System Design Specification

For eApps

This document contains proprietary information that is confidential to the Monetary Authority of Singapore (MAS). Disclosure of this document in full or in part, may result in material damage to MAS. Written permission must be obtained prior to the disclosure of this document to a third party. Any hardcopy of this document, printed or photocopied, is considered an uncontrolled copy. The original, signed-off document is the controlled copy.

UPDATED BY

ROLE	NAME	SIGNATURE	DATE (DD-MM-YYYY)
Team Lead	DEEPAK CHAKRAWANI		

REVIEWED BY

ROLE	NAME	SIGNATURE	DATE (DD-MM-YYYY)
PM	LIH FEN (CRIMSONLOGIC)		

MDD

APPROVED BY

ROLE	NAME	SIGNATURE	DATE (DD-MM-YYYY)
User PM	YIP EE XIU (MDD)		
User PM	EUGENE LEONG (MDD)		

Change History

Version	Changes made	Date
2.0	1. SafeWord Configuration and authentication with SafeWord 2. Database structure 3. Deployment to WAS 4. Some screenshots 5. Appendix – SGS Daily Price calculation formulae and rules for T+1, T+2 and Repo T+1 price calculations	27/02/2006 (ZG)
3.0	1. Java classes for daily closing price and public announcement 2. JSP pages for daily closing price and public announcement 3. Remove SGS Decryption Module 4. New encryption/decryption method 5. Remove backend job for security update, auction result, producing encrypted bid data file	27/09/2008 (ZG)
4.0	1. Java classes for ATM submission 2. JSP pages for ATM submission 3. RSA authentication methods	03/03/2010 (ZG)
5.0	1. Modification for HA, i.e. WAS clustering 2. Modification for MAS Bills	14/04/2011 (ZG, THL)
6.0	1. Modification of the Infrastructure Diagram 2. Modification for iBanking project 3. Modification to cater for negative yield closing prices	12/09/2011 (THL)
6.1	1. Modification to the Daily Prices Spread table	15/03/2012 (THL)
6.2	1. File format changes for interface with Primary Dealers, to include sign for yields (Section 5.4, 5.5)	29/10/2012 (THL)
6.3	Daily Prices CSR 1. To include additional info in submission status 2. To remove the modify rights to spread data 3. To remove the modify rights to previous days Daily Prices	31/01/2013 (THL)
6.4	Encryption Key Changes	09/04/2013 (THL)

6.5	Logic for authentication, check if token belongs to PD	05/08/2013 (THL)
6.6	Daily Prices CSR a) New table to capture Actions for Closing Price/Announcements b) Combine the tasks for release of Closing Price c) Restrict re-generation/re-release of Closing Price d) Updated screenshots	24/10/2013 (BL/THL)
6.7	2013Q4 Enhancements 1. Removal of 6M T-bill (29 Mar 2014) and addition of 24W MAS Bill (2 Jan 2014) as benchmarks for closing prices. 2. Removal of comm bill and repo rate from closing prices submission and final closing prices report. 3. Updated Screenshots 4. Updates to include file format for closingprices.txt	31/12/2013 (THL)
6.8	2014Q4 Enhancement 1. Withdrawal of eApps bids by PD 2. Publishing Auction Results to PDs and the Public (eApps)	19/11/2014 (FL)
6.9	Savings Bond module Annex 7.12	23/09/2015 (THL)
6.10	Changes for the SSB Phase 3 1. Online web application for SGS Bonds and T-Bills to include SRS. 2. Upload bid file to include SRS 3. Standard the flow to retrieve the subscription file from MEFT-Access with acknowledgement file. 4. PGP encryption for all files 5. Auction results files for the banks to generate by application type	23/03/2018 (CL)
6.11	Section 4.1.56 and 3.3.16 Added new for Issuance Calendar. Section 5.4.2 and 5.5.1 Update of UOB file name with “SFTP” Section 7.3 and 5.4.6 Include the CPF validation logic. Section 5.4.2 Acknowledgment file generation for Input empty file and Input file with No Details records from PD’s	26/07/2018 (CL) 25/04/2019 (CL) 09/05/2019 (CL) 30/07/2019 (CL)

	Section 5.4.2 and 5.5.1 Update of the OCBC return filename – include "SSB" New section 7.13 The amended Announcement file for bonds and bills New section 7.14 The signal files and messages New section 7.12.1 The Signal file message response for Announcement to website New section 7.12.2 The Signal file message response for Result Released to website. New section 4.1.37.1 The Signal file message response for Daily Price Release. New section 4.1.57 a) Non-Benchmark Indicator message response. New section 4.1.57 b) The Signal file message response for MAS / T Bills for Announcement Section 4.1.53 Include the Signal file message response for Auction Release to Website. Section 5.8.2 Included the error messages for UI and Batch Section 3.3.20, 4.1.8, 4.1.9, 4.1.11 Updated the screens Section 7.15 WOGAA Changes	
6.12	Changes for CR - CuR-MAS-SBA-14 Section 2, 3.3.12, 7.7 To include USD Facility T3 Date and T3 Dirty Price in DB/Closing Price report Section 4.1.35, 4.1.44, 4.1.45 To add/update US Public Holiday details	09/04/2020 (CL)
6.13	Changes for CR - CuR-MAS-SBA-11 Section 3.4, 3.4.4, 5.3 bank.txt to include MEPS+ participant indicator Changes for CR - CuR-MAS-SBA-16 Section 2, 5.4.6, 7.10.2 To accept Negative Yield for Bid submission Changes for CR - CuR-MAS-SBA-13 Section 3.4.1, 5.8 To generate a Closing price file for AMMO (closingprice_ammo.txt)	03/06/2020 (CL)
	Changes for CR - CuR-MAS-SBA-15	01/09/2020

	Section 5.7, 7.15, 7.16 FRN and SORA	(CL)
6.14	Changes for CR - CuR-MAS-SBA-17 Section 2, 3, 3.1.3, 3.1.65, 3.2, 3.2.1.2, 3.2.1.8, 3.2.1.12, 3.2.3, 3.3.21, 4.1.1, 4.1.2, 4.1.3, 4.1.25, 6, 6.1.1, 7.1, 7.8, 7.11.2, 7.17, 7.18 eApps CorpPass MAF integration	03/09/2020 (CL)
6.15	Changes for CR - CuR-MAS-SBA-18 Section 7.16 FRN and SORA Section 3.3.22, 7.19 Antivirus scan for all file uploads Section 7.20 PDF files used in eApps application	05/10/2020 (CL)
6.16	Changes for CR - CuR-MAS-SBA-19 Section 4.1.1, 4.1.56, 7.21 SingPass Login Button and Footer	08/03/2021 (CL)
	Section 3.3.4, 3.4.3, 5.2 secint.txt file changes to add Coupon Payment Indicator for FRN	26/04/2021 (CL)
	Changes for CR - CuR-MAS-SBA-13 Section 3.4.1, 5.8 AMMO Closing price file (closingprice_ammo.txt) - should contain only issue codes which are issued and to schedule the job to run in morning for previous working day Closing Price.	24/06/2021 (CL)
6.17	Changes for CR – CuR-MAS-SBA-21 MQ Revamp Section 7.22 Steps to create the keystore/renew certificate/CA certs change	24/03/2022 (CL)
6.18	Changes for CR – CuR-MAS-SBA-20 Section 7.23 CMTB and SGS (Infra) Changes for CR – CuR-MAS-SBA-24 Section 7.24 Syndication Section 7.25 Upload File Size Restrictions in eApps	14/06/2022 (CL)

6.19	Changes for CR – CuR-MAS-SBA-25 Section 3.3.3, 3.3.23, 3.3.24, 5.4.4, 5.4.5, 5.4.6, 5.5.4, 5.9.1, 5.9.2, 7.26, 7.27, 7.28 Retail Bid File Process, Submission Inquiry, Unpublish results at MAS Website Section 7.22, 7.22.1 Steps to create the keystore/renew certificate/CA certs change	26/10/2022 (CL)
6.20	Changes for CR - CuR-MAS-SBA-30-eApps_SBA_MASNET-Replacement Section 5.5.1, 5.4.2, 5.4.4, 5.5.1, 5.5.5, 5.6 Decommission of MASNET-Access and migration to MEFT-Access, SFTP downstream file name modification	15/12/2023 (CL)
6.21	Changes for CR - CuR-MAS-SBA-26 - eApps 1.2 Release 1 and Release 2 Section 3.1.19, 3.1.20, 3.1.59, 3.1.60, 3.1.66, 3.1.67, 3.2.1.44, 3.2.1.45, 3.2.1.46, 3.3.25, 3.3.26, 4.1.21, 4.1.22, 4.1.23 Submission Inquiry, Unpublish results at MAS Website	04/01/2024 (CL)
6.22	Changes for CR – CuR-MAS-SBA-31 – eApps MASNET MAF Decom Section 3.1.3, 3.1.23, 3.2.1.2, 3.2.7, 4.1.1, 4.1.2, 4.1.26, 7.18, 7.29, 7.29.1, 7.29.1.1, 7.29.1.2, 7.29.1.3, 7.29.1.4, 7.29.1.5 CorpPass TOPPAN Integration	14/05/2024 (CL)

1 INTRODUCTION 1

1.1 OVERVIEW 1
1.2 AUDIENCE 1
1.3 SDS STRUCTURE 1

2 PROCESS 2

3 SYSTEM DESIGN 18

3.1 WEB FRONT-END INTERFACE DESIGN 19
3.1.1 MAS SGS HOME 19
3.1.2 SGSHELP.HTML 19
3.1.3 SGSLOGIN.JSP 19
3.1.4 AUTHENTICATIONFAILED.JSP 20
3.1.5 REDIRECT.JSP 20
3.1.6 SGSMENU.JSP 20
3.1.7 SGSSUBLIST.JSP 20
3.1.8 ISSUECLOSED.JSP 20
3.1.9 SGSSUBSCRIBE.JSP 21
3.1.10 SGSMASBILLSSUBSCRIBE.JSP 21
3.1.11 SGSRESUBMIT.JSP 21
3.1.12 SGSMASBILLSRESUBMIT.JSP 22
3.1.13 SGSSUBCONFIRM.JSP 22
3.1.14 SGSMASBILLSSUBCONFIRM.JSP 22
3.1.15 TRANSACTIONSUCCEEDED.JSP 22
3.1.16 MASBILLSTRANSACTIONSUCCEEDED.JSP 23
3.1.17 TRANSACTIONFAILED.JSP 23
3.1.18 MASBILLSTRANSACTIONFAILED.JSP 23
3.1.19 SGSINQUIRYLIST.JSP 23
3.1.20 SGSINQUIRYDISPLAY.JSP 24
3.1.21 SGSAUCTIONLIST.JSP 24
3.1.22 SGSAUCTIONDISPLAY.JSP 24
3.1.23 LOGOFF.JSP 25
3.1.24 TIMEOUT.JSP 25
3.1.25 THE ERROR PAGE 25
3.1.26 SGSMASDAILYPRICEMENU.JSP 25
3.1.27 SGSMASDAILYPRICESUBLIST.JSP 25
3.1.28 SGSMASDAILYPRICEDISPLAY.JSP 26
3.1.29 SGSMASDAILYPRICEINPUT.JSP 26
3.1.30 SGSMASDAILYPRICEREINPUT.JSP 26
3.1.31 SGSMASDAILYPRICECONFIRM.JSP 26
3.1.32 MASDPTRANSACTIONSUCCEEDED.JSP 26
3.1.33 MASDPTRANSACTIONFAILED.JSP 27
3.1.34 SGSMASDPEXCEPTIONREPORT.JSP 27
3.1.35 SGSMASPRICESPREADLIST.JSP 27
3.1.36 MASPRICESPREADSUCCEEDED.JSP 27
3.1.37 MASPRICESPREADFAILED.JSP 27
3.1.38 SGSMASPUBLICHOLIDAY.JSP 28

3.1.39 MASPUBLICHOLIDAYSUCCEEDED.JSP 28
3.1.40 MASPUBLICHOLIDAYFAILED.JSP 28
3.1.41 SGSUSERDAILYPRICEMENU.JSP 28
3.1.42 SGSUSERDAILYPRICEDISPLAY.JSP 28
3.1.43 SGSUSERDAILYPRICEFILEATTACH.JSP 28
3.1.44 SGSUSERDAILYPRICEINPUT.JSP 28
3.1.45 SGSUSERDAILYPRICEREINPUT.JSP 29
3.1.46 SGSUSERDAILYPRICECONFIRM.JSP 29
3.1.47 USERDPTRANSACTIONSUCCEEDED.JSP 29
3.1.48 USERDPTRANSACTIONFAILED.JSP 29
3.1.49 SGSPUBLICANNOUNCELIST.JSP 30
3.1.50 SGSPUBLICBONDANNOUNCE.JSP 30
3.1.51 SGSPUBLICTBILLANNOUNCE.JSP 30
3.1.52 SGSPUBLICTMASBILLANNOUNCE.JSP 30
3.1.53 SGSPUBLICRESULTLIST.JSP 30
3.1.54 SGSPUBLICBONDRESULT.JSP 30
3.1.55 SGSPUBLICTBILLRESULT.JSP 31
3.1.56 SGSPUBLICTMASBILLRESULT.JSP 31
3.1.57 MESSAGE.JSP 31
3.1.58 SGSSUBMITFILEATTACH.JSP 31
3.1.59 SGSUPLOADCONFIRM.JSP 31
3.1.60 SGSUPLOADRESULT.JSP 31
3.1.61 SGSANNOUNCEMENTMENU.JSP 32
3.1.62 SGSMASISSUECALENDARMENU.JSP 32
3.1.63 SGSMASISSUECALENDAR.JSP 32
3.1.64 SGSUSERISSUECALENDARMENU.JSP 32
3.1.65 CORPPASSAUTHORIZATIONFAILED.JSP 32
3.1.66 SGSAUCTIONMENU.JSP 33
3.1.67 SGSACKNOWLEDGEMENTFILEDOWNLOAD.JSP 33
3.2 WEB BACK-END DESIGN 34
3.2.1 JAVA SERVLETS AND JSP LOGIC DESIGN 34
3.2.2 DATABASE ACCESS 47
3.2.3 INTEGRATION WITH RSA/MAF AUTHENTICATION SERVER 47
3.2.4 ERROR LOGGING 47
3.2.5 INTEGRATION WITH ENCRYPTION MODULE 47
3.2.6 SOME MISCELLANEOUS JAVA CLASSES 48
3.2.7 INTEGRATION WITH TOPPAN AUTHENTICATION 48
3.3 ORACLE DATABASE DESIGN 49
3.3.1 "ABA0001_SECURITY_MASTER" 49
3.3.2 "ABA0005_RSA_TEXT_ENC_TRANS" 50
3.3.3 "ABA0006_AUCTION_RESULT" 51
3.3.4 “ABA0008_STAGE_SECURITY_MASTER” 52
3.3.5 “ABA0009_BANK_MASTER” 53
3.3.6 “ABA0010_ANNOUNCE_TEXT” 54
3.3.7 “ABA0011_DAILY_PRICE” 55
3.3.8 “ABA0012_DP_STATUS” 57
3.3.9 “ABA0013_PRIMARY_DEALER” 57
3.3.10 “ABA0015_PRICE_SPREAD” 58
3.3.11 “ABA0016_DAILY_EXTRA_DATA” (NO LONGER IN USE) 58
3.3.12 “ABA0017_FINAL_DAILY_PRICE” 59
3.3.13 “ABA0018_FINAL_EXTRA_PRICE” (NO LONGER IN USE) 60
3.3.14 “ABA0019_PUBLIC_HOLIDAY” 61
3.3.15 “ABA0020_STAGING_BANK_MASTER” 61

3.3.16 "ABA0021_ISSUE_CALENDAR" 62
3.3.17 "ABA0022_NON_BENCHMARK" 62
3.3.18 "ABA0502_PUB_KEY" 63
3.3.19 "ABA0023_AUDIT_ACTION" 63
3.3.20 "ABA0214_SB_CD_FILE_TYPE" 63
3.3.21 "ABA0030_CORP_PASS_MAPPING" 64
3.3.22 "ABA0032_EAPPS_CONFIG" 65
3.3.23 "ABA0007_DETAIL_AUCTION_RESULT" 65
3.3.24 "ABA0004_RETAIL_BID_TRANS" 66
3.3.25 "AQA0020_AUCTION_RESULTS_REPORT" 67
3.3.26 "ABA0035_RETAILBID_FILE" 68
3.4 BACK-END AUTOMATION DESIGN 70
3.4.1 AUTOSYS JOBS 71
3.4.2 LOAD NEW SECURITY SUB-JOB 74
3.4.3 LOAD SECURITY DAILY UPDATE SUB-JOB 75
3.4.4 LOAD BANK LIST 78

4 USER INTERFACE 79

4.1 WEB INTERFACE 79
4.1.1 LOGON PAGE TO SGS EAPPS 79
4.1.2 AUTHENTICATION FAILED PAGE 81
4.1.3 SGS EAPPS HOME PAGE 82
4.1.4 SGS ONGOING ISSUE LIST PAGE 83
4.1.5 SGS ISSUE CLOSED PAGE 84
4.1.6 SGS APPLICATION FORM PAGE 85
4.1.7 HELP ON FILLING UP FORM 86
4.1.8 ERROR PAGE FOR SGS APPLICATION FORM 87
4.1.9 SGS APPLICATION CONFIRMATION PAGE 88
4.1.10 POP-UP PRINTER WINDOW ON TRANSACTION SUCCEEDED PAGE 90
4.1.11 SGS APPLICATION SUCCEEDED PAGE 91
4.1.12 SGS APPLICATION FAILED PAGE 92
4.1.13 BID FILE UPLOAD MENU 93
4.1.14 BROWSE UPLOAD FILE 94
4.1.15 CONFIRM SUBMISSION – FILE REJECTED 95
4.1.16 CONFIRM SUBMISSION – ALL RECORDS REJECTED 96
4.1.17 CONFIRM SUBMISSION – PARTIAL RECORDS REJECTED 98
4.1.18 CONFIRM SUBMISSION – ALL RECORDS ACCEPTED 99
4.1.19 SUBMISSION RESULT – PARTIAL RECORDS REJECTED (31-10-2013 DOES NOT DISPLAY) 100
4.1.20 SUBMISSION RESULT – ALL RECORDS ACCEPTED 101
4.1.21 SGS INQUIRY LIST PAGE 102
4.1.22 SGS INQUIRY DISPLAY PAGE 103
4.1.23 SGS AUCTION RESULT 105
4.1.24 SGS AUCTION RESULT LIST PAGE 107
4.1.25 SGS AUCTION RESULT DISPLAY PAGE 107
4.1.26 LOGOFF PAGE 109
4.1.27 TIMEOUT PAGE 110
4.1.28 HELP PAGE 110
4.1.29 DAILY CLOSING PRICE MENU FOR BANKS 111
4.1.30 UPLOAD DAILY CLOSING PRICE FILE 111
4.1.31 RE-ENTER DAILY CLOSING PRICE 112
4.1.32 CONFIRM DAILY CLOSING PRICE 113

4.1.33 SUBMISSION SUCCEEDED PAGE FOR DAILY CLOSING PRICE FILE 114
4.1.34 INQUIRE DAILY CLOSING PRICE 115
4.1.35 MODIFY DAILY CLOSING PRICE 115
4.1.36 DAILY CLOSING PRICE MENU FOR MAS 117
4.1.37 DISPLAY TODAY'S SUBMISSION STATUS 118
4.1.38 VIEW SUBMISSION 120
4.1.39 EDIT SUBMISSION 121
4.1.40 CONFIRM SUBMISSION 122
4.1.41 SUBMISSION SUCCEED 123
4.1.42 RUN EXCEPTION REPORT 124
4.1.43 VIEW PRICE SPREAD 124
4.1.44 INPUT PRICE SPREAD SUCCEED 125
4.1.45 MANAGE PUBLIC HOLIDAY 125
4.1.46 MANAGE PUBLIC HOLIDAY SUCCEED 126
4.1.47 RUN DAILY CLOSING PRICE REPORT 127
4.1.48 NEW ISSUE ANNOUNCEMENT LIST 129
4.1.49 MAS BILL NEW ISSUE ANNOUNCEMENT 129
4.1.50 T-BILL NEW ISSUE ANNOUNCEMENT 130
4.1.51 BOND NEW ISSUE ANNOUNCEMENT 131
4.1.52 AUCTION RESULT LIST 132
4.1.53 T-BILL AUCTION RESULT 135
4.1.54 BOND AUCTION RESULT 136
4.1.55 ISSUANCE CALENDAR 137
4.1.56 ANNOUNCEMENT 138
4.1.57 FOOTER 140

5 DATA FORMAT 142

5.1 LOAD NEW SECURITY ISSUES INTO ORACLE TABLE 142
5.2 LOAD SECURITY DAILY UPDATES (SECINT.TXT) INTO ORACLE TABLE 143
5.3 LOAD BANK LIST INTO ORACLE TABLE 145
5.4 FILE FORMAT OF APPLICATION DATA FILE FOR BANK UPLOAD FILE 146
5.4.1 OVERVIEW 146
5.4.2 ASSUMPTION 146
5.4.3 PROCESS FLOW 149
5.4.4 PROCESS 150
5.4.5 FORMAT 155
5.4.6 FIELD DEFINITIONS 156
5.5 THE FORMAT OF AUCTION RESULT FILE FOR BANKS 160
5.5.1 OVERVIEW 160
5.5.2 PROCESS FLOW 161
5.5.3 PROCESS 161
5.5.4 FORMAT 162
5.5.5 FIELD DEFINITIONS 163
5.6 THE FORMAT OF ISSUANCE CALENDAR FILE 166
5.6.1 FORMAT 166
5.6.2 FIELD DEFINITIONS 167
5.7 CLOSING PRICE FILE FOR MEPS+ AND FMBS 168
5.8 CLOSING PRICE FILE FOR AMMO 169
5.9 FILE FORMAT OF APPLICATION DATA ACKNOWLEDGEMENT FILE FROM BANK UPLOAD FILE 170
5.9.1 FORMAT 170

5.9.2 FIELD DEFINITIONS 172

6 DEPLOYMENT 176

6.1 DEPLOY .EAR FILE USING WAS ADMIN CONSOLE 176
6.1.1 CONFIGURING INTERFACE WITH RSA 176
6.2 COPYING UNIX SCHEDULED JOB AND ORACLE JOB FILES TO PRDSYS 177
6.2.1 PREREQUISITES: 177
6.2.2 DIRECTORY STRUCTURE ON PRDSYS: 177
6.2.3 COPY FILES 177
6.2.4 SCRAMBLE ORACLE ID AND PASSWORD FILE 177
6.2.5 UNIX RUN-TIME SETTING 178
6.2.6 SCHEDULE JOBS 178
6.3 CREATING ORACLE TABLES 179

7 APPENDIX 180

7.1 ETENDER NETWORKED COMPONENTS 180
AUCTION SYSTEM 180
INTERNAL SERVER SEGMENT 180
7.2 NRIC NO. VALIDATION ALGORITHM 181
7.3 CPF NO. VALIDATION ALGORITHM 185
7.4 BID CHECKSUM ALGORITHM 190
7.5 WARNING MESSAGES OF BACKEND SCHEDULED JOBS FOR OPERATORS 191
7.6 FORMULAE USED IN SGS DAILY PRICE CALCULATION 192
7.6.1 FORMULAE FOR T-BILL 192
7.6.2 FORMULAE FOR BOND 193
7.7 CALCULATION OF T+1, T+2, T+3 AND REPO T+1 PRICES, T+3 PRICES 197
7.8 OTHERS 199
7.9 DATA SCRAMBLE/UNSCRAMBLE ALGORITHM FOR BANK FILES (THIS IS NO LONGER IN USED) 200
7.9.1 REQUIREMENT 200
7.9.2 ENCRYPTION 200
7.9.3 DECRYPTION 202
7.9.4 CONVERT THE HEXADECIMAL (HEX) STRING INTO ITS CHARACTER EQUIVALENT 204
7.10 VALIDATION OF APPLICATION FORM 205
7.10.1 REQUIRED FIELDS 205
7.10.2 VALIDATION RULES 205
7.11 CONFIGURATION AND DEPLOYMENT OF RSA AUTHENTICATION 206
7.11.1 CONFIGURATION OF RSA AUTHENTICATION AGENT 206
7.11.2 DEPLOYMENT OF RSA AUTHENTICATION 206
7.11.3 POSSIBLE FAILURE 206
7.12 SAVINGS BOND MODULE 207
7.12.1 ANNOUNCEMENT 207
7.12.2 OVERALL RESULTS 209
7.13 ANNOUNCEMENT FILE. 210
7.14 SIGNAL FILE AND MESSAGES 211
7.15 WOGAA CHANGES 211
7.16 FRN AND SORA 212
7.17 CORPPASS AND MAF INTEGRATION 213
7.17.1 WEBSPHERE CONFIGURATION CHANGES 213

7.17.2 JAVA CODE CHANGES 215
7.17.3 MAF CONFIGURATION 221
7.17.4 NETWORK CONFIGURATION 222
7.18 CODE CHANGES FOR NON-CORPPASS LOGIN 222
7.19 ANTIVIRUS SCAN 223
7.20 PDF FILES USED IN EAPPS 226
7.21 FOOTER – PRIVACY STATEMENT AND TERMS OF USE 227
7.22 MQ REVAMP – STEPS TO CREATE KEYSTORE/RENEW CERTIFICATE/CA CERTS CHANGE 227
7.22.1 STEPS TO GENERATE ROOT/INTERMEDIATE/SIGNED CA CERTS 232
7.23 CMTB AND SGS (INFRA) 239
7.24 SYNDICATION 239
7.25 UPLOAD FILE SIZE RESTRICTIONS IN EAPPS 239
7.26 SUBMISSION INQUIRY CHANGES IN EAPPS 241
7.27 UNPUBLISH RESULTS AT MAS WEBSITE 249
7.28 WAF CHANGES 253
7.29 CORPPASS TOPPAN INTEGRATION 254
7.29.1 JAVA CODE CHANGES 254

1 Introduction

1.1 Overview

The System Design Specifications (SDS) document was prepared as an output of the SGS eApps System Design. This document includes the detail description of some of the major processes of the SGS eApps System.

1.2 Audience

This document is intended for use by any person involved in the development, approval and maintenance of the SGS eApps System Project. It is also targeted for the user who requires a high-level view of the application.

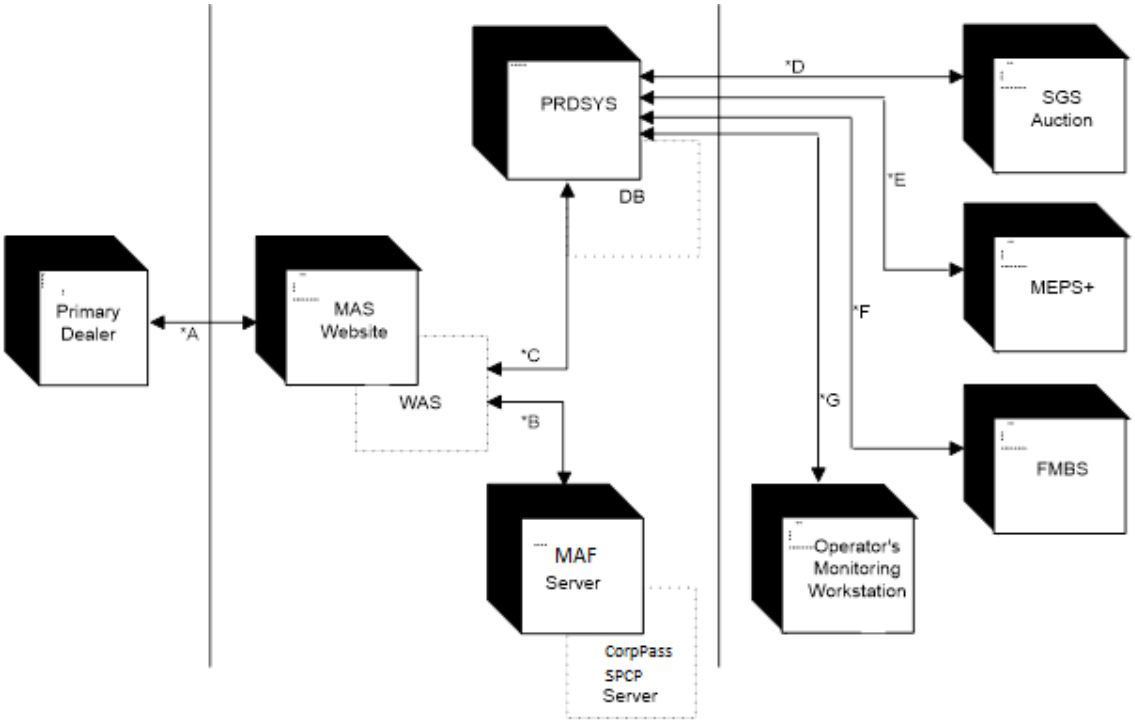
1.3 SDS Structure

This document contains:

Section 2	Processes This section focuses on the application architecture and overall system flow of the SGS eApps System.
Section 3	System design Programs of each component of the SGS eApps System.
Section 4	User interface Screen format of the SGS eApps System.
Section 5	Data format Table structures, and other data formats.
Section 6	Deployment System setup and program installation.

2 Process

Overall Process of SGS eApps System



Legend:

*A For PD and MAS users: 1. Logon 2. View help documents in pdf format 3. Select security to subscribe 4. Submit application 5. Confirm submission 6. View submission result 7. Upload application files 8. Confirm upload 9. View upload result 10. Select security to inquire bid submission 11. View all bid submissions 12. Select security to inquire auction result 13. View auction result 14. Submit daily closing price 15. Confirm daily closing price submission 16. View daily closing price submission result 17. Inquire today's daily closing price 18. Timeout For MAS users: 19. Submit daily closing price for bank 20. Confirm daily closing price submission 21. View daily closing price submission result 22. Inquire daily closing price 23. Inquire today's submission status 24. Run today's exception report 25. Run Today's daily closing price report 26. View daily closing spread 27. Manage public holiday 28. Update Financial Database (SGS Website) 29. Send today's price to MEPS+ and FMBS[1] For public: 30. View new issue announcement page 31. View auction result announcement page	*B 1. Authenticate with MAF server *C 1. Encrypt data 2. Save data in Oracle (prdsys) 3. Retrieve data from Oracle (prdsys) upon request 4. Format data for display *D 1. SGS Auction system reads data from eApps 2. SGS Auction system updates auction result *E 1. MEPS+ sends security master file to eApps 2. MEPS+ sends security update file to eApps 3. MEPS+ sends bank list file to eApps 4. eApps sends closing price file to MEPS+ 5. eApps sends previous month's closing price file to MERIT (added Feb 2013) *F 1. eApps sends closing price file to MEPS+ *G 1. Notify operator if data loading error

[1] Functions 28/29 have been combined to one function "Release today's final price (includes update financial database and sending files to MEPS+ and FMBS)"

A.1. Logon

1. Primary Dealer visits MAS web page to subscribe Singapore Government Securities or MAS Bills, view bid submissions or view auction result.
2. Primary Dealer accesses SGS eApps Logon web page.
3. Primary Dealer clicks on the CorpPass Login URL.
4. Primary Dealer inputs their CorpPass UEN/CorpPass ID/Password.
5. If Primary Dealer logons successfully, he/she will be redirected to SGS eApps menu page.
6. If Primary Dealer fails to logon, a logon failure page will be displayed.

Business Rules:
√ Primary Dealer will be required to logon for subscribing Singapore Government Securities or MAS Bills, viewing bid submissions and viewing the auction result.
√ Each Primary Dealer has to register their CorpPass User details with eApps to access eApps application
√ System generated token user id will be assigned in the format 'CxxxxT01' and once reaches 'CxxxxT99', the next Token User ID generated would be 'DxxxxT01'. Similar pattern would be followed until the Token User ID reaches 'ZxxxxT99', where xxxx is the bank code
√ Primary Dealer can login using their registered CorpPass User details UEN/CorpPass ID/Password

A.2. View help documents in pdf format

1. PDs can click on links on the logon page to view 3 help documents in pdf format – Info on SGS eApps, User Guide, and form T1.
2. PDs can click on links on the SGS help page to view 3 help documents in pdf format – Info on SGS eApps, User Guide, and form T1.
3. PDs can click on link on SGS page header to view FAQ document in pdf format.
4. PDs can click on link on bid application form page to view help document of how to fill up the form

Business Rules:
√ All the help documents are published in pdf format.
√ ITD, IAD, FD and MDD will all participate in the maintenance of help documents and MDD will finalize the changes.
√ PDs need to have Acrobat Reader installed in order to view those documents.

A.3. Select security to subscribe

1. After Primary Dealer logons and selects "Subscription", he/she will be shown a list of ongoing Singapore Government Securities and MAS Bills.
2. Primary Dealer selects one security to subscribe.

Business Rules:
√ If one security's tender date is later than today's date, it is considered as ongoing issue.
√ If one security's tender date is same as today's date, and the current time is earlier than 12:00pm, it is considered as ongoing issue.
√ The ongoing Singapore Government Securities and MAS Bills will be listed according to alphabetic order.
√ After Primary Dealer selects a security from the list, it assumes that Primary Dealer is subscribing the selected security code with the LATEST issue no.

A.4. Submit application

1. After Primary Dealer selects a security, a blank application form will be shown.
2. Primary Dealer fills up the application form.
3. Primary Dealer clicks on the "Submit" button.
4. The validation procedures will take place.
5. The application will be submitted if it passes the validation.

Business Rules:

√ The list of field names on the application form is shown in Section 4.

√ The values of issue code, issue name, issue date, and Primary Dealer's token user id with MAS are displayed automatically. Those fields are read only.

