

API

TimeToChat

V1.0



SOMMAIRE

1.	Définition du TimeToChat.....	3
2.	Présentation	3
2.1	Points d'entrée.....	3
2.2	Utilisation	4
2.3	SMS long	4
2.4	Limitation.....	4
2.5	SMS entrants et Comptes rendus	5
3.	Recevoir un SMS	6
4.	Obtenir un Token	7
5.	Envoyer un SMS	8
6.	Compte rendu	11
7.	Liste des états	13
8.	Tester les API	14
8.1	Connexion	14
8.2	Générer un token	14
8.3	Utiliser un token.....	14
8.4	Envoyer un SMS	15
8.5	Comptes rendus d'envoi.....	15
9.	Connecteur SMPP	17
9.1	Paramètres de connexion.....	17
9.2	Limitation.....	17
9.3	Commandes écoutées	17
9.4	Commande renvoyées	19

1. Définition du TimeToChat

Le TimeToChat est une offre de messagerie conversationnelle par SMS de SFR Business spécialement dédiée aux échanges entre les marques et leurs clients. Cette offre est dédiée aux services d'échange bidirectionnel entre une entreprise et sa clientèle avec une numérotation non surtaxée et spécifique de type 09 3X XX XX XX.

2. Présentation

Ce document décrit les étapes détaillées afin que vous puissiez recevoir et envoyer des SMS au travers des API TimeToChat.

Afin de pouvoir envoyer des requêtes sur les API TimeToChat, vous devez, au préalable, souscrire à l'offre TimeToChat de SFR. Vous obtiendrez ainsi un couple identifiant / mot de passe qui vous permettra d'envoyer des SMS.

2.1 Points d'entrée

Le Push_Cra qui vous permet de recevoir en temps réel les SMS envoyé vers un numéro 09 3X XX XX XX sur une URL de votre choix.

L'API TimeToChat qui vous permet de répondre aux SMS reçu en utilisant un numéro 09 3X XX XX XX.

Un connecteur SMPP qui vous permet de gérer vos SMS sur un 09 3X XX XX XX.

L'url d'accès aux API TimeToChat est : <https://www.dmc.sfr-sh.fr/mug/api/V1/>

L'interface Swagger est accessible à l'adresse : <https://www.dmc.sfr-sh.fr/mug/api/V1/api-docs/>

SFR Business - API MUG 1.0 OAS3
API SMS REST - Messages Unitaires Garantis

Authorize

Authentification

POST /services/{service_id}/token Génération d'un token

Messages

POST /services/{service_id}/messages Envoi d'un SMS

GET /services/{service_id}/messages/{message_id} Compte-rendu d'un SMS

Le connecteur SMPP est accessible au travers d'un tunnel utilisant le protocole TLS :

- Host : www.dmc.sfr-sh.fr
- Port : 5772

2.2 Utilisation

L'API TimeToChat requièrent une authentification basée sur les standards (oAuth 2.0) au format REST.

Cela consiste à effectuer des requêtes HTTP (GET ou POST) en utilisant la norme SSL, avec des paramètres en en-tête pour sécuriser vos appels.

2.3 SMS long

La taille standard d'un SMS est de 160 caractères. Au-delà de 160 caractères, les SMS sont automatiquement transformés en SMS longs.

La taille maximale d'un SMS long est de 1224 caractères.

Attention, les SMS standards étant envoyés avec l'alphabet GSM, les caractères suivants comptent double : ^, ~, |, {, }, [,], \, €.

Le SMS long est reçu par le destinataire comme un message unique et sans surcoût. Le coût total du SMS dépend de sa longueur :

- 0 à 160 caractères => 1 SMS
- 161 à 306 caractères => 2 SMS (7 caractères servent à la concaténation)
- 307 à 459 caractères => 3 SMS
- 460 à 612 caractères => 4 SMS
- 613 à 755 caractères => 5 SMS
- 756 à 918 caractères => 6 SMS
- 919 à 1071 caractères => 7 SMS
- 1072 à 1224 caractères => 8 SMS

Si l'option Unicode est cochée, la taille d'un SMS est de 70 caractères, et la taille maximale d'un SMS long est de 528 caractères. Aucun caractère ne compte double. Le coût total d'un SMS Unicode dépend de sa longueur :

