DE CONVERSION

3.1. Généralités

Règles sur les identifiants :

les identifiants Neptune étant long, un traitement est proposé pour les convertir dans le cadre de GTFS

Un identifiant Neptune est composé de 3 items :

[préfixe]:[type d'objet]:[identifiant technique]

Un paramètre d'échange : préfixe d'identifiant Neptune est demandé lors des échanges.

A l'export :

- si le préfixe est celui fourni en paramètre (par défaut celui du référentiel), alors seul l'identifiant technique est exploité ; (Si le préfixe n'est pas fourni (chaîne vide) alors ce mécanisme est neutralisé
- dans le cas contraire, l'identifiant GTFS sera composé de [préfixe].[identifiant technique]

Exemple

ObjectId	Prefix = REZO1	Prefix=<vide>
REZO1:StopArea:12	12	REZO1.12
REZO2:StopArea:12	REZO2.12	REZO2.12
REZO1:StopArea:AZ_34	AZ_34	REZO1.AZ_34

À l'import,

- si l'identifiant GTFS respecte l'un des patterns de l'export, celui-ci sera exploité pour reformer l'identifiant Neptune,
- dans le cas contraire, le format [préfixe]:[type d'objet] :[identifiantGTFS] où le préfixe est celui fourni en paramètre
- si l'identifiant contient un caractère invalide pour la syntaxe ObjectId ; ce caractère sera remplacé par '-'



Règles sur les longueurs de texte :

à l'import, les textes sont tronqués à 255 caractères (limite des champs dans la base)

3.2. Réseau

La notion de réseau n'existe pas en GTFS :

- À l'import, le réseau est créé à partir du préfixe d'identifiant Neptune fourni en paramètre.

évolution possible : le réseau pourra être fourni en paramètre ; il devra alors être préalablement créé

- À l'export, cet objet est ignoré, mais il peut service de filtre pour la sélection des lignes

3.3. Transporteur

Le transporteur (company) est mis en équivalence avec l' « agency »

Agency.txt		
Champ	Import	export
agency_id	Cf §3.1 en cas d'absence, un identifiant interne sera créé	Cf §3.1
agency_name	company.name	company.name
agency_url	Nouvel attribut company.url	company.url
agency_timezone *	Nouvel attributs company.time_zone	Si company.time_zone est renseigné, sa valeur est exploitée, sinon la valeur fournie en paramètre sera utilisée
agency_lang	Non exploité	Non renseigné
agency_phone	company.phone	company.phone
agency_fare_url	Non exploité	Non renseigné



3.4. Arrêts

Les arrêts de chouette (StopArea) sont de 4 types, la mise en équivalence est faite sur les arrêts GTFS (stops)

Type Chouette	Type GTFS
SP=StopPlace (pôle d'échange)	Sans correspondance
CSP=CommercialStopPoint (arrêt commercial)	Station (location_type = 1)
BP=BoardingPosition (point d'embarquement)	Stop (location_type = 0)
Q=Quay (quai)	Stop (location_type = 0)

- À l'import :

Les « stations » GTFS n'étant pas systématiquement fournies, l'import intègre un algorithme de création d'arrêt commercial afin que tout arrêt physique ou quai en ait un en fin de traitement :

- si le « stop » est lié à une « station », la relation est conservée
- dans le cas contraire : un arrêt commercial est créé de même nom que l'arrêt physique (BP ou Q) ; si cet arrêt existe, alors l'arrêt physique lui est associé

une fois les arrêts commerciaux créés, leur position est définie comme le barycentre des arrêts physiques ; si des arrêts physiques sont trop loin de ce barycentre (distance maximale fournie en paramètre), alors ceux-ci sont retirés, et le barycentre est recalculé.

Les arrêts détachés sont alors regroupés au sein de nouveaux arrêts commerciaux et le traitement de distance est appliqué dessus (ce traitement pouvant conduire à une nouvelle séparation).

Cet algorithme est neutralisé si la distance maximale vaut 0.

La distinction entre Quay et BoardingPosition se fait sur présence de desserte ferrée :

- si une course qui passe par l'arrêt est d'un mode ferré (), alors l'arrêt est de type Quay.