√

√ The application form validation rules for T-bills and bonds are as follows:

- There must be at least one row of bid.
- Bids must be inputted row by row in order.
- The **Nominal Amt** must be in multiple of one thousand with a minimum **Nominal Amt** of 1,000.
- **Yield** must be numeric up to 2 decimal places.
- Yield range accepted differs based on the "**mnet.acceptNegativeYield**" property value defined in sgs.properties file
  - If the property value is true, then Yield range accepted is -99.99 to 99.99 (0 is a valid yield)
  - If the property value is empty or false, then Yield range accepted is 0.01 to 99.99 (0 is not a valid yield)
- If **Competitive** Bid selected, then **Yield** must not be empty.
- If **Yield** entered, the **Competitive** must be 'Y'.
- If **Nominal Amt** is empty, then **Competitive** must be empty and **Yield** must be empty.
- **Nationality** must be S/F/_.
- **Type of applicant** must be:
  - Primary Dealer (PD)
  - Full Bank (FB)
  - Restricted Bank (RB)
  - Offshore Bank (OB)
  - Other Bank (NB)
  - Merchant Bank (MB)
  - Investment Advisor (IA)
  - Finance Company (FC)
  - Insurance Company (INS)
  - Statutory Board, Government-Linked Company (SB)
  - Individual (IND)
  - Singapore Exchange (SGX)
  - Monetary Authority of Singapore (MAS)
  - Others (OTH).
- If **Type of applicant** is "PD", "FB", "RB", "OB" or "MAS", **Securities account** type must be "Bank".
- If **Type of applicant** is not "PD", "FB", "RB", "OB" or "MAS", **Securities account** type must be "Bank's customer".
- If **Type of applicant** is "PD", **Securities account No** must be the same as the account no in the token id.
- **Name of applicant** - not compulsory.
- **Applicant's address** – not compulsory.
- **NRIC No/Passport No**[2] – not compulsory.
- **Application Type** – List box selection, CASH/SRS/CPF
  If **"SRS" or "CPF"** selected, then **CPF/SRS account No** – is compulsory
  else CPF/SRS account NO – not compulsory
- **CDP account No** – not compulsory.

[2] S-Series and T Series NRIC numbers for Singaporeans will be validated

- **Token user id of primary dealer** – PD's logon token id
- **Cust_Bank** must be B/C
- If **Cust_Bank** = "C", **Name of Applicant** must not be empty
- If **Name of Applicant** = empty, **Nationality** must be blank
- If **Type of Applicant** = "IND",
  - **Nationality**,
  - **NRIC No/Passport No**, must not be blank
- If **Type of Applicant** != "IND",
  - **Nationality**,
  - **NRIC No/Passport No**,
  - Selection of Application type – not compulsory
  - **CPF/SRS Acc No** must be blank
- If **Nationality** not empty, **Type of Applicant** must be "IND"
- If **NRIC No/Passport No** not empty, **Type of Applicant** must be "IND"
- ~~If **Using CPF Funds** is "Y",~~
- If Application type is selected (CASH/SRS/CPF)
  - **Type of Applicant** must be "IND"
  - First Character of **Issue Code** must be "N" or "B" (i.e. bond or T-Bill)
  - If Application type is SRS or CPF
    - **CPF/SRS Acc No** must not be empty
  - If **Nationality** is 'S', **CPF Acc No** must be same as the right 8 characters of **NRIC No**
  - If **Nationality** is not 'S', **CPF Acc No** must pass validation.
- If **CPF/SRS Acc No** not empty,
  -
  - **Type of Applicant** must be "IND"
  - **Application Type must be "SRS" or "CPF"**
  - First Character of **Issue Code** must be "N" or "B" (i.e. bond or T-Bill)
  - Below validation logic needs to be applied to validate the CPF number in eApps and eAuction
    - For Singapore Citizen/PR, CPF No should match the same as NRIC number (Length – 9 characters)
    - For Foreigner, CPF No starts with S or T, followed by the letter 'A', 'B', 'C', 'D,' 'F', or 'Y' (Length – 9 characters). Refer Section 7.3 for validation logic
  - If the CPF Acct No. is empty, then display the error "The CPF Account No. should not be empty."

- **AccMAS** must be entered.
- **AccMAS** must be all digits.
- **AccMAS** must be bigger than 7000, or 2500.
- **AccMAS** must be validated against backend auto-debit bank list.
- For **Token id** = 2500, **TypeOfApplicant** must be MAS, **AccMAS** must be 2500
- **NRIC No/Passport No** – Singapore NRIC no's checksum must be correct
- **Bank's Ref. No** must be entered and must be unique.

√ The application form validation rules for MAS Bills are as follows:

- There must be at least one row of bid.
- Bids must be inputted row by row in order.
- The **Nominal Amt** must be in multiple of one thousand with a minimum **Nominal Amt** of 1,000.
- **Yield** must be numeric up to 2 decimal places.
- Yield range accepted differs based on the "**mnet.acceptNegativeYield**" property value defined in sgs.properties file
  - If the property value is true, then Yield range accepted is -99.99 to 99.99 (0 is a valid yield)
  - If the property value is empty or false, then Yield range accepted is 0.01 to 99.99 (0 is not a valid yield)
- Only **Competitive** Bids are accepted.
- If **Nominal Amt** is empty, then **Yield** must be empty.
- **Type of applicant** must be:
  - Primary Dealer (PD)

- Full Bank (FB)
- Restricted Bank (RB)
- Offshore Bank (OB)
- Other Bank (NB)
- Merchant Bank (MB)
- Finance Company (FC)
- Insurance Company (INS)
- Statutory Board, Government-Linked Company (SB)
- Singapore Exchange (SGX)
- Others (OTH).

- If **Type of applicant** is "PD", "FB", "RB", "OB", **Securities account** type must be "Bank".
- If **Type of applicant** is not "PD", "FB", "RB", "OB", **Securities account** type must be "Bank's customer".
- If **Type of applicant** is "PD", **Securities account No** must be the same as the account no in the token id.
- **Name of applicant** - not compulsory.
- **Applicant's address** – not compulsory.
- **Token user id of primary dealer** – PD's logon token id
- **Cust_Bank** must be B/C
- If **Cust_Bank** = "C", **Name of Applicant** must not be empty
- **AccMAS** must be entered.
- **AccMAS** must be all digits.
- **AccMAS** must be bigger than 7000.
- **AccMAS** must be validated against backend auto-debit bank list.
- **Bank's Ref. No** must be entered and must be unique.

A.5. Confirm submission

1. After Primary Dealer submits the application, a confirmation page of the submission will be displayed.
2. PD clicks on "Confirm" to confirm the submission.
3. PD clicks on "Cancel" to make amendments to the bid application.

Business Result:

√ The confirmation page has all the bid details that PD inputted just now and the transaction date and time.

A.6. View submission result

1. After Primary Dealer confirms the submission, the result of whether the submission is successful will be displayed.

Business Result:

√ The Primary Dealer will be given a submission failure message if there is any problem with back-end system (e.g. encryption problem).

√ If the submission is successful, the Primary Dealer will be given a "Transaction Succeeded" page.

√ The "Transaction Succeeded" page has all the bid details, transaction date and time, and reference no. for the transaction.

√ Reference no. is unique for each transaction. It is a 5-digit number and the range is from 00001 to 99999. It starts from 00001 and is auto-incremented after each transaction.

A.7. Upload application file

- After Primary Dealer selects a security, a file upload form will be shown.

- Primary Dealer selects a file.
- Primary Dealer clicks on the "Submit" button.
- The System will decrypt the file on the fly.
- The validation procedures will take place.
- The file will be submitted if it passes the validation.

Business Rules:
√ Only 3 local banks are able to use this function.
√ The file format is as section 5.4.
√ The file should be PGP encrypted.
√ The file may contain bids for multiple issues.
√ The file validation rules are as section 7.10.

A.8. Confirm upload

1. After Primary Dealer submits the file, a confirmation page of the submission will be displayed.
2. PD clicks on "Confirm" to confirm the submission.
3. PD clicks on "Cancel" to make amendments to the bid application.

Business Result:
√ The confirmation page has all the bid details that PD inputted just now and the transaction date and time.
√ The confirmation page will display all bids accepted and all bids rejected.
√ If a bid is rejected, the reason for rejection will be shown.

A.9. View submission result

i) After Primary Dealer confirms the submission, the result of whether the submission is successful will be displayed.

Business Result:
√ The Primary Dealer will be given a submission failure message if there is any problem with back-end system (e.g. encryption problem).
√ If the submission is successful, the Primary Dealer will be given a "Transaction Succeeded" page.
√ The "Transaction Succeeded" page has all the bid details, transaction date and time, and reference no. for the transaction.
√ Reference no. is unique for each transaction. It is a 5-digit number and the range is from 00001 to 99999. It starts from 00001 and is auto-incremented after each transaction.

A.10. Select security to Inquire

1. After Primary Dealer logons and selects "Submission Inquiry", he/she will be shown a list of ongoing Singapore Government Securities and MAS Bills.
2. Primary Dealer selects one security to inquire.

Business Rules:
√ If one security's tender date is later than today's date, it is considered as ongoing issue.
√ If one security's tender date is same as today's date, and the current time is earlier than 12:00pm, it is considered as ongoing issue.
√ The ongoing Singapore Government Securities and MAS Bills will be listed according to alphabetic order.
√ After Primary Dealer selects a security from the list, it assumes that Primary Dealer is inquiring the selected security code with the LATEST issue no.

A.11. View bid submissions

1. Security Code, Bank's Ref No, Transaction Ref No, and Received Date and Time are shown on the display page of bid submissions.

Business Rules:
√ PD is only able to view bid submissions of its own bank.
√ MAS MDD can view bid submissions of all banks.
√ Bid submissions are display according to the alphabetic order of Bank's Ref No.

A.12. Select security to inquire auction result

1. After Primary Dealer logons and selects "Auction Result", he/she will be shown a list of Singapore Government Securities and MAS Bills in two sections – today's result and recent one month's result.
2. Primary Dealer selects one security to inquire.

Business Rules:
√ If there is no auction result for today, display "No auction result for today".
√ If there is no auction result for recent one month, display "No auction result for recent one month"
√ Results for PDs will only be available after MAS users publish the result.

A.13. View auction result

1. Auction result is displayed.

Business Rules:
√ PD is only able to view auction result of its own bank.
√ Use monospace font to display auction result for better alignment.

A.14. Submit daily closing price

1. PD submit daily closing price by attaching a .csv file
2. PD clicks on the "Submit" button.
3. The validation procedures will take place.
4. The application will be submitted if it passes the validation.

Business Rules:
√ Banks may upload daily price in ".csv" file format.
√ The first line contains text "Date:" and today's date. Date should be in "dd/mm/yyyy" format.
√ Every SGS issue and MAS Bills issue should occupy one line.
√ For SGS and MAS Bills issues, each line should start with issue code, followed by maturity date, bid/offer prices, and high/low price if the issue is a benchmark one.
√ For Bonds, the prices are being input. For T-bills and MAS Bills, the yield is being input. Negative yields are allowed.
√ For High/Low Benchmark prices, banks may input '-' if no data is available. Likewise negative yields are allowed.

A.15. Confirm daily closing price submission

4. After PD uploads the daily closing price file, a confirmation page of the submission will be displayed.
5. PD clicks on "Confirm" to confirm the submission.
6. PD clicks on "Cancel" to make amendments to the daily closing price submission.

Business Result:
√ The confirmation page has all the daily closing price details that PD uploaded in the file just now and the transaction date and time.

A.16. View daily closing price submission result

1. After PD confirms the submission, the result of whether the daily closing price submission is successful will be displayed.

Business Result:
√ If the submission is successful, the PD will be given a "Transaction Succeeded" page.
√ The "Transaction Succeeded" page has all the daily closing price details, transaction date and time.

A.17. Inquire daily closing price submission

1. After PD logons and selects "Input/View/Modify today's data", and if the PD has already submitted today's daily closing price, he/she will be shown the details of today's daily closing price submission.

Business Rules:
√ If there is no submission today, display the file upload page instead.

A.18. Timeout

1. If the Primary Dealer has been idle for 15 minutes after logon, the session will automatically timeout. He/she has to logon again for any further transactions.

A.19. Submit daily closing price for bank

1. MAS MDD user submits daily closing price using web online form
2. User clicks on the "Submit" button.
3. The validation procedures will take place.
4. The application will be submitted if it passes the validation.

Business Rules:
√ MAS MDD users input bank code first to identify the bank.
√ MAS MDD users can only modify for today's data
√ Users must input bid/ask prices for all ongoing (not matured) issues.
√ Users must input highest/lowest done prices for all benchmark issues.
√ For Bonds, the prices are being input. For T-bills and MAS Bills, the yield is being input. Negative yields are allowed.
√ For High/Low Benchmark prices, users may input '-' if no data is available. Likewise negative yields are allowed.

A.20. Confirm daily closing price submission

7. After MAS MDD user submit daily closing price, a confirmation page of the submission will be displayed.
8. User clicks on "Confirm" to confirm the submission.
9. User clicks on "Cancel" to make amendments to the daily closing price submission.

Business Result:
√ The confirmation page has all the daily closing price details that MAS MDD user entered in the online form just now and the transaction date and time.

A.21. View daily closing price submission result

1. After MAS MDD user confirms the submission, the result of whether the daily closing price submission is successful will be displayed.

Business Result:
√ If the submission is successful, the MAS MDD user will be given a "Transaction Succeeded" page.
√ The "Transaction Succeeded" page has all the daily closing price details, transaction date and time.

A.22. Inquire daily closing price submission

1. After MAS MDD user logons and selects "Input/View/Modify today's data for bank xxxx", and if the bank or MDD user has already submitted today's daily closing price for the bank, he/she will be shown the details of today's daily closing price submission.

Business Rules:
√ If there is no submission today, display an empty online form instead.

A.23. Inquire today's submission status

1. After MAS MDD user logons and selects "Display today's submission status", a page will display all PD's bank code, whether the bank has submitted, and whether the submission is editable. It will also include the last updated date/time and last updated by token id.

Business Rules:
√ MAS MDD user can click the bank code to modify the submission status from final to draft.
√ MAS MDD user can also enter/modify data for the bank.

A.24. Run today's exception report

1. After MAS MDD user logons and selects "Runt today's exception report", a page will display all the exceptions if any. This is to check whether there is any abnormality in today's submission.

Business Rules:
√ Banks that have not submitted today's price will be displayed.
√ Any data that exceeds the predefined high/low spread will be displayed.

A.25. Run today's daily closing price report

1. After MAS MDD user logons and selects "Download today's report", the system will consolidate all the daily price submitted by banks and generate today's report in .csv format.

Business Rules:
√ System will calculate average price for each issue.
√ System will calculate T-bill price based on the average yield.
√ System will calculate Bond's yield based on the average price.
√ System will calculate accrued interest for bonds.
√ System will calculate dirty price for bonds.
√ Closing Prices must not be released.

A.26. View daily price spread

1. After MAS MDD user logons and selects “View daily price spread”, the system displays an online form.

Business Rules:
√ If there is stored daily price spread, system will display those values in the form. Otherwise, system will display an empty form.

A.27. Manage public holiday

1. After MAS MDD user logons and selects “Manage public holiday”, the system displays an online form.

Business Rules:
√ User needs to input a year first and select the country code from drop down.
√ If there is stored value for public holiday, system will display those values in the form. Otherwise, system will display an empty form.

A.28-29. Release today's final price (includes update financial database and sending files to MEPS+ and FMBS)

i. After MAS MDD user logons and selects “Release today's final price”, the system will generate daily closing price files and put them in Unix file system. AutoSys jobs will them pick up those file and send them to MEPS+ and FMBS.
ii. For FMBS, the sign for the yield will also be passed before the yield value. For values >= 0, ‘+’ will be passed. For values <0, ‘-‘ will be passed.
iii. The system will also update the database tables in Financial Database System.

Business Rules:
√ SGS eApps uses MQ to send those files.
√ SGS eApps updates Financial Database System via database link.

A.30. View new issue announcement page

- When there is a new issue, public can visit this page for the particulars of this issue.

Business Rules:
√ This page display the issue name, issue description, issue code, ISIN code, issue date, issue amount, method of submission, and closing time, etc.

A.31. View auction result announcement page

3 After an auction, public can visit this page for the auction result.

Business Rules:
√ This page display the average yield, cutoff yield, average price, cutoff price, and cutoff percentage, etc.

B.1. Authenticate with MAF server

1. Once the Primary Dealer inputs CorpPass login details in CorpPass Login page, CorpPass SPCP server authenticates and send the authentication details to MAF server
2. If the authentication is failed, CorpPass shows the logon failure message
3. After servlet on MAS Web site receives Primary Dealer's logon information from MAF server, it will authorize against the Corp Pass Mapping table in eApps Database .
4. If the authorization is failed, servlet will give Primary Dealer an authorization failure web page.

Business Rules:

√ Servlet only uses Primary Dealer's CorpPass logon details received from MAF for authorization. No digital certificate is involved.

C.1. Encrypt data

1. After servlet receives the application data from Primary Dealer, it will be decrypted using RSA algorithm.

Business Rules:
√ The servlet will assemble the application data to a string according to the format of Section 5.
√ The servlet will retrieve the public key stored in database.
√ The public key is used to encrypt the application data.

C.2. Save data in Oracle

1. After data encryption, the encrypted data will be saved in Oracle (on prdsys).
2. The bank's reference no. is stored in clear format.
3. The reference no. for the current security will be incremented by 1.

Business Rules:
√ A common Oracle id is used.

C.3.Retrieve data from Oracle upon request

1. Servlet issues SQL command to retrieve data from Oracle upon Primary Dealer's request of view bid submissions, auction result, view daily closing price submission, etc.

Business Rules:
√ A common Oracle id is used.

C.4.Format data for display

1. After servlets get data from Oracle, the data is formatted for display.

Business Rules:
√ Use monospace font for auction result.
√ Bid submissions are displayed in a table with 4 columns: security code, bank's ref no, transaction ref no, and received date and time.
√ Daily closing price is display in an online form.

D.1. SGS Auction system reads data from eApps

1. When SGS Auction System performs some functions, it reads data from SGS eApps database, such as security master, bank list, etc.

Business Rules:
√ SGS Auction System access SGS eApps database via database link.

D.2. SGS Auction system updates auction result

1. After an auction, FD user releases auction result. SGS Auction system will update auction result tables in SGS eApps database.

Business Rules:
√ SGS Auction System access SGS eApps database via database link.

E.1. MEPS+ sends security master file to eApps (prior to any announcements)

1. After FD user enters a new issue in MEPS+ and triggers sending, a security master file will be sent to PRDSYS.
2. AutoSys job will load the file into database.

Business Rules:
√ MEPS+ sends file to SGS eApps via MQ.
√ AutoSys uses a dedicated Unix id to run the loading job.

E.2. MEPS+ sends security update file to eApps (Daily)

1. MEPS+ end of day job (EOD) will generate a security update file, which contains updates for all ongoing securities, and send it to PRDSYS.
2. AutoSys job will load the file into database.

Business Rules:
√ MEPS+ sends file to SGS eApps via MQ.
√ AutoSys uses a dedicated Unix id to run the loading job.

E.3. MEPS+ sends bank list file to eApps (Daily)

1. MEPS+ end of day job (EOD) will generate a bank list file, which contains updates for all banks, and send it to PRDSYS.
2. AutoSys job will load the file into database.

Business Rules:
√ MEPS+ sends file to SGS eApps via MQ.
√ AutoSys uses a dedicated Unix id to run the loading job.

E.4. eApps sends closing price file to MEPS+ (Daily)

- After MAS MDD user runs today's daily closing price report, and triggers sending daily price to MEPS+, a file is generated and put under Unix file system on PRDSYS.
- AutoSys job will send the file to MEPS+.
- An OnDemand report will be generated upon the receipt of the Daily Price by MEPS+. The closing price data is required by MEPS+ to process forward dated transactions during MEPS+ EOD.

Business Rules:
√ SGS eApps sends file to MEPS+ via MQ.

E.5. eApps sends previous month's closing price file to MERIT (every 1st day of the month)

i) On the first working day of the month (8pm), eApps will transmit a file containing the closing price details for all issues (e.g. yield) for everyday of the previous month, regardless if it is a working day or not. If no data is available, the value transmitted will be zero.
ii) An AutoSys job will send the file to the designated MERIT ftp data folder for the first leg. The second leg of picking the file will be executed by MERIT.

Business Rules:
√ SGS eApps generates the file and deposits it in PRDSYS.

F.1. eApps sends closing price file to FMBS

- After MAS MDD user runs today's daily closing price report, and triggers sending daily price to MEPS+ and FMBS, a file is generated and put under Unix file system on PRDSYS.
- AutoSys job will send the file to FMBS.
- The file is processed the next working day in FMBS.

Business Rules:
√ SGS eApps sends file to FMBS via MQ.

G.1. Notify operator if data loading error

1. When AutoSys calls Unix job to load data to Oracle, it will always create log file.
2. If there is any error in the loading, it will send an error message to Unicenter TNG's console.

Business Rules:
√ Operator will inform person in charge..

3 System design

The design of SGS eApps system consists of five parts:

i) Web front-end interface design - what is the PDs see when they use web browser to access MAS SGS eApps website. It includes the design of HTML and JSP pages, graphics, and JavaScript.

ii) Web back-end design - which includes the design of Java Servlet, JSP logic, database access, interaction with MAF authentication server, and data encryption module.

iii) Oracle database design - which includes table creation, constraint definition, and access right definition, etc.

iv) Workflow design - which includes the design of AutoSys jobs, UNIX kch scripts, Java programs for data parsing, Oracle SQL/Loader scripts for data loading, PL/SQL scripts for data manipulation and error reporting.

v) Data encryption/decryption - which includes calling of RSA algorithm to encrypt/decrypt data.

vi) File encryption/decryption – which is using PGP encryption.

3.1 Web front-end interface design

There are three different kinds of web pages that PDs will see:

i) Static HTML pages.
ii) JSP pages - which are directed by back-end servlets. Normally PDs don't access JSP pages directly. They access servlets instead.
iii) Web pages produced by error handling servlets.

All the web pages except MAS home should be protected under SSL mode.

Many web pages have a header on the top of the pages. Four links are provided on the header, which are “Menu”, “Help”, “FAQ” and “Logoff”. A JavaScript file “header.js” is used to automatically generate the header.

3.1.1 MAS SGS Home

There is a menu at the left side of MAS SGS home or MAS Bills home. One menu item is added to the menu, which is “SGS eApps for Primary Dealers” and links to “https://masnetsvc2.mas.gov.sg/sgseapps/sgseapps.SGSLogin.”

3.1.2 SGSHelp.html

On this page, 4 links are provided for help files that are in pdf format.

3.1.3 SGSLogin.jsp

Users can choose to access SGS eApps directly rather than to follow the link on the MAS home. "SGSLogin.jsp" provides another place for users to logon.

This page consists of two frames.

For RSA Simulator or RSA Login - On the upper frame, a logon form is also provided. Some JavaScript code checks whether the user has provided userid and password after the user clicks on "Submit".

For CorpPass Simulator login – On the upper frame, a logon form will have input fields to accept UEN, NRIC and password. JavaScript code checks whether the user has provided UEN, NRIC and password after the user clicks on "Submit".

The result of logon will be displayed in the same window.

For CorpPass login – On the upper frame, URL link to connect to CorpPass Login page with the CorpPass logo would be provided.
“mnet.corpPass.url.login” property in sgs.properties file will have the CorpPass Login URL.

If a user logons successfully, the SGS eApps home will be shown. There is also a "Cancel" button, which will direct user to MAS home page if user clicks on it.

For CorpPass TOPPAN Integration login – On the upper frame, instead of redirecting to the CorpPass login page directly, URL link to connect to TOPPAN EIC Proxy would be provided which in turn handles the redirection to the CorpPass login page.

"mnet.toppan.eicProxy.auth.url" property in sgs.properties file will have the TOPPAN EIC Proxy Authentication URL.

If a user logons successfully, the SGS eApps home will be shown.

The bottom frame contains links to some help files in PDF format.

3.1.4 AuthenticationFailed.jsp

After the servlet "masnet.etender.SGSHome" receives a logon request, it will authenticate the userid and password with the RSA Authentication server. If the logon fails, it will send "AuthenticationFailed.jsp" to the PD. It will also check against the ABA0013_PRIMARY_DEALER table. If the PD bank code does not exist in the table, the logon will also fail.

The page "AuthenticationFailed.jsp" tells the user the possible reasons for logon failure. It also provides a link to the logon page.

3.1.5 Redirect.jsp

After PD successfully logons, this page will be shown. After 2 seconds, this page will automatically re-direct PD to SGS eApps Menu.

3.1.6 SGSMenu.jsp

This is the home page for SGS eApps System. The menu of all available SGS eApps services is shown at the left part. And a brief explanation of each service is provided at the right part.

For the first phase of SGS eApps system, three services are available, which are "*Subscription*", "*Submissions Inquiry*", "*Auction Result*", "*Daily Closing Price*", "*Enhanced Repo Facility*", "*Repo Result*", "*Announcement*", "*Download Issuance Calendar*", and "*Admin Issuance Calendar*". Nine graphic buttons are provided as the menu items. User can click on the buttons for the desired services.

3.1.7 SGSSubList.jsp

On this page, a list of ongoing SGS securities and MAS Bills is displayed in a table. If there is no ongoing security, a message of "No ongoing security is available" will be displayed.

In the table, there are five columns – Security Code, Security Description, Issue Date, Closing Date and Issue Size. User can click on the security code to subscribe the issue.

The servlet "sgseapps.etender.SGSSubList" passes the information of ongoing securities to the JSP file "SGSSubList.jsp". The code on the JSP page will automatically format those issues in table format and provide hype-links.

3.1.8 IssueClosed.jsp

According to the business rule, an issue is closed if it is after 12:00pm in the closing date. If a user accesses the list of available issues before 12:00pm and subscribes an issue after 12:00pm, the message of "Issue has closed for submission" will be displayed to user on the page "IssueClosed.jsp".

On the page "IssueClosed.jsp", a link to the page of available issues is provided.

3.1.9 SGSSubscribe.jsp

On this page, a form is displayed for user to input subscription details for T-bills and bonds.

The following fields are static fields, of which the values are generated by the Java code on the page: security name, security code, issue date, and user id.

The following fields are editable and require input from user: nominal amount, competitive indicator, yield, type of applicant, name of applicant, applicant's address, NRIC/passport no., nationality, Application Type, CPF/SRS account no., CDP account no., settlement bank, settlement bank's account code with MAS, and bank or bank's customer indicator.

JavaScript code on the page will automatically convert user's input to upper case.

3.1.10 SGSMasBillsSubscribe.jsp

On this page, a form is displayed for user to input subscription details for MAS Bills.

The following fields are static fields, of which the values are generated by the Java code on the page: security name, security code, issue date, and user id.

The following fields are editable and require input from user: nominal amount, yield, type of applicant, name of applicant, applicant's address, settlement bank, settlement bank's account code with MAS, and bank or bank's customer indicator.

JavaScript code on the page will automatically convert user's input to upper case.

3.1.11 SGSReSubmit.jsp

This jsp is for T-bills and bonds.

The static fields and editable fields on this page are same as those of "SGSSubscribe.jsp".

After a user submits an application (either via SGSSubscribe.jsp or via SGSReSubmit.jsp itself), the back-end program will do some validation using the business rule spelled out in Section 2. If the validation fails, this page will be shown. Java code on the page automatically formats the error information to HTML format, i.e. to display the error message and to highlight some fields in red color.

This page starts by showing user the error message. All the fields, which caused validation failure, are highlighted in red color.

3.1.12 SGSMasBillsReSubmit.jsp

This jsp is for MAS Bills.

The static fields and editable fields on this page are same as those of "SGSSubscribe.jsp".

After a user submits an application (either via SGSMasBillsSubscribe.jsp or via SGSMasBillsReSubmit.jsp itself), the back-end program will do some validation using the business rule spelled out in Section 2. If the validation fails, this page will be shown. Java code on the page automatically formats the error information to HTML format, i.e. to display the error message and to highlight some fields in red color.

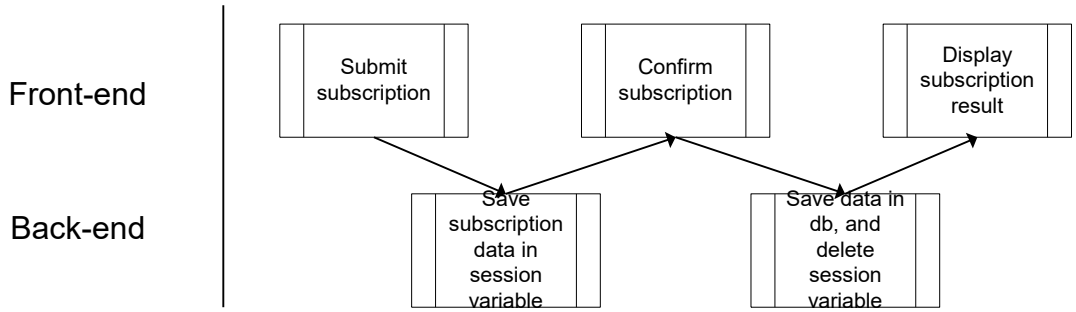
This page starts by showing user the error message. All the fields, which caused validation failure, are highlighted in red color.

3.1.13 SGSSubConfirm.jsp

This is the page for user to confirm the application for T-bills and bonds.

On the page there is a "Confirm" button. User can click on the button to confirm the subscription.

Another purpose of providing a "confirm" page is to solve the "Refresh" or "Reload" problem. This can be illustrated in the following diagram:



When the user refreshes the result page, it will not cause re-posting of form data. The "confirm" page successfully prevents re-posting.

3.1.14 SGSMasBillsSubConfirm.jsp

This is the page for user to confirm the application for MAS Bills.

On the page there is a "Confirm" button. User can click on the button to confirm the subscription.

3.1.15 TransactionSucceeded.jsp

If an application from the user for T-bills and bonds has been accepted, this page will be shown.

On this page, the userid who submitted the bid application, the security code, the subscription date and time, and a unique transaction reference no is shown.

There is also a table showing the details of user's application. User may print this page for future reference.

The information of userid, security code, transaction date and time, and reference no are provided by the servlet "sgseapps.etender.SGSSubResult". The information is automatically formatted to HMTL format by the Java code on the "TransactionSucceeded.jsp".

3.1.16 MasBillsTransactionSucceeded.jsp

If an application from the user for MAS Bills has been accepted, this page will be shown.

On this page, the userid who submitted the bid application, the security code, the subscription date and time, and a unique transaction reference no is shown.

There is also a table showing the details of user's application. User may print this page for future reference.

The information of userid, security code, transaction date and time, and reference no are provided by the servlet "sgseapps.etender.SGSMasBillsSubResult". The information is automatically formatted to HMTL format by the Java code on the "MasBillsTransactionSucceeded.jsp".

3.1.17 TransactionFailed.jsp

A transaction may fail due to some unknown exceptional conditions, e.g. database is down. In this situation, "TransactionFailed.jsp" will be shown.

On this page, a link is provided to the "confirm" page.

3.1.18 MasBillsTransactionFailed.jsp

A transaction may fail due to some unknown exceptional conditions, e.g. database is down. In this situation, "MasBillsTransactionFailed.jsp" will be shown.

On this page, a link is provided to the "confirm" page.

3.1.19 SGSInquiryList.jsp

On this page, a list of ongoing SGS securities and MAS Bills is displayed in a table. If there is no ongoing security, a message of "No ongoing security is available" will be displayed.

"Download Retail Bid Acknowledgement Submitted via eApps" link will be shown for 3 local PDs to get into Acknowledgement File Download Form.

In the table, there are five columns – Security Code, Security Description, Issue Date, Closing Date and Issue Size. User can click on the security code to inquire bid submissions.

The servlet “sgseapps.etender.SGSInquiryList” passes the information of ongoing securities to the JSP file “SGSInquiryList.jsp”. The code on the JSP page will automatically format those issues in table format and provide hype-links.

3.1.20 SGSInquiryDisplay.jsp

This page displays the bid submissions of the selected security code for the bank of current user. The back-end servlet "masnet.etender.SGSInquire" passes the result information to the JSP file. The Java code on the JSP page will automatically format the result in HTML format.

For MDD and three Local banks (DBS, UOB, OCBC), both “Download Retail Bid Submissions” and “Download Online Submissions” links will be displayed if data exist else will not be shown.

MDD will be able to download the “Download Online Submissions” which are submitted by all PD and MDD. MDD will be able to download the “Download Retail Bid Submissions” which are uploaded by three local banks.

Local PD will be able to download the “Download Retail Bid Submissions” and “Download Online Submissions” which are submitted/uploaded by the own. Local PD will be able to see the Online Submissions which are submitted by their own.

Non-local PD can see “Download Online Submissions” link to down the submission by their own bank. Non-local PD can’t see “Download Retail Bid Submissions” link.

At the bottom of this page, user can click on the button “>>” to navigate through the next page. User can also click on the button “<<” to go back to the previous page.

3.1.21 SGSAuctionList.jsp

On this page, two lists of auction result are displayed in two tables. One table displays today's auction result and the other displays the result of recent one month. If there is no auction result for today or recent one month, a message of “No auction result” will be displayed.

In the table, there are two columns – Security Code, and the closing date. User can click on the security code to show the auction result for his/her bank.

The servlet “sgseapps.etender.SGSAuctionList” passes the information of available auction results to the JSP file “SGSAuctionList.jsp”. The code on the JSP page will automatically format those issues in table format and provide hype-links.

3.1.22 SGSAuctionDisplay.jsp

This page displays the result of the selected security code for the bank of current user. The back-end servlet "sgseapps.etender.SGSAuctionDisplay" passes the result information to the JSP file. The Java code on the JSP page will automatically format the result in HTML format.

This page uses HTML tag <PRE> and </PRE>, which means the result is displayed using the original format from mainframe.

3.1.23 Logoff.jsp

If a user finishes using SGS eApps system, he/she can click on the "Logoff" button on the top frame of SGS home page. Then the page "Logoff.jsp" is displayed for Non-CorpPass Login.
On the page "Logoff.jsp", a link to the logon page is provided.

For CorpPass Login – MAF global logout page would be displayed using the below URL.
"Continue" button in the global logout page takes to the eApps logon page.

UAT – MAF Global logout URL - https://ueservices.mas.gov.sg/sgseapps/?logout
PROD – MAF Global logout URL - https://eservices.mas.gov.sg/sgseapps/?logout

For CorpPass TOPPAN Integration Login – eApps will invalidate its own session.

3.1.24 Timeout.jsp

If a user logons to SGS eApps system and opens a web page for 15 minutes, this user's session will automatically timeout and the page "Timeout.jsp" will be shown.

On the page "Timeout.jsp", a link is provided to the logon page.