- 0 à 70 caractères => 1 SMS
- 71 à 132 caractères => 2 SMS (4 caractères servent à la concaténation)
- 133 à 198 caractères => 3 SMS
- 199 à 264 caractères => 4 SMS
- 265 à 330 caractères => 5 SMS
- 331 à 396 caractères => 6 SMS
- 397 à 462 caractères => 7 SMS
- 463 à 528 caractères => 8 SMS

2.4 Limitation

Par défaut, le nombre d'appels simultanés des différentes méthodes des API TimeToChat est limité à 20 par seconde. Toute demande supplémentaire sera rejetée avec l'erreur « Too many request ».

Les trois méthodes « Token », « POST Message » et « GET Message » comptent pour un appel.

Si vous rencontrez des besoins particuliers nécessitant d'augmenter cette limite, nous vous invitons à vous rapprocher de nos équipes pour étudier une offre sur mesure.

2.5 SMS entrants et Comptes rendus

Tous les SMS entrants reçu sur le 09 3X XX XX XX sont envoyés en temps réel sur l'URL que vous avez mentionnée lors de la souscription du service. Ce mécanisme s'appelle le « Push_CRA ».

Le Push_CRA est également utilisé pour remonter les comptes-rendus des SMS que vous avez envoyé, en réponse à un SMS entrant.

3. Recevoir un SMS

Que ce soit pour recevoir les SMS MO envoyés vers un numéro en 09 3X XX XX XX ou pour connaître l'état de l'envoi d'un SMS MT, il est nécessaire d'utiliser le Push_Cra.

Le Push_Cra consiste à envoyer en temps réel tous les MO et les comptes rendus sur une URL de votre choix.

Chaque élément du JSON contient :

- **Username** : l'identifiant du service utilisé pour envoyer le SMS
- **To** : numéro du destinataire au format international
- **From** : le numéro au format 09 3X XX XX XX utilisé pour envoyer le SMS
- **Date** : date de l'évènement auquel correspond le compte rendu
- **Call_id** : identifiant de l'envoi (à fournir pour diagnostic en cas de problème)
- **Ref_externe** : référence externe fournie lors de la soumission du SMS
- **status** : état du SMS au moment de la génération du compte rendu
- **Status_list** : liste des états horodatés par lesquels sont passés le SMS
 - **Date** : date à laquelle le SMS est passé dans cet état
 - **Type** : Etat du SMS (cf liste ci-dessous)

Les comptes rendus sont au format json et contiennent l'ensemble des états par lesquels est passé le SMS :

- **SENT** : Le SMS a été pris en compte par l'opérateur
- **RECEIVED** : Le SMS a été reçu sur le mobile du destinataire
- **INCOMING** : SMS reçu de la part d'un destinataire
- **ANSWERED** : Le destinataire a répondu à un SMS précédemment envoyé
- **ERROR** : Une erreur est survenue lors de l'envoi du SMS

Exemple de SMS entrant :

```
2024-03-25 14:30:52 {"status_report":{"username":"123456",
"sms":[{"to":"+336XXXXXXXXX","from":"093XXXXXXXXX","date":"2024-03-25T14:30:31Z",
"call_id":"","status_list":[{"date":"2024-03-25T14:30:31Z","type":"INCOMING",
"info":"Ceci est un test"}]}, "status":"INCOMING"}]}
```

Exemples de comptes-rendus :

```
2024-03-25 14:32:37 {"status_report":{"username":"123456",
"sms":[{"to":"+336XXXXXXXXX","from":"093XXXXXXXXX","date":"2024-03-25T14:32:25Z",
"call_id":"66018af8c8ab4e0ca97532f1","ref_externe":"abc123",
"status_list":[{"date":"2024-03-25T14:32:25Z","type":"SENT"}]}, "status":"SENT"}]}
```

```
2024-03-25 14:32:48
{"status_report":{"username":"123456","sms":[{"to":"+336XXXXXXXXX","from":"093XXXXXX
X", "date":"2024-03-25T14:32:27Z","call_id":"66018af8c8ab4e0ca97532f1",
"ref_externe":"abc123","status_list":[{"date":"2024-03-
25T14:32:27Z","type":"RECEIVED"}]}, "status":"RECEIVED"}]}
```

```
2024-03-25 14:33:37
{"status_report":{"username":"123456","sms":[{"to":"+336XXXXXXXXX","from":"093XXXXXX
X", "date":"2024-03-25T14:33:09Z","call_id":"66018af8c8ab4e0ca97532f1",
"ref_externe":"abc123","status_list":[{"date":"2024-03-25T14:33:09Z",
"type":"ANSWERED","info":"Merci c'est super"}]}, "status":"ANSWERED"}]}
```

