****

**NESD-QA: Gherkin Scripting Guideline**

Document Revision # : Version 1.0

Data classification : C2

**Executive Summary**:

The document details the procedure or details for SDQA Test Analysts to better capture Gherkin tests in ALM Octane.

Document Owner: Pheta Moloi  
Approved By:   
 14 February 2023

**APPROVAL**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Designation | Company | Date | Signature |
| Pheta Moloi | Senior Specialist : NESD-QA | Vodacom |  |  |
| Nthabiseng Mokoena | Specialist : NESD-QA | Vodacom |  |  |

**CONTRIBUTORS**

|  |  |  |
| --- | --- | --- |
| Name | Designation | Company |
| Masiza Qutu | Manager : NESD-QA |  |
| Nthabiseng Mokoena | Specialist : NESD-QA |  |
| Thandi Khoza | Specialist : NESD-QA |  |
| Katlego Sepeng | Specialist : NESD-QA |  |
| Phikolomzi Genge | Specialist : NESD-QA |  |
| Nare Mathatho | Specialist : NESD-QA |  |
| Wilheminah Ngqola | Specialist : NESD-QA |  |
| Hilary Steyn | Specialist : NESD-QA |  |
| Dineo Moloi | Specialist : NESD-QA |  |
|  |  |  |

**DOCUMENT HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Name | Date | Comments |
| 1.0 | Pheta Moloi | 18 October 2021 | Initial draft |
| 1.1 | Pheta Moloi | 02 Nov 2021 | Update - ALM Octane Gherkin Script – Data and Voice 30 Days BundleAdded - ALM Octane Gherkin Script – SMS 30 Days BundleAdded - ALM Octane Gherkin Script – SMS 30 Days Bundle |
|  | Nthabiseng Mokoena | 08 February 2022 | Postpaid account credit verification line added |
|  | Nthabiseng Mokoena | 09 March 2022 | Purchase line update to generic line  * ViewThrough GT line update * SMS allocation line fixed |
|  | Pheta Moloi | 22 March 2022 | * Added credit verification for Prepaid/TopUp |
|  | Nthabiseng Mokoena | 12 April 2023 | * Balance adjust and verification updates |

**Table of Contents**

[ Update - ALM Octane Gherkin Script – Data and Voice 30 Days Bundle 3](#_Toc120611586)

[ Added - ALM Octane Gherkin Script – SMS 30 Days Bundle 3](#_Toc120611587)

[ Added - ALM Octane Gherkin Script – SMS 30 Days Bundle 3](#_Toc120611588)

[ Postpaid account credit verification line added 3](#_Toc120611589)

[ Purchase line update to generic line 3](#_Toc120611590)

[ Introduction 5](#_Toc120611591)

[.1 Definitions, Abbreviations and Acronyms 5](#_Toc120611592)

[.2 References 5](#_Toc120611593)

[ Assumptions 6](#_Toc120611594)

[ ALM Octane Gherkin Script 6](#_Toc120611595)

[.1 ALM Octane Gherkin Script Details 6](#_Toc120611596)

[.1.1 Create MSISDN 6](#_Toc120611597)

[.1.2 Airtime balance adjust 6](#_Toc120611598)

[.1.3 Fusion purchase request 6](#_Toc120611599)

[.1.4 Verify CCS 7](#_Toc120611600)

[.1.5 Verify Fusion 7](#_Toc120611601)

[.1.6 Verify Service Balances (ViewThrough) 7](#_Toc120611602)

[.1.7 VOMS 9](#_Toc120611603)

[ ALM Octane Gherkin Script – Data 30 Days Bundle 10](#_Toc120611604)

[ ALM Octane Gherkin Script – Data 30 Calendar Days Bundle 10](#_Toc120611605)

[ ALM Octane Gherkin Script – Data and Voice 30 Days Bundle 11](#_Toc120611606)

[ ALM Octane Gherkin Script – Voice 30 Days Bundle 11](#_Toc120611607)

[ ALM Octane Gherkin Script – SMS 30 Days Bundle 12](#_Toc120611608)

[ ALM Octane Gherkin Script – Voice and Data 30 Calendar Days Bundle 12](#_Toc120611609)