3.1.25 The error page

If there is any exception, which occurs in the back-end processing, an error page will be provided by the servlet "masnet.common.SvtMnetError" and shown to the user. On the page, a unique error identification code and the time when the error occurred are shown. The user may call MAS Helpdesk and tell the Helpdesk technician the error code and the time for error tracing.

Only WAS run-time errors are shown on the error page. Other errors such as "File not found", "Internal server error" are handled by the HTTP server itself.

3.1.26 SGSMASDailyPriceMenu.jsp

This is the daily closing price menu page for MAS user. The menu of all available daily closing price services is shown. These services include "Display today's submission status", "Input/View/Modify today's data for bank", "View historical data for bank", "Run today's exception report", "Download today's report", "Download historical report", "View daily price spread", "Manage public holiday", "Update financial database", and "Send today's daily price to MEPS+ and FMBS".

3.1.27 SGSMASDailyPriceSubList.jsp

On this page, a table displays banks' submission status. The table has three columns – Bank Code, Submitted, and Allow Edit. User can click on the bank code to view bank's submitted data.

The servlet “sgseapps.etender.SGSMASDailyPrice” reads from database and passes the information of banks’ submission to the JSP.

3.1.28 SGSMASDailyPriceDisplay.jsp

This page displays the detailed data of a bank’s submission. The back-end servlet "sgseapps.etender.SGSMASDailyPrice" passes the bank’s daily closing price data to the JSP file. The Java code on the JSP page will automatically format the result in HTML format.

MAS MDD user is able to see the daily closing price data from all banks.

3.1.29 SGSMASDailyPriceInput.jsp

On this page, a form is displayed for MAS user to input daily closing price for a bank. The issue codes, maturity dates of all ongoing securities will be displayed, and are not editable.

The following fields are editable and require input from user: bid price, offer price, highest done price for benchmark issues, and lowest done price for benchmark issues.

User must input values for all fields.

3.1.30 SGSMASDailyPriceReInput.jsp

The static fields and editable fields on this page are same as those of "SGSMASDailyPriceInput.jsp".

After a user submits a daily closing price data (either via SGSMASDailyPriceInput.jsp or via SGSMASDailyPriceReInput.jsp itself), the back-end program will do some validation using the business rule spelled out in Section 2. If the validation fails, this page will be shown. Java code on the page automatically formats the error information to HTML format, i.e. to display the error message and to highlight some fields in red color.

This page starts by showing user the error message. All the fields, which caused validation failure, are highlighted in red color.

3.1.31 SGSMASDailyPriceConfirm.jsp

This is the page for user to confirm the daily closing price submission.

On the page there is a "Confirm" button. User can click on the button to confirm the submission.

Another purpose of providing a "confirm" page is same as explained in 3.1.11.

3.1.32 MASDPTransactionSucceeded.jsp

If a submission from the user has been accepted, this page will be shown.

On this page, the date of the data, name of the bank, userid who submitted the bid application, and date and time is shown.

There is also a table showing the details of user's submission. User may print this page for future reference.

The information of userid, bank name, and transaction date and time are provided by the servlet "sgseapps.etender.SGSMASDailyPriceResult". The information is automatically formatted to HMTL format by the Java code on the page.

3.1.33 MASDPTransactionFailed.jsp

A transaction may fail due to some unknown exceptional conditions, e.g. database is down. In this situation, this page will be shown.

On this page, a link is provided to the "confirm" page.

3.1.34 SGSMASDPExceptionReport.jsp

After MAS user runs today's exception report, the result will be passed to this page for display.

There are two sections on this page, "Banks that have not submitted" and "High/Low spread exception". MAS user uses this report to check whether there is any data abnormality.

3.1.35 SGSMASPriceSpreadList.jsp

This page allows user to view spread parameters.

Three kinds of spread parameters for front-end & back-end validations: against yesterday's, bid/ask, and high/low.

Three kinds of spread parameters for back-end validations only: highest/lowest & average, highest/lowest, high/low & average.

3.1.36 MASPriceSpreadSucceeded.jsp

If the submission from the user has been accepted, this page will be shown.

On this page, only a message will be shown.

3.1.37 MASPriceSpreadFailed.jsp

A transaction may fail due to some unknown exceptional conditions, e.g. database is down. In this situation, this page will be shown.

On this page, a link is provided to the previous page.

3.1.38 SGSMASPublicHoliday.jsp

This page allows user to input public holidays for a particular year of a specific country selected. It allows maximum 15 public holidays for a year.

User will input the date, and the description of the holidays.

3.1.39 MASPublicHolidaySucceeded.jsp

If the submission from the user has been accepted, this page will be shown.

On this page, only a message will be shown.

3.1.40 MASPublicHolidayFailed.jsp

A transaction may fail due to some unknown exceptional conditions, e.g. database is down. In this situation, this page will be shown.

On this page, a link is provided to the previous page.

3.1.41 SGSUserDailyPriceMenu.jsp

This is the daily closing price menu page for banks' users. There is only one menu item: Input/View/Modify today's data.

3.1.42 SGSUserDailyPriceDisplay.jsp

This page displays the detailed data of this bank's submission. The back-end servlet "sgseapps.etender.SGSUserDailyPrice" passes the bank's daily closing price data to the JSP file. The Java code on the JSP page will automatically format the result in HTML format.

Bank user is only able to see the daily closing price data from his/her own bank.

3.1.43 SGSUserDailyPriceFileAttach.jsp

This page allows user to attach the daily closing price file and to submit it to MAS via file upload. There is a "Browse" button, which user uses it to locate the daily closing price file. User clicks on "Submit" to submit the file.

3.1.44 SGSUserDailyPriceInput.jsp

On this page, a form is displayed for user to input daily closing price for a bank. The issue codes, maturity dates of all ongoing securities will be displayed, and are not editable.

The following fields are editable and require input from user: bid price, offer price, highest done price for benchmark issues, and lowest done price for benchmark issues.

User must input values for all fields.

3.1.45 SGSUserDailyPriceReInput.jsp

The static fields and editable fields on this page are same as those of "SGSUserDailyPriceInput.jsp".

After a user submits a daily closing price data (either via SGSUserDailyPriceFileAttach.jsp, SGSUserDailyPriceInput.jsp or via SGSUserDailyPriceReInput.jsp itself), the back-end program will do some validation using the business rule spelled out in Section 2. If the validation fails, this page will be shown. Java code on the page automatically formats the error information to HTML format, i.e. to display the error message and to highlight some fields in red color.

This page starts by showing user the error message. All the fields, which caused validation failure, are highlighted in red color.

3.1.46 SGSUserDailyPriceConfirm.jsp

This is the page for user to confirm the daily closing price submission.

On the page there is a "Confirm" button. User can click on the button to confirm the submission.

Another purpose of providing a "confirm" page is same as explained in 3.1.11.

3.1.47 UserDPTransactionSucceeded.jsp

If a submission from the user has been accepted, this page will be shown.

On this page, the date of the data, name of the bank, userid who submitted the bid application, and date and time is shown.

There is also a table showing the details of user's submission. User may print this page for future reference.

The information of userid, bank name, and transaction date and time are provided by the servlet "sgseapps.etender.SGSUserDailyPriceResult". The information is automatically formatted to HMTL format by the Java code on the page.

3.1.48 UserDPTransactionFailed.jsp

A transaction may fail due to some unknown exceptional conditions, e.g. database is down. In this situation, this page will be shown.

On this page, a link is provided to the "confirm" page.

Only WAS run-time errors are shown on the error page. Other errors such as "File not found", "Internal server error" are handled by the HTTP server itself.

3.1.49 SGSPublicAnnounceList.jsp

This page shows the latest announced security list. It has three sections: one for MAS Bills, one for T-bill, and another for 1 year T-bill and bond.

The back-end servlet "sgseapps.etender.SGSPublicAnnounceList" passes the latest announced new security info to the JSP file. The Java code on the JSP page will automatically format the announcement in HTML format.

3.1.50 SGSPublicBondAnnounce.jsp

This page displays a bond announcement.

The back-end servlet "sgseapps.etender.SGSPublicAnnounce" passes two kinds of info to the page. One is the details of the new security. The Java code on the JSP page will automatically format the info in HTML format. The other is the appropriate announcement text to be displayed at the bottom of the page.

3.1.51 SGSPublictBillAnnounce.jsp

This page displays a T-bill announcement.

The back-end servlet "sgseapps.etender.SGSPublicAnnounce" passes two kinds of info to the page. One is the details of the new security. The Java code on the JSP page will automatically format the announcement in HTML format. The other is the appropriate announcement text to be displayed at the bottom of the page.

3.1.52 SGSPublictMasBillAnnounce.jsp

This page displays a MAS Bill announcement.

The back-end servlet "sgseapps.etender.SGSPublicAnnounce" passes two kinds of info to the page. One is the details of the new security. The Java code on the JSP page will automatically format the announcement in HTML format. The other is the appropriate announcement text to be displayed at the bottom of the page.

3.1.53 SGSPublicResultList.jsp

This page shows the latest auctioned security list. It has three sections: one for MAS Bills, one for T-bill, and another for 1 year T-bill and bond.

The back-end servlet "sgseapps.etender.SGSPublicResultList" passes the latest auctioned new security info to the JSP file. The Java code on the JSP page will automatically format the info in HTML format.

3.1.54 SGSPublicBondResult.jsp

This page displays a bond auction result.

The back-end servlet "sgseapps.etender.SGSPublicResult" passes the details of the auction result to the page. The Java code on the JSP page will automatically format the info in HTML format. The other is the appropriate announcement text to be displayed at the bottom of the page.

3.1.55 SGSPublictBillResult.jsp

This page displays a T-bill auction result.

The back-end servlet "sgseapps.etender.SGSPublicResult" passes the details of the auction result to the page. The Java code on the JSP page will automatically format the info in HTML format. The other is the appropriate announcement text to be displayed at the bottom of the page.

3.1.56 SGSPublictMasBillResult.jsp

This page displays a MAS Bill auction result.

The back-end servlet "sgseapps.etender.SGSPublicResult" passes the details of the auction result to the page. The Java code on the JSP page will automatically format the info in HTML format. The other is the appropriate announcement text to be displayed at the bottom of the page.

3.1.57 message.jsp

This page displays the system message to the user. Many servlets call this page.

3.1.58 SGSSubmitFileAttach.jsp

On this page, a form is displayed for user to upload bid file.

3.1.59 SGSUploadConfirm.jsp

This is the page for user to confirm the upload.

This page will display message to confirm file submission to be saved.

If file is already existing, this page will display alert that by confirming the action, existing file will be replaced with newly uploaded file.

File processing date will be decided based on the submission cutoff time checking.

On the page there is a "Confirm" button. User can click on the button to confirm the upload.

3.1.60 SGSUploadResult.jsp

This page shows the upload result.

This page displays details of the userid who uploaded the bid application file, the file name, file processing date, and uploaded date/time.

User may print this page for future reference.

The information of userid, security code, transaction date and time, and reference no are provided by the servlet "sgseapps.etender.SGSSubmitUploadResult". The information is automatically formatted to HMTL format by the Java code on the "SGSUploadResult.jsp".

3.1.61 SGSAnnouncementMenu.jsp

This is the announcement menu page for MAS user. There are three menu items:

1. Add/remove of Non-benchmark indicator, which user will input issue code and issue no.
2. "View announcement page for issue ()", which user will input issue code and issue no.
3. "Release announcement for issue ()", which user will also input issue code and issue no.

The whole work process is: FD user inputs a new issue in MEPS+; FD user releases the issue in MEPS+; MEPS+ sends the issue to SGS eApps; SGS eApps loads the issue into database; MDD user verifies the details of the announcement; MDD user releases the issue.

There will be some validation checks at release. The issue must be valid and not a past issue, that is result load date is not empty.

3.1.62 SGSMASIssueCalendarMenu.jsp

This is the issuance calendar menu page for MAS user. There are two menu items: "Input/View/Modify Issuance Calendar for year ()", which user will input the year, and "Download Issuance Calendar for year ()", which user will also input the year.

3.1.63 SGSMASIssueCalendar.jsp

This page displays all issues in the selected year, or is empty if there is no issue inputted by MDD user. MDD user uses this page to input issues for the year, and save to the database.

3.1.64 SGSUserIssueCalendarMenu.jsp

This is the issuance calendar menu page for three local banks' user. There are one menu item: "Download Issuance Calendar for year ()", which user will also input the year.

3.1.65 CorpPassAuthorizationFailed.jsp

The page "CorpPassAuthorizationFailed.jsp" tells the user the possible reasons for logon failure for CorpPass or CorpPass Simulator Login. It also provides a link to the login page

3.1.66 SGSAuctionMenu.jsp

The page "SGSAuctionMenu.jsp" will display the Overall Results link and Option to download auction results in CSV format

3.1.67 SGSAcknowledgementFileDownload.jsp

The is the page for user to download the Acknowledgment only for the files uploaded in eApps.

Acknowledgment file can be downloaded based on input date (processing date) and CASH/CPF/SRS radio options.

3.2 Web back-end design

The back-end design consists of Java servlets and JSP logic design, database access, integration with MAF Authentication server, error logging, integration with encryption module, and some miscellaneous Java classes.

3.2.1 Java servlets and JSP logic design

There is one common logic, which user needs to logon first before accessing any of those servlets, imposed to all Java servlets except "sgseapps.SGSLogin", "sgseapps.etender.SGSPublicAnnounceList", "sgseapps.etender.SGSPublicAnnounce". "sgseapps.etender.SGSPublicResultList", and "sgseapps.etender.SGSPublicResult". If a user has not logged on yet, he/she will be automatically directed to the logon page.

Some general-purpose servlets belong to the package "sgseapps", so that a servlet in this group is represented as "sgsapps.servletname".

Some servlets are specially developed for SGS eTender, daily closing price, and repo. They belong to the package "sgseapps.etender". A servlet in this group is represented as "sgseapps.etender.servletname".

The error-handling servlet "masnet.common.SvtMnetError" belongs to the package "masnet.common". It is represented as "masnet.common.SvtMneterror".

All the servlets make use of servlet "masnet.common.SvtMnetError" for run-time exception displaying.

3.2.1.1 sgseapps.SGSLogin

This is the logon page. This servlet always directs user to the page "SGSLogin.jsp".

3.2.1.2 sgseapps.SGSAuthentication

eApps application supports 5 type of login as below based on the "authentication.mechanism" property value in sgs.properties

1. No authentication mechanism (RSA simulator) - authentication.mechanism=
2. RSA - authentication.mechanism=RSA
3. CorpPass Simulator - authentication.mechanism=CorpPassSimulator
4. CorpPass - authentication.mechanism=CorpPass
5. CorpPass TOPPAN Integration – authentication.mechanism=Toppan

The format of userid must be AxxxxTyy, where xxxx is the bank code and yy is the serial no for RSA simulator and RSA.

For CorpPass Simulator – UEN and NRIC are the inputs to login
For CorpPass – UEN and CorpPass ID are the inputs to login

This servlet handles user's logon request. If a user logons successfully, the servlet will direct user to the page "Redirect.jsp", which will in turn re-direct user to SGS eApps Home. If a user fails in the logon, the servlet will direct user to the page

1. "AuthenticationFailed.jsp" for RSA Simulator and RSA login.
2. CorpPassAuthorizationFailed.jsp for CorpPass Simulator and CorpPass and CorpPass TOPPAN Integration login

The servlet uses the following steps to authenticate with RSAAuthentication Server.

- Create SGSAuthIntf object, which returns the correct authentication class, RSA or Safeword, according to the authentication.mechanism property in sgs.properties file in /was_share
- Init the object
- Pass in userid and password for authentication
- Get authentication result

The servlet uses the following steps to authenticate with CorpPass via MAF integration

- Create SGSCorpPassAuth object and call authenticate() method which retrieves the user login details CPEntID (UEN) and CPUID (NRIC) using NinjaTokenPrincipal – Refer Section 7.17.2.5 for more details
- If NinjaTokenPrincipal not found or didn't receive the CorpPass details CPEntID and CPUID, below error message would be shown to the user

"System error occurred. Please contact MAS Helpdesk."

- Authorization steps for CorpPass or CorpPass Simulator login
- Retrieve the Bank code and Token User ID from the ABA0030_CORP_PASS_MAPPING table with input UEN and Base64 encoded NRIC value where Active Indicator ='Y'
  - If no record found in the DB, below authorization error message would be shown to the user "Your account is not authorized to access eApps application, please contact MAS Helpdesk for user registration."
  - If more than one record found in the DB, below authorization error message would be shown to the user "Your account is not authorized to access eApps application, please contact MAS Helpdesk to verify your CorpPass Entity (UEN) details."
  - If only one record found in the DB, then the Authorization is successful and ABA0030_LAST_LOGIN_DT would be updated to current date timestamp
- Validate the Bank code against the ABA0013_PRIMARY_DEALER table. If the PD bank code does not exist in the table, the logon will also fail
- The Token User ID will be in the pattern 'CxxxxT01' and once reaches 'CxxxxT99', the next Token User ID would be 'DxxxxT01'. Similar pattern would be followed until the Token User ID reaches 'ZxxxxT99', where xxxx is the Bank code
- This Token User ID is considered as the userid instead of the input UEN and NRIC and this will be referenced in Other tables where we store User ID/Submitted By.

For all type of login, the UserID is displayed in the eApps page header along with the Bank name
'Login as <Bank Name> (<UserID>)'

The servlet uses the following steps to authenticate with CorpPass via TOPPAN Integration

- Validate the query parameter value (eic_state) against the session value (session_eic_state).
- If eic_state value is unmatched, user will be redirected to page CorpPassAuthorizationFailed.jsp with message "System error occurred. Please contact MAS Helpdesk." and error will be logged.
- If eic_state value is matched, checks if it received the query parameter 'error' to identify whether the authentication to EIC Proxy was success or failure.
  - If failure/received query parameter 'error', user will be redirected to page CorpPassAuthorizationFailed.jsp with message "System error occurred. Please contact MAS Helpdesk." and error will be logged.
  - If successful, get userToken from query parameter and call REST API to EIC OIDC Relying Party (RP). API calls need to include an EIC security token in the request headers based on the "mnet.toppan.ecquaria-

authToken" property value in sgs.properties. These security tokens are encrypted in a JWE format and can only be generated and issued by using the EIC Management Console.

- On failure REST API response, user will be redirected to page CorpPassAuthorizationFailed.jsp with message "System error occurred. Please contact MAS Helpdesk." and error will be logged.
- On successful REST API response, the servlet will take userInfo.uuid and userInfo.cpEntId from JSON response to validate against ABA0030_CORP_PASS_MAPPING table for authorization.

- Authorization steps for CorpPass TOPPAN Integration is the same as per current implementation.

3.2.1.3 sgseapps.SGSMenu

This servlet always directs user to the page "SGSMenu.jsp".

3.2.1.4 sgseapps.Logoff

This servlet is called when user clicks the "Logoff" button on SGS eApps web pages. It will invalidate the current session and direct user to the page of "Logoff.jsp" for Non-CorpPass login.

For CorpPass Login – User will be redirected to MAF global logout page using the below URL.

UAT – MAF Global logout URL - https://ueservices.mas.gov.sg/sgseapps/?logout
PROD – MAF Global logout URL - https://eservices.mas.gov.sg/sgseapps/?logout

If a user wants to use any of SGS eApps service after logoff, he/she has to logon again.

3.2.1.5 sgseapps.Timeout

This servlet is called when an SGS eApps web page has been opened for 880 seconds. It will invalidate the current session and direct user to the page of "Timeout.jsp".

If a user wants to use any of SGS eApps service after timeout, he/she has to logon again.

3.2.1.6 sgseapps.etender.SGSSubList

This servlet reads data from Oracle database and passes the data to "SGSSubList.jsp".

The selection criteria are that if a security's tender date is late than today, it is considered as an ongoing issue. Or if a security's tender date is same as today, but the current time is earlier than 12:00pm.

3.2.1.7 sgseapps.etender.SGSSubscribe

When user selects a security to subscribe in the table on the page of "SGSSubList.jsp", this servlet will be called with a query string in the format of "sgseapps.etender.SGSSubscribe?securitycode".

This servlet will use the security code, which is got in the query string, to look up in the Oracle database for the details information of the security, i.e. security name, issue date, closing date, reference no., etc. If no security is found in the database using the security code, the user will be directed to the page of "SGSSubList".

The servlet will then check whether the security is closed for subscription. If the closing date of the selected security is today and the time passes 12:00pm, then the security is considered as closed for subscription and the servlet will direct user to the page of "IssueClosed.jsp".

If the security is found in the Oracle database, the servlet will construct an object of "sgseapps.etender.Securities" and pass it to the page of "SGSSubscribe.jsp" for T-bill and bond or to the page of "SGSMasBillSubscribe.jsp" for MAS Bills.

3.2.1.8 sgseapps.etender.SGSSubConfirm

The servlet will first check whether the security is closed for subscription. If the closing date of the selected security is today and the time passes 12:00pm, then the security is considered as closed for subscription and the servlet will direct user to the page of "IssueClosed.jsp".

This servlet then validates the application using the business rules spelled in Section 2. If the application fails in the validation, this servlet will pass a Boolean array to the page "SGSReSubmit.jsp". The Boolean array indicates which fields are not acceptable in the form.

The relationship of the Boolean array and the fields in the table is as follows:

bl(0)	first line of bid section
bl(1)	second line of bid section
bl(2)	third line of bid section
bl(3)	fourth line of bid section
bl(4)	fifth line of bid section
bl(5)	type of applicant
bl(6)	name of applicant
bl(7)	applicant's address
bl(8)	NRIC/passport no.
bl(9)	nationality
bl(10)	Application Type
bl(11)	CPF/SRS Acc No
bl(12)	account with MAS
bl(13)	bank/custodian indicator
Bl(14)	Bank’s reference no
Bl(15)	CDP Acc No

If the application passes the validation, the servlet will assemble a string which represent the application data. The format of the string is shown in the following table. The servlet will then save the string as a session variable and direct user to the page of "SGSSubConfirm.jsp".

Field Name	Length	e.g.	Comment
ISSUE_CODE	8	NX00100H	
TRANS_REF	5	Reference No	
RECEIVED_DATE	8	YYYYMMDD	
RECEIVED_TIME	6	HHMMSS	
USERID	8		ABAOLTRN will be stored in this field instead of Login UserID
TRANS_TYPE	3	TE_	
TENDER_DATE	8	YYYYMMDD	From security master
TRANS_DATE	8	YYYYMMDD	Same as rec date
NOMINAL_AMT1	11	00500000000	
COMP_NOCOMP1	1		

BID_YIELD1	3.2	00305	Length is 5
NOMINAL_AMT2	11		
COMP_NOCOMP2	1		
BID_YIELD2	3.2		Length is 5
NOMINAL_AMT3	11		
COMP_NOCOMP3	1		
BID_YIELD3	3.2		Length is 5
NOMINAL_AMT4	11		
COMP_NOCOMP4	1		
BID_YIELD4	3.2		Length is 5
NOMINAL_AMT5	11		
COMP_NOCOMP5	1		
BID_YIELD5	3.2		Length is 5
NAME_OF_APPLN	30		
NATIONALITY	1		
IC_PASSPORT	14		
CPF_SRS_ACCOUNT	16		
PRI_DLR_CODE	4	7171	Primary dealer b-code
CUST_BANK_CODE	4		Settlement bank code
CUST_BANK_BC	1		Bank or custodian
TYPE_OF_INSTN	3		
REC_STATUS	1	_	
BANK_REF_NO	12		
CDP_ACCOUNT	16		
SUBMISSION_MTD	1		M, A, or E
APPLICATION_TYPE	3		
FILLER	5		

3.2.1.9 sgseapps.etender.SGSSubResult

After a user confirms an application, this servlet will be called.

The servlet will first check whether the security is closed for subscription. If the closing date of the selected security is today and the time passes 12:00pm, then the security is considered as closed for subscription and the servlet will direct user to the page of "IssueClosed.jsp".

The servlet now is ready to call the encryption module. It reads the public key from database, and then uses the public key to encrypt the application data, which is saved as a session variable previously.

Because the encrypted data is in binary format, another conversion is needed. The servlet will convert the encrypted data into an ASCII string.

The servlet will then save the ASCII string in the table aba0005_RSA_encrypted_transaction and increment the reference no in the table aba0001_security_master. The ASCII string is tailored into 4 pieces, which are saved in 4 fields in the database.

The servlet also saves bank's reference no, transaction reference no, security code, issue no and other audit trail information in clear format for banks' submission inquiry.

Finally, the servlet will remove the session variable, which stores the application data.

If the transaction is successfully finished, the servlet will direct user to the page "TransactionSucceeded.jsp". It stores the details of user's application in session variables and the page "TransactionSucceeded.jsp" uses the variable values to show the application summary to user.

If the transaction fails, the servlet will direct user to the page "TransactionFailed.jsp".

3.2.1.10 sgseapps.etender.SGSWithdraw

The servlet first checks if the tender date cutoff time has passed. If not, it will set the withdrawal time and the id performing the withdrawal to the bid application record.

3.2.1.11 sgseapps.etender.PublishAuctionResult

The servlet will first check if there are unreleased results. If there are, it will first publish the results by updating the MAS website database via dblink, and then insert an audit trail record into the eApps database.

3.2.1.12 sgseapps.etender.SGSMasBillsSubConfirm

The servlet will first check whether the security is closed for subscription. If the closing date of the selected security is today and the time passes 12:00pm, then the security is considered as closed for subscription and the servlet will direct user to the page of "IssueClosed.jsp".

This servlet then validates the application using the business rules spelled in Section 2. If the application fails in the validation, this servlet will pass a Boolean array to the page "SGSMasBillReSubmit.jsp". The boolean array indicates which fields are not acceptable in the form.

The relationship of the boolean array and the fields in the table is as follows:

bl(0)	first line of bid section
bl(1)	second line of bid section
bl(2)	third line of bid section
bl(3)	fourth line of bid section
bl(4)	fifth line of bid section
bl(5)	type of applicant
bl(6)	name of applicant
bl(7)	applicant's address
bl(8)	NRIC/passport no.
bl(9)	nationality
bl(10)	Application Type
bl(11)	CPF/SRS Acc No
bl(12)	account with MAS
bl(13)	bank/custodian indicator
Bl(14)	Bank’s reference no
Bl(15)	CDP Acc No

If the application passes the validation, the servlet will assemble a string which represent the application data. The format of the string is shown in the following table. The servlet will then save the string as a session variable and direct user to the page of "SGSMasBillSubConfirm.jsp".

Field Name	Length	e.g.	Comment
ISSUE_CODE	8	NX00100H	
TRANS_REF	5	Reference No	
RECEIVED_DATE	8	YYYYMMDD	
RECEIVED_TIME	6	HHMMSS	
USERID	8		ABAOLTRN will be stored in this field instead of Login UserID
TRANS_TYPE	3	TE	
TENDER_DATE	8	YYYYMMDD	From security master
TRANS_DATE	8	YYYYMMDD	Same as rec date
NOMINAL_AMT1	11	00500000000	
COMP_NOCOMP1	1		

BID_YIELD1	3.2	00305	Length is 5
NOMINAL_AMT2	11		
COMP_NOCOMP2	1		
BID_YIELD2	3.2		Length is 5
NOMINAL_AMT3	11		
COMP_NOCOMP3	1		
BID_YIELD3	3.2		Length is 5
NOMINAL_AMT4	11		
COMP_NOCOMP4	1		
BID_YIELD4	3.2		Length is 5
NOMINAL_AMT5	11		
COMP_NOCOMP5	1		
BID_YIELD5	3.2		Length is 5
NAME_OF_APPLN	30		
NATIONALITY	1		
IC_PASSPORT	14		
CPF_ACCOUNT	16		
PRI_DLR_CODE	4	7171	Primary dealer b-code
CUST_BANK_CODE	4		Settlement bank code
CUST_BANK_BC	1		Bank or custodian
TYPE_OF_INSTN	3		
REC_STATUS	1	_	
BANK_REF_NO	12		
CDP_ACCOUNT	16		
SUBMISSION_MTD	1		M, A, or E
APPLICATION_TYPE	3		
FILLER	5		

3.2.1.13 sgseapps.etender.SGSMasBillSubResult

After a user confirms an application, this servlet will be called.

The servlet will first check whether the security is closed for subscription. If the closing date of the selected security is today and the time passes 12:00pm, then the security is considered as closed for subscription and the servlet will direct user to the page of "IssueClosed.jsp".

The servlet now is ready to call the encryption module. It reads the public key from database, and then uses the public key to encrypt the application data, which is saved as a session variable previously.

Because the encrypted data is in binary format, another conversion is needed. The servlet will convert the encrypted data into an ASCII string.

The servlet will then save the ASCII string in the table aba0005_RSA_encrypted_transaction and increment the reference no in the table aba0001_security_master. The ASCII string is tailored into 4 pieces, which are saved in 4 fields in the database.

The servlet also saves bank's reference no and transaction reference no in clear format for banks' submission inquiry.

Finally, the servlet will remove the session variable, which stores the application data.

If the transaction is successfully finished, the servlet will direct user to the page "MasBillTransactionSucceeded.jsp". It stores the details of user's application in session variables and the page "MasBillTransactionSucceeded.jsp" uses the variable values to show the application summary to user.

If the transaction fails, the servlet will direct user to the page "MasBillTransactionFailed.jsp".

3.2.1.14 sgseapps.etender.SGSInquiryList

This servlet reads data from Oracle database and passes the data to "SGSInquiryList.jsp".

Only ongoing securities are to be selected from Oracle.

3.2.1.15 sgseapps.etender.SGSInquire

From Oracle database, this servlet retrieves all bid submissions of the user's bank for the selected security. It then passes the information to the page of "SGSInquiryDisplay.jsp".

MAS MDD users are able to see bid submissions from all banks.

3.2.1.16 sgseapps.etender.SGSAuctionList

This servlet reads data from Oracle database and passes the data to "SGSAuctionList.jsp".

It passes two lists of securities to the JSP file - today's auction results and recent one month's auction results.

3.2.1.17 sgseapps.etender.SGSAuctionDisplay

From Oracle database, this servlet retrieves the auction result for the current user and for the selected security. It then passes the information to the page of "SGSAuctionDisplay.jsp".

3.2.1.18 masnet.common.SvtMnetError

This servlet is used for exception handling. It generates a unique exception identification code for each exception and logs the exception in a log file.
It also automatically produces HTML code, which is eventually shown on user's browser.

3.2.1.19 sgseapps.etender.SGSMASDailyPrice

This servlet handles all daily closing price services for MAS users, Such as:

1. Edit/View/Modify bank's data. If bank's data is final, it directs to SGSMASDailyPriceDisplay.jsp. If the data is in draft, it directs to SGSMASDailyPriceInput.jsp.
2. Bank's submission status. It directs to SGSMASDailyPriceSubList.jsp, which shows a list of banks and their submission status.
3. Run today's exception report. It runs the report in the backend and passes the result to SGSMASDPExceptionReport.jsp.
4. Generate today's report. It calculates yield/price based on the formula in Appendix 7.6 and generates a .csv file. It will then output the .csv file to user.
   i) Today's report can only be re-generated if today's prices have not been released
   ii) Run of today's prices will be audit log in ABA0023_audit_action table
5. Download today/historical report. It reads data from database and generate a .csv file. It will then output the .csv file to user.
6. Edit price spread. It directs to SGSMASPriceSpreadList.jsp.
7. Manage public holiday. It direct to SGSMASPublicHoliday.jsp.

8. Release today's final prices. The following 3 tasks are done:
   i) Update Financial Database. It calls database stored procedure and updates tables in Financial Database.
   ii) Send today's daily price to MEPS+ and FMBS. It creates a request file and saves it in Unix file system. The backend AutoSys job will then generate price file.
   iii) Release of today's prices will be audit log in ABA0023_audit_action table. Today's prices can only be released if today's prices have not been released before.

3.2.1.20 sgseapps.etender.SGSMASDailyPriceMenu

This servlet always directs user to the page "SGSMASDailyPriceMenu.jsp".

3.2.1.21 sgseapps.etender.SGSMASDailyPriceConfirm

The servlet checks whether all fields are filled up and the numbers are in correct format. It also validates those numbers against the spread parameters in database. It generates error and warning messages accordingly.

If the submission passes validates, the servlet directs to SGSMASDailyPriceConfirm.jsp. Otherwise, it directs to SGSMASDailyPriceReInput.jsp.

3.2.1.22 sgseapps.etender.SGSMASDailyPriceResult

After a user confirms a submission, this servlet will be called.

The servlet stores data in database. If there is error occurred, it directs to MASDPTransactionFailed.jsp. Otherwise, it directs to MASDPTransactionSucceeded.jsp.

3.2.1.23 sgseapps.etender.SGSMASDailyPriceSpread

This servlet is used for handling user's price spread submission. It validates whether all fields are filled up and all numbers are in correct format. If the submission passes validation, the spread data is stored in database, and the servlet directs to MASPriceSpreadSucceeded.jsp. Otherwise, it directs to MASPriceSpreadFailed.jsp.

3.2.1.24 sgseapps.etender.SGSMASPublicHoliday

This servlet is used for handling user's public holiday submission. It only does very basic validation. If the submission passes validation, the spread data is stored in database, and the servlet directs to MASPublicHolidaySucceeded.jsp. Otherwise, it directs to MASPublicHolidayFailed.jsp.

3.2.1.25 sgseapps.etender.SGSUserDailyPrice

This servlet handles all daily closing price services for bank users. If the bank has already submitted data, it directs to SGSUserDailyPriceDisplay.jsp. If the bank has not submitted, it directs to SGSUserDailyPriceFileAttach.jsp. If the bank's submission is in draft mode, it directs to SGSUserDailyPriceInput.jsp.

3.2.1.26 sgseapps.etender.SGSUserDailyPriceMenu

This servlet always directs user to the page "SGSUserDailyPriceMenu.jsp".

3.2.1.27 sgseapps.etender.SGSUserDailyPriceInput

It reads from database to get all ongoing issues and bank's submitted data, then pass all data to SGSUserDailyPriceInput.jsp.

3.2.1.28 sgseapps.etender.SGSUserDailyPriceConfirm

The servlet checks whether all fields are filled up and the numbers are in correct format. It also validates those numbers against the spread parameters in database. It generates error and warning messages accordingly.

If the submission passes validates, the servlet directs to SGSUserDailyPriceConfirm.jsp. Otherwise, it directs to SGSUserDailyPriceReInput.jsp.