4. Obtenir un Token

Pour commencer, vous devez générer un token valide afin d'appeler les API TimeToChat. Pour cela, vous devez utiliser la méthode POST /services/{service_id}/token :

POST

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ \
  "password": "12a34b56c" \
}' 'https://www.dmc.sfr-sh.fr/mug/api/V1/services/123456/token'
```

URL

```
https://www.dmc.sfr-sh.fr/mug/api/V1/services/123456/token
```

Si la méthode POST est un succès, un code 201 est retourné et la réponse (JSON) doit ressembler à :

```
{
  "token":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzZXJ2aWNlX2lkIjoibjY2OTY5IiwiaWF0IjoxNTM4NDYzMjY2LCJleHAiOiE1Mzg0NjUwNjZ9.Xccd6UPvOWgJ_Q1T0l4xqmF1P1sNXD5o772icdV6iZk"
}
```

Si la méthode POST est un échec, un code 401 est retourné et la réponse (JSON) doit ressembler à :

```
{
  "code": "AUTHENTICATION_FAILED"
}
```

Un token reste valide pendant une durée de 30 minutes. Passé ce délai, un nouveau token doit être généré en rappelant la méthode POST /services/{service_id}/token.

5. Envoyer un SMS

Afin d'envoyer un SMS depuis l'API Mug, vous devez utiliser le token que vous venez de générer dans l'en-tête "Authorization" et appeler la méthode POST /services/{service_id}/messages :

POST

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzZXJ2aWNlX2lkIjoibG9ja20TY5IiwiaWF0IjoxNTM4NDY0NzY5LCJleHAiOiJlMzg0NjY1Nj19.QYW0_sShRCPgdRWA1-AOf-2nSW85wJgjPuWMymI1HH0' -d '{\n  "from": "093XXXXXXX", \n  "to": "0601020304", \n  "extRef": "123456", \n  "validityDate": "2018-12-27T09:06:22.620Z", \n  "text": "Test MUG" \n}' https://www.dmc.sfr-sh.fr/mug/api/V1/services/123456/messages'
```

URL

<https://www.dmc.sfr-sh.fr/mug/api/V1/services/123456/messages>

Si la méthode POST est un succès, un code 201 est retourné et la réponse (JSON) doit ressembler à :

```
{\n  "id": "5bb31c2f911ea924bdd42d75"\n}
```

Si la méthode POST est un échec, un code de type 400 est retourné et la réponse (JSON) doit ressembler à :

Code 401 (token expiré)

```
{\n  "code": "AUTHENTICATION_FAILED"\n}
```

Code 400 (parameter manquant)

```
{\n  "code": "JSON_INVALID",\n  "params": {"error": "should have required property 'text'"}\n}
```


Code 400 (date de validité)

```
{
  "code": "JSON_INVALID",
  "params": {
    "error": "validityDate should NOT be in the past"
  }
}
```

Code 400 (date de validité)

```
{
  "code": "JSON_INVALID",
  "params": {
    "error": "validityDate should match format \"date-time\""
  }
}
```

Pour les comptes de test, il existe une limitation sur la durée de validité et sur le nombre de SMS autorisé par mois :

Code 403 (délai expiré)

```
{
  "errorCode": "SERVICE_EXPIRED",
  "errorText": "service has expired"
}
```

Code 403 (limite mensuelle)

```
{
  "errorCode": "MONTHLY_LIMIT",
  "errorText": "monthly message limit reached"
}
```

Code 403 (Emetteur interdit)

```
{
  "errorCode": "FROM_OADC_NOT_AUTHORIZED",
  "params": {"oadc": "<liste des OADC aurotisés>"}
}
```

Les paramètres obligatoires sont :