- À l'export, tous les arrêts à l'exception des Pôles d'échange sont exportés selon le type GTFS correspondant.



stops.txt		
Champ	Import	export
stop_id	Cf §3.1	Cf §3.1
stop_code	Non exploité	Non renseigné
	<i>évolution possible : ajouter un attribut: StopArea.public_code</i>	
stop_name	stop_area.name	stop_area.name
stop_desc	stop_area.comment	stop_area.comment
stop_lat	stop_area.latitude	stop_area.latitude
stop_lon	stop_area.longitude	stop_area.longitude
zone_id	Non exploité	Non renseigné
	<i>évolution possible : ajouter un attribut: fare_zone.(l'attribut fare_code existant étant un entier, il ne convient pas pour l'usage GTFS)</i>	
stop_url	nouvel attribut : stop_area.url	stop_area.url
location_type	Voir début du chapitre	Voir début du chapitre
parent_station	Voir début du chapitre	Voir début du chapitre
stop_timezone	nouvel attribut : stop_area.time_zone	stop_area.time_zone
wheelchair_boarding	stop_area.mobility_restricted_suitability	stop_area.mobility_restricted_suitability

3.5. Correspondances

Les correspondances de chouette (ConnectionLink) sont mise en équivalence avec les « transfers » GTFS

Toutefois, Google Transit prévoit la notion de « transfert » interdit, ce qui laisse supposer que les correspondances doivent pouvoir être implicites.



À l'import, ce caractère 'implicite' est traduit par une construction de correspondances selon un rayon fourni en paramètre :

Les correspondances sont alors créées en complément des correspondances construites sur les « transferts » GTFS en y excluant celles marquées comme interdites.

Un rayon de 0 ne générera pas de correspondance implicite.

À l'export, les correspondances de la base chouette sont exportées .

Différentes possibilités peuvent être étudiées en évolution :

- **Option 1** : les correspondances correspondant à un rayonnement fourni en paramètre seront exclues de l'export
- **Option 2** : en plus de l'option 1 : si un rayonnement calcule des correspondances qui ne sont pas définies, celles-ci seront exportées comme correspondance interdite
- **Option 3** : en plus de l'option 1 : enrichir le modèle chouette pour gérer les correspondances interdites

transfers.txt		
Champ	Import	export
from_stop_id	Cf §3.1 : l'identifiant technique sera composé des 2 identifiants d'arrêt séparés par un « _ »	connection_link.start_of_link.objectId
to_stop_id		connection_link.end_of_link.objectId
transfer_type	0,1,2 : connection_link créé 3 : interdiction de générer une correspondance par rayonnement	0 si connection_link.default_duration n'est pas renseigné et si la correspondance est hors rayonnement (option) 2 si connection_link.default_duration est renseigné
min_transfer_time	connection_link.default_duration	connection_link.default_duration si renseigné

Note 1 : la notion de correspondance planifiée (type = 1) est perdue à l'import.

Note 2 : la notion de correspondance interdite (type = 2) est potentiellement perdue à l'import.

Note 3 : les correspondances importées et exportées ne sont pas forcément identiques (selon la valeur du rayonnement)



3.6. Calendriers

Les calendriers de chouette (TimeTables) sont mis en équivalence avec les «services» GTFS (fichiers calendar.txt et calendar_dates.txt)

Les principales différences de modélisation sont :

- un calendrier GTFS ne peut avoir qu'une période
- une course GTFS ne peut avoir qu'un calendrier
- une course GTFS peut commencer après minuit, auquel cas dans chouette, il faut la basculer au lendemain

À l'import :

- une course commençant après minuit sera affectée à un calendrier décalé d'un jour sur la définition du calendrier GTFS, ce calendrier aura comme identifiant celui d'origine postfixé de '_after_midnight'

À l'export :

- Un calendrier Chouette à plusieurs périodes sera traduit en GTFS par :
 - soit un calendrier à 1 période et des exclusions de dates sur les intervalles entre les périodes
 - soit un calendrier uniquement composé de dates si la liste obtenue ainsi est plus petite que la première conversion
- Une course attachée à plusieurs calendriers sera attachée à un calendrier GTFS combinant les jours effectifs de chacun des calendriers d'origine ; le calendrier portera un identifiant combiné des identifiants des calendriers d'origine.