3.2.1.29 sgseapps.etender.SGSUserDailyPriceResult

After a user confirms a submission, this servlet will be called.

The servlet stores data in database. If there is error occurred, it directs to UserDPTransactionFailed.jsp. Otherwise, it directs to UserDPTransactionSucceeded.jsp.

3.2.1.30 sgseapps.etender.SGSPublicAnnounceList

This servlet reads data from Oracle database and passes the data to "SGSPublicAnnounceList.jsp".

It passes 3 securities to the JSP file – the most recent announced MAS Bills, 3-month T-bill, and 1 year T-bill or bond.

3.2.1.31 sgseapps.etender.SGSPublicAnnounce

This servlet reads the details information about the issue. If it is a T-bill, it directs to SGSPublictBillAnnounce.jsp. If it is a bond, it directs to SGSPublicBondAnnounce.jsp.

It also reads the announcement text from database and passes it to the jsp.

3.2.1.32 sgseapps.etender.SGSPublicResultList

This servlet reads data from Oracle database and passes the data to "SGSPublicResultList.jsp".

It passes 3 securities to the JSP file – the most recent auctioned MAS Bills, 3-month T-bill, and 1 year T-bill or bond.

3.2.1.33 sgseapps.etender.SGSPublicResult

This servlet reads the details information about the issue. If it is a T-bill, it directs to SGSPublictBillResult.jsp. If it is a bond, it directs to SGSPublicBondResult.jsp.

3.2.1.34 sgseapps.etender.SGSSubmitMethodSelect

This servlet allows users of 3 local banks to select submission methods – by web form or by file upload. If user selects by web form, this servlet re-direct the user to the ongoing security list page, and user selects a security to fill up the web application form. If user selects by file upload, this servlet re-direct the user to the file attach form.

3.2.1.35 sgseapps.etender.SGSSubmitUploadConfirm

The servlet will first decrypt the file using MAS private key. Then it will check whether the security is closed for subscription. If the closing date of the selected security is today and the time passes 12:00pm, then the security is considered as closed for subscription and all bids for the security will be rejected.

This servlet then validates the uploaded file using the business rules spelled in Section 7.10. There are some fatal errors, e.g. the file's checksum is wrong. If the servlet detects there is fatal error, the whole file will be rejected.

The servlet parses the uploaded file line by line. It finds header line, detail header line, bid details line, and trailer line. The servlet parse bid line, it uses the rules as section 7.10. If the bid passes validation, the servlet will store the bid in a Vector. If the bid fails in validation, the servlet will store the bid in another Vector.

After finishes parsing bid file, the servlet stores the 2 Vectors and messages in session variables, and re-direct to page "SGSSubmitUploadConfirm.jsp".

3.2.1.36 sgseapps.etender.SGSSubmitUploadResult

After a user confirms the upload, this servlet will be called.

The servlet reads the 2 Vectors from session variables. It reads each bid in the successful bid Vector, and forms a string as the format in section 3.2.1.8. The servlet checks whether the bank reference number in the bid exists in database. If yes, the servlet will move this bid to the unsuccessful bid Vector.

If no, it now is ready to call the encryption module. It reads the public key from database, and then uses the public key to encrypt the string, which is saved as a session variable previously.

Because the encrypted data is in binary format, another conversion is needed. The servlet will convert the encrypted data into an ASCII string.

The servlet will then save the ASCII string in the table aba0005_RSA_encrypted_transaction and increment the reference no in the table aba0001_security_master. The ASCII string is tailored into 4 pieces, which are saved in 4 fields in the database.

The servlet also saves bank's reference no and transaction reference no in clear format for banks' submission inquiry.

Finally, the servlet will save the 2 Vectors (successful and unsuccessful bids) to session variables, and re-direct to page "SGSSubmitUploadResult.jsp".

3.2.1.37 sgseapps.etender.SGSAnnouncementMenu

This servlet first checks whether the user is a MAS user. If yes, it always directs user to the page "SGSAnnouncementMenu.jsp".

3.2.1.38 sgseapps.etender.SGSAnnouncement

This servlet first checks whether the user is a MAS user.

This servlet processes user's request from "SGSAnnouncementMenu.jsp".

If the user selected to view the announcement page for an issue, this servlet will read the details of the issue and pass the information to "sgseapps.etender.SGSPublicAnnounce" for display.

If the user selected to release an issue, this servlet will change an indicator for the issue in database, and show a message on the screen. The release of announcement will be audit log in ABA0023_AUDIT_ACTION table.

3.2.1.39 sgseapps.etender.SGSMASIssueCalendarMenu

This servlet first checks whether the user is a MAS user. If yes, it always directs user to the page "SGSMASIssueCalendarMenu.jsp".

3.2.1.40 sgseapps.etender.SGSMASIssueCalendar

This servlet first checks whether the user is a MAS user.

This servlet processes user's request from "SGSMASIssueCalenderMenu.jsp".

If the user selected to input/view/modify the issuance calendar for a selected year, this servlet displays the inputted issues in a table, or an empty table if there is no inputted issues.

If the user selected to download issuance calendar for a selected, this servlet will read data from database and assemble a file for user to download.

3.2.1.41 sgseapps.etender.SGSMASIssueCalendarResult

This servlet first checks whether the user is a MAS user.

This servlet validates the issuance calendar inputted by user from “SGSMASIssueCalendar.jsp”. If the data can pass validation, this servlet will save data in database and display “Transaction succeeded” message. Otherwise, this servlet will display “Transaction failed” message.

3.2.1.42 sgseapps.etender.SGSUserIssueCalendarMenu

This servlet always directs user to the page "SGSUserIssueCalendarMenu.jsp".

3.2.1.43 sgseapps.etender.SGSUserIssueCalendar

This servlet will read issuance calendar data for a selected year from database and assemble a file for user to download.

3.2.1.44 sgseapps.etender.OneTransaction.java

This is an Model class of Transaction entity

3.2.1.45 sgseapps.etender.SGSAuctionMenu.java

This servlet will handle auction menu with new method to download auction results report in csv format

3.2.1.46 sgseapps.etender.SGSAuctionResult.java

This servlet will handle auction results.

3.2.2 Database access

Servlets use WAS managed connection pool to access Oracle database.

The Java class of "sgseapps.SQLManager" is used to get a connection from the connection pool, and to return the connection to the pool after use it.

The Oracle database instance name for SGS eApps is "mnet". The owner of this instance is "ms9aba". In the SQL statements implanted in Java servlets, all the table names should be prefixed by "ms9aba.". And the database instance owner should grant the access right to the Oracle database userid “ms7aba”, which is used by the Web application.

3.2.3 Integration with RSA/MAF Authentication server

The authentication part uses Factory Pattern. When there is authentication request, the servlet SGSAuthentication calls SGSAuthFactory.java to get an object of SGSAuthIntf, which could be instance of SGSNoAuthImpl.java, SGSRSAAuthImpl.java, or SGSSWAuthImpl.java, depending on the settings in the property file.

If it’s no authentication, userid with format of AxxxxTyy and any password can pass authentication. This is only used in Dev environment.

If it’s RSA authentication, SGSRSAAuthImpl.java will authenticate with RSA server and get result. RSA Java API is used to interact with RSA Authentication server.

If it’s CorpPass authentication, SGSCorpPassAuth.java will retrieve the NinjaTokenPrincipal to get the CorpPass authenticated login details from MAF.

If it’s CorpPass Simulator authentication, UEN, NRIC and any password can pass authentication. This is only used in Dev environment.

The configuration and deployment of RSA authentication are as Appendix 7.11.
The configuration and deployment of MAF authentication are as Appendix 7.17.

3.2.4 Error logging

When there is any exception, the servlet "masnet.common.SvtMnetError" will be called. It will in turn to call "masnet.common.MnetLogger" to write error messages to the log file. The file name of the log file is specified in the configuration file, which is represented by an instance of class "masnet.common.MnetConfig".

Each error message stored in the log file has an associated unique error identification code. Programmer, system administrator or helpdesk technician can open the log file, locate the error using the error id, and find the details of the error.

3.2.5 Integration with encryption module

The class “Encryptor” in package “sgseapps.data” are used for data encryption using RSA algorithm. This class makes use of singleton design pattern to ensure there is always only one instance running.

When other program calls this class, this class will first read the public key stored in database and do the encryption. The key size is of 2048 bit. The genkey.job will generate the key pair and store in the database tables. The public key is stored in MNET and private key is stored in PRI1. It is recommended to re-generate the key periodically and the historical keys are stored as well.

3.2.6 Some miscellaneous Java classes

There are several miscellaneous Java classes: "sgseapps.etender.AuctionResults", "sgseapps.etender.Securities" and "sgseapps.Utils".

The Java class "sgseapps.etender.AuctionResults" is used to represent one line of the auction result for the current user and for the selected security. The whole result may consist of many instances of "sgseapps.etender.AuctionResults".

The Java class "sgseapps.etender.Securities" is used to represent a security. Many "get" and "set" methods are provided to read or write a security's attribute.

The Java class "sgseapps.Utils" provides some static methods, such as date/time formatting, string formatting, etc.

3.2.7 Integration with TOPPAN Authentication

EICHandler.java will call REST API to EIC OIDC Relying Party to resolve login user information. The information will be provided by EIC OIDC RP’s response in JSON format and will be used for authentication.

3.3 Oracle database design

Oracle database is used to store security reference information, user's application data, auction result, and user's logon information. For SGS eApps system, it uses Oracle database instance "mnet" and table space "netapp".

Eighteen tables are created for SGS eApps system - "aba0001_security_master", "aba0005_rsa_text_enc_trans", "aba0006_auction_result", “aba0008_stage_security_master”, “aba0009_bank_master”, “aba0010_announce_text”, “aba0011_daily_price”, “aba0012_dp_status”, “aba0013_primary_dealer”, “aba0015_price_spread”, “aba0016_daily_extra_data”, “aba0017_final_daily_price”, “aba0018_final_extra_price”, “aba0019_public_holiday”, “aba0020_staging_bank_master”, “aba0021_issue_calendar”, “aba0022_non_benchmark”, and “aba0502_pub_key”. After the creation of those tables, the access rights of “insert”, “update” and “deleted” should be granted to the ms7aba user.

3.3.1 "aba0001_security_master"

This table stores security's reference information. The table creation statement is as below:

```
CREATE TABLE aba0001_security_master
     (aba0001_security_code          CHAR(8),
      aba0001_issue_no               CHAR(1),
      aba0001_issue_type             CHAR(1),
      aba0001_curr                   CHAR(3),
      aba0001_security_name          VARCHAR2(30)       NOT NULL,
      aba0001_issue_date             DATE               NOT NULL,
      aba0001_tender_date            DATE               NOT NULL,
      aba0001_issue_size             NUMBER(13)         NOT NULL,
      aba0001_qty_applied            NUMBER(13),
      aba0001_ave_yield              NUMBER(5,2),
      aba0001_cutoff_yield           NUMBER(5,2),
      aba0001_maturity_date          DATE               NOT NULL,
      aba0001_percent_coy            NUMBER(5,2),
      aba0001_percent_sub            NUMBER(5,2),
      aba0001_interest_rate          NUMBER(7,4),
      aba0001_tax_status             CHAR(1),
      aba0001_ave_price              NUMBER(7,4),
      aba0001_coy_price              NUMBER(7,4),
      aba0001_closing_price          NUMBER(7,4),
      aba0001_reference_no           NUMBER(5),
      aba0001_isin_code              CHAR(12),
      aba0001_announce_date          DATE,
      aba0001_result_load_date       DATE,
      aba0001_last_int_date          DATE,
      aba0001_next_int_date          DATE,
      aba0001_accrued_int_days       NUMBER(3),
      aba0001_int_date1              DATE,
      aba0001_int_date2              DATE,
      aba0001_int_paid_ind           CHAR(1)            DEFAULT 'N',
      aba0001_tenor                  NUMBER(3),
      aba0001_etender_ind            CHAR(1)            DEFAULT 'Y',
      aba0001_mas_applied            NUMBER(13),
      aba0001_mas_allotted           NUMBER(13),
      aba0001_nc_percent             NUMBER(5,2),
      aba0001_nc_qty_allot           NUMBER(13),
      aba0001_ex_int_date            DATE,
      aba0001_qty_app_comp           NUMBER(13),
```

```
      aba0001_qty_app_noncomp       NUMBER(13),
      aba0001_accrued_int           NUMBER(7,4)        DEFAULT 0,
      aba0001_announce_indicator    CHAR(1)            DEFAULT 'N',
      aba0001_median_yield          NUMBER(5,2),
      aba0001_median_price          NUMBER(7,4),
      CONSTRAINT pk_aba0001_security_master PRIMARY KEY
      (
            aba0001_security_code,
            aba0001_issue_no)
)
      TABLESPACE netapp
      STORAGE (
            INITIAL       20K
            NEXT          20K
            MINEXTENTS      1
            MAXEXTENTS    100
            PCTINCREASE     0);
```

"aba0001_security_code" and "aba0001_issue_no" are the primary of this table. Columns "aba0001_security_name", "aba0001_issue_date", "aba0001_tender_date", "aba0001_issue_size" and "aba0001_maturity_date" are required as "NOT NULL".

When there is a new issue, mainframe will generate a text file, which contains all the details about the issue. Oracle SQL/Loader script will load the data into this table.

Column "aba0001_reference_no" should be incremented after inserting a new record in table "aba0004_encrypted_transaction".

3.3.2 "aba0005_rsa_text_enc_trans"

This table is to store user's bid applications. The table creation statement is as below:

```
CREATE TABLE aba0005_RSA_text_enc_trans
     (aba0005_security_code         CHAR(8)            NOT NULL,
      aba0005_issue_no              CHAR(1)            NOT NULL,
      aba0005_bank_ref_no           VARCHAR(12)        NOT NULL,
      aba0005_ref_no                VARCHAR(5)         NOT NULL,
      aba0005_encrypted_data1       VARCHAR2(600)      NOT NULL,
      aba0005_encrypted_data2       VARCHAR2(600),
      aba0005_encrypted_data3       VARCHAR2(600),
      aba0005_encrypted_data4       VARCHAR2(600),
      aba0005_data_checksum         VARCHAR2(300),
      aba0005_received_date         DATE               DEFAULT SYSDATE,
      aba0005_received_time         DATE               DEFAULT SYSDATE,
      aba0005_submission_method     CHAR(1)            DEFAULT 'E',
      aba0005_withdrawn_time        DATE,
      aba0005_withdrawn_by          VARCHAR2(8),
      aba0005_submitted_by          VARCHAR2(8)
      CONSTRAINT fk_aba0005 FOREIGN KEY
      (
            aba0005_security_code,
            aba0005_issue_no
      )
      REFERENCES aba0001_security_master
      (
            aba0001_security_code,
```

```
            aba0001_issue_no
      ),
      CONSTRAINT pk_aba0005 PRIMARY KEY
      (
            aba0005_security_code,
            aba0005_issue_no,
            aba0005_bank_ref_no
      )
      CONSTRAINT SK_ABA0005 UNIQUE
      (
            ABA0005_SECURITY_CODE,
            ABA0005_ISSUE_NO,
            ABA0005_REF_NO
      ),
      )
      TABLESPACE netapp
      STORAGE (
                  INITIAL       500K
                  NEXT          500K
                  MINEXTENTS      1
                  MAXEXTENTS  1000
                  PCTINCREASE     0);
```

Columns "aba0001_security_code" and "aba0001_issue_no" in table "aba0001_security_master" are foreign keys of this table.

Columns "aba0004_security_code", "aba0004_issue_no" and “aba004_bank_ref_no” are primary keys of this table.

Columns “ABA0005_SECURITY_CODE”, “ABA0005_ISSUE_NO”, and “ABA0005_REF_NO” are unique for this table. This is to ensure in clustered server mode, there is no records with the same ref no (taken from ABA0001_security_master table) for the same issue.

SGS eApps system will insert a new record into this table after a bid application is successfully received.

The reason why the encrypted data is truncated to 4 columns is that the PL/SQL script has problem in handling long-length VARCHAR2 data when it generates text file using the data. After an issue is closed for subscription, PL/SQL script will load application data in this table and generate a text for mainframe to perform auction.

3.3.3 "aba0006_auction_result"

This table is used to store auction result. The table creation statement is as below:

```
CREATE TABLE aba0006_auction_result
(
      aba0006_security_code         CHAR(8)      NOT NULL,
      aba0006_issue_no              CHAR(1)      NOT NULL,
      aba0006_bank_acc_code         NUMBER(4)    NOT NULL,
      aba0006_tender_date           DATE         NOT NULL,
      aba0006_line_no               NUMBER(9)    NOT NULL,
      aba0006_line_content          VARCHAR2(200),
      aba0006_update_time           DATE         DEFAULT SYSDATE,
      CONSTRAINT fk_aba0006_auction_result FOREIGN KEY
      (
            aba0006_security_code,
            aba0006_issue_no
```

```
        )
        REFERENCES aba0001_security_master
        (
              aba0001_security_code,
              aba0001_issue_no
        ),
        CONSTRAINT pk_aba0006_auction_result PRIMARY KEY
        (
              aba0006_security_code,
              aba0006_issue_no,
              aba0006_bank_acc_code,
              aba0006_tender_date,
              aba0006_line_no
        )
)
        TABLESPACE netapp
        STORAGE
        (
              INITIAL       500K
              NEXT          500K
              MINEXTENTS      1
              MAXEXTENTS  1000
              PCTINCREASE     0
        );
```

Columns "aba0001_security_code" and "aba0001_issue_no" in table "aba0001_security_master" are foreign keys of this table.

Columns "aba0006_security_code", "aba0006_issue_no", "aba0006_bank_acc_code", "aba0006_tender_date", and "aba0006_line_no" are the unique key of this table.

When a new issue is closed for subscription and the auction is performed, mainframe will generate a text file, which contains the auction result. Oracle SQL/Loader script will load the data into this table.

3.3.4 "aba0008_stage_security_master"

Some fields of a security need to be updated after auction. The table "aba0008_stage_security_master" is used for temporarily storing of data in the data loading.

When loading security updated data, there are three steps:

- delete all old records in table "aba0008_stage_security_master"
- load data into table "aba0008_stage_security_master"
- update table "aba0001_security_master" using data in table "aba0008_stage_security_master"

```
CREATE TABLE aba0008_stage_security_master
(
        aba0008_security_code         CHAR(8),
        aba0008_issue_no              CHAR(1),
        aba0008_issue_type            CHAR(1),
        aba0008_curr                  CHAR(3),
        aba0008_security_name         VARCHAR2(30)       NOT NULL,
        aba0008_issue_date            DATE               NOT NULL,
        aba0008_tender_date           DATE               NOT NULL,
        aba0008_issue_size            NUMBER(13)         NOT NULL,
        aba0008_qty_applied           NUMBER(13),
```

```
        aba0008_ave_yield             NUMBER(5,2),
        aba0008_cutoff_yield          NUMBER(5,2),
        aba0008_maturity_date         DATE              NOT NULL,
        aba0008_percent_coy           NUMBER(5,2),
        aba0008_percent_sub           NUMBER(5,2),
        aba0008_nc_percent            NUMBER(5,2),
        aba0008_nc_qty_allot          NUMBER(13),
        aba0008_interest_rate         NUMBER(7,4),
        aba0008_tax_status            CHAR(1),
        aba0008_ave_price             NUMBER(7,4),
        aba0008_coy_price             NUMBER(7,4),
        aba0008_closing_price         NUMBER(7,4),
        aba0008_reference_no          NUMBER(5),
        aba0008_isin_code             CHAR(12),
        aba0008_announce_date         DATE,
        aba0008_result_load_date      DATE,
        aba0008_last_int_date         DATE,
        aba0008_next_int_date         DATE,
        aba0008_accrued_int_days      NUMBER(3),
        aba0008_int_date1             DATE,
        aba0008_int_date2             DATE,
        aba0008_int_paid_ind          CHAR(1)           DEFAULT 'N',
        aba0008_tenor                 NUMBER(3),
        aba0008_etender_ind           CHAR(1)           DEFAULT 'Y',
        aba0008_mas_applied           NUMBER(13),
        aba0008_mas_allotted          NUMBER(13),
        aba0008_nc_percent            NUMBER(5,2),
        aba0008_nc_qty_allot          NUMBER(13),
        aba0008_ex_int_date           DATE,
        aba0008_qty_app_comp          NUMBER(13),
        aba0008_qty_app_noncomp       NUMBER(13),
        aba0008_accrued_int           NUMBER(7,4)       DEFAULT 0,
        aba0008_announce_indicator    CHAR(1)           DEFAULT 'N',
        aba0008_cpn_paym_ind          CHAR(1),
        CONSTRAINT fk_aba0008_stage FOREIGN KEY
        (
              aba0008_security_code,
              aba0008_issue_no
        )
        REFERENCES aba0001_security_master
        (
              aba0001_security_code,
              aba0001_issue_no
        )
)
        TABLESPACE netapp
        STORAGE
        (
              INITIAL       20K
              NEXT          20K
              MINEXTENTS      1
              MAXEXTENTS    100
              PCTINCREASE     0
        );
```

3.3.5 “aba0009_bank_master”

This table is used to store bank list with auto-debit indicator. Only banks that authorized MAS to debit from their accounts are allowed to subscribe Singapore Government Securities and MAS Bills.

```
CREATE TABLE aba0009_bank_master
(
      aba0009_bank_code             CHAR(4)           NOT NULL,
      aba0009_bank_name             VARCHAR2(40)      NOT NULL,
      aba0009_bank_shortname        VARCHAR2(15)      NOT NULL,
      aba0009_autodebit_indicator   CHAR(1),
      aba0009_participant_indicator CHAR(1)           DEFAULT 'Y',
      CONSTRAINT pk_aba0009_bank_master PRIMARY KEY
      (
            aba0009_bank_code
      )
)
      TABLESPACE netapp
      STORAGE
      (
            INITIAL           10K
            NEXT              10K
            MINEXTENTS        1
            MAXEXTENTS        5
            PCTINCREASE       0
     );
```

The column “aba0009_bank_code” is the primary key of this table.

3.3.6 “aba0010_announce_text”

This stores announcement text, which is in the public announcement web pages. “aba0010_name” is either “bond” or “T-bill”.

```
CREATE TABLE aba0010_announce_text
(
      aba0010_name            VARCHAR2(8)       NOT NULL,
      aba0010_used_before     DATE              DEFAULT '01/JAN/2099',
      aba0010_text            VARCHAR2(4000)    NOT NULL,
      aba0010_last_updated    DATE              DEFAULT SYSDATE,
      CONSTRAINT pk_aba0010_announce_text PRIMARY KEY
      (
            aba0010_name,
            aba0010_used_before
      )
)
      TABLESPACE netapp
      STORAGE
      (
            INITIAL     10K
            NEXT        10K
            MINEXTENTS  1
            MAXEXTENTS  5
            PCTINCREASE 0
      );
```

The columns “aba0010_name” and “aba0010_used_before” are the primary keys of this table.

3.3.7 "aba0011_daily_price"

This table stores daily prices submitted by banks.

```
CREATE TABLE aba0011_daily_price
(
      aba0011_security_code         CHAR(8)           NOT NULL,
      aba0011_issue_no              CHAR(1)           NOT NULL,
      aba0011_bank_acc_code         NUMBER(4)         NOT NULL,
      aba0011_submission_date       DATE              NOT NULL,
      aba0011_bid_price             NUMBER(7,4)       DEFAULT 0,
      aba0011_offer_price           NUMBER(7,4)       DEFAULT 0,
      aba0011_high_price            NUMBER(7,4)       DEFAULT 0,
      aba0011_low_price             NUMBER(7,4)       DEFAULT 0,
      aba0011_volume                NUMBER(13)        DEFAULT 0,
      aba0011_update_time           DATE              DEFAULT SYSDATE,
      aba0011_BENCH_PRICE_FLAG      VARCHAR2(1),
      CONSTRAINT fk_aba0011_daily_price FOREIGN KEY
      (
            aba0011_security_code,
            aba0011_issue_no
      )
      REFERENCES aba0001_security_master
      (
            aba0001_security_code,
            aba0001_issue_no
      ),
      CONSTRAINT pk_aba0011_daily_price PRIMARY KEY
      (
            aba0011_security_code,
            aba0011_issue_no,
            aba0011_bank_acc_code,
            aba0011_submission_date
      )
)
      TABLESPACE netapp
      STORAGE
      (
            INITIAL     1000K
            NEXT        1000K
            MINEXTENTS  1
            MAXEXTENTS  1000
            PCTINCREASE 0
      );
```

Columns "aba0001_security_code" and "aba0001_issue_no" in table "aba0001_security_master" are foreign keys of this table.

Columns "aba0011_security_code", "aba0011_issue_no", "aba0011_bank_acc_code" and "aba011_submission_date" are primary keys of this table.

Columns aba0011_BENCH_PRICE_FLAG will have values 'Y' or 'N', to indicate whether the values aba0011_high_price and aba 0011_low_price should be taken as their value (value 'Y'). 'N' to indicate either there is no submission or it is not a benchmark issue.

One record in this table represents the daily price for one on-going SGS issue or MAS Bills issue, i.e. not matured issue, submitted by one bank.

3.3.8 "aba0012_dp_status"

This table is used to store the status of daily price submitted by banks. The column "aba0012_allow_edit_ind" is either "Y" or "N". Status of "Y" means the daily price submitted by bank "aba0012_bank_acc_code" is in draft mode. This bank is allowed to modify the data.

```
CREATE TABLE aba0012_dp_status
(
      aba0012_bank_acc_code         NUMBER(4)         NOT NULL,
      aba0012_submission_date       DATE              NOT NULL,
      aba0012_allow_edit_ind        CHAR(1)           DEFAULT 'N',
      aba0012_update_time           DATE              DEFAULT SYSDATE,
      ABA0012_USERID                VARCHAR2(8),
      CONSTRAINT pk_aba0012_DP_status PRIMARY KEY
      (
            aba0012_bank_acc_code,
            aba0012_submission_date
      )
)
      TABLESPACE netapp
      STORAGE
      (
            INITIAL     50K
            NEXT        50K
            MINEXTENTS  1
            MAXEXTENTS  1000
            PCTINCREASE 0
      );
```

The columns "aba0012_bank_acc_code" and "aba0012_submission_date" are the primary keys of this table.

3.3.9 "aba0013_primary_dealer"

This table is used to store primary dealer list with auto-debit indicator. The column "aba0013_masrepo_ind" represent whether this primary dealer is allowed to participate in repo submission.

```
CREATE TABLE aba0013_primary_dealer
(
      aba0013_bank_code             CHAR(4)           NOT NULL,
      aba0013_bank_name             VARCHAR2(40)      NOT NULL,
      aba0013_bank_shortname        VARCHAR2(15)      NOT NULL,
      aba0013_masrepo_ind           CHAR(1)           DEFAULT 'Y',
      CONSTRAINT pk_aba0013_primary_dealer PRIMARY KEY
      (
            aba0013_bank_code
      )
)
      TABLESPACE netapp
      STORAGE
      (
            INITIAL     10K
            NEXT        10K
            MINEXTENTS  1
            MAXEXTENTS  5
            PCTINCREASE 0
```

```
);
```

The column “aba0013_bank_code” is the primary key of this table.

3.3.10 “aba0015_price_spread”

This table stores various price spread ranges, which used in the validation before calculating daily closing price. For column “aba0015_type”, values “-1” (> 3 months) and “-2” (<= 3 months) represents price spreads for T-bill/MAS Bills, while values “0” – “29” (0-1 year to >29 years) are for bonds.

```
CREATE TABLE aba0015_price_spread
(
      aba0015_type                  NUMBER(2),
      aba0015_check_1               NUMBER(3,2),
      aba0015_check_2               NUMBER(3,2),
      aba0015_check_3               NUMBER(3,2),
      aba0015_check_4               NUMBER(3,2),
      aba0015_check_5               NUMBER(3,2),
      aba0015_check_6               NUMBER(3,2),

      CONSTRAINT pk_aba0015_price_spread PRIMARY KEY
      (
            aba0015_type
      )
)
      TABLESPACE netapp
      STORAGE
      (
            INITIAL     50K
            NEXT        50K
            MINEXTENTS  1
            MAXEXTENTS  1000
            PCTINCREASE 0
      );
```

The column “aba0015_type” is the primary key of this table.

3.3.11 “aba0016_daily_extra_data” (no longer in use)

This table stores values of overnight repo and 3-month comm bills submitted by banks.

```
CREATE TABLE aba0016_daily_extra_data
(
      aba0016_bank_acc_code         NUMBER(4)         NOT NULL,
      aba0016_submission_date       DATE              NOT NULL,
      aba0016_repo_bid_price        NUMBER(7,4)       DEFAULT 0,
      aba0016_repo_offer_price      NUMBER(7,4)       DEFAULT 0,
      aba0016_comm_bid_price        NUMBER(7,4)       DEFAULT 0,
      aba0016_comm_offer_price      NUMBER(7,4)       DEFAULT 0,
      aba0016_update_time           DATE              DEFAULT SYSDATE,
      CONSTRAINT pk_aba0016_daily_extra_data PRIMARY KEY
      (
            aba0016_bank_acc_code,
            aba0016_submission_date
```

```
        )
)
        TABLESPACE netapp
        STORAGE
        (
                INITIAL     100K
                NEXT        100K
                MINEXTENTS  1
                MAXEXTENTS  1000
                PCTINCREASE 0
        );
```

The columns “aba0016_bank_acc_code” and “aba0016_submission_date” are the primary keys of this table.

3.3.12 “aba0017_final_daily_price”

This table stores daily closing prices (average prices), dirty prices, MLA prices, yields, T+1 day dirty prices, Repo T+1 dirty prices , benchmark issues highest prices, and benchmark issues lowest prices.

```
CREATE TABLE aba0017_final_daily_price
(
        aba0017_security_code         CHAR(8)           NOT NULL,
        aba0017_issue_no              CHAR(1)           NOT NULL,
        aba0017_submission_date       DATE              NOT NULL,
        aba0017_high_price            NUMBER(7,4)       DEFAULT 0,
        aba0017_low_price             NUMBER(7,4)       DEFAULT 0,
        aba0017_ave_price             NUMBER(7,4)       DEFAULT 0,
        aba0017_dirty_price           NUMBER(7,4)       DEFAULT 0,
        aba0017_mla_price             NUMBER(7,4)       DEFAULT 0,
        aba0017_yield                 NUMBER(7,4)       DEFAULT 0,
        aba0017_update_time           DATE              DEFAULT SYSDATE,
        aba0017_t1_date               DATE,
        aba0017_t1_dirty_price        NUMBER(7,4)       DEFAULT 0,
        aba0017_repot1_date           DATE,
        aba0017_repot1_dirty_price    NUMBER(7,4)       DEFAULT 0,
        aba0017_t2_date               DATE,
        aba0017_repot2_date           DATE,
        aba0017_repot2_dirty_price    NUMBER(7,4)       DEFAULT 0,
        aba0017_bench_high_price      NUMBER(7,4)       DEFAULT 0,
        aba0017_bench_low_price       NUMBER(7,4)       DEFAULT 0,
        aba0017_BENCH_PRICE_FLAG      VARCHAR2(1),
        aba0017_t3_date_non_us        DATE,
        aba0017_t3_dirty_price_non_us NUMBER(7,4)       DEFAULT 0,
        CONSTRAINT fk_aba0017_final_daily_price FOREIGN KEY
        (
                aba0017_security_code,
                aba0017_issue_no
        )
        REFERENCES aba0001_security_master
        (
                aba0001_security_code,
                aba0001_issue_no
        ),
        CONSTRAINT pk_aba0017_final_daily_price PRIMARY KEY
        (
                aba0017_security_code,
```

```
            aba0017_issue_no,
            aba0017_submission_date
      )
)
      TABLESPACE netapp
      STORAGE
      (
            INITIAL     1000K
            NEXT        1000K
            MINEXTENTS  1
            MAXEXTENTS  1000
            PCTINCREASE 0
      );
```

Columns "aba0001_security_code" and "aba0001_issue_no" in table "aba0001_security_master" are foreign keys of this table.

Columns "aba0017_security_code", "aba0017_issue_no", and "aba017_submission_date" are primary keys of this table.

Columns aba0017_BENCH_PRICE_FLAG will have values 'Y' or 'N', to indicate whether the values aba0017_bench_high_price and aba 0017_bench_low_price should be taken as their value (value 'Y'). 'N' to indicate either there is no submission or it is not a benchmark issue.

One record in this table represents the daily price for one on-going SGS issue and MAS Bills, i.e. not matured issue.

3.3.13 "aba0018_final_extra_price" (no longer in use)

This table stores daily final prices for overnight repo rate and 3-month comm. bills. The column "aba0018_security_type" is either "comm." or "repo".

```
CREATE TABLE aba0018_final_extra_price
(
      aba0018_security_type         CHAR(4)           NOT NULL,
      aba0018_submission_date       DATE              NOT NULL,
      aba0018_high_price            NUMBER(7,4)       DEFAULT 0,
      aba0018_low_price             NUMBER(7,4)       DEFAULT 0,
      aba0018_ave_price             NUMBER(7,4)       DEFAULT 0,
      aba0018_update_time           DATE              DEFAULT SYSDATE,
      CONSTRAINT pk_aba0018_final_extra_price PRIMARY KEY
      (
            aba0018_security_type,
            aba0018_submission_date
      )
)
      TABLESPACE netapp
      STORAGE
      (
            INITIAL     1000K
            NEXT        1000K
            MINEXTENTS  1
            MAXEXTENTS  1000
            PCTINCREASE 0
      );
```

The columns "aba0018_security_type" and "aba0018_submission_date" are the primary keys of this table.

3.3.14 "aba0019_public_holiday"

This table stores public holidays.

```
CREATE TABLE aba0019_public_holiday
(
      aba0019_date                  DATE              NOT NULL,
      aba0019_description           VARCHAR(30),
      aba0019_country               VARCHAR(10)       DEFAULT 'SG',
      aba0019_update_time           DATE              DEFAULT SYSDATE,
      CONSTRAINT pk_aba0019_public_holiday PRIMARY KEY
      (
            aba0019_date,
            aba0019_country
      )
)
      TABLESPACE netapp
      STORAGE
      (
            INITIAL     50K
            NEXT        50K
            MINEXTENTS  1
            MAXEXTENTS  1000
            PCTINCREASE 0
     );
```

The columns "aba0019_date" and "aba0019_country" are the primary keys of this table.