- **To** : Numéro de téléphone du destinataire au format 06/07xxxxxxx ou +32xxxxxxx pour l'international
- **Text** : Contenu du message à envoyer
- **From** : Un numéro 09 3X XX XX XX autorisé pour le service

Les paramètres optionnels sont :

- **extRef** : La référence externe issue de votre SI. Cette référence sera remontée dans les comptes rendus (150 caractères maximum)
- **validityDate** : Date de validité du message, au format «2019-07-01T08:00:00.123Z »
- **unicode** : Permet d'envoyer des SMS encodés en utf8. Par défaut l'encodage utilisé est l'ISO8859-1
- **flash** : Permet d'envoyer des SMS flash, SMS non stockés dans la mémoire de l'appareil

6. Compte rendu

Pour connaître le résultat de l'envoi d'un SMS, vous devez utiliser le token que vous venez de générer dans l'en-tête "Authorization", indiquer l'identifiant du message reçu lors de l'envoi du SMS et appeler la méthode GET /services/{service_id}/messages/{message_id} :

GET

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzZXJ2aWNlX2lkIjoibG8F6rF3XzIzsQ6c2JJWu1-N4Ib5g'jM3MjYyLCJleHAiOiJlMzg2MzkwNjJ9.hUIF7IHePOp2HjLG8F6rF3XzIzsQ6c2JJWu1-N4Ib5g'  
https://www.dmc.sfr-sh.fr/mug/api/V1/services/123456/messages/5bb5bdce911ea924bddaa243'
```

URL

```
https://www.dmc.sfr-sh.fr/mug/api/V1/services/123456/messages/5bb5bdce911ea924bddaa243
```

Si la méthode GET est un succès, un code 200 est retourné et la réponse (JSON) doit ressembler à :

```
{  
  "from": "093XXXXXXX",  
  "to": "0601020304",  
  "text": "Test TimeToChat",  
  "unicode": false,  
  "flash": false,  
  "id": "5bb5bdce911ea924bddaa243",  
  "date": "2023-10-04T07:14:22.000Z",  
  "history": [  
    {  
      "date": "2023-10-04T07:14:22.000Z",  
      "status": "NEW"  
    },  
    {  
      "date": "2023-10-04T07:14:23.000Z",  
      "status": "SENT"  
    },  
    {  
      "date": "2023-10-04T07:14:24.000Z",  
      "status": "RECEIVED"  
    }  
  ]  
}
```

Dans le cas d'un SMS long, la réponse contiendra en plus les numéros des parties de SMS :

```
{
  "from": "093XXXXXXX",
  "to": "0601020304",
  "text": "Test TimeToChat",
  "unicode": false,
  "flash": false,
  "id": "5bb5bdce911ea924bddaa243",
  "date": "2023-10-04T07:14:22.000Z",
  "history": [
    {
      "date": "2023-10-04T07:14:22.000Z",
      "status": "NEW"
    },
    {
      "part": "1/2",
      "date": "2023-10-04T07:14:23.000Z",
      "status": "SENT"
    },
    {
      "part": "2/2",
      "date": "2023-10-04T07:14:23.000Z",
      "status": "SENT"
    },
    {
      "part": "1/2",
      "date": "2023-10-04T07:14:24.000Z",
      "status": "RECEIVED"
    },
    {
      "part": "2/2",
      "date": "2023-10-04T07:14:24.000Z",
      "status": "RECEIVED"
    }
  ]
}
```

Si la méthode GET est un échec, un code de type 400 est retourné et la réponse (JSON) doit ressembler à :

Code 401 (token expiré)