[ ALM Octane Gherkin Script – Intergrated Voice Data and Whatsapp 7 Days Bundle 13](#_Toc120611610)

[ ALM Octane Gherkin Script – Fusion Validation 13](#_Toc120611611)

[.1 Fusion Response Validation (DEPRICATED) 13](#_Toc120611612)

[.2 Fusion Service Balance (VIEWTHRU) Validation 14](#_Toc120611613)

# Introduction

As NESD-QA Test Automation team, we have identified a gap whereby there is some misalignment between Test Analysts and Test Automators when it comes to capturing on Gherkin scripts and automation thereof. The is a particular way that Test Automators expect Test Analysts would capture Gherkin scripts and this document is meant to detail exactly that.

## Definitions, Abbreviations and Acronyms

Table 2 - Definitions, Abbreviations and Acronyms

| Term | Definition |
| --- | --- |
| UNITS | Unit code for SMS, used during SMS bundle purchase on Fusion |
| MINUTES | Unit code for Voice, used during Voice bundle purchase on Fusion |
| MB,GB | Unit code for Data, used during Data bundle purchase on Fusion |
|  |  |

## 

## References

Table 3 - Document References

| No | Document | Author | Identification | Version | Date |
| --- | --- | --- | --- | --- | --- |
| 1. |  |  |  |  |  |
| 2. |  |  |  |  |  |

# 

# Assumptions

The document assumes that Test Analysts are well conversant with BDD and Gherkin scripting in general. That some level of manual testing has been conducted to verify the need for EFS response verification.

# ALM Octane Gherkin Script

## ALM Octane Gherkin Script Details

Below will detail the basic ALM Octane Gherkin script. When building a script, you will mainly choose from the statements below to complete your script for test automation.

### Create MSISDN

For every test we would normally delete and recreate an MSISDN profile on CCS.

Below two tests have now been combined to one function

Given New subscriber <Subscriber\_type> profile is <Offerring\_Code>

- Identifies an MSISDN to be used during test execution (<Subscriber\_type> = MSISDN to be used during execution).

- Delete and the recreate the MSISDN profile on CCS as identified by <Offerring\_Code> parameter. E.g. <Offerring\_Code> = “NOF5”, “WF5” etc.

Given Multiple New subscribers <Hybrid\_Subscriber> profile is <Hybrid\_Offerring\_Code> and <Prepaid\_Subscriber> profile is <Prepaid\_Offerring\_Code>

* The function will create two subscriber profiles with provided offering ceodes.
* Hybrid\_Subscriber/Hybrid\_Offerring\_Code – MSISDN to be created
* Hybrid\_Offerring\_Code/ Prepaid\_Offerring\_Code – subscriber type, WF12/NOF7

Given Existing subscriber is <Subscriber\_type> with offering code <Offerring\_Code>

* This will be used when you want to reuse an already existing subscriber and there is a not need to create a new subscriber.

### Airtime balance adjust

This will only apply to either prepaid or hybrid MSISDN. It will be required when there is a need to adjust the airtime balance counter.

And Subscriber <MonetaryType\_Code> balance is <Recharge\_amount>

- Allows for balance adjust to the specified Counter by <MonetaryType\_Code > String parameter.

NB: monetary codes are listed on the pricebook within the Monetary Type sheet. E.g “C\_VZA\_PPS\_MainAccount” OR “C\_VZA\_AirtimeAdvance”  
 Please do take note that the amount must be in Rands. E.g. <Recharge\_amount> = R20.00

### Fusion purchase request

**When** I purchase via fusion soid is **<SOID>** price is **<Purchase\_Amount>** bundle size is **<Bundle\_Siz**e> unit code is **<unit\_code>** validity period is **<validity\_period>**

- This will send a purchase/provisioning request to CCS via Fusion. This will work for both dynamic and static bundles.

**<unit\_code>** - unit code for the bundle that being purchased/provisioned. E.g. “GB”, “MB”,”MINUTES”,”UNITS”

### Verify CCS

**Verify Bundle Provisioned:**

When verifying an offer which is not prorated you can use either of the below

T**hen Data bundle for OfferingID <OfferingID> allocated is <Bundle\_Siz**e **>**

- This will verify if CCS bundle provisioning is correct or not. The bundle being verified will be identified by <OfferingID> parameter.