3.3.15 "aba0020_staging_bank_master"

This table stores bank master data temporarily. It is used when loading bank.txt from MEPS+.

```
CREATE TABLE aba0020_staging_bank_master
(
      aba0020_bank_code             CHAR(4)           NOT NULL,
      aba0020_bank_name             VARCHAR2(40)      NOT NULL,
      aba0020_bank_shortname        VARCHAR2(15)      NOT NULL,
      aba0020_autodebit_indicator   CHAR(1),
      aba0020_participant_indicator CHAR(1)           DEFAULT 'Y',
      CONSTRAINT pk_aba0020_bank_master PRIMARY KEY
      (
            aba0020_bank_code
      )
)
      TABLESPACE netapp
      STORAGE
      (
            INITIAL       10K
            NEXT          10K
            MINEXTENTS      1
            MAXEXTENTS      5
            PCTINCREASE     0
```

```
);
```

The columns “aba0020_bank_code” is the primary key of this table.

3.3.16 “aba0021_issue_calendar”

This table stores issuance calendar.

```
CREATE TABLE aba0021_issue_calendar
(
      aba0021_security_code         VARCHAR2(8)       NOT NULL,
      aba0021_issue_no              VARCHAR2(1)       NOT NULL,
      aba0021_isin_code             VARCHAR2(12),
      aba0021_issue_type            VARCHAR2(1),
      aba0021_tenor                 NUMBER(3,0),
      aba0021_tenor_unit            VARCHAR2(1),
      aba0021_curr                  VARCHAR2(3),
      aba0021_new_reopen            VARCHAR2(1),
      aba0021_open_datetime         DATE,
      aba0021_close_datetime        DATE,
      aba0021_issue_datetime        DATE,
      aba0021_flag                  VARCHAR2(1)
      CONSTRAINT pk_aba0021_issue_calendar PRIMARY KEY
      (
            aba0021_security_code,
            aba0021_issue_no
      )
)
      TABLESPACE netapp
      STORAGE
      (
            INITIAL       20K
            NEXT          20K
            MINEXTENTS      1
            MAXEXTENTS      50
            PCTINCREASE     0
      );
```

3.3.17 “aba0022_non_benchmark”

This table stores non benchmark issues.

```
CREATE TABLE aba0022_non_benchmark
(
      aba0022_security_code         CHAR(8)     NOT NULL,
      aba0022_issue_no              CHAR(1)     NOT NULL,
      aba0022_update_time           DATE,
      CONSTRAINT pk_aba0022_non_benchmark PRIMARY KEY
      (
            aba0022_security_code,
            aba0022_issue_no
      ),
      CONSTRAINT fk_aba0022_non_benchmark FOREIGN KEY
      (
```

```
            aba0022_security_code,
            aba0022_issue_no
      )
      REFERENCES aba0001_security_master
      (
            aba0001_security_code,
            aba0001_issue_no))
      TABLESPACE netapp
      STORAGE
      (
            INITIAL       20K
            NEXT          20K
            MINEXTENTS      1
            MAXEXTENTS      50
            PCTINCREASE     0
      );
```

3.3.18 "aba0502_pub_key"

```
CREATE TABLE "MS9ABA"."ABA0502_PUB_KEY"
(
      "ABA0502_PUBLIC_KEY"    VARCHAR2(2048 BYTE),
      "ABA0502_UPDATED_DT"    DATE,
      "ABA0502_CURRENT_KEY"   VARCHAR2(1 BYTE)
);
```

3.3.19 "aba0023_audit_action"

This table stores the actions done in eApps. The trigger of ERF allotment will only proceed if the action of releasing the Closing Price has been taken.

```
CREATE TABLE "MS9ABA"."ABA0023_AUDIT_ACTION"
  (
    "ABA0023_MODULE"          VARCHAR2(20 BYTE) NOT NULL    ENABLE,
    "ABA0023_ACTION"          VARCHAR2(50 BYTE) NOT NULL    ENABLE,
    "ABA0023_USERID"          VARCHAR2(8 BYTE)  NOT NULL    ENABLE,
    "ABA0023_UPDATE_TIME"     DATE              DEFAULT     SYSDATE
  );
```

3.3.20 "ABA0214_SB_CD_FILE_TYPE"

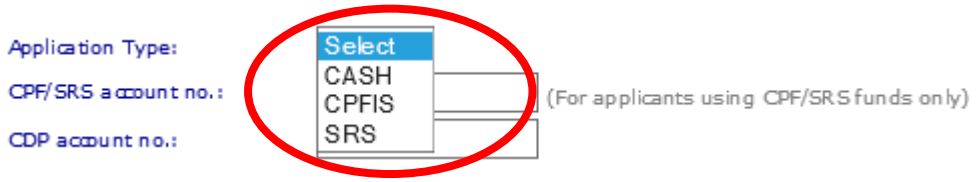
Enhance the existing file type table to include fields for eApps & eAuction usage.

```
Update TABLE ABA0214_SB_CD_FILE_TYPE add
(
      ABA0214_SB_TYPE_DISPLAY       VARCHAR(100) NOT NULL,
      ABA0214_SB_OBS_IND                  Boolean      NOT NULL DEFAULT N,
      ABA0214_SB_TYPE_CATEGORY            VARCHAR(25)
)
```

Data Info

ABA0214_SB_CD_FILE_TYPE	ABA0214_SB_DESCRIPTION	ABA0214_SB_TYPE_DISPLAY	ABA0214_SB_OBS_IND	ABA0214_SB_TYPE_CATEGORY
AP1	Application For Cash	CASH	N	Subscription
AP2	Application For CPFIS	CPFIS	Y	Subscription
AP3	Application For SRS	SRS	N	Subscription
RA1	Auction Result for CASH	CASH	N	AuctionRst
RA2	Auction Result for CPFIS	CPFIS	Y	AuctionRst
RA3	Auction Result for SRS	SRS	N	AuctionRst

1. In-Use Indicator denotes the state of File Type. Is it active or inactive.
2. UI interface program will read this field to display in the list box.



3.3.21 "ABA0030_CORP_PASS_MAPPING"

This table stores the CorpPass UEN, NRIC, Bank Code and Token User ID. CorpPass User details are inserted to this table using eAuction -> eApps User Maintenance

```
CREATE TABLE "MS9ABA"."ABA0030_CORP_PASS_MAPPING"
  (       "ABA0030_BANK_CODE" CHAR(4 BYTE) NOT NULL ENABLE,
          "ABA0030_CP_ENTITY_ID" VARCHAR2(10 BYTE) NOT NULL ENABLE,
          "ABA0030_CP_UID" VARCHAR2(100 BYTE) NOT NULL ENABLE,
          "ABA0030_TOKEN_USER_ID" VARCHAR2(8 BYTE) NOT NULL ENABLE,
          "ABA0030_ACTIVE_IND" VARCHAR2(1 BYTE) NOT NULL ENABLE,
          "ABA0030_LAST_LOGIN_DT" TIMESTAMP (6),
          "ABA0030_LAST_MODIFIED_DT" TIMESTAMP (6) DEFAULT SYSDATE NOT NULL ENABLE,
          "ABA0030_UPDATED_BY" VARCHAR2(20 BYTE) NOT NULL ENABLE,
           CONSTRAINT "ABA0030_CORP_PASS_MAPPING_PK" PRIMARY KEY ("ABA0030_BANK_CODE",
"ABA0030_CP_ENTITY_ID", "ABA0030_CP_UID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "NETAPP"  ENABLE,
           CONSTRAINT "ABA0030_CORP_PASS_MAPPING_UK" UNIQUE ("ABA0030_TOKEN_USER_ID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "NETAPP"  ENABLE
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
```

```
 STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
 TABLESPACE "NETAPP" ;

grant select,update,insert,delete on "MS9ABA"."ABA0030_CORP_PASS_MAPPING" to MS7ABA;
grant select,update,insert,delete on "MS9ABA"."ABA0030_CORP_PASS_MAPPING" to
MS_ABA_EDITOR_ROLE;
```

3.3.22 "ABA0032_EAPPS_CONFIG"

This table stores the eApps Configuration details

```
 CREATE TABLE "MS9ABA"."ABA0032_EAPPS_CONFIG"
  (       "ABA0032_KEY" VARCHAR2(50 BYTE) NOT NULL ENABLE,
          "ABA0032_VALUE" VARCHAR2(100 BYTE) NOT NULL ENABLE,
           CONSTRAINT "PK_ABA0032_EAPPS_CONFIG" PRIMARY KEY ("ABA0032_KEY")
 USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
 STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
 TABLESPACE "NETAPP"  ENABLE
  ) SEGMENT CREATION IMMEDIATE
 PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
 STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
 TABLESPACE "NETAPP" ;

GRANT SELECT, UPDATE, INSERT, DELETE ON "MS9ABA"."ABA0032_EAPPS_CONFIG" to MS7ABA;
GRANT SELECT, UPDATE, INSERT, DELETE ON "MS9ABA"."ABA0032_EAPPS_CONFIG" to
MS_ABA_EDITOR_ROLE;
```

3.3.23 "ABA0007_DETAIL_AUCTION_RESULT"

This table stores the Detail auction results.

```
 CREATE TABLE "MS9ABA"."ABA0007_DETAIL_AUCTION_RESULT"
  (       "ABA0007_SECURITY_CODE" VARCHAR2(8 BYTE) NOT NULL ENABLE,
          "ABA0007_ISSUE_NO" VARCHAR2(1 BYTE) NOT NULL ENABLE,
          "ABA0007_REF_NO" NUMBER(5,0),
          "ABA0007_BANK_REF_NO" VARCHAR2(12 BYTE),
          "ABA0007_FORM_NO" NUMBER(6,0) NOT NULL ENABLE,
          "ABA0007_TENDER_DATE" DATE NOT NULL ENABLE,
          "ABA0007_PRI_DLR_CODE" NUMBER(4,0) NOT NULL ENABLE,
          "ABA0007_NAME_OF_APPLN" VARCHAR2(30 BYTE),
          "ABA0007_NATIONALITY" VARCHAR2(1 BYTE),
          "ABA0007_IC_PASSPORT" VARCHAR2(14 BYTE),
          "ABA0007_COMP_NONCOMP" VARCHAR2(1 BYTE),
```

```
        "ABA0007_BID_YIELD" NUMBER(5,2),
        "ABA0007_APPLIED_AMT" NUMBER(11,0),
        "ABA0007_ALLOTED_AMT" NUMBER(11,0),
        "ABA0007_SETTLEMENT_AMT" NUMBER(15,2),
        "ABA0007_ACCRUED_INT" NUMBER(11,2),
        "ABA0007_CUST_BANK_CODE" NUMBER(4,0),
        "ABA0007_CUST_BANK_BC" VARCHAR2(1 BYTE),
        "ABA0007_TYPE_OF_APPLN" VARCHAR2(3 BYTE),
        "ABA0007_CDP_ACC_NO" VARCHAR2(16 BYTE),
        "ABA0007_SUB_MTD" CHAR(1 BYTE),
        "ABA0007_UPDATED_DATETIME" DATE DEFAULT SYSDATE,
        "ABA0007_CPF_ACNO" CHAR(16 BYTE),
        "ABA0007_FILE_TYPE" VARCHAR2(3 BYTE) DEFAULT 'AP1' NOT NULL ENABLE,
         CONSTRAINT "PK_ABA0007_IND_AUCT_RESULT" PRIMARY KEY
("ABA0007_SECURITY_CODE", "ABA0007_ISSUE_NO", "ABA0007_FORM_NO")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "NETAPP"  ENABLE
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 2097152 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "NETAPP" ;
```

3.3.24 “ABA0004_RETAIL_BID_TRANS”

This table stores the metadata details while processing the Retail bid file using eAuction Batch Retail Bid file process job.

```
  CREATE TABLE "MS9ABA"."ABA0004_RETAIL_BID_TRANS"
   (      "ABA0004_SECURITY_CODE" CHAR(8 BYTE) NOT NULL ENABLE,
        "ABA0004_ISSUE_NO" CHAR(1 BYTE) NOT NULL ENABLE,
        "ABA0004_BANK_REF_NO" VARCHAR2(12 BYTE) NOT NULL ENABLE,
        "ABA0004_FORM_NO" NUMBER(6,0) NOT NULL ENABLE,
        "ABA0004_RECEIVED_DATE" DATE DEFAULT SYSDATE,
        "ABA0004_RECEIVED_TIME" DATE DEFAULT SYSDATE,
        "ABA0004_SUBMISSION_METHOD" CHAR(1 BYTE) DEFAULT 'E',
        "ABA0004_SUBMITTED_BY" VARCHAR2(20 BYTE),
        "ABA0004_FILE_TYPE" VARCHAR2(3 BYTE) DEFAULT 'AP1' NOT NULL ENABLE,
         CONSTRAINT "PK_ABA0004_RETAIL_BID_TRANS" PRIMARY KEY
("ABA0004_SECURITY_CODE", "ABA0004_ISSUE_NO", "ABA0004_BANK_REF_NO")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "NETAPP"  ENABLE,
         CONSTRAINT "FK_ABA0004_RETAIL_BID_TRANS" FOREIGN KEY
("ABA0004_SECURITY_CODE", "ABA0004_ISSUE_NO")
```

```
        REFERENCES "MS9ABA"."ABA0001_SECURITY_MASTER" ("ABA0001_SECURITY_CODE",
"ABA0001_ISSUE_NO") ENABLE
  ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 10485760 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "NETAPP" ;

  grant select,update,insert,delete on "MS9ABA"."ABA0004_RETAIL_BID_TRANS" to MS7ABA;
  grant select,update,insert,delete on "MS9ABA"."ABA0004_RETAIL_BID_TRANS" to
MS_ABA_EDITOR_ROLE;
```

3.3.25 "AQA0020_AUCTION_RESULTS_REPORT"

This table stores auction results reports data will be populate on release auction results from eAuction for a specific date

```
CREATE TABLE "MS9ABA"."AQA0020_AUCTION_RESULTS_REPORT"(
        "AQA0020_RECORD_TYPE" CHAR(2 BYTE) ,
        "AQA0020_SECURITY_CODE" CHAR(8 BYTE) ,
        "AQA0020_ISSUE_NO" CHAR(1 BYTE) ,
        "AQA0020_FORM_NO" NUMBER(6,0),
        "AQA0020_REC_STATUS" CHAR(1 BYTE) ,
        "AQA0020_TENDER_DATE" DATE,
        "AQA0020_DEALER_CODE" NUMBER(4,0),
        "AQA0020_CUS_BANK" NUMBER(4,0),
        "AQA0020_CUS_CD" CHAR(1 BYTE) ,
        "AQA0020_CUS_SUB_CD" CHAR(3 BYTE) ,
        "AQA0020_QTY_APP" NUMBER(11,0),
        "AQA0020_BID_YL" NUMBER(5,2),
        "AQA0020_COMP_CD" CHAR(1 BYTE) ,
        "AQA0020_CUST_ACNO" VARCHAR2(20 BYTE) ,
        "AQA0020_APP_NAME" VARCHAR2(40 BYTE) ,
        "AQA0020_APP_TY" CHAR(3 BYTE) ,
        "AQA0020_QTY_ALLOT" NUMBER(11,0),
        "AQA0020_AMT_ALLOT" NUMBER(13,2),
        "AQA0020_PRICE" NUMBER(7,4),
        "AQA0020_NATIONALITY" CHAR(11 BYTE) ,
        "AQA0020_NRIC_PPNO" CHAR(14 BYTE) ,
        "AQA0020_CPF_ACNO" CHAR(16 BYTE) ,
        "AQA0020_BANK_REF_NO" VARCHAR2(12 BYTE) ,
        "AQA0020_CDP_ACNO" VARCHAR2(16 BYTE) ,
        "AQA0020_SUB_MTD" CHAR(1 BYTE) ,
        "AQA0020_FILE_TYPE" VARCHAR2(3 BYTE)  NOT NULL ENABLE,
        "AQA0020_ALLOT_IND" CHAR(1 BYTE) ,
        "AQA0020_EOVA_NRIC_PPNO" CHAR(1 BYTE) ,
        "AQA0020_ACCR_INT" NUMBER(11,2),
        "AQA0020_EOVA_CPF_ACNO"         CHAR(1 BYTE),
        "AQA0020_USERID" VARCHAR2(20 BYTE),
```

```
	"AQA0020_PAYMT_MOD" CHAR(1 BYTE) , CONSTRAINT
PK_AQA0009_SEC_AUCTION_PARA_STG PRIMARY KEY
(AQA0020_SECURITY_CODE,AQA0020_ISSUE_NO, AQA0020_FORM_NO))
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 32768 NEXT 32768 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "ABAAPP"  ENABLE
  ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 57344 NEXT 32768 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "ABAAPP" ;

grant select,update,insert,delete on "MS9ABA"."AQA0020_AUCTION_RESULTS_REPORT" to MS7ABA;
grant select,update,insert,delete on "MS9ABA"."AQA0020_AUCTION_RESULTS_REPORT" to
MS_ABA_EDITOR_ROLE;
```

3.3.26 "ABA0035_RETAILBID_FILE"

This table stores store encrypted retail bid files.

```
CREATE TABLE "MS9ABA"."ABA0035_RETAILBID_FILE"(
	"ABA0035_BANK_ACC_CODE" NUMBER(4,0) NOT NULL ENABLE,
	"ABA0035_PROCESSING_DATE" DATE NOT NULL ENABLE,
	"ABA0035_FILE_NAME" VARCHAR2(50 BYTE) COLLATE "USING_NLS_COMP" NOT NULL
ENABLE,
	"ABA0035_FILE" BLOB NOT NULL ENABLE,
	"ABA0035_MODIFIED_TIME" TIMESTAMP (6) NOT NULL ENABLE,
	"ABA0035_MODIFIED_USER_ID" VARCHAR2(8 BYTE) COLLATE "USING_NLS_COMP" NOT
NULL ENABLE,
	"ABA0035_ACKNOWLEDGEMENT_FILE" BLOB,
	"ABA0035_ACKNOWLEDGEMENT_FILE_NAME" VARCHAR2(150 BYTE) COLLATE
"USING_NLS_COMP",
	CONSTRAINT "UK_ABA0035" UNIQUE ("ABA0035_BANK_ACC_CODE",
"ABA0035_PROCESSING_DATE", "ABA0035_FILE_NAME")
	USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
	STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
	PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
	BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
	TABLESPACE "NETAPP"  ENABLE) SEGMENT CREATION IMMEDIATE
	PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
	NOCOMPRESS LOGGING STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1
MAXEXTENTS 2147483645
	PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
	BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
	TABLESPACE "NETAPP" ;

GRANT SELECT, UPDATE, INSERT, DELETE ON "MS9ABA"."ABA0035_RETAILBID_FILE" TO
MS_ABA_EDITOR_ROLE;
```

GRANT SELECT, UPDATE, INSERT, DELETE ON "MS9ABA"."ABA0035_RETAILBID_FILE" TO MS7ABA;

3.4 Back-end automation design

The back-end automation design consists of data loading into Oracle database, text file generating from the data in Oracle database, and error reporting to system operators.

When there is a new issue, MEPS+ will generate a text file, which contains all the details about the security. The file will be transferred to the Oracle database machine via MQ, then AutoSys will invokes a UNIX Shell program will load data into database.

MEPS+ calculates the interest dates for all issues, generate a text file and transfer it to SGS eApps every day. The file will be loaded to update the security master table in the database.

MEPS+ transfers a bank list file, which contains the auto-debit and participant indicator information, to PRDSYS every day. The data in the file will be loaded to replace the old content of the bank list table.

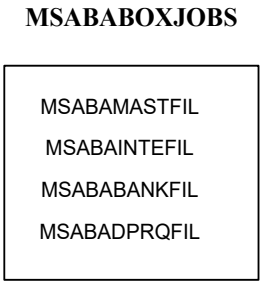
When MDD user triggers sending daily closing price file to MEPS+, a request file is generated and stored on PRDSYS. AutoSys will invoke a UNIX Shell program to generate a daily closing price file.

SGS eApps back-end also has error reporting function. It makes use of UNICENTRER TUNG for error reporting.

3.4.1 AutoSys jobs

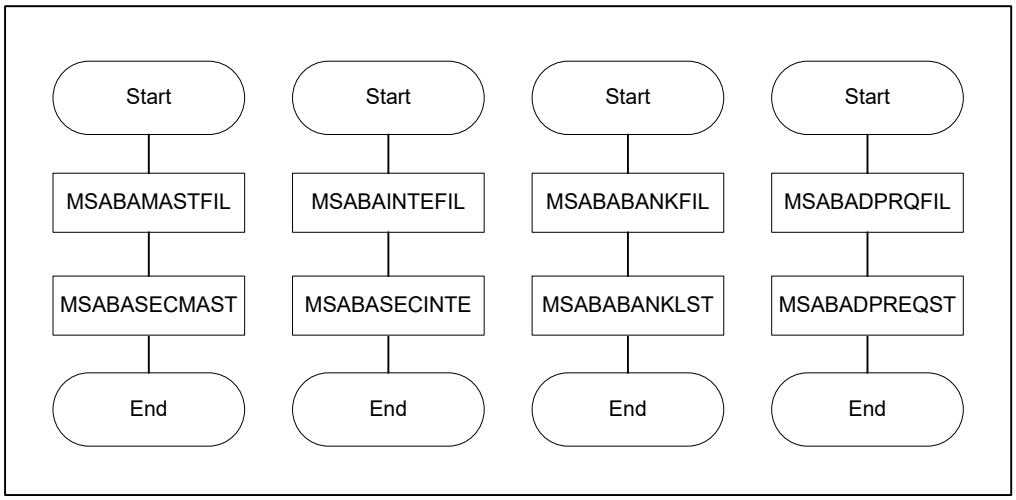
3.4.1.1 Summary

To define one box job MSABABOXJOBS and **three four watcher sub jobs**: MSABAMASTFIL, MSABAINTEFIL, MSABABANKFIL, and MSABADPRQFIL.

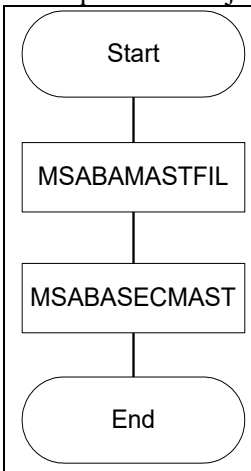


Job Name: MSABABOXJOBS
Description: Box job to control sub jobs
Command: None
Starting Command: From 7:00am – 9:00pm, Mon. – Fri
Machine to run: PRDSYS
Unix id: ms1ab

To define 4 sub jobs, which are all **file watcher** jobs.

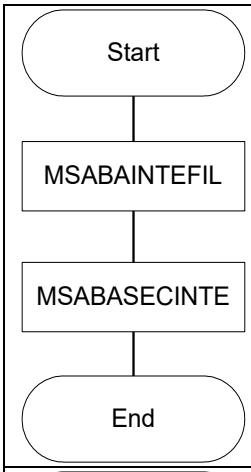
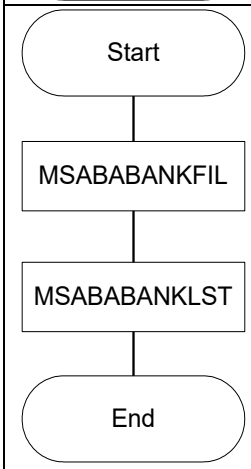
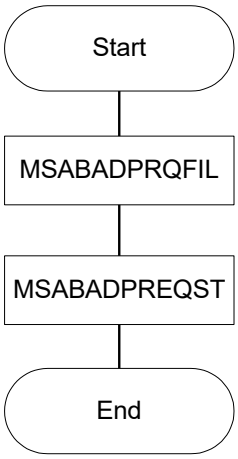


Description of sub jobs

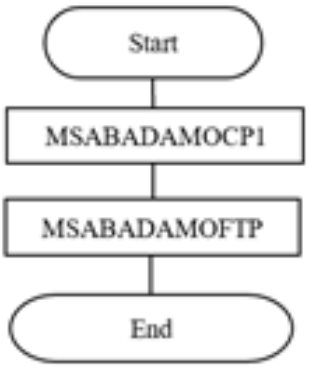


Job Name: MSABAMASTFIL
Description: Watch for secmast file and bring up loading job
File to watch: /mas_appl/aba/data/host/secmast.txt
Starting Command: Depends on master box job
Machine to run: PRDSYS
Unix id: ms1ab

Job Name: MSABASECMAST
Description: To load security master into database
Command: /mas_appl/aba/jobs/secmast.job
Starting Command: Success (MSABAMASTFIL)
Machine to run: PRDSYS
Unix id: ms1ab

	Job Name: MSABAINTEFIL Description: Watch for secint file and bring up loading job File to watch: /mas_appl/aba/data/host/secint.txt Starting Command: Depends on master box job Machine to run: PRDSYS Unix id: ms1ab Job Name: MSABASECINTE Description: To load security daily update into database Command: /mas_appl/aba/jobs/secint.job Starting Command: Success (MSABAINTEFIL) Machine to run: PRDSYS Unix id: ms1ab
	Job Name: MSABABANKFIL Description: Watch for bank list file and bring up loading job File to watch: /mas_appl/aba/data/host/bank.txt Starting Command: Depends on master box job Machine to run: PRDSYS Unix id: ms1ab Job Name: MSABABANKLST Description: To load bank list into database Command: /mas_appl/aba/jobs/bank.job Starting Command: Success (MSABABANKFIL) Machine to run: PRDSYS Unix id: ms1ab
	Job Name: MSABADPRQFIL Description: Watch for daily price request file and bring up extraction job File to watch: /was_share/SGSAppSrv/SGSeApps/SGSWeb/autosys/dpreq.txt Starting Command: From 7:00am – 9:00pm, Mon. – Fri Machine to run: PRDSYS* Unix id: ms1ab Job Name: MSABADPREQST Description: To generate daily price text file for MEPS+ Command: /mas_appl/aba/jobs/DailyPrice.job Starting Command: Success (MSABADPRQFIL) Machine to run: PRDSYS* Unix id: ms1ab *: For development environment, this one runs on DCSSYS3.

3.4.1.2 AMMO Closing Price Autosys Job

	Job Name: MSABADAMOCP1 Description: To generate closing price text file for AMMO Command: /mas_apps/aba/job/closingprice_ammo.job Starting Command: 7:30 am, Mon to Fri excluding Public Holiday Machine to run: PCSSYS90* Unix id: ms1ab *: For development environment, this one runs on DCSSYS2. Job Name: MSABADAMOFTP Description: To send closing price text file to AMMO Command: /mas_apps/aba/job/closingprice_ammo_ftp.job Starting Command: Success (MSABADAMOCP1) Machine to run: PCSSYS90* Unix id: ms2ab *: For development environment, this one runs on DCSSYS2.

Note:
To send AMMO closing price file to secondary SFTP server, use the below job
/mas_apps/aba/job/closingprice_ammo_ftp_secondary.job

3.4.2 Load new security sub-job

After MEPS+ transfers a text file named "secmast.txt" to the Oracle machine under directory "$App_dir/aba/data/host/", the loading new security sub-job “secmast.job” will be executed.

The sub-job uses the following logic:

- look for the file "secmast.txt" under the directory "$App_dir/aba/data/host/"
- if there is no such file, exit
- unscramble a file which contains encrypted password, and get Oracle id, password and connection string
- call Oracle SQL/Loader to load the text file into database
- backup “secmast.txt”
- analyze log file and notify computer operators using UNICENTER TUNG if there is any error
- backup log file

Oracle SQL/Loader uses "security.ctl". The content "security.ctl" is as below:

```
load data
infile "/mas_appl/aba/data/host/secmast.txt"
into table aba0001_security_master
APPEND(
      aba0001_security_code         POSITION(1-8)     CHAR,
      aba0001_issue_no              POSITION(9-9)     CHAR,
      aba0001_issue_type            POSITION(10-10)   CHAR,
      aba0001_curr                  POSITION(11-13)   CHAR,
      aba0001_security_name         POSITION(14-43)   CHAR,
      aba0001_issue_date            POSITION(44-51)   DATE 'YYYYMMDD'
":aba0001_issue_date",
      aba0001_tender_date           POSITION(52-59)   DATE 'YYYYMMDD'
":aba0001_tender_date",
      aba0001_issue_size            POSITION(60-72)   INTEGER EXTERNAL,
      aba0001_qty_applied           POSITION(73-85)   INTEGER EXTERNAL,
      aba0001_ave_yield       POSITION(86-90)   DECIMAL EXTERNAL
 ":aba0001_ave_yield/100",
      aba0001_cutoff_yield          POSITION(91-95)   DECIMAL EXTERNAL
":aba0001_cutoff_yield/100",
      aba0001_maturity_date         POSITION(96-103)  DATE 'YYYYMMDD'
":aba0001_maturity_date",
      aba0001_percent_coy           POSITION(104-108) DECIMAL EXTERNAL
 ":aba0001_percent_coy/100",
      aba0001_percent_sub           POSITION(109-113) DECIMAL EXTERNAL
 ":aba0001_percent_sub/100",
      aba0001_nc_percent            POSITION(114-118) DECIMAL EXTERNAL
 ":aba0001_nc_percent/100",
      aba0001_interest_rate         POSITION(119-125) DECIMAL EXTERNAL
 ":aba0001_interest_rate/10000",
      aba0001_tax_status            POSITION(126-126) CHAR,
      aba0001_coy_price       POSITION(127-133) DECIMAL EXTERNAL
 ":aba0001_coy_price/10000",
      aba0001_ave_price       POSITION(134-140) DECIMAL EXTERNAL
 ":aba0001_ave_price/10000",
      aba0001_closing_price         POSITION(141-147) DECIMAL EXTERNAL
 ":aba0001_closing_price/10000",
      aba0001_isin_code       POSITION(148-159) CHAR,
      aba0001_tenor                 POSITION(160-162) INTEGER EXTERNAL
,
      aba0001_etender_ind           POSITION(163-163) CHAR,
```

```
      aba0001_mas_applied           POSITION(164-176) INTEGER EXTERNAL
,
      aba0001_mas_allotted          POSITION(177-189) INTEGER EXTERNAL
,
      aba0001_int_paid_ind          POSITION(190-190) CHAR,
      aba0001_last_int_date         POSITION(191-198) DATE 'YYYYMMDD'
":aba0001_last_int_date",
      aba0001_next_int_date         POSITION(199-206) DATE 'YYYYMMDD'
":aba0001_next_int_date",
      aba0001_accrued_int_days      POSITION(207-209) INTEGER EXTERNAL
,
      aba0001_int_date1       POSITION(210-213) DATE 'DDMM' ":aba0001_int_date1",
      aba0001_int_date2       POSITION(214-217) DATE 'DDMM' ":aba0001_int_date2",
      aba0001_announce_date         SYSDATE)
```

The "security.ctl" defines the name of the input file, the name of the destination table, the rule that how data is loaded into corresponding fields, and the formula of data format transformation, etc.

The result of loading will be saved in the file "security.log" and bad data will be kept in the file "security.bad".

Oracle table will automatically impose primary key checking for the table "aba0001_security_master". The table always rejects duplicated records.

3.4.3 Load security daily update sub-job

After MEPS+ transfers a text file named "secint.txt" to the Oracle machine under directory "$App_dir/aba/data/host", the loading security daily update sub-job “secint.job” will be executed.

The sub-job uses the following logic:

- look for the file "secint.txt" under the directory "$App_dir/aba/data/"
- if there is no such file, exit
- unscramble a file which contains encrypted password, and get Oracle id, password and connection string
- call Oracle SQL/Plus to delete all records in the staging table “aba0008_stage_security_master”
- call Oracle SQL/Loader to load the text file into the staging table
- call Oracle SQL/Plus to update table “aba0001_security_master” using data in the staging table and update the interest rate in “aba0001_security_master” table only if the Coupon Payment Indicator is ‘Y’
- back up “secint.txt”
- analyze log file and notify computer operators using UNICENTER TUNG if there is any error
- backup log file
- generate a signal file “wcm_secint_cpn_paym.dat” in the path “/was_share/SGSAppSrv/signal” only if any of the issue code in secint.txt file has Coupon Payment Indicator ‘Y’

3.4.3.1 Delete data in staging table

Oracle SQL/Plus first uses “delete.sql”, which looks like follows:

```
delete from aba0008_stage_security_master;
quit
```

The script above simply deletes all the records in the table “aba0008_stage_security_master”.