```
{  
  "code": "AUTHENTICATION_FAILED"  
}
```

7. Liste des états

Un SMS envoyé via l'API TimeToChat passera successivement par plusieurs états :
NEW : réception par les API d'une nouvelle demande d'envoi de SMS.

CRA_SMS_SENT : le SMS a été soumis à l'opérateur.

Après l'état « **CRA_SMS_SENT** », le SMS peut passer en état « **CRA_SMS_RECEIVED** » (qui indique que le SMS a été délivré sur le mobile du destinataire) ou en état « **ERREUR** » dont les différents cas de figure sont décrits ci-dessous :

- **CRA_SMS_ERROR** : Une erreur est survenue lors de l'envoi du SMS
- **CRA_SMS_OPINVALID** : Opérateur du destinataire invalide
- **CRA_SMS_BLACKLIST** : Destinataire blacklisté sur la plateforme
- **CRA_SMS_OPBLACKLIST** : Destinataire blacklisté par l'opérateur
- **CRA_SMS_INVALIDFROM** : Champ OADC invalide
- **CRA_SMS_WRONGNUMBER** : Format de numéro invalide
- **CRA_SMS_FALSENUMBER** : Abonné absent ou numéro non attribué
- **CRA_SMS_OPERROR** : Problème lors de la communication vers l'opérateur
- **CRA_SMS_OPFAILURE** : Erreur de transmission du message à l'opérateur
- **CRA_SMS_OPFULL** : Surcharge du SMS-C (lien opérateur)
- **CRA_SMS_COUNTRY** : Pays de destination non supporté
- **CRA_SMS_FULLSIM** : Carte SIM du destinataire pleine
- **CRA_SMS_LOTOFMSG** : Trop de messages envoyés vers ce numéro
- **CRA_SMS_REJECTED** : Message rejeté par l'opérateur
- **CRA_SMS_BLOCKED** : Appel bloqué par l'opérateur

8.4 Envoyer un SMS

Cliquer sur le bouton **POST** de la méthode « /services/{service_id}/messages »

Cliquer sur le bouton **Try it out**

Renseigner l'identifiant du service et les données du message et valider en cliquant sur **Execute**

service_id * required
string
(path)

Identifiant du service MUG utilisé pour envoyer le message

Request body

Données nécessaires à l'envoi d'un nouveau message

Edit Value Model

```
{
  "to": "+33600000000",
  "text": "Test de SMS depuis le Mug",
  "extRef": "123456",
  "from": "Mug",
  "unicode": false,
  "flash": false
}
```

Cancel

Execute

Récupérer l'identifiant du message dans la réponse :

Response body

Download

```
{
  "id": "5bc9a8751a34791aff429efe",
  "extRef": "123456"
}
```

8.5 Comptes rendus d'envoi

Cliquer sur le bouton **GET** de la méthode « /services/{service_id}/messages/{message_id} »

Cliquer sur le bouton **Try it out**

Renseigner l'identifiant du service et l'identifiant du message précédemment récupéré et valider en cliquant sur **Execute**

service_id * required string (path)	Identifiant du service Mug utilisé pour l'envoi du message
	123456789
message_id * required string (path)	Identifiant du message pour lequel on souhaite récupérer le compte-rendu
	5bc9a8751a34791aff429efe
Execute Clear	

Le compte rendu d'envoi du message s'affiche dans la réponse :

Download