**NB**: This is strictly for **NON-PRORATED** bundles.

T**hen Data bundle for OfferingID <OfferingID> allocated is <Bundle\_Siz**e **> free unit is <FreeUnit> <Prorated>**

- This will verify if CCS bundle provisioning is correct or not. The bundle being verified will be identified by <FreeUnit> parameter.

- When the bundle being verified is not prorated, then the parameter <Prorated> should be set to “NO”

- When the bundle being verified is prorated, then the parameter <Prorated> should be set to “YES”

**Verify Airtime Balances:**

Then amount <Purchase\_Amount>ZAR is deducted from <Recharge\_amount> <MonetaryType\_Code> money counter

- This will verify if the expected amount has been deducted from CCS correct airtime counter for prepaid and hybrid plans.

Then amount <Purchase\_Amount>ZAR is credited into <MonetaryType\_Code> money counter

- This will verify if the expected amount has been credited into the money counter on CCS for prepaid and hybrid plans.

Then amount <Purchase\_Amount> is credited into <MonetaryType\_Code> airtime balance.

* This will verify if the expected amount was added into the credit\_limit counter strictly for **PostPaid**

**NB: monetary codes are listed on the pricebook within the Monetary Type sheet. E.g “C\_VZA\_PPS\_MainAccount” OR “C\_VZA\_AirtimeAdvance”**

And Verify instant Recharge limit is <InstantRechargeLimit>cents

**Verify Bundle Allocation Expiry Period:**

**And Expiry period is <Expiry\_Date> with Offerring ID <OfferingID>**

- This will verify if CCS bundle expiry period is correct as per the expected expiry date which is identified by the parameter <Expiry\_Date>. - The bundle being verified will be identified by <OfferingID> parameter.

- For this these bundles, the expiry period will be defined by number of days. **<Expiry\_Date>** = “30 days”,”14 days”, “1 days” etc.

**And Expiry period is <Expiry\_Date> with Offerring ID <OfferingID> and free unit <FreeUnit> calendar month <Prorated**>

- This will verify if CCS bundle expiry date is correct or not. The bundle being verified will be identified by <FreeUnit> parameter.

- At this point the assumption is that the bundle being verified is prorated, then the parameter <Prorated> should be set to “YES”

- In this case, it most likely that the expiry period is defined Calendar Month periods. **<Expiry\_Date>** = “1 months”, “2 months”, etc.

### Verify Fusion

This will verify fusion purchase responses.

This has been identified as a major course for test automation failure. **Test Analysts are encouraged to be certain if this is a requirement or not for the specific test.**

### Verify Service Balances (ViewThrough)

Then verify "C\_VZA\_PPS\_MainAccount" service balances remaining airtime

***Then verify fusion <FreeUnit> <OfferingID> <Product\_Name> <Product\_desc>***

*- This will perform a fusion viewthrough query and verify the product description.*

***Then fusion service bundle <FreeUnit> <OfferingID> for product <Product\_Name> is <Bundle\_Size> <Prorated>***

***-*** *This will verify if the provisioned bundle is as per the expected.*

*- This will verify the bundle based on the provided product name.*

***<Product\_Name> =*** *“1.2GB - 30 Days recurring bundle”*

***Then fusion service validity <FreeUnit> <OfferingID> <Validity\_Type> for product <Product\_Name> is <Expiry\_Date>***

***-*** *This will verify if the expiry date is as per the expected.*

*- This will verify the expiry date based on the provided product name.*  
***<Validity\_Type> =*** *“validityPeriod”*

**<Product\_Name> =** “1.2GB - 30 Days recurring bundle”

VOMS

When Subscriber redeems valid voucher for voucher type <Voucher\_type> via IVR

When Subscriber redeems valid voucher for voucher type <Voucher\_type> via USSD

When Subscriber redeems valid voucher for voucher type <Voucher\_type> via SMS

* For all three VOMS redeems functions, this will perform VOMS voucher redeem/recharge based on the provided Voucher\_type
* The function will automatically perform a voucher PIN generation if there is no existing and active voucher in batch stored in the database.