3.4.3.2 load data into staging table

Oracle SQL/Loader uses "secint.ctl". The content " secint.ctl" is as below:

```
load data
infile "/mas_appl/aba/data/host/secint.txt"
into table aba0008_stage_security_master
APPEND
WHEN(01:02) <> 'GX'(
      aba0008_security_code         POSITION(1-8)     CHAR,
      aba0008_issue_no              POSITION(9-9)     CHAR,
      aba0008_issue_type            POSITION(10-10)   CHAR,
      aba0008_curr                  POSITION(11-13)   CHAR,
      aba0008_security_name         POSITION(14-43)   CHAR,
      aba0008_issue_date            POSITION(44-51)   DATE 'YYYYMMDD'
":aba0008_issue_date",
      aba0008_tender_date           SYSDATE,
      aba0008_issue_size            POSITION(60-72)   INTEGER EXTERNAL,
      aba0008_qty_applied           POSITION(73-85)   INTEGER EXTERNAL,
      aba0008_ave_yield       POSITION(86-90)   DECIMAL EXTERNAL
 ":aba0008_ave_yield/100",
      aba0008_cutoff_yield          POSITION(91-95)   DECIMAL EXTERNAL
":aba0008_cutoff_yield/100",
      aba0008_maturity_date         POSITION(96-103)  DATE 'YYYYMMDD'
":aba0008_maturity_date",
      aba0008_percent_coy           POSITION(104-108) DECIMAL EXTERNAL
 ":aba0008_percent_coy/100",
      aba0008_percent_sub           POSITION(109-113) DECIMAL EXTERNAL
 ":aba0008_percent_sub/100",
      aba0008_nc_percent            POSITION(114-118) DECIMAL EXTERNAL
 ":aba0008_nc_percent/100",
      aba0008_nc_qty_allot          POSITION(119-131) DECIMAL EXTERNAL,
      aba0008_interest_rate         POSITION(132-138) DECIMAL EXTERNAL
 ":aba0008_interest_rate/10000",
      aba0008_tax_status            POSITION(139-139) CHAR,
      aba0008_coy_price       POSITION(140-146) DECIMAL
EXTERNAL ":aba0008_coy_price/10000",
      aba0008_ave_price       POSITION(147-153) DECIMAL
EXTERNAL ":aba0008_ave_price/10000",
      aba0008_closing_price         POSITION(154-160) DECIMAL EXTERNAL
 ":aba0008_closing_price/10000",
      aba0008_isin_code       POSITION(161-172) CHAR,
      aba0008_tenor                 POSITION(173-175) INTEGER EXTERNAL
,
      aba0008_etender_ind           POSITION(176-176) CHAR,
      aba0008_mas_applied           POSITION(177-189) INTEGER EXTERNAL
,
      aba0008_mas_allotted          POSITION(190-202) INTEGER EXTERNAL
,
      aba0008_int_paid_ind          POSITION(203-203) CHAR,
      aba0008_last_int_date         POSITION(204-211) DATE 'YYYYMMDD'
":aba0008_last_int_date",
      aba0008_next_int_date         POSITION(212-219) DATE 'YYYYMMDD'
":aba0008_next_int_date",
      aba0008_accrued_int_days      POSITION(220-222) INTEGER EXTERNAL
,
      aba0008_int_date1       POSITION(223-226) DATE 'MMDD' ":aba0008_int_date1",
      aba0008_int_date2       POSITION(227-230) DATE 'MMDD' ":aba0008_int_date2",
```

```
        aba0008_ex_int_date          POSITION(231-238)       DATE 'YYYYMMDD'
        ":aba0008_ex_int_date",
        aba0008_cpn_paym_ind    POSITION(239-239) CHAR,
        aba0008_result_load_date      SYSDATE)
```

The "secint.ctl" defines the name of the input file, the name of the destination table, the rule that how data is loaded into corresponding fields, and the formula of data format transformation, etc.

The result of loading will be saved in the file "secint.log" and bad data will be kept in the file "secint.bad".

3.4.3.3 update security master table

Oracle SQL/Plus uses “secint.sql”, which looks like follows:

```
SET serveroutput on;
DECLARE
        result_count  NUMBER(2);
        CURSOR c1 IS
        SELECT * FROM aba0008_stage_security_master;
        result_count_sb     NUMBER(2);
        CURSOR c2 IS
        SELECT * FROM ABA0108_SB_STAGE_SEC_MASTER;
BEGIN
        SELECT COUNT(*) INTO result_count FROM aba0008_stage_security_master;
        IF result_count > 0 THEN
                FOR result_rec IN c1 LOOP
                        UPDATE aba0001_security_master
                        SET    aba0001_last_int_date = result_rec.aba0008_last_int_date,
                               aba0001_next_int_date = result_rec.aba0008_next_int_date,
                               aba0001_int_paid_ind = result_rec.aba0008_int_paid_ind,
                               aba0001_ex_int_date = result_rec.aba0008_ex_int_date,
                               aba0001_result_load_date = SYSDATE
                        WHERE aba0001_security_code = result_rec.aba0008_security_code AND
                               aba0001_issue_no = '1';

                        UPDATE aba0001_security_master
                        SET    aba0001_interest_rate = result_rec.aba0008_interest_rate
                        WHERE aba0001_security_code = result_rec.aba0008_security_code AND
                               aba0001_issue_no = '1' AND result_rec.aba0008_cpn_paym_ind = 'Y';
                END LOOP;
                COMMIT;
        END IF;

        SELECT COUNT(*) INTO result_count_sb FROM ABA0108_SB_STAGE_SEC_MASTER;
        IF result_count_sb > 0 THEN
                FOR result_rec_sb IN c2 LOOP
                        UPDATE aba0101_sb_security_master
                        SET    aba0101_sb_last_int_date = result_rec_sb.aba0108_sb_last_int_date,
                               aba0101_sb_next_int_date = result_rec_sb.aba0108_sb_next_int_date,
                               aba0101_sb_int_paid_ind = result_rec_sb.aba0108_sb_int_paid_ind,
                               aba0101_sb_ex_int_date = result_rec_sb.aba0108_sb_ex_int_date

                        WHERE aba0101_sb_security_code = result_rec_sb.aba0108_sb_security_code AND
                               aba0101_sb_issue_no = '1';
                END LOOP;
                COMMIT;
        END IF;
END;
/
exit;
```

The above script updates security master table using data in the staging table.

3.4.4 Load bank list

After mainframe transfers a text file named "bank.txt" to the Oracle machine under directory "$App_dir/aba/data/", the loading bank list sub-job will be executed.

The sub-job uses the following logic:

- look for the file "bank.txt" under the directory "$App_dir/aba/data/"
- if there is no such file, exit
- unscramble a file which contains encrypted password, and get Oracle id, password and connection string
- call Oracle SQL/Loader to load the text file into database and replace the old content in the table
- backup “bank.txt”
- analyze log file and notify computer operators using UNICENTER TUNG if there is any error
- backup log file

Oracle SQL/Loader uses "bank.ctl". The content "bank.ctl" is as below:

```
load data
infile "/mas_appl/aba/data/bank.txt"
into table aba0009_bank_master
REPLACE
(
        aba0009_bank_code               POSITION(1-4)   CHAR,
        aba0009_bank_name               POSITION(5-44)  CHAR,
        aba0009_bank_shortname          POSITION(45-59) CHAR,
        aba0009_autodebit_indicator     POSITION(60-60) CHAR,
        aba0009_participant_indicator   POSITION(61-61) CHAR)
```

The "bank.ctl" defines the name of the input file, the name of the destination table, the rule that how data is loaded into corresponding fields, and the formula of data format transformation, etc.

The result of loading will be saved in the file "bank.log" and bad data will be kept in the file "bank.bad".

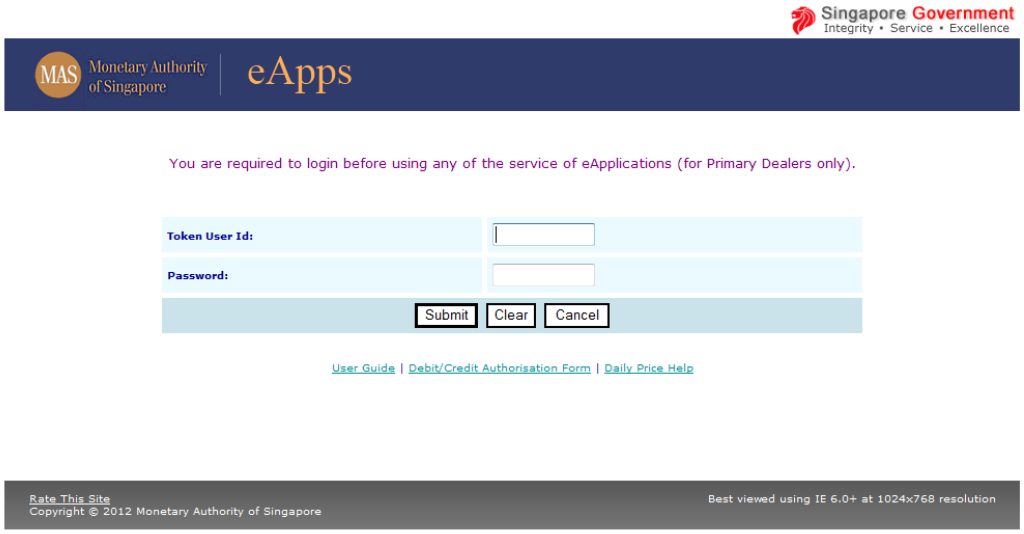
Oracle table will automatically impose primary key checking for the table "aba0009_bank_code". The table always rejects duplicated records.

4 User interface

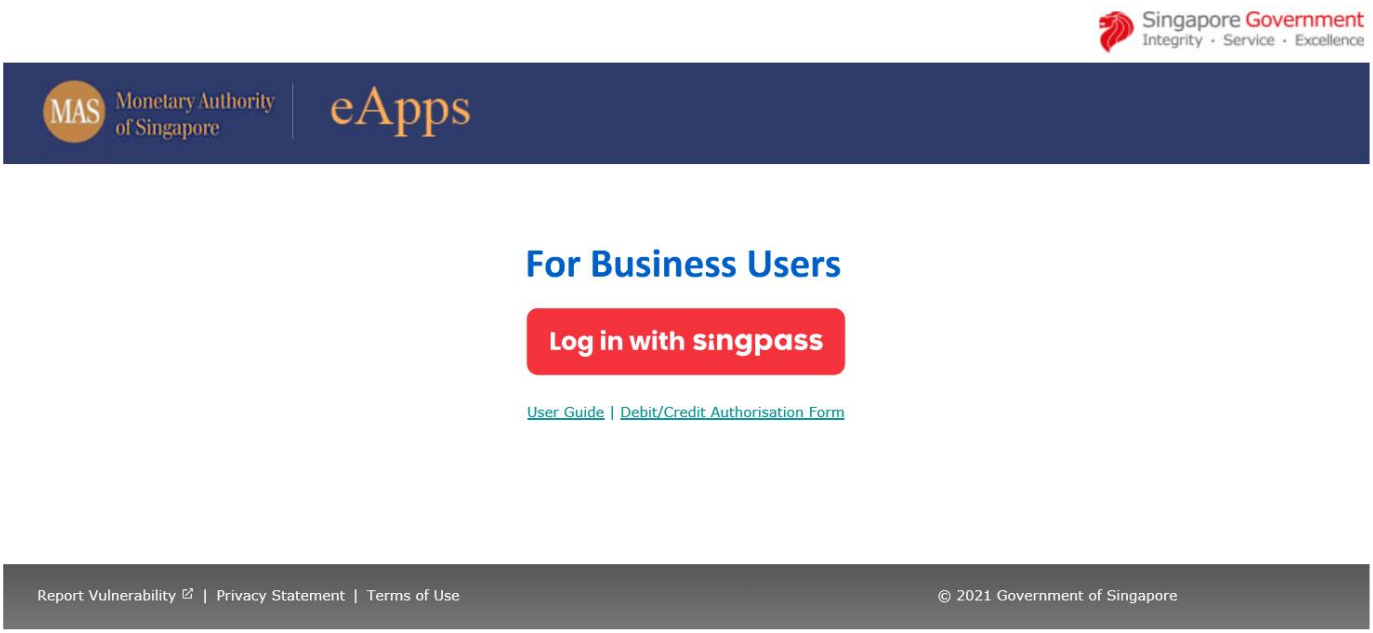
4.1 Web interface

4.1.1 Logon page to SGS eApps

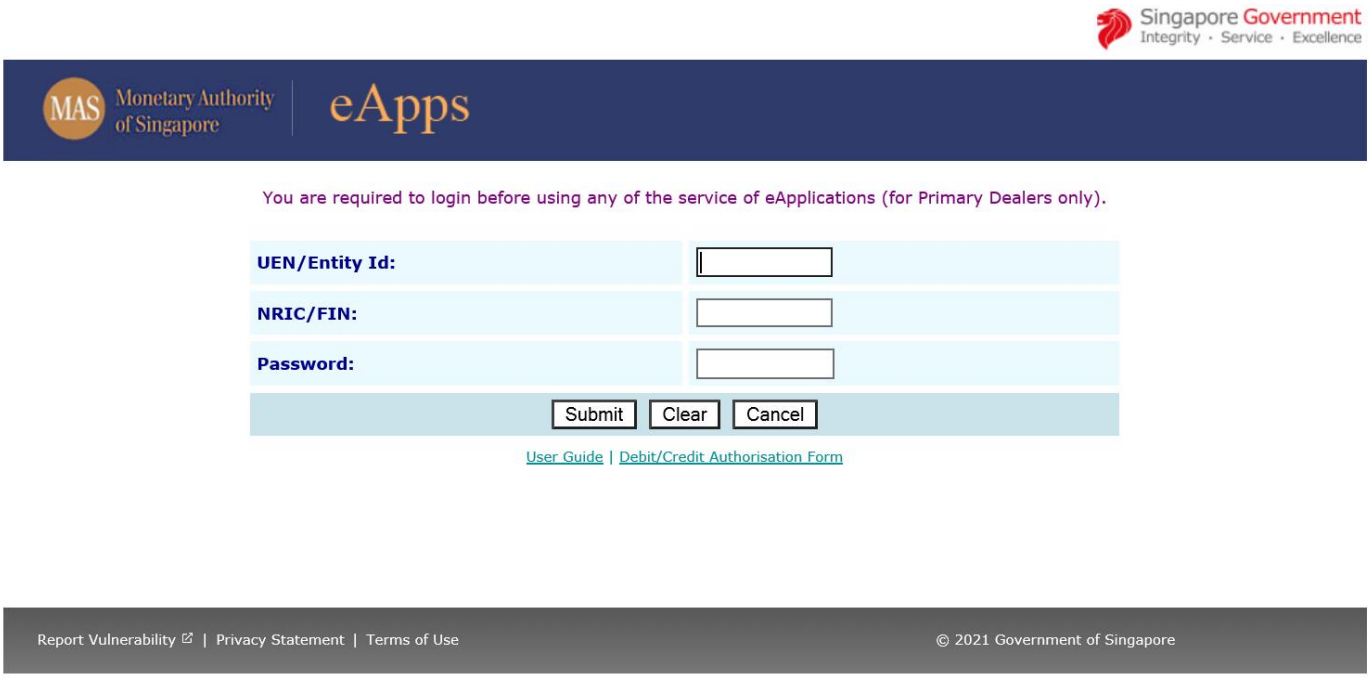
RSA/RSA Simulator Login



CorpPass/CorpPass TOPPAN Integration Login

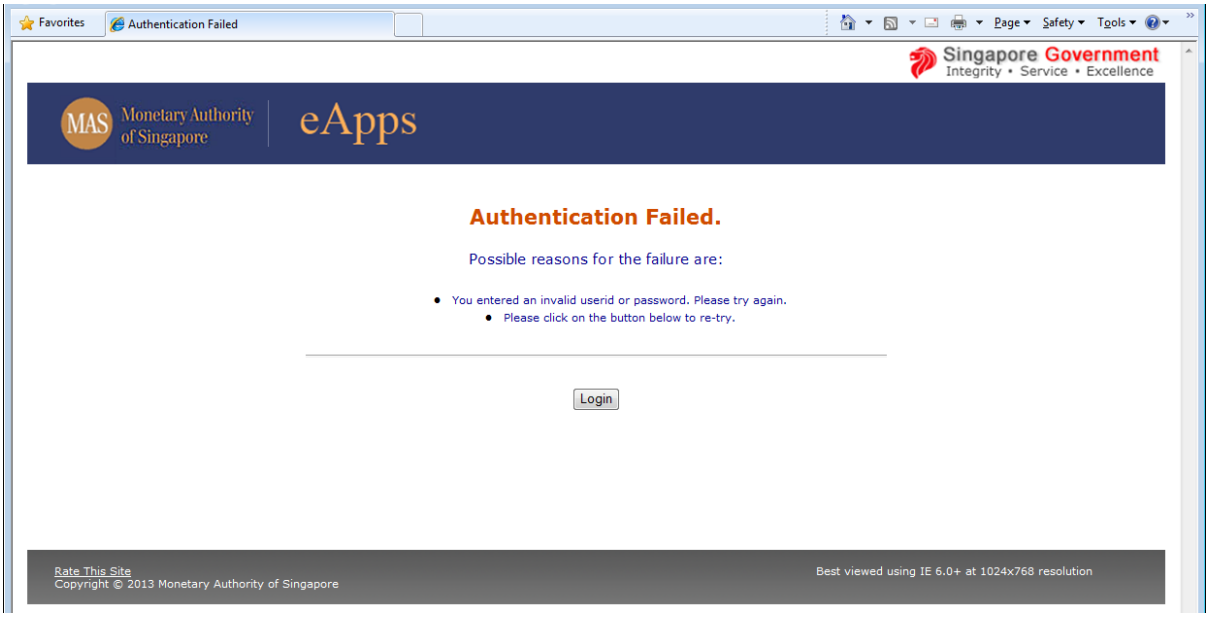


CorpPass Simulator Login

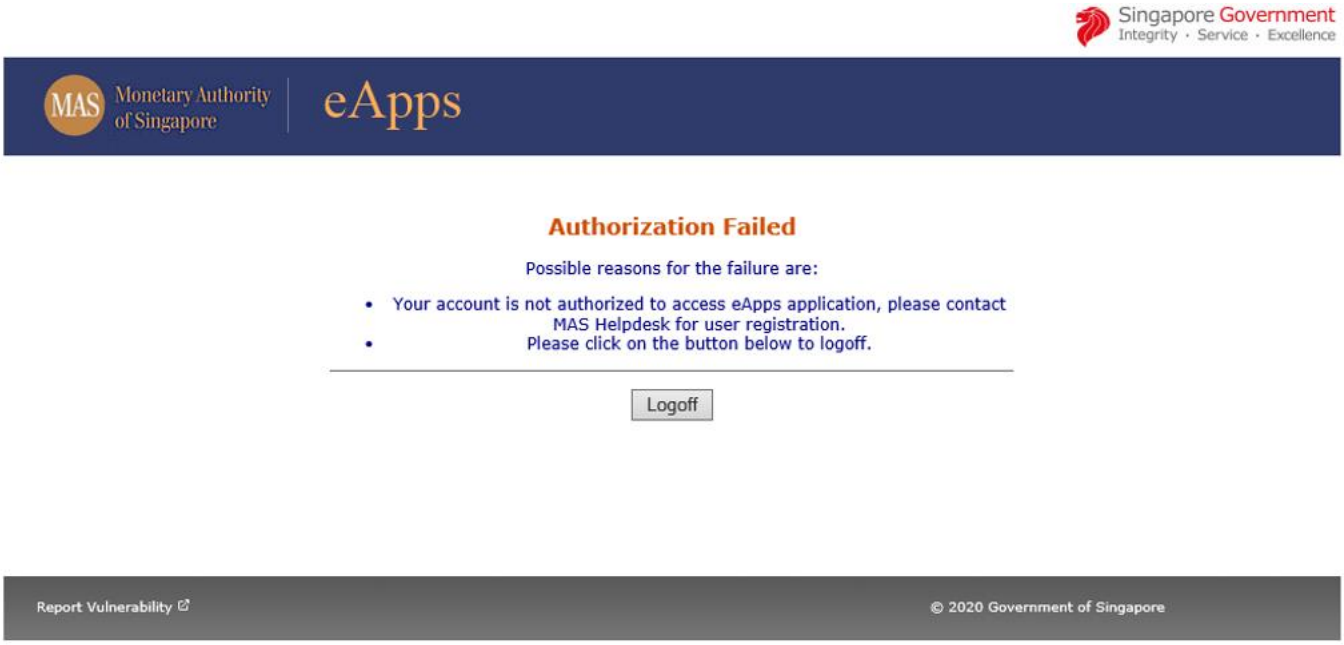


4.1.2 Authentication failed page

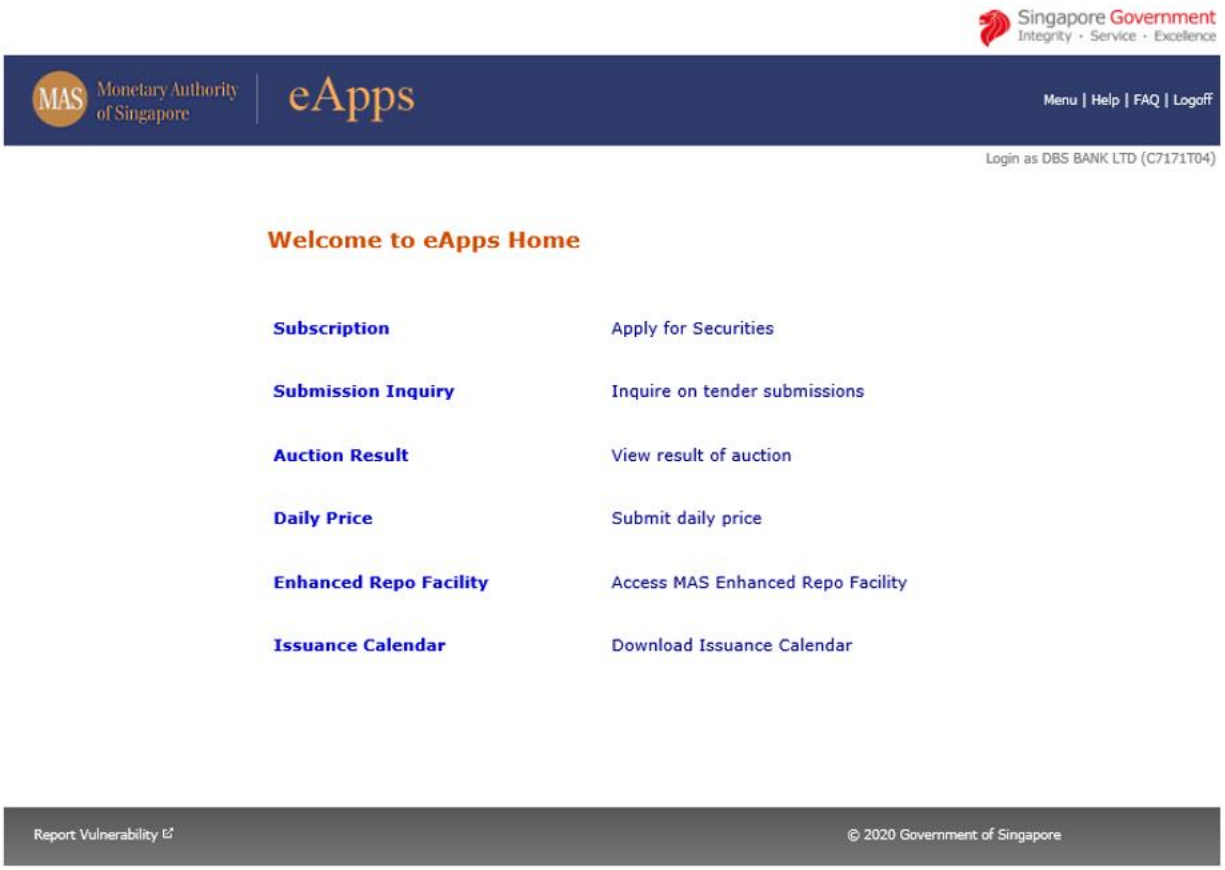
RSA/RSA Simulator Authentication failed



CorpPass/CorpPass Simulator/CorpPass TOPPAN Integration Authorization failed page



4.1.3 SGS eApps home page



4.1.4 SGS ongoing issue list page

MAS Bills

Security Code	Security Description	Issue Date	Closing Date	Issue Size
ML13529U	12 wk Mbill	01/06/2013	29/05/2013	1,000,000,000
ML13530U	12 wk Mbill	01/06/2013	30/05/2013	1,000,000,000
ML13531U	12 wk Mbill	01/06/2013	31/05/2013	1,000,000,000

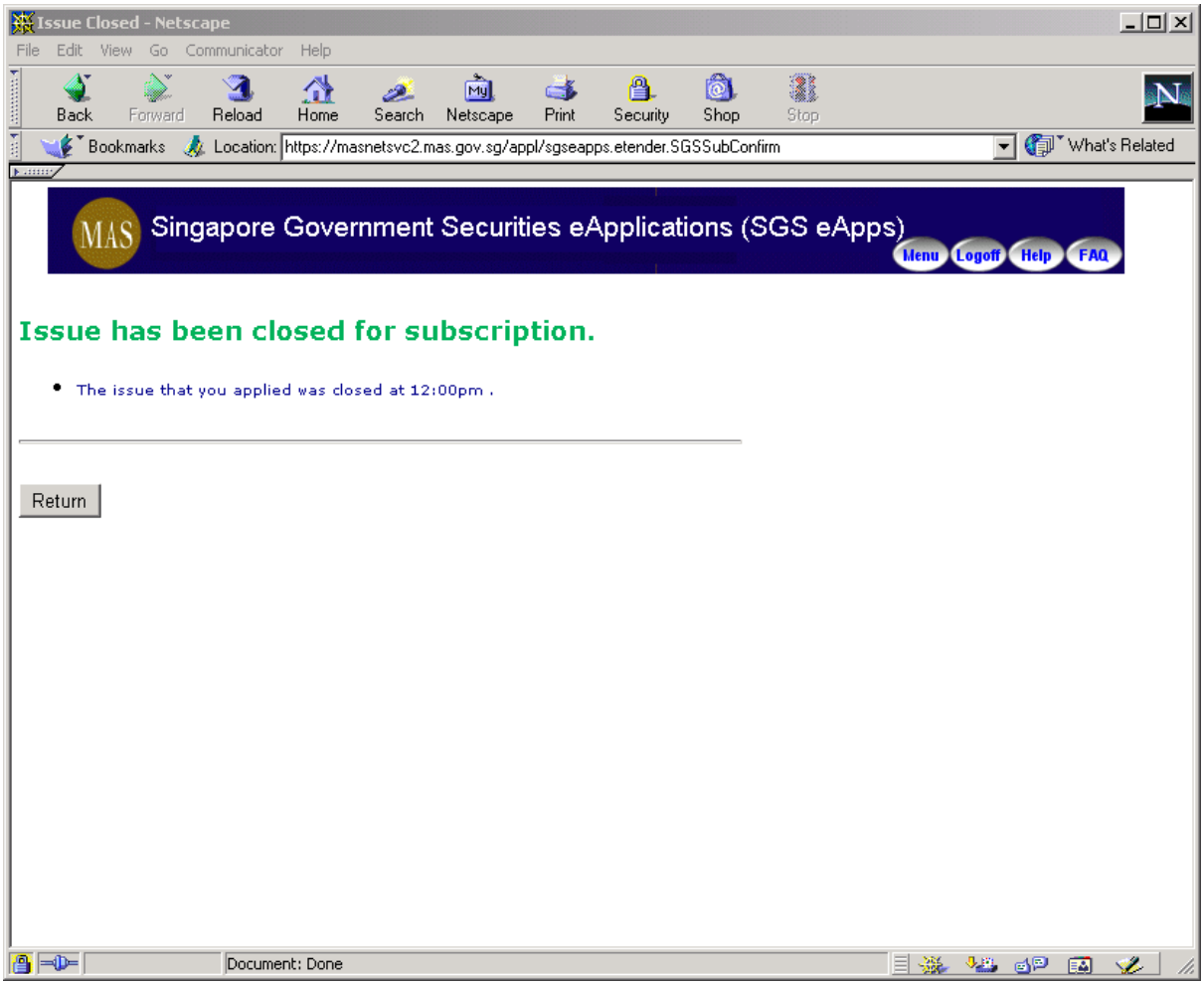
T Bills

Security Code	Security Description	Issue Date	Closing Date	Issue Size
BS13529U	6 mths Tbill	01/06/2013	29/05/2013	1,000,000,000
BS13530U	6 mths Tbill	01/06/2013	30/05/2013	1,000,000,000
BS13531U	6 mths Tbill	01/06/2013	31/05/2013	1,000,000,000

SGS Bonds

Security Code	Security Description	Issue Date	Closing Date	Issue Size
N213201D	2-yr Bond	28/03/2014	27/03/2014	2,000,000,000
N213529U	2-yr Bond	01/06/2013	29/05/2013	2,000,000,000
N213530U	2-yr Bond	01/06/2013	30/05/2013	2,000,000,000
N213531U	2-yr Bond	01/06/2013	31/05/2013	2,000,000,000

4.1.5 SGS issue closed page



4.1.6 SGS application form page

Singapore Government
Integrity • Service • Excellence

Monetary Authority of Singapore | eApps

Menu | Help | FAQ | Logoff

Login as DBS BANK LTD

APPLICATION FOR MAS BILLS

Help

12 wk Mbill ML13529U to be issued on 01/06/2013

	Nominal amount ($)	Competitive	Yield
1.		Y	
2.		Y	
3.		Y	
4.		Y	
5.		Y	

(The yield for competitive bids must be in percentage terms up to 2 decimal places.)

Type of applicant: NIL

Full name of applicant: (For non-bank only)

Applicant's address:

Securities account no. with MAS: (e.g.9999)

The alloted securities will be transfered to the Bank account.

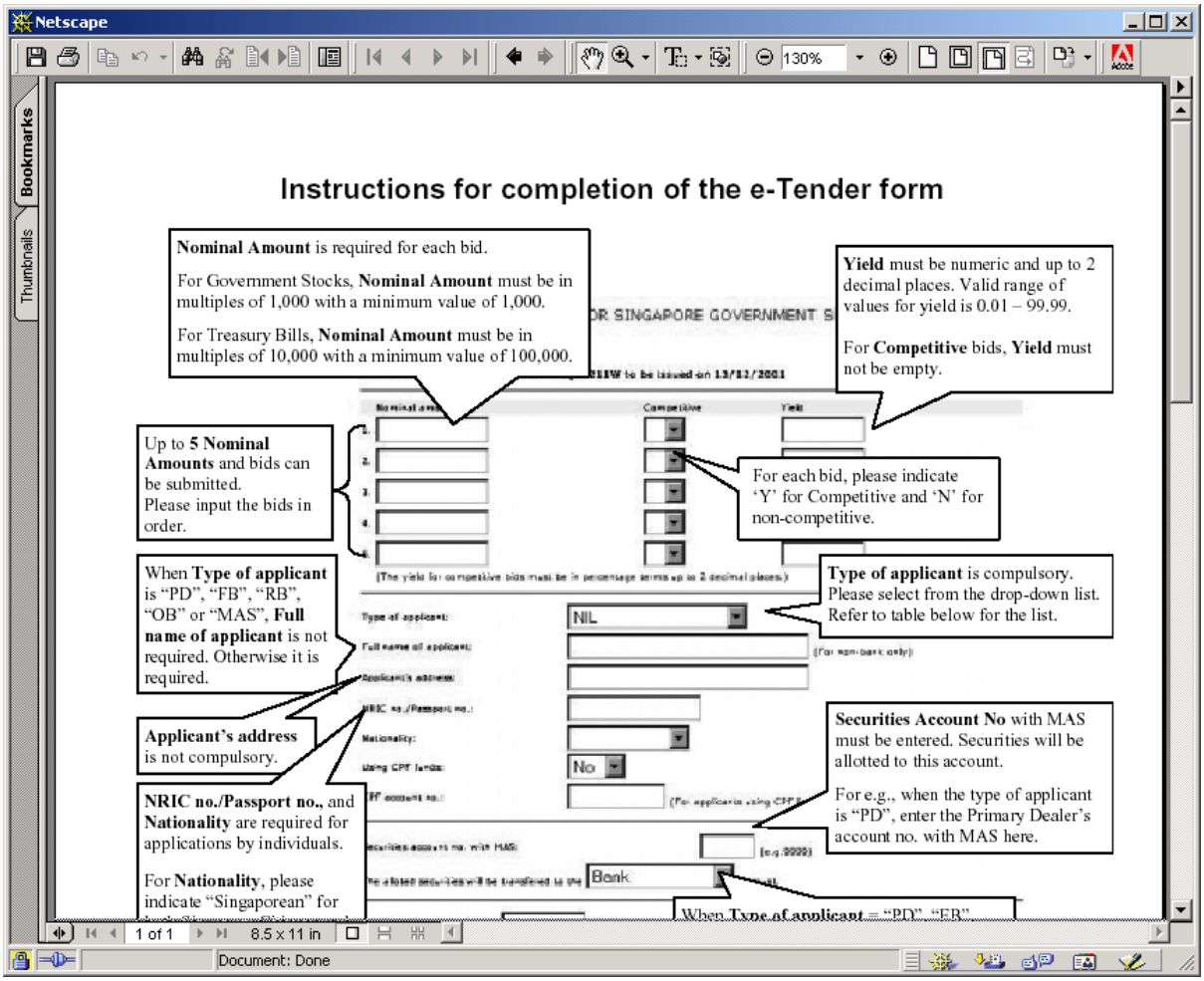
Bank's reference no.: 7171-

Primary Dealer's token user id: A7171T01

Important: By clicking the 'Submit' button below, you are indicating your agreement to be bound by the terms and conditions set out in the SGS Electronic Application Service Agreement.