calendar.txt		
Champ	Import	export
service_id	Cf §3.1	Cf §3.1
monday	time_table.day_type contient mon- day	time_table.day_type contient monday
tuesday	time_table.day_type contient tues- day	time_table.day_type contient tuesday
wednesday	time_table.day_type contient wed- nesday	time_table.day_type contient wednes- day
thursday	time_table.day_type contient thurs- day	time_table.day_type contient thursday
friday	time_table.day_type contient friday	time_table.day_type contient friday
saturday	time_table.day_type contient satur- day	time_table.day_type contient saturday
sunday	time_table.day_type contient sun- day	time_table.day_type contient sunday
start_date	Timetable.period[0].start_date	Voir ci-dessus
end_date	Timetable.period[0].end_date	

calendar_dates.txt		
Champ	Import	export
service_id	Cf §3.1	Cf §3.1
date	time_table.date.date	time_table.date.date
exception_type	1 : time_table.date.in_out = vrai 2 : time_table.date.in_out = faux	1 si time_table.date.in_out est vrai 2 si time_table.date.in_out est faux



3.7. Lignes, séquences d'arrêts, missions et courses

Les lignes Chouette sont organisées selon un modèle plus riche que la route GTFS :

- Une ligne Chouette est composée de séquences d'arrêt, de missions et de courses ;
- La ligne GTFS (Route) n'est composée que de courses.

À l'import, la route GTFS est importée en ligne Chouette, les séquences d'arrêt et missions sont construites en regroupant les courses (trips) de même direction et desservants les mêmes arrêts.

- Une mission est construite selon la liste ordonnée des arrêts desservis par une course
- Chaque mission est associée à une séquence d'arrêt possédant la même liste d'arrêts.
- Evolution possible : si une mission contient un sur-ensemble des arrêts d'une autre dans le même ordre, alors elle partage sa séquence d'arrêts.

À l'export, toutes les courses d'une ligne sont exportées ; le sens des courses est celui défini dans la séquence d'arrêts (Aller → direction_id = 1 ; retour → direction_id = 2) ; l'ordre des arrêts dans les horaires de la course est celui défini dans la séquence d'arrêts.

Note : la notion de mission et de séquence d'arrêt étant inexistante en GTFS, les lignes chouette exportées en GTFS pourront être réorganisées lors d'un import inverse.

Important : GTFS permet de fournir des courses avec des horaires de passage non définis sur certains arrêts ; ce type de course n'est pas traité dans l'import et se traduira par une erreur.

Une évolution possible est d'importer ces courses en interpolant les horaires de passage en se servant de la distance linéaire ou de la position des arrêts. Ce type de course ne sera donc pas restitué à l'export à l'identique.



routes.txt		
Champ	Import	export
route_id	Cf §3.1 route_id construit directement line.objectid il entre dans la composition des identifiants des objets suivants : route, journey_pattern, vehicle_journey et stop_point	Cf §3.1 route_id est construit à partir de line.object_id
agency_id	Relation line.company	Relation line.company
route_short_name	Line.number et Line.name si route_long_name n'est pas fourni	Line. number ou Line.name si number non renseigné
route_long_name	Line.published_name et Line.name	Line.published_name ou Line.name le tout si différent de route_short_name
route_desc	Line.comment	Line.comment si différent de route_short_name et route_long_name
route_type	Line.transport_mode (voir table de correspondance ci- après)	Line.transport_mode
route_url *	Nouvel attribut : line.url	line.url
route_color *	Nouvel attribut : line.color	line.color
route_text_color *	Nouvel attribut : line.text_color	line.text_color

Correspondance des modes de transport :

GTFS vers Chouette	
0 = Tram, Streetcar, Light rail	Tramway
1= Subway, Metro	Metro
2= Rail	Train
3= Bus	Bus



4= Ferry	Ferry
5 = Cable car	Other
6= Gondola, Suspended cable car	Other
7- Funicular	Other