```
{
  "to": "+33617612030",
  "text": "Test de SMS depuis le Mug",
  "extRef": "123456",
  "unicode": false,
  "flash": false,
  "id": "5bc9a8751a34791aff429efe",
  "date": "2018-10-19T09:48:37.659Z",
  "history": [
    {
      "date": "2018-10-19T09:48:37.659Z",
      "status": "NEW"
    },
    {
      "date": "2018-10-19T09:48:38.000Z",
      "status": "CRA_SMS_SENT"
    },
    {
      "date": "2018-10-19T09:48:39.000Z",
      "status": "CRA_SMS_RECEIVED"
    }
  ]
}
```


9. Connecteur SMPP

L'API MUG dispose d'un connecteur SMPP. Ce connecteur se base sur la norme 3.4 du protocole standard SMPP.

Documentation officielle : http://docs.nimta.com/SMPP_v3_4_Issue1_2.pdf

Le connecteur SMPP du MUG se base sur les commandes (pdu) du protocole standard.

Les informations fournies ci-dessous concernent les commandes (pdu) de requêtes et de réponses implémentées dans le connecteur SMPP du MUG.

Pour une description détaillée des commandes, se référer à la documentation officielle.

9.1 Paramètres de connexion

Le connecteur SMPP est accessible au travers d'un tunnel utilisant le protocole TLS :

- Host : www.dmc.sfr-sh.fr
- Port : 5772

9.2 Limitation

Comme pour le Mug, le connecteur SMPP est limité à 20 requêtes par seconde. Passé ce seuil, les requêtes seront rejetées avec l'erreur « Too many request ».

9.3 Commandes écoutées

Pour la signification ou le code hexadécimal des codes dans le champ **command_status** se référer à la page 112 §5.1.3 de la [documentation officielle](#)

Toute autre commande envoyée que celle décrite ci-dessous sera ignorée et ne recevra pas de réponse de la part serveur.

Commande	Statut	Pdu de requête	Pdu de réponse
bind_transmitter	Non supporté		command_status = ESME_RBINDFAIL
bind_transceiver	supporté	system_id (obligatoire): identifiant de connexion password (obligatoire): Mot de passe associé au sytem_id	Succès: pdu standard Echec: command_status = ESME_RBINDFAIL
outbind	Non supporté		command_status = ESME_RBINDFAIL
unbind	supporté		
enquire_link	supporté		

Commande	Statut	Pdu de requête	Pdu de réponse
submit_sm	supporté	destination_addr(obligatoire): Numéro de téléphone de destination au format international ou au format national français Contenu du message (obligatoire): message_payload: Si ce champ est renseigné alors il est pris en compte et le champ short_message est ignoré. A utiliser pour envoyer un message 'long' short_message: Contenu du message source_addr: Destinataire, si rempli il doit correspondre à un OADC autorisé pour ce service data_coding: S'il vaut 0x08 alors le message sera considéré comme unicode. S'il vaut 0xF0 alors le message sera considéré comme étant un message flash dest_addr_subunit: Si le champ vaut 0x01 alors le message sera considéré comme un message flash validity_period: Période de validité du message. Les formats acceptés sont les suivants: - une date absolue au format UTC YYMMDDhhmmss voir la documentation officielle page 164 §7.1.1 - une période au format : YYMMDDhhmmss000R* external_reference: Référence du SI client Toute autre information dans la trame sera ignorée.	Succès: message_id = ID_DU_MESSAGE Echec: Si le pdu en entrée est mal formé : command_status = ESME_RQUERYFAIL Sinon: command_status = ESME_RSUBMITFAIL
submit_sm_multi	non supporté		command_status = ESME_RSUBMITFAIL

Commande	Statut	Pdu de requête	Pdu de réponse
data_sm			command_status = ESME_RSUBMITFAIL
query_sm			command_status = ESME_RQUERYFAIL
cancel_sm			command_status = ESME_RCANCELFAIL
replace_sm			command_status = ESME_RREPLACEFAIL
deliver_sm			command_status = ESME_RDELIVERYFAILURE

9.4 Commande renvoyées

Commande	Pdu de requête	Pdu de réponse
enquire_link		Pdu standard avec command_status = 0 Sinon la connexion est fermée. Cette commande est envoyée toute les minutes au client
deliver_sm	2 cas sont possibles : Un MESSAGE_ORIGINATED ou un STATUS_REPORT. La différence sera dans le champ esm_class qui sera positionné à 0x00 pour un MO et à 0x04 pour un SR. Cas du MO esm_class: 0x00 source_addr: Adresse de l'émetteur du MO (personne ayant répondu au message) short_message: Texte du MO Cas du SR esm_class: 0x04 source_addr: Adresse du destinataire du SMS initial auquel se rapporte ce SR receipted_message_id: Id du SMS initial auquel se rapporte ce SR message_state: Etat du message, Voir la documentation officielle page 130 §5.2.28 short_message: Chaîne de caractères formatée de la façon suivante : submit date:YYMMDDhhmm done date:YYMMDDhhmm stat: __STATUS__ text: __TEXTE__ id: Id du SMS initial correspondant au DLR	

Commande	Pdu de requête	Pdu de réponse
	• sub: "001" • dlvr: "001" • submit date: Date d'envoi du sms • done date: Date de réception du DLR • stat: Etat du message • err: "000" ou le code d'erreur • text: Texte du SR external_reference: Référence du SI client reçu dans le submit_sm sar_msg_ref_num: Indentifiant sur 2 octets du message généré à partir de receipted_message_id sar_segment_seqnum: Numéro de la partie sar_total_segments: Nombre total de parties ce format suit la documentation officielle page 167 Appendice B	