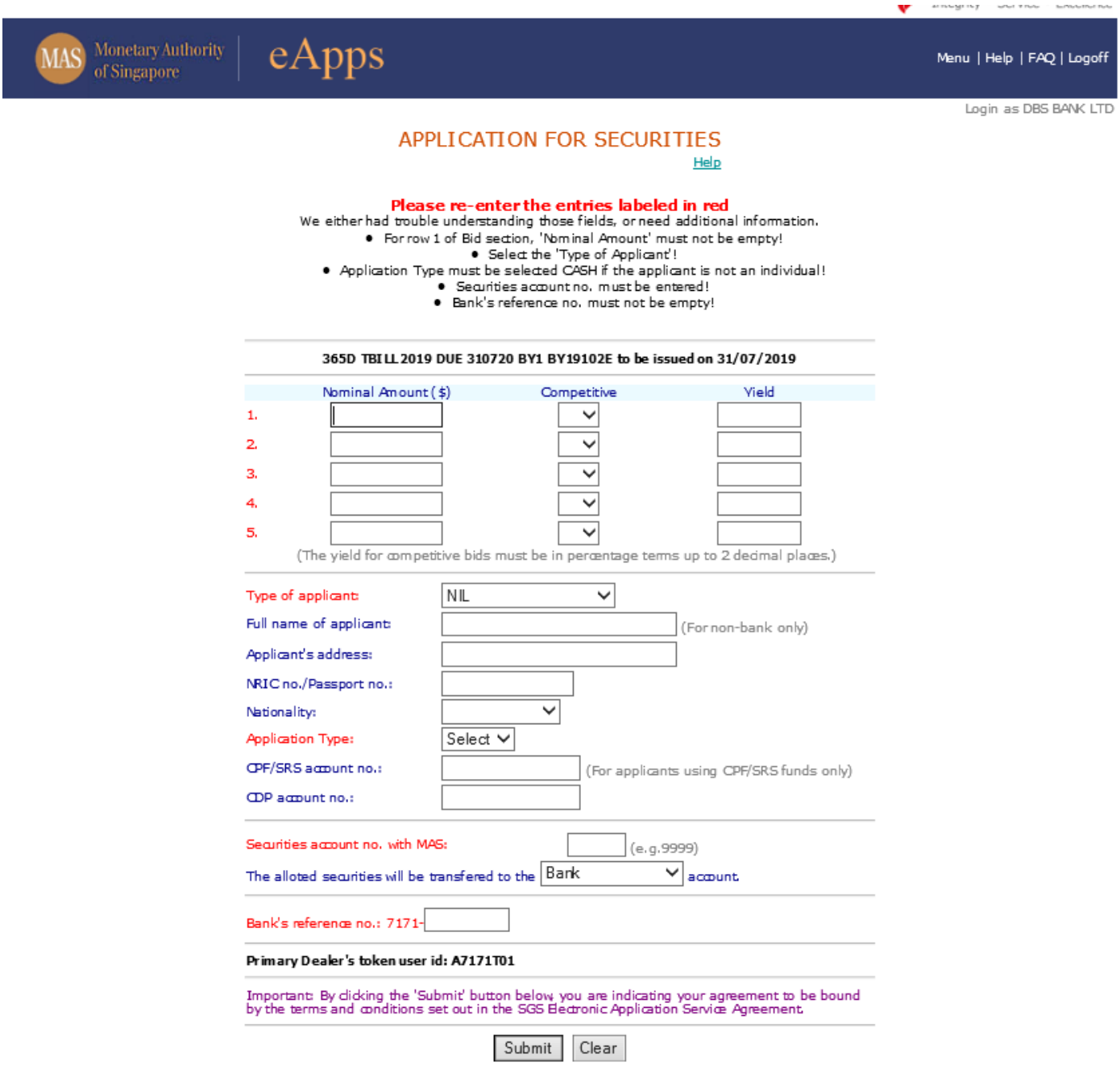
Submit Clear

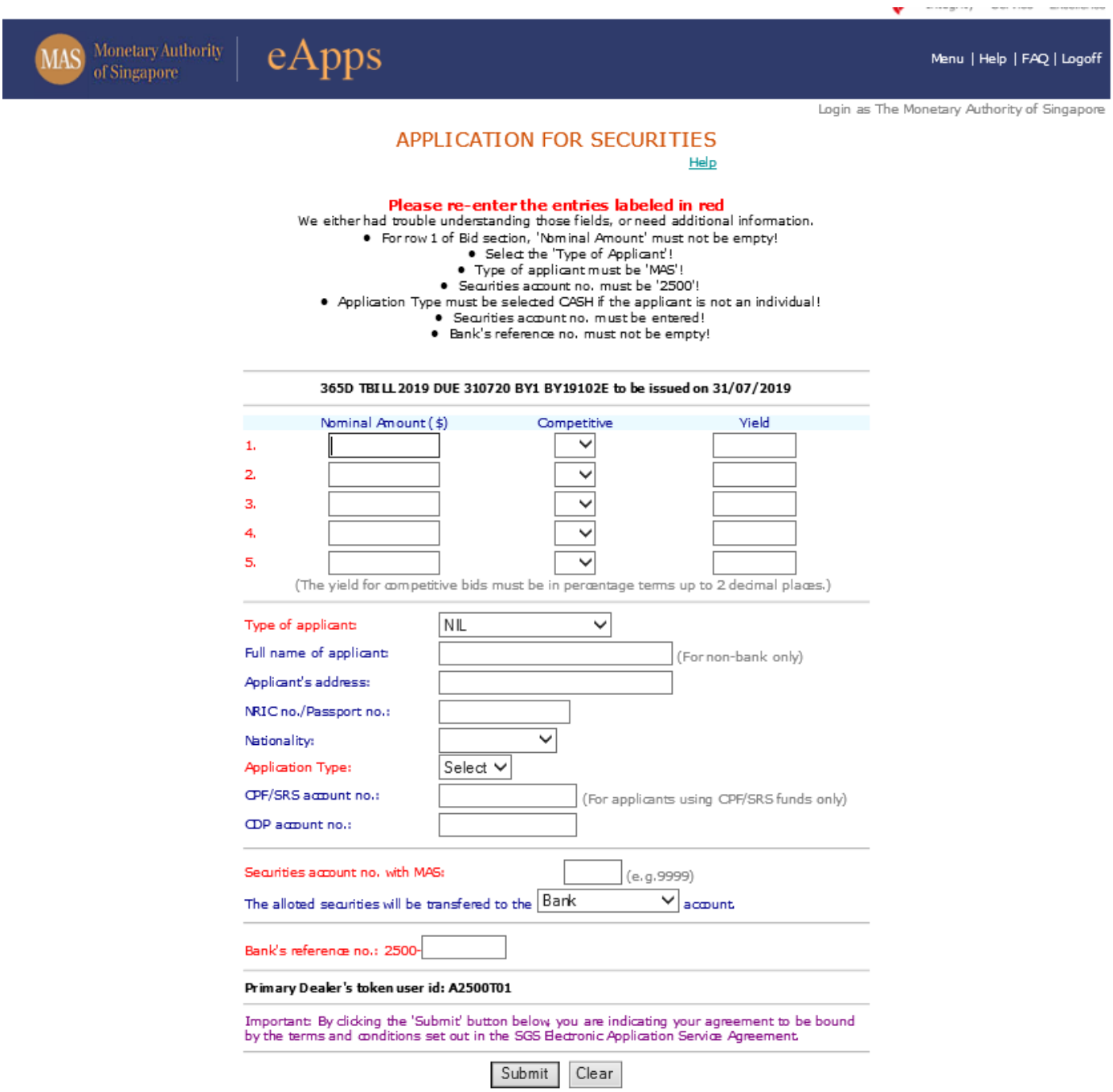
4.1.7 Help on filling up form



Input of SRS Acct No. – No validation required.

4.1.8 Error page for SGS application form





4.1.9 SGS application confirmation page

Integrity · Service · Excellence

Monetary Authority of Singapore | eApps

Menu | Help | FAQ | Logoff

Login as DBS BANK LTD

Please Confirm Your Application.

- You submitted an application for security code: BY19102E
- Your user id with eApps is: A7171T01.
- Date/Time: 10-07-2019 13:03:03.
- Please click on the button below to confirm your application.

365D TBILL 2019 DUE 310720 BY1 BY19102E to be issued on 31/07/2019

	Nominal amount ($)	Competitive	Yield
1.	10,000	C	1.96
2.	20,000	C	1.50
3.	120,000	N	
4.	180,000	N	
5.	100,000	C	1.45

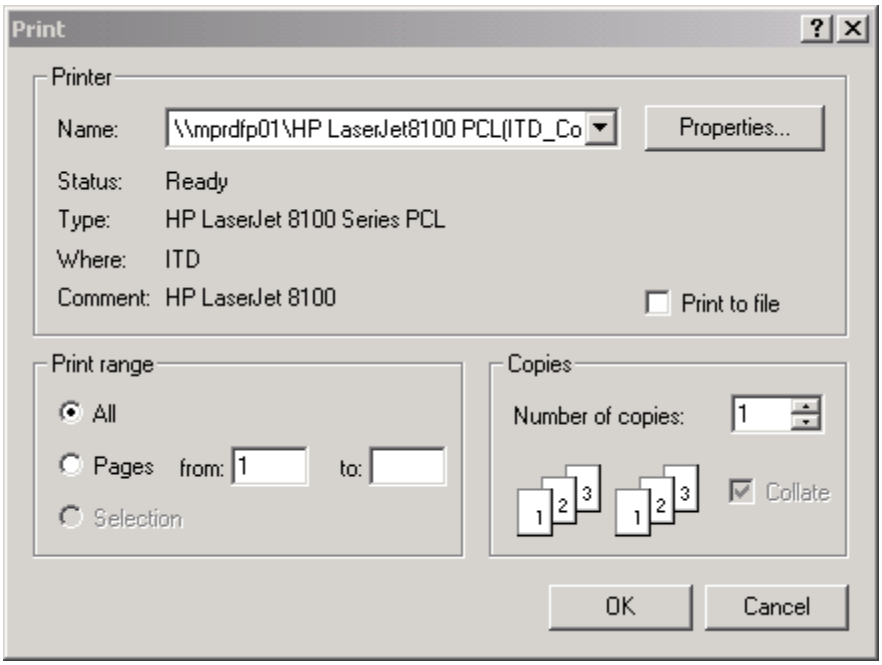
Type of applicant: Primary Dealer
Application Type: CASH

Securities account no. with MAS: 7171
Bank name: DBS BANK LTD
The alloted securities will be transfered to the bank account with MAS.

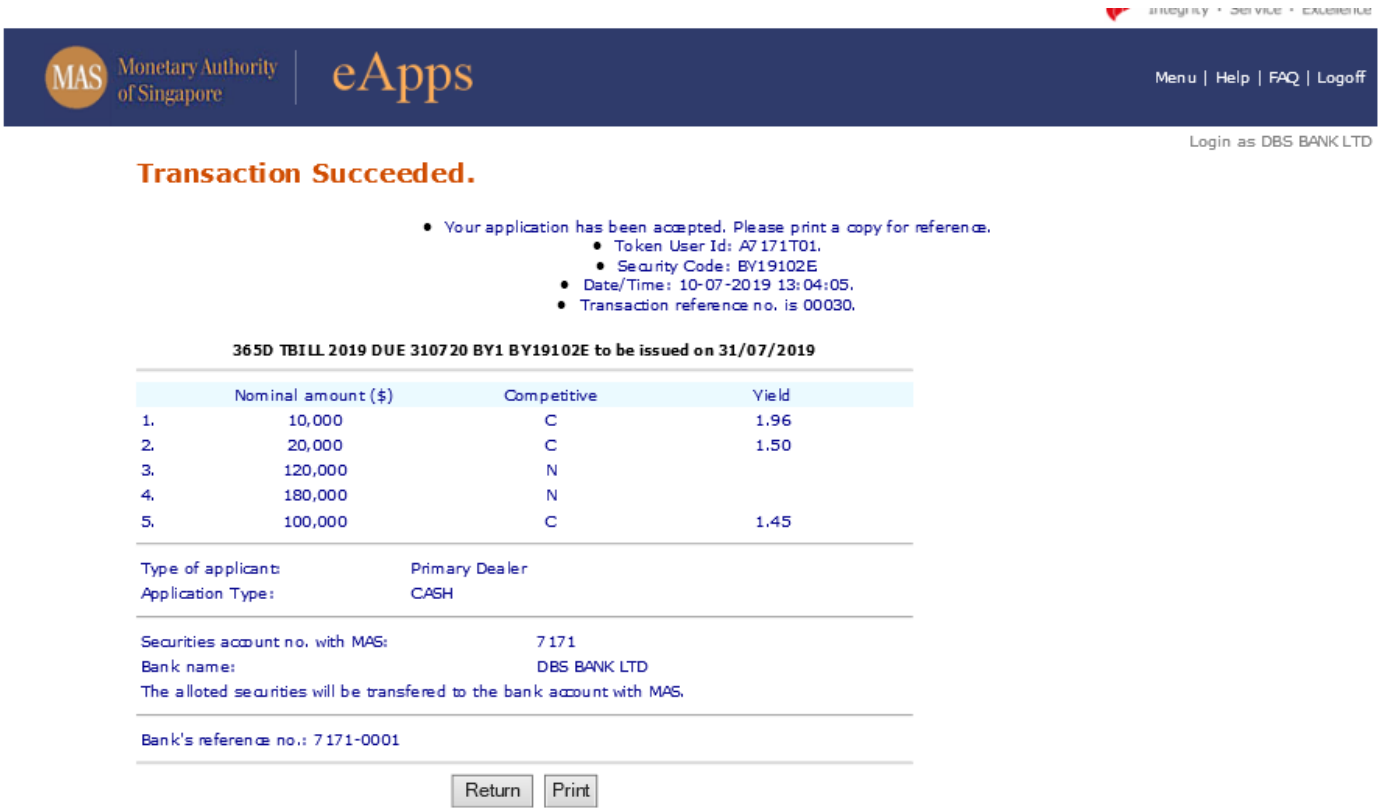
Bank's reference no.: 7171-0001

Confirm | Cancel

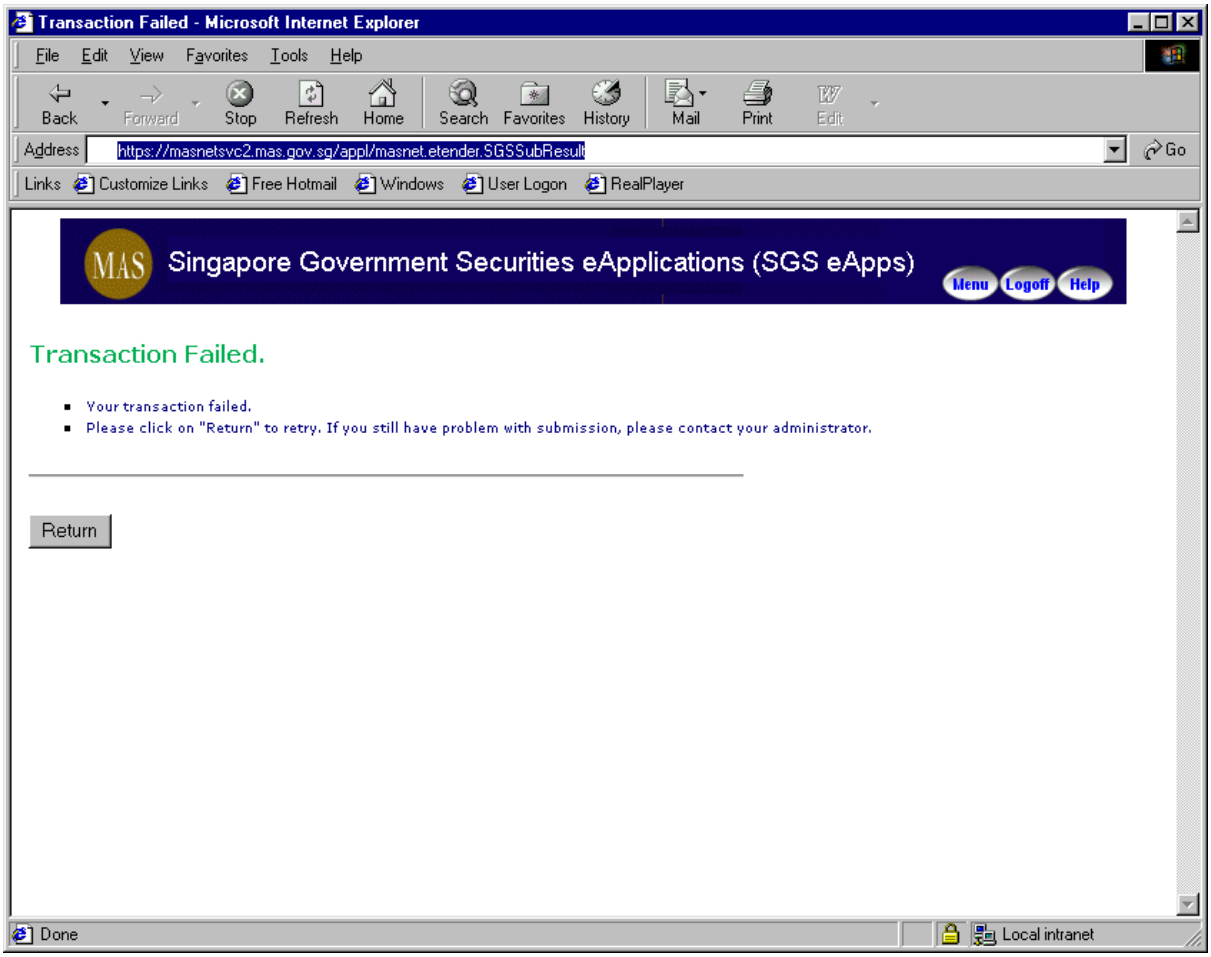
4.1.10 Pop-up printer window on transaction succeeded page



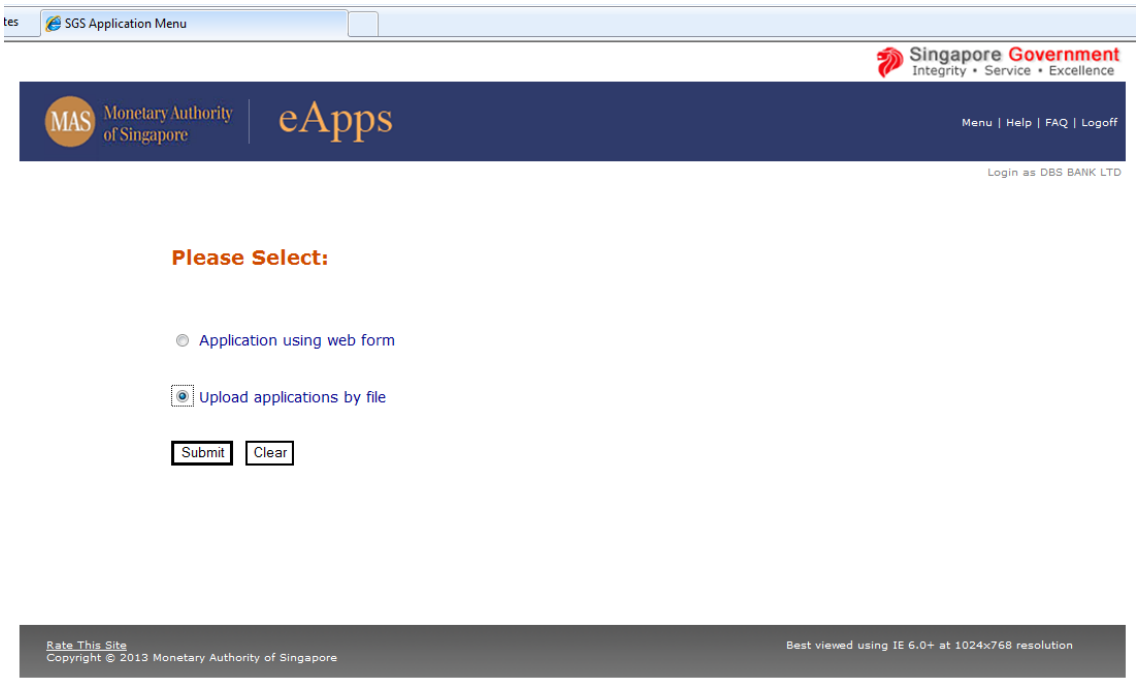
4.1.11 SGS application succeeded page



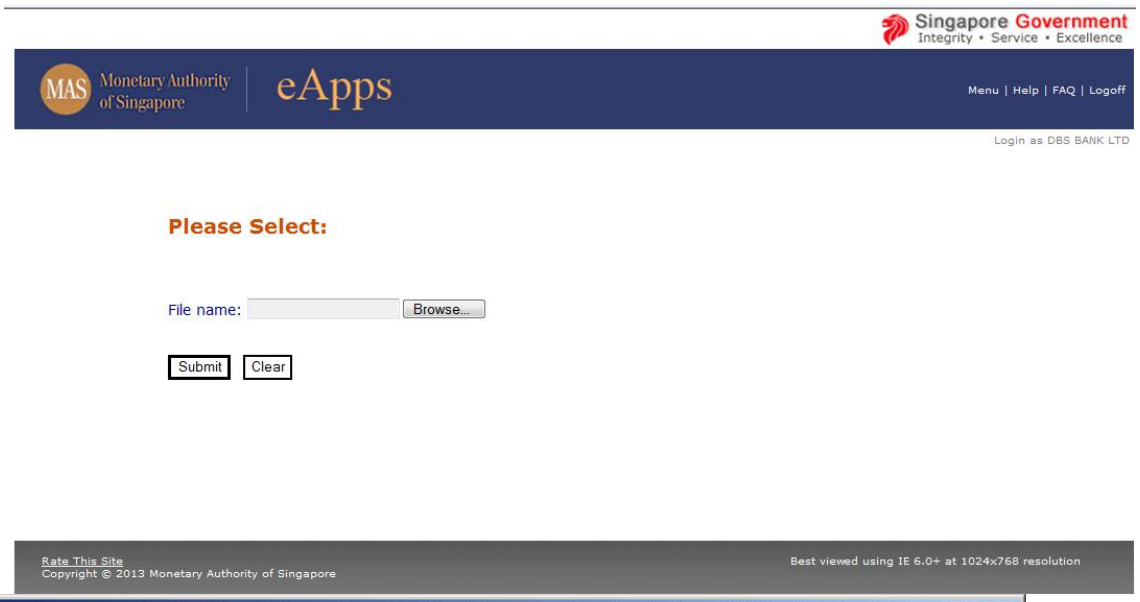
4.1.12 SGS application failed page



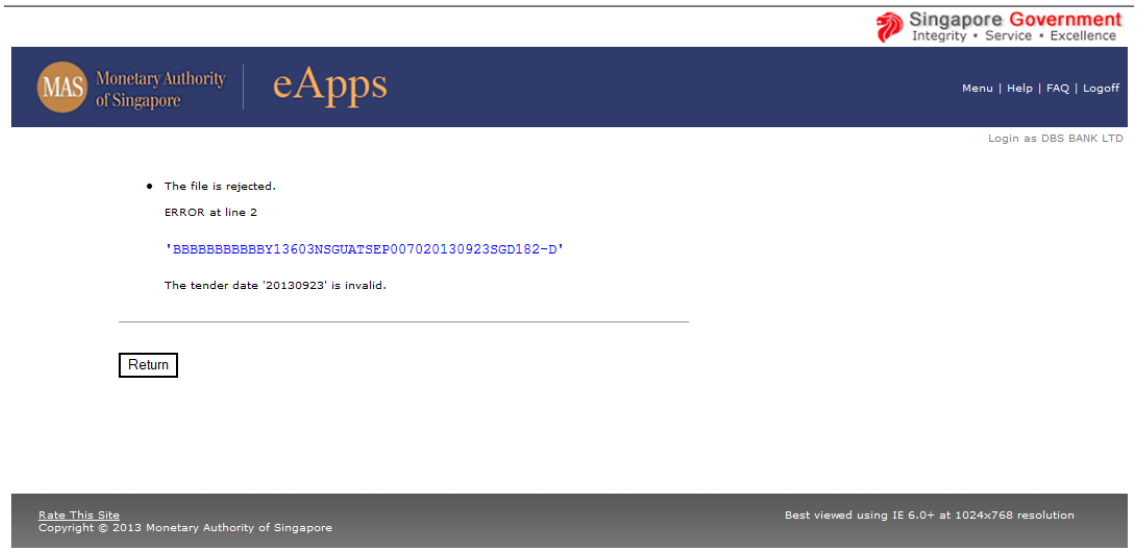
4.1.13 Bid file upload menu



4.1.14 Browse upload file



4.1.15 Confirm submission – file rejected



4.1.16 Confirm submission – all records rejected

APPLICATION FOR BOOK-ENTRY GOVERNMENT...

Singapore Government
Integrity • Service • Excellence

Monetary Authority of Singapore | eApps

Menu | Help | FAQ | Logoff

Login as DBS BANK LTD

APPLICATION FOR SECURITIES

Total number of rejected bids: 3

Line No.	Line content and reject reason
3	DDDDDDDDDDBS13530UDBS C0012013053000000TE_00000100000C+00111DBS Cust 1 SS0001166E 7171CIND 123456789012A00000020237812 BS13530U is closed for application.
4	DDDDDDDDDDBS13530UDBS C0022013053000000TE_00000200000N+00000DBS Cust 2 SS0206673D 7171CIND 123456789012A00000020337701 BS13530U is closed for application.
5	DDDDDDDDDDBS13530UDBS C0032013053000000TE_00000300000C+00130DBS Cust 3 SS1213056B 7171CIND 123456789012A00000020437831 BS13530U is closed for application.

Total number of accepted bids: 0, Total accepted amount: 0.

Line No.	Security Code	Bank's Ref No.	Applicant's Name	IC/Passport No.	Applied Amt	Comp/Non-Comp	Bid Yield	Submit Method
No accepted record.								

Important: By clicking the 'Submit' button below, you are indicating your agreement to be bound by the terms and conditions set out in the SGS Electronic Application Service Agreement.

Submit Return

Singapore Government
Integrity • Service • Excellence

Monetary Authority of Singapore | eApps

Menu | Help | FAQ | Logoff

Login as OVERSEA-CHINESE BKG CORP LTD

APPLICATION FOR SECURITIES

Total number of rejected bids: 4

Line No.	Line content and reject reason
3	DDDDDDDDDDBS13202UOCBC C0120131023000000TE_00000100000C+00111OCBC Cust 1 ST0199779Z 7171CIND 123456789012A00000020238305 Invalid Singapore NIRC No. Securities bank account no. must be 7339.
4	DDDDDDDDDDBS13202UOCBC C0220131023000000TE_00000200000N+00000OCBC Cust 2 SS0206673D 7171CIND 123456789012A00000020338194 Securities bank account no. must be 7339.
5	DDDDDDDDDDBS13202UOCBC C0320131023000000TE_00000300000C+00080 FF1234567A 7171CIND 123456789012A00000020438274 The name of applicant should not be empty.
6	DDDDDDDDDDBS13202UOCBC C0420131023000000TE_00000100000C+00130OCBC Cust 4 FF1234267A 7171CIND 123456789012A00000020238324 Securities bank account no. must be 7339.

Total number of accepted bids: 0, Total accepted amount: 0.

Line No.	Security Code	Bank's Ref No.	Applicant's Name	IC/Passport No.	Applied Amt	Comp/Non-Comp	Bid Yield	Submit Method
No accepted record.								

Important: By clicking the 'Submit' button below, you are indicating your agreement to be bound by the terms and conditions set out in the SGS Electronic Application Service Agreement.

Submit Return

Singapore Government Securities eApplications (SGS eApps)

Menu Logoff He

APPLICATION FOR SINGAPORE GOVERNMENT SECURITIES

Total number of rejected bids: 7

Line No.	Line content and reject reason
3	DDDDDDDDDDBQ08135He-07 20081223000000TE_00001000000C00240BEN SS0214378Z 7339C Bank's reference no. 'e-07' already exists.
4	DDDDDDDDDDBQ08135HC-01 20081223000000TE_00035000000C00240MARY SS0001166E 7339C Bank's reference no. 'C-01' already exists.
5	DDDDDDDDDDBQ08135HC-02 20081223000000TE_00000002000N00000PAUL SS0046999H 7339C Bank's reference no. 'C-02' already exists.
7	DDDDDDDDDDBQ08135He-05 20081223000000TE_00000090000C00240ANDY SS0095028I 7339C Bank's reference no. 'e-05' already exists.
9	DDDDDDDDDDBQ08135He-06 20081223000000TE_00001000000C00240BEN SS0214378Z 7339C Bank's reference no. 'e-06' already exists.
8	DDDDDDDDDDBQ08135He-03 20081223000000TE_00000100000C00240JOHN SS0206673D 7339C Bank's reference no. 'e-03' is duplicated in the application file.

Total number of accepted bids: 0

Line No.	Security Code	Bank's Ref No.	Applicant's Name	IC/Passport No.	Applied Amt	Comp/Non-Comp	Bid Yield
No accepted record.							

Important: By clicking the 'Submit' button below, you are indicating your agreement to be bound by the terms and conditions set out in the SGS Ele

Submit Return

4.1.17 Confirm submission – partial records rejected

Singapore Government Securities Auction (SGSAuction)

Main | New Tender | Browse Tenders | Update Tender | View UnRel | Verify Tender | Reactivate | Exit

APPLICATION FOR SINGAPORE GOVERNMENT SECURITIES

- Bank Code: 7339.

Total number of rejected bids: 1

Line No.	Line content and reject reason
5	DDDDDDDDDDBS13202UOCBC C0320131023000000TE_00000300000C+00080 FF1234567A 7339CIND 123456789012A00000020438442 The name of applicant should not be empty, and be a maximum of 30 characters long.

Total number of accepted bids: 3

Line No.	Security Code	Bank's Ref No.	Applicant's Name	IC/Passport No.	Applied Amt	Comp/Non-Comp	Bid Yield
3	BS13202U	7339OCBC C01	OCBC Cust 1	T0199779Z	100000	C	1.11
4	BS13202U	7339OCBC C02	OCBC Cust 2	S0206673D	200000	N	0.0
6	BS13202U	7339OCBC C04	OCBC Cust 4	F1234267A	100000	C	1.3

Important: By clicking the 'Submit' button below, you are indicating your agreement to be bound by the terms and conditions set out in the SGS Electronic Application Service Agreement.

Submit Return

APPLICATION FOR SINGAPORE GOVERNMENT SECURITIES

Total number of rejected bids: 2

Line No.	Line content and reject reason
8	DDDDDDDDDDBQ08135He-03 20081223000000TE_00000100000C00240JOHN SS0206673D 7339 Bank's reference no. 'e-03' is duplicated in the application file.
6	DDDDDDDDDDBQ08135He-03 20081223000000TE_00000006000N00000LILY SS0089907J 7339 Bank's reference no. 'e-03' is duplicated in the application file.

Total number of accepted bids: 5

Line No.	Security Code	Bank's Ref No.	Applicant's Name	IC/Passport No.	Applied Amt	Comp/Non-Comp	Bid Yield
3	BQ08135H	e-07	BEN	S0214378Z	1000000	C	2.4
4	BQ08135H	C-01	MARY	S0001166E	35000000	C	2.4
5	BQ08135H	C-02	PAUL	S0046999H	2000	N	0.0
7	BQ08135H	e-05	ANDY	S0095028I	90000	C	2.4
9	BQ08135H	e-06	BEN	S0214378Z	1000000	C	2.4

Important: By clicking the 'Submit' button below, you are indicating your agreement to be bound by the terms and conditions set out in the SGS Electronic Applicat

Submit Return

4.1.18 Confirm submission – all records accepted

Singapore Government
Integrity • Service • Excellence

Monetary Authority of Singapore | eApps

Menu | Help | FAQ | Logoff

Login as OVERSEA-CHINESE BKG CORP LTD

APPLICATION FOR SECURITIES

Total number of rejected bids: 0

Line No.	Line content and reject reason
No rejected record.	

Total number of accepted bids: 4, Total accepted amount: 700,000.

Line No.	Security Code	Bank's Ref No.	Applicant's Name	IC/Passport No.	Applied Amt	Comp/Non-Comp	Bid Yield	Submit Method
3	BS13202U	OCBC C01	OCBC Cust 1	T0199779Z	100000	C	1.11	A
4	BS13202U	OCBC C02	OCBC Cust 2	S0206673D	200000	N	0.0	A
5	BS13202U	OCBC C03	OCBC Cust 3	F1234567A	300000	C	0.8	A
6	BS13202U	OCBC C04	OCBC Cust 4	F1234267A	100000	C	1.3	A

Important: By clicking the 'Submit' button below, you are indicating your agreement to be bound by the terms and conditions set out in the SGS Electronic Application Service Agreement.

Submit Return

4.1.19 Submission result – partial records rejected (31-10-2013 Does not display)

4.1.20 Submission result – all records accepted

Singapore Government
Integrity • Service • Excellence

Monetary Authority of Singapore | eApps

Menu | Help | FAQ | Logoff

Login as OVERSEA-CHINESE BKG CORP LTD

APPLICATION FOR SECURITIES

Transaction Completed.

- Token User Id: A7339T01.
- Date/Time: 23-10-2013 17:24:08.

Total number of accepted bids: 4, total accepted amount: 700,000.

Line No.	Security Code	Bank's Ref No.	Transaction Ref No.	Applicant's Name	IC/Passport No.	Applied Amt	Comp/Non-Comp	Bid Yield	Submit Method
3	BS13202U	OCBC C01	2	OCBC Cust 1	T0199779Z	100000	C	1.11	A
4	BS13202U	OCBC C02	3	OCBC Cust 2	S0206673D	200000	N	0.0	A
5	BS13202U	OCBC C03	4	OCBC Cust 3	F1234567A	300000	C	0.8	A
6	BS13202U	OCBC C04	5	OCBC Cust 4	F1234267A	100000	C	1.3	A

Return Print

Rate This Site
Copyright © 2013 Monetary Authority of Singapore

Best viewed using IE 6.0+ at 1024x768 resolution

Singapore Government Securities eApplications (SGS eApps)

Menu Logoff Help FAQ

APPLICATION FOR SINGAPORE GOVERNMENT SECURITIES

Transaction Completed.

- Token User Id: A7339T05.
- Date/Time: 29-01-2009 20:16:20.