Chouette vers GTFS	
<u>Tramway</u>	0 = Tram, Streetcar, Light rail
Trolleybus Coach Bus	3= Bus
<u>Metro</u> <u>Val</u>	1= Subway, Metro
RapidTransit LocalTrain LongDistanceTrain Train	2= Rail
Ferry	4= Ferry
Air Waterborne PrivateVehicle Walk Bicycle Shuttle Taxi Other	3= Bus



trips.txt		
Champ	Import	export
route_id	Relation VehicleJourney.Route.Line	Relation VehicleJourney.Route.Line
service_id	Relation VehicleJourney → TimeTable Cf §3.6	Relation vehicle_journey → TimeTable Cf §3.6
trip_id	Cf §3.1	Cf §3.1
trip_headsign	JourneyPattern.published_name	JourneyPattern.published_name
trip_short_name	VehicleJourney.published_journey_name	VehicleJourney.published_journey_name ou VehicleJourney.number
direction_id	VehicleJourney.route.way_back	VehicleJourney.route.way_back
block_id	Non exploité	Non renseigné
	<i>évolution possible : ajouter un attribut : VehicleJourney .vehicle_service_id</i>	
shape_id	Non exploité	Non renseigné
	<i>La prise en compte des tracés est à l'étude pour Janvier 2015</i>	
wheelchair_accessible	VehicleJourney.mobility_restricted_suitability	VehicleJourney.mobility_restricted_suitability
bikes_allowed	Non exploité	Non renseigné
	<i>évolution possible : ajouter un attribut : VehicleJourney .bike_suitability</i>	

stop_times.txt		
Champ	Import	export
trip_id	Relation VehicleJourney	Relation VehicleJourney
arrival_time	VehicleJourneyAtStop.arrival_time	VehicleJourneyAtStop.arrival_time



stop_times.txt		
Champ	Import	export
departure_time	<i>VehicleJourneyAtStop.departure_time</i>	<i>VehicleJourneyAtStop.departure_time</i>
stop_id	Relation <i>VehicleJourneyAtStop.StopPoint.contained_in</i>	Relation <i>VehicleJourneyAtStop.StopPoint.-contained_in</i>
stop_sequence	<i>VehicleJourneyAtStop.StopPoint.position</i>	<i>VehicleJourneyAtStop.StopPoint.position</i>
stop_headsign	Non exploité	Non renseigné
pickup_type	Non exploité	Non renseigné
	<i>évolution possible: définir un mapping sur VehicleJourneyAtStop.boarding_alighting_possibility</i>	
drop_off_type	Non exploité	Non renseigné
	<i>évolution possible : définir un mapping sur VehicleJourneyAtStop.boarding_alighting_possibility</i>	
shape_dist_traveled	Non exploité	Non renseigné
	<i>La prise en compte des tracés est à l'étude pour Janvier 2015</i>	

Mapping possible sur boarding_alighting_possibility :

boarding_alighting_possibility	pickup_type	drop_off_type
BoardAndAlight	0	0
AlightOnly	1	0
BoardOnly	0	1
NeitherBoardOrAlight	1	1
BoardAndAlightOnRequest	2 ou 3 (2 à l'export)	2 ou 3 (2 à l'export)
AlightOnRequest	0 ou 1(0 à l'export)	2 ou 3 (2 à l'export)
BoardOnRequest	2 ou 3 (2 à l'export)	0 ou 1 (0 à l'export)



3.8. Périodes de fréquence

Les périodes de fréquence ne sont pas actuellement modélisées dans Chouette, bien que les structures soient prêtes à accueillir les informations.

A l'import, le traitement consiste alors à convertir la course associée à une période de fréquence en autant de courses nécessaires pour couvrir la période ; plusieurs périodes peuvent être associées à la même course.

A l'export, cette notion n'existe pas dans les données de Chouette. Il n'y a pas de possibilité de génération de Périodes de fréquences.

Une évolution est en cours pour prendre en compte les courses à fréquences dans Chouette ; cette évolution comprend la reprise du mapping présenté ici.

frequencies.txt		
Champ	Import	export
trip_id	Relation vers Trip	Sans objet
start_time	Heure de départ de la première course	Sans objet
end_time	Limite en heure de départ pour la génération des courses	Sans objet
headway_secs	Décalage entre 2 courses générées	Sans objet
exact_times	Non exploité, la génération des courses choisit des départs cadencés	Sans objet