Total number of accepted bids: 5

Line No.	Security Code	Bank's Ref No.	Transaction Ref No.	Applicant's Name	IC/Passport No.	Applied Amt	Comp/Non-Comp	Bid Yield
3	BQ08135H	e-07	16	BEN	S0214378Z	1000000	C	2.4
4	BQ08135H	C-01	17	MARY	S0001166E	35000000	C	2.4
5	BQ08135H	C-02	18	PAUL	S0046999H	2000	N	0.0
7	BQ08135H	e-05	19	ANDY	S0095028I	90000	C	2.4
9	BQ08135H	e-06	20	BEN	S0214378Z	1000000	C	2.4

Return Print

4.1.21 SGS inquiry list page

For local PD users:

MAS Bills				
Security Code	Security Description	Issue Date	Closing Date	Issue Size
MA23075A	4W MBILL A TEST MA2	08/07/2023	06/07/2023	1,400,000,000
MD23117S	4W MBILL TEST 202307 MD2	07/07/2023	05/07/2023	1,400,000,000

Floating Rate Notes				
Security Code	Security Description	Issue Date	Closing Date	Issue Size
M123075A	2YR FRN 2023 A TEST M12	08/07/2023	06/07/2023	1,400,000,000
M162316E	2YR FRN 2020 DUE 150637 M16	07/07/2023	05/07/2023	1,400,000,000

T Bills				
Security Code	Security Description	Issue Date	Closing Date	Issue Size
B230705A	6M TBILL A TEST B23	08/07/2023	06/07/2023	1,400,000,000
BS23101S	6M TBILL TEST 202307 BS2	07/07/2023	05/07/2023	1,400,000,000

For MDD users:

T Bills				
Security Code	Security Description	Issue Date	Closing Date	Issue Size
BS00001X	20YR TBILL TEST DUE 20436 NZ16	22/02/2023	21/02/2023	1,000,000,000
BS72222D	20YR TEST20230221	21/02/2023	22/02/2023	1,300,000,000
BS73333D	20YR BOND TEST20230221 NZ16	21/02/2023	22/02/2023	1,400,000,000

SGS Bonds				
Security Code	Security Description	Issue Date	Closing Date	Issue Size
N200001V	20YR BOND 2020 DUE 150637 NZ16	01/03/2023	23/02/2023	1,000,000,000
NZ72222D	20YR BOND TEST20230221 NZ21	21/02/2023	22/02/2023	1,400,000,000

For non-local PD users:

4.1.22 SGS inquiry display page

For local PD users:

Monetary Authority of Singapore | eApps

Menu | Help | FAQ | Logoff

Login as DBS BANK LTD (C7171T02)

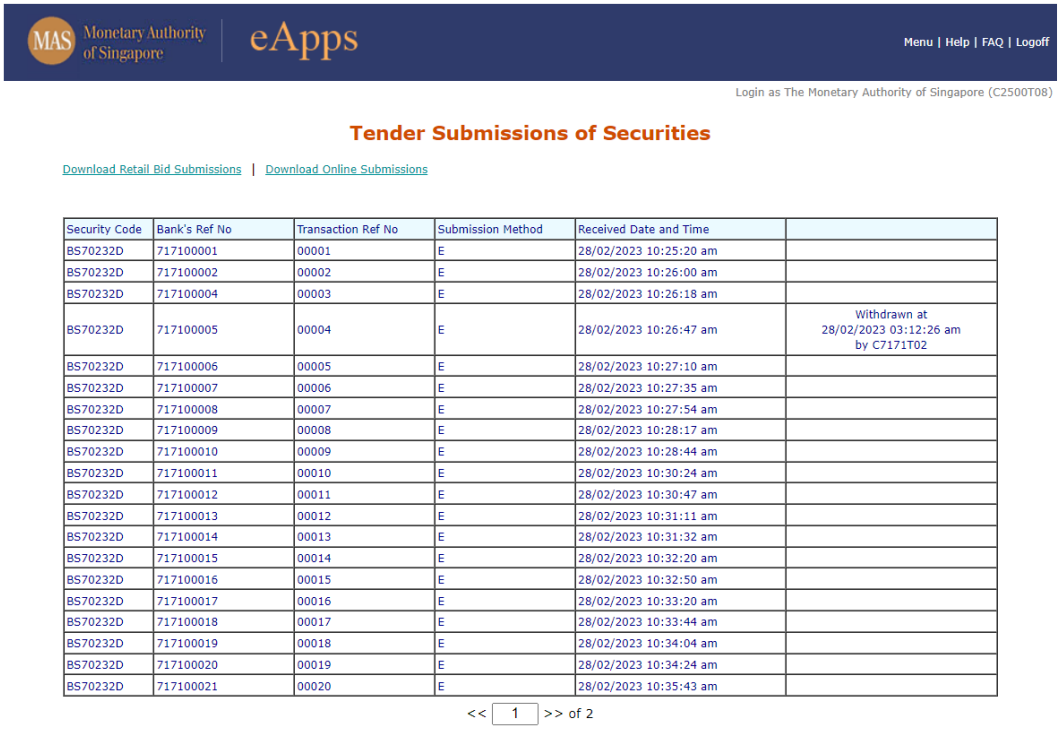
Tender Submissions of Securities

Download Retail Bid Submissions | Download Online Submissions

Security Code	Bank's Ref No	Transaction Ref No	Submission Method	Received Date and Time	
BS70232D	717100001	00001	E	28/02/2023 10:25:20 am	Withdraw
BS70232D	717100002	00002	E	28/02/2023 10:26:00 am	Withdraw
BS70232D	717100004	00003	E	28/02/2023 10:26:18 am	Withdraw
BS70232D	717100005	00004	E	28/02/2023 10:26:47 am	Withdrawn at 28/02/2023 03:12:26 am by C7171T02
BS70232D	717100006	00005	E	28/02/2023 10:27:10 am	Withdraw
BS70232D	717100007	00006	E	28/02/2023 10:27:35 am	Withdraw
BS70232D	717100008	00007	E	28/02/2023 10:27:54 am	Withdraw
BS70232D	717100009	00008	E	28/02/2023 10:28:17 am	Withdraw
BS70232D	717100010	00009	E	28/02/2023 10:28:44 am	Withdraw
BS70232D	717100011	00010	E	28/02/2023 10:30:24 am	Withdraw
BS70232D	717100012	00011	E	28/02/2023 10:30:47 am	Withdraw
BS70232D	717100013	00012	E	28/02/2023 10:31:11 am	Withdraw
BS70232D	717100014	00013	E	28/02/2023 10:31:32 am	Withdraw
BS70232D	717100015	00014	E	28/02/2023 10:32:20 am	Withdraw
BS70232D	717100016	00015	E	28/02/2023 10:32:50 am	Withdraw
BS70232D	717100017	00016	E	28/02/2023 10:33:20 am	Withdraw
BS70232D	717100018	00017	E	28/02/2023 10:33:44 am	Withdraw
BS70232D	717100019	00018	E	28/02/2023 10:34:04 am	Withdraw
BS70232D	717100020	00019	E	28/02/2023 10:34:24 am	Withdraw
BS70232D	717100021	00020	E	28/02/2023 10:35:43 am	Withdraw

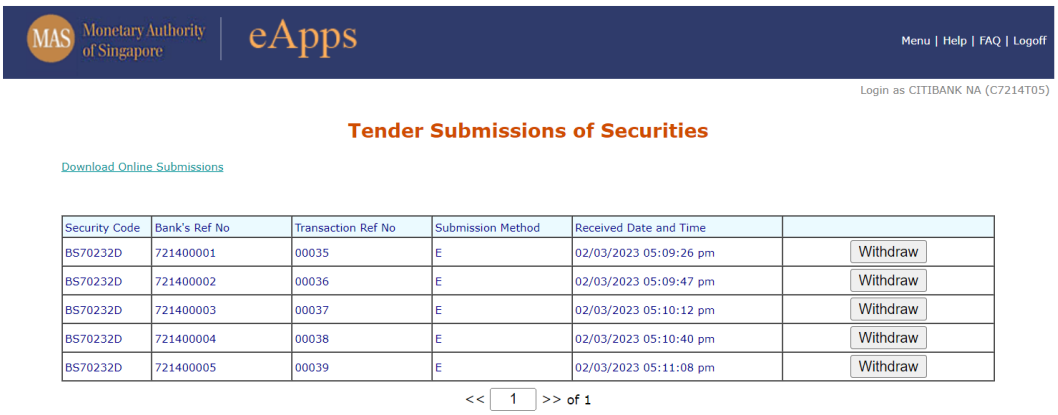
<< 1 >> of 2

For MAS users:



Security Code	Bank's Ref No	Transaction Ref No	Submission Method	Received Date and Time	
BS70232D	717100001	00001	E	28/02/2023 10:25:20 am	
BS70232D	717100002	00002	E	28/02/2023 10:26:00 am	
BS70232D	717100004	00003	E	28/02/2023 10:26:18 am	
BS70232D	717100005	00004	E	28/02/2023 10:26:47 am	Withdrawn at 28/02/2023 03:12:26 am by C7171T02
BS70232D	717100006	00005	E	28/02/2023 10:27:10 am	
BS70232D	717100007	00006	E	28/02/2023 10:27:35 am	
BS70232D	717100008	00007	E	28/02/2023 10:27:54 am	
BS70232D	717100009	00008	E	28/02/2023 10:28:17 am	
BS70232D	717100010	00009	E	28/02/2023 10:28:44 am	
BS70232D	717100011	00010	E	28/02/2023 10:30:24 am	
BS70232D	717100012	00011	E	28/02/2023 10:30:47 am	
BS70232D	717100013	00012	E	28/02/2023 10:31:11 am	
BS70232D	717100014	00013	E	28/02/2023 10:31:32 am	
BS70232D	717100015	00014	E	28/02/2023 10:32:20 am	
BS70232D	717100016	00015	E	28/02/2023 10:32:50 am	
BS70232D	717100017	00016	E	28/02/2023 10:33:20 am	
BS70232D	717100018	00017	E	28/02/2023 10:33:44 am	
BS70232D	717100019	00018	E	28/02/2023 10:34:04 am	
BS70232D	717100020	00019	E	28/02/2023 10:34:24 am	
BS70232D	717100021	00020	E	28/02/2023 10:35:43 am	

For non-local PD users:



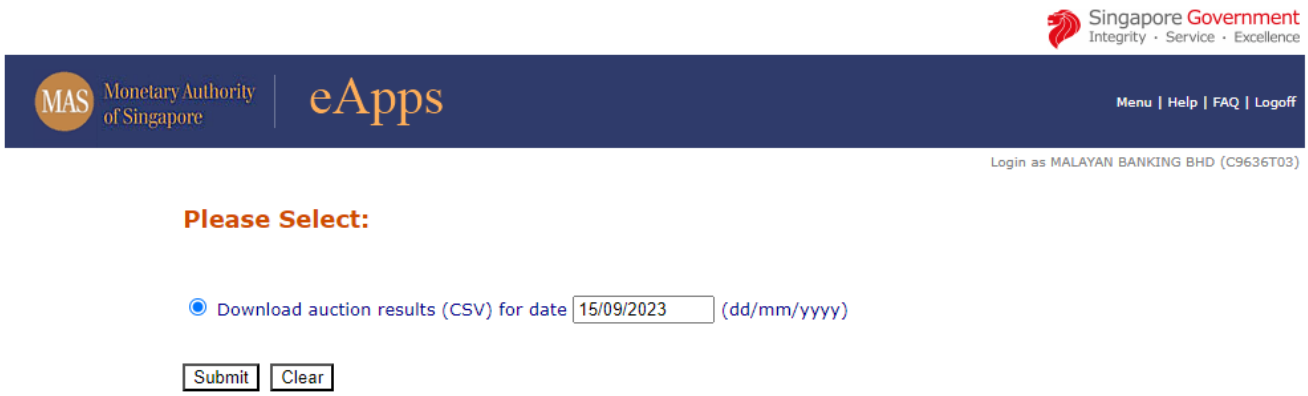
Security Code	Bank's Ref No	Transaction Ref No	Submission Method	Received Date and Time	
BS70232D	721400001	00035	E	02/03/2023 05:09:26 pm	Withdraw
BS70232D	721400002	00036	E	02/03/2023 05:09:47 pm	Withdraw
BS70232D	721400003	00037	E	02/03/2023 05:10:12 pm	Withdraw
BS70232D	721400004	00038	E	02/03/2023 05:10:40 pm	Withdraw
BS70232D	721400005	00039	E	02/03/2023 05:11:08 pm	Withdraw

4.1.23 SGS auction result

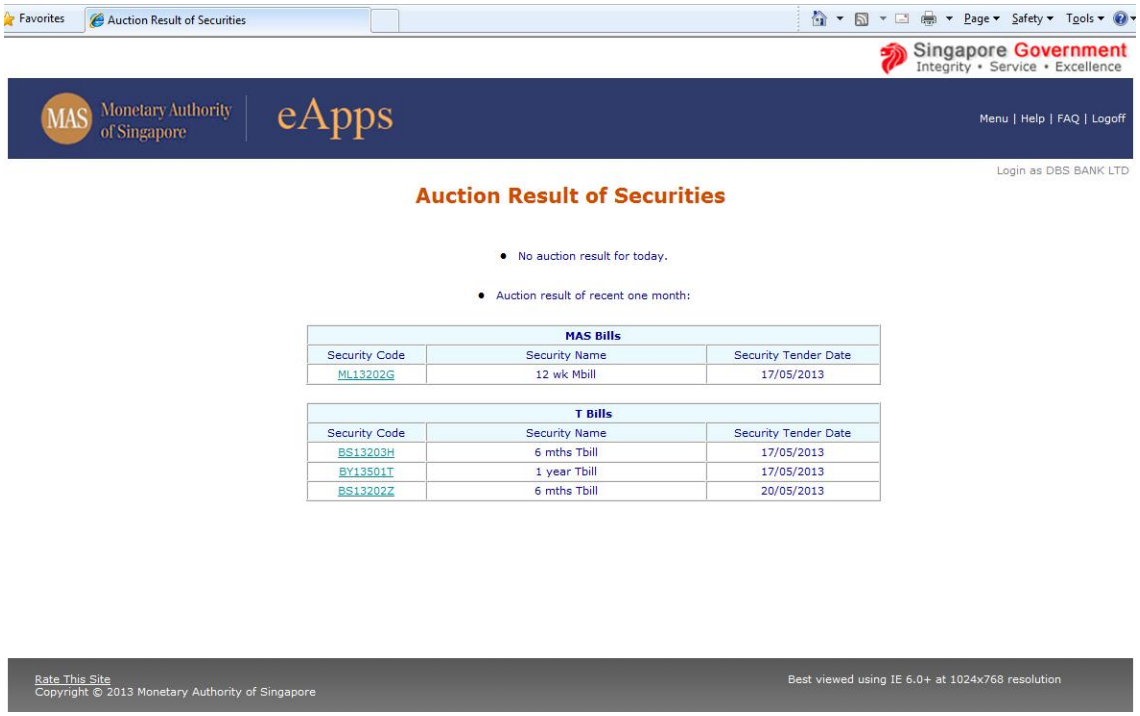
For MAS users:

For local PD users:

For non-local PD users:



4.1.24 SGS auction result list page



4.1.25 SGS auction result display page

For PD users:

For MAS users:

Result of Auctions for Securities

29 May 14

Latest

Publish To publish result of auctions conducted on 29/05/2014.

- **MAS Bills**

Issue Code	Description	Issue Date	Auction Date
MD14800T	Mbill 2014Q1 UAT	10/02/2014	06/02/2014

- **Treasury Bill (6-Mth)**

Issue Code	Description	Issue Date	Auction Date
BS14100H	6 Mth test Tbill	03/02/2014	30/01/2014

- **Treasury Bill (1-Yr)/SGS Bond**

Issue Code	Description	Issue Date	Auction Date
BY14101H	1 Yr test Tbill	01/04/2014	21/03/2014
N514800V	5-yr Bond	05/06/2014	29/05/2014

Auction Result - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: https://masnetsvc2.mas.gov.sg/appl/sgseapps.etender.SGSAuctionDisplay?BY01100H226/12/2001 What's Related

Singapore Government Securities eApplications (SGS eApps)

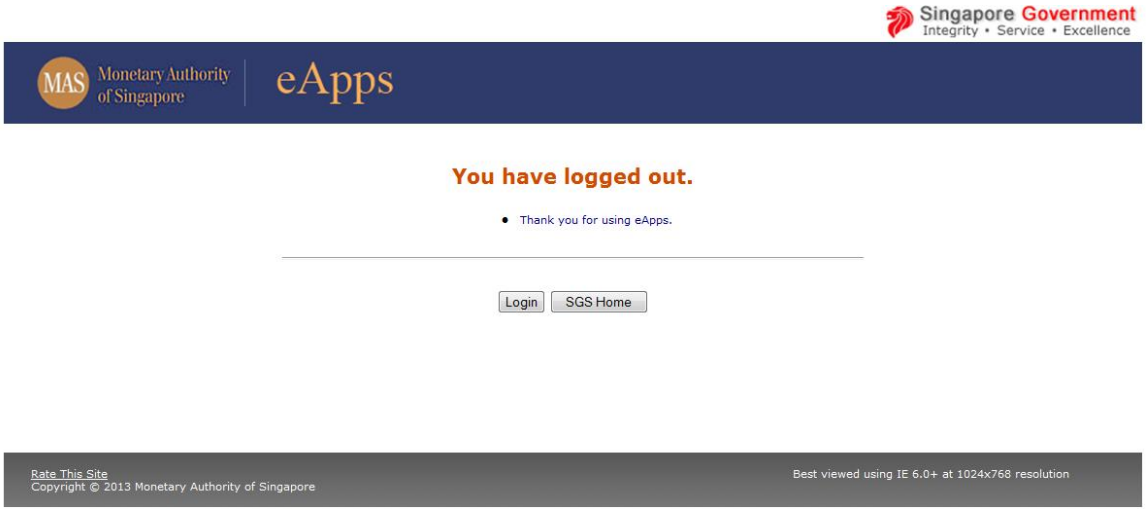
Menu Logoff Help FAQ

```
The Monetary Authority of Singapore                ACCOUNT NO:  2500
ISSUE: BY01100H 364-DAY TREA BILL DUE 14052002
     WE ARE PLEASED TO INFORM YOU THAT YOUR TENDER(S)/APPLICATION(S)
FOR GOVERNMENT SECURITIES HAS/HAVE BEEN ACCEPTED AS FOLLOWS:-
                              PRICE      NOMINAL
                            PER SGD100/   AMOUNT          AMOUNT
APPLICANT                     YIELD      ALLOTTED        PAYABLE
---------                   --------  ------------  --------------
BANKS/CORPORATIONS
------------------
The Monetary Authority of Sing  98.365 /   40,000,000   39,346,000.00
                                 1.95
The Monetary Authority of Sing  98.323 /   40,000,000   39,329,200.00
                                 2.00
The Monetary Authority of Sing  98.348 /   20,000,000   19,669,600.00
                                 1.97
                                          ------------- ---------------
                            SUB-TOTAL    100,000,000    98,344,800.00
                                TOTAL    100,000,000    98,344,800.00
                                         ============= ===============
DOMESTIC OPERATIONS DIVISION
FINANCE DEPARTMENT
THE MONETARY AUTHORITY OF SINGAPORE
```

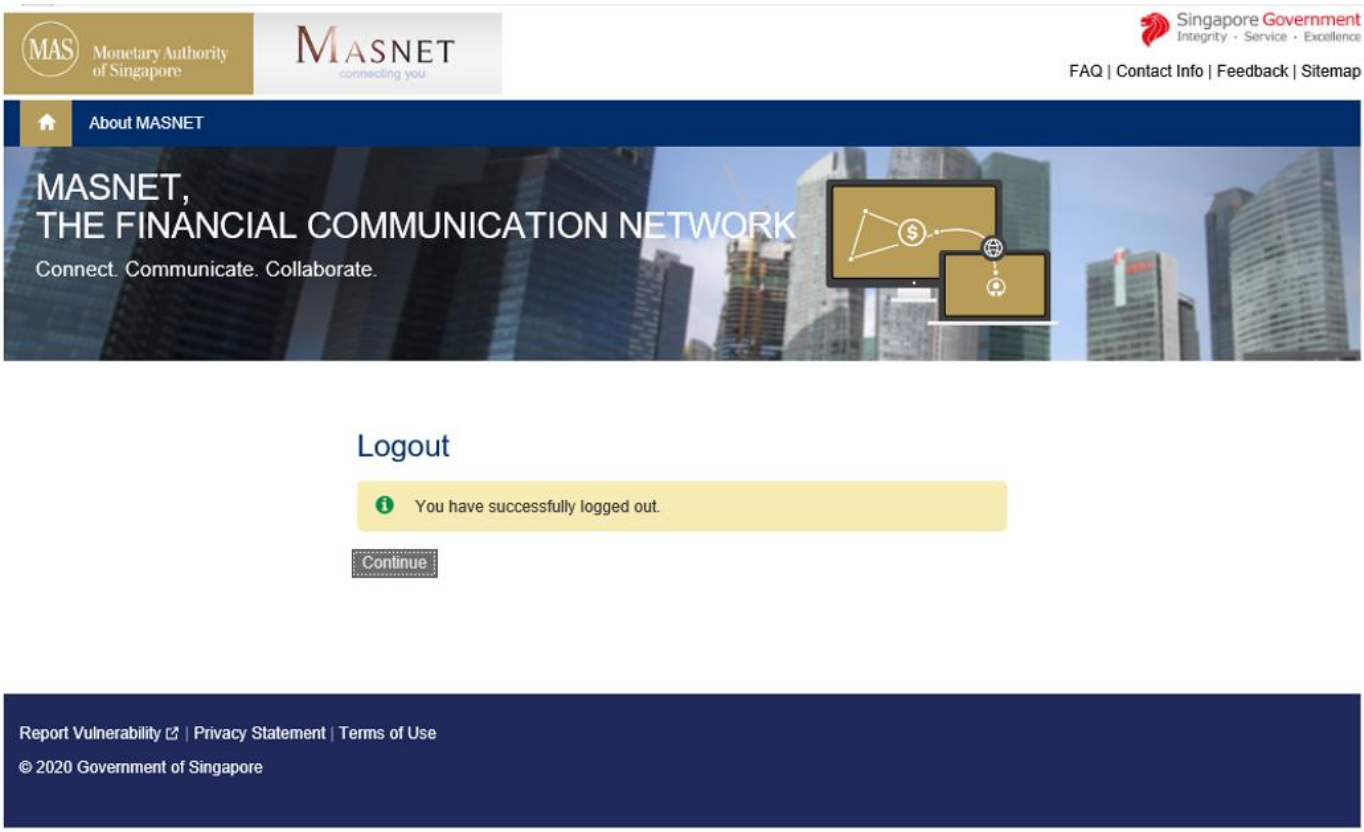
Document: Done

4.1.26 Logoff page

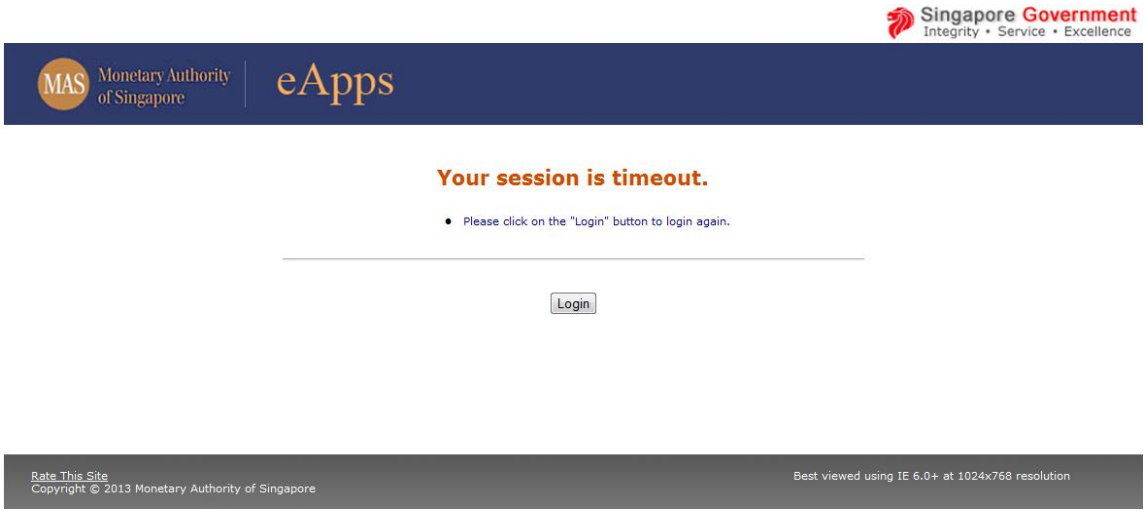
Non-CorpPass/CorpPass TOPPAN Integration Login – Application logout page



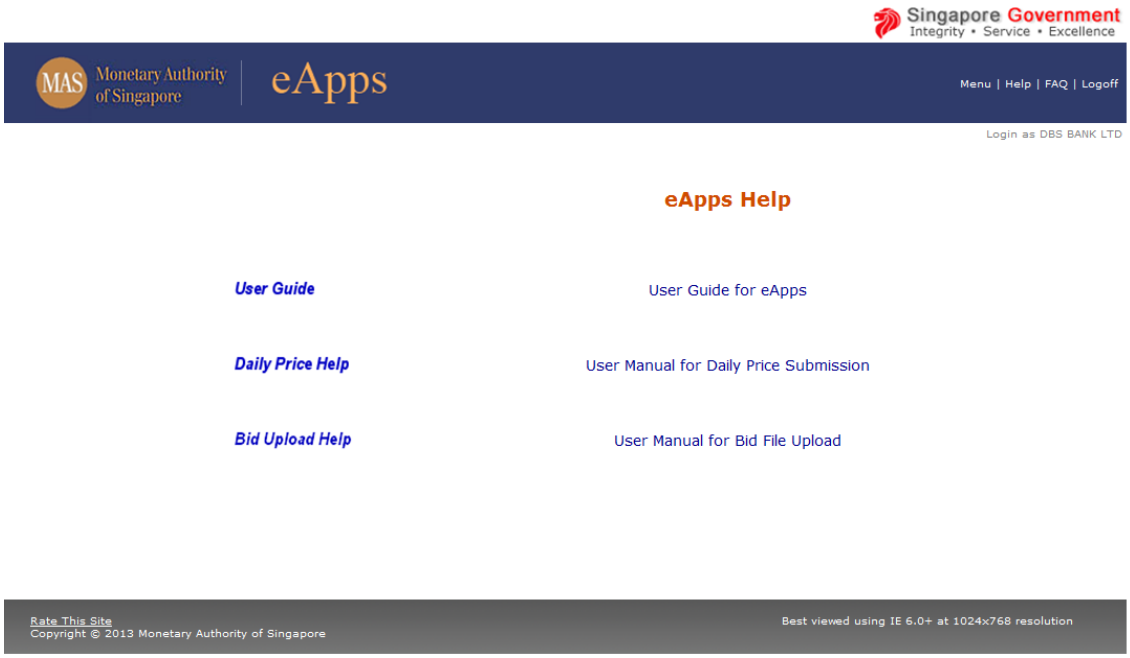
CorpPass Login – MAF Global logout page



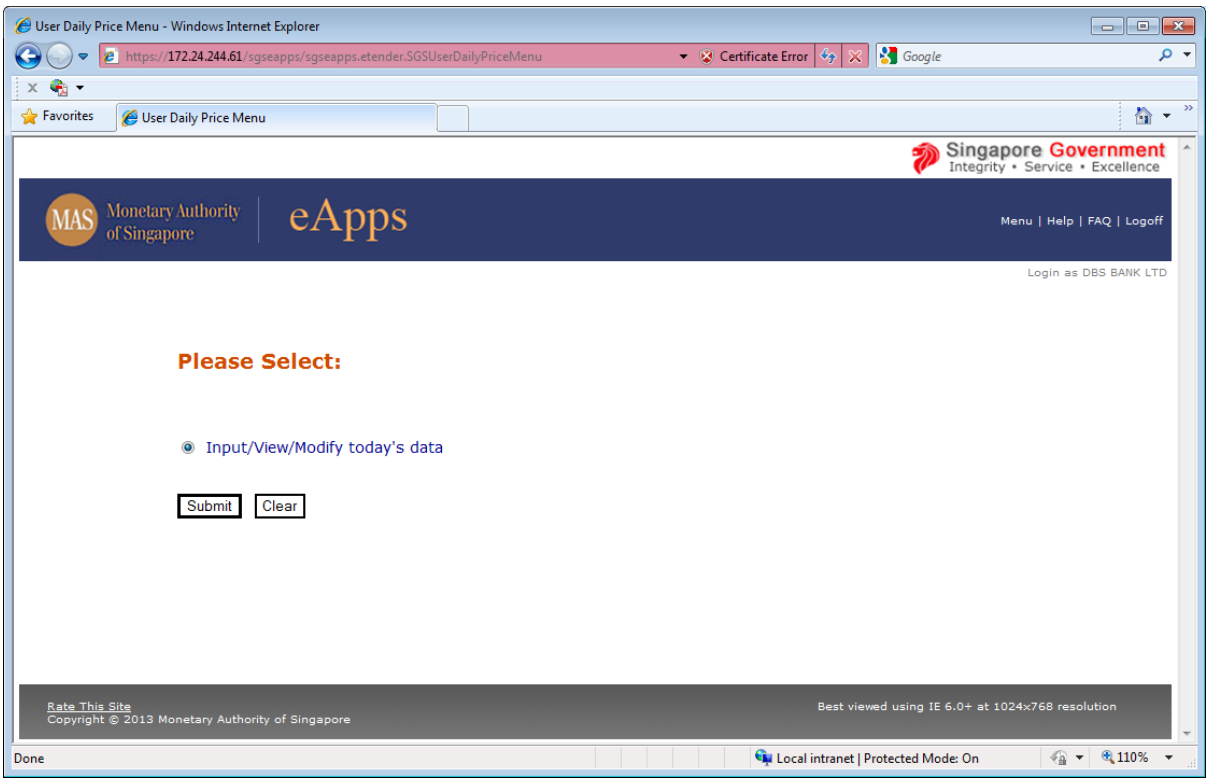
4.1.27 Timeout page



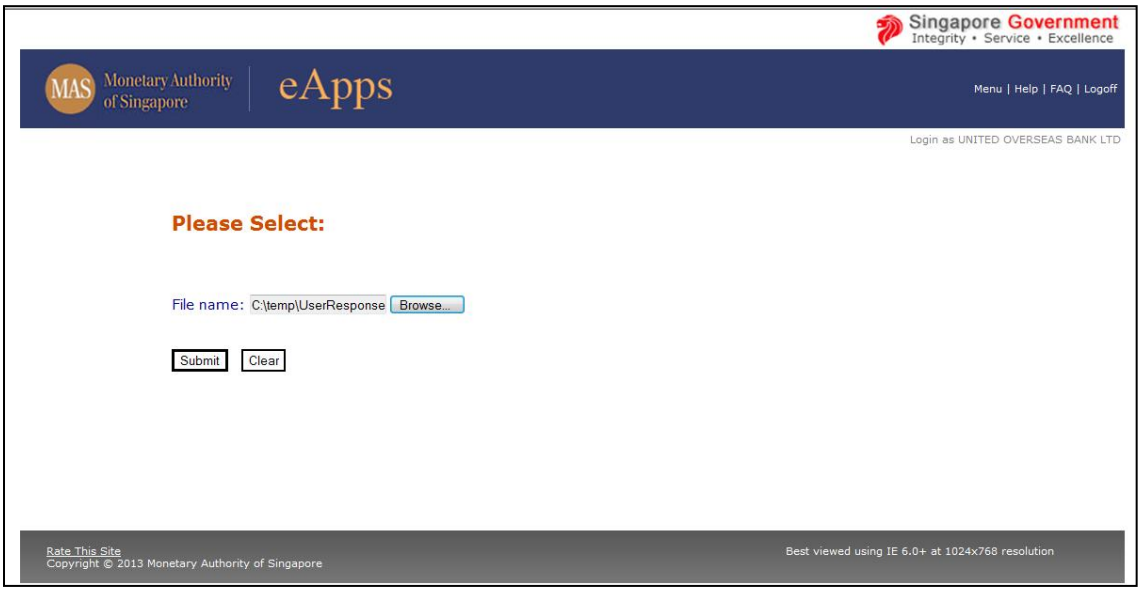
4.1.28 Help page



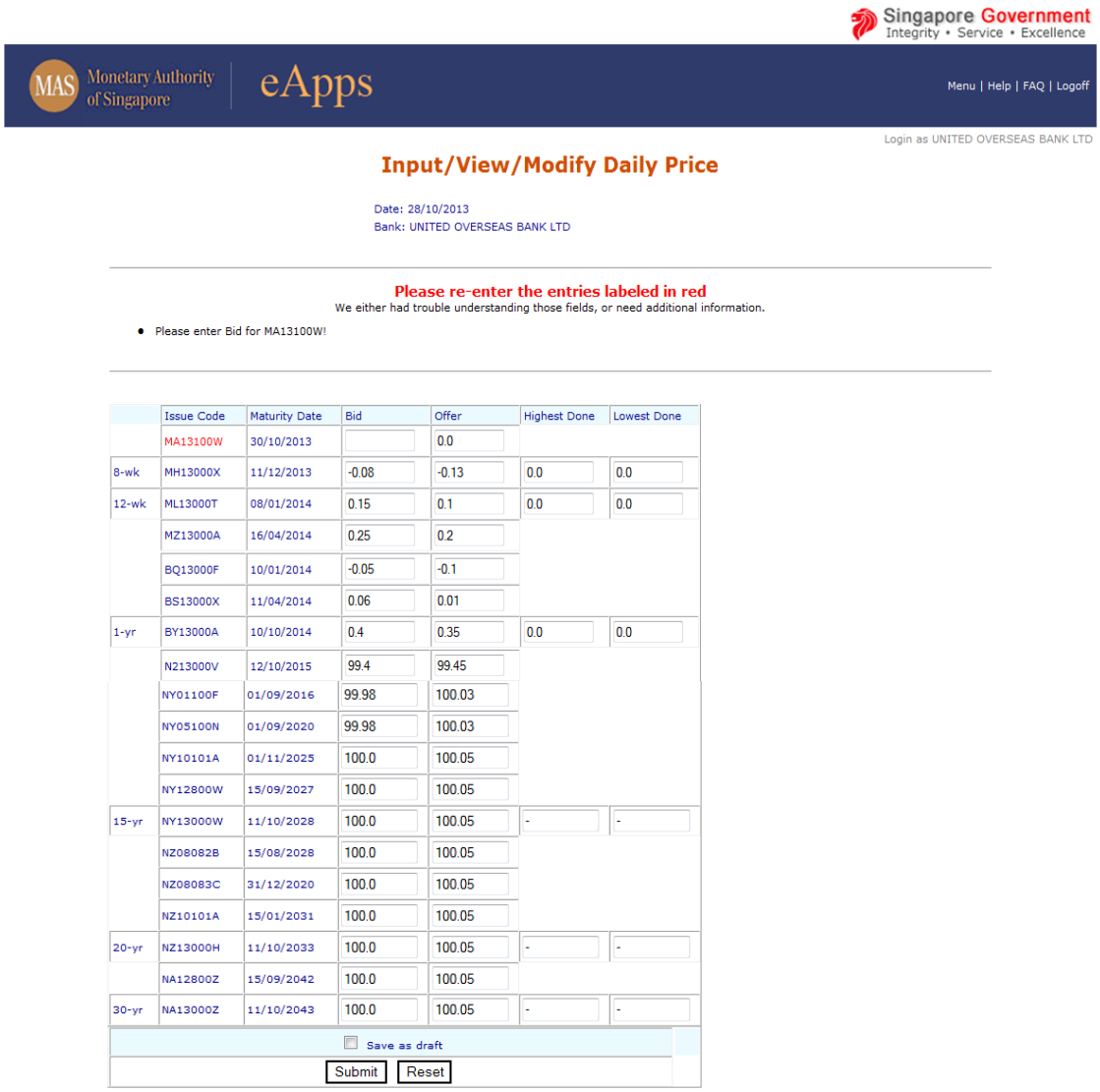
4.1.29 Daily closing price menu for banks



4.1.30 Upload daily closing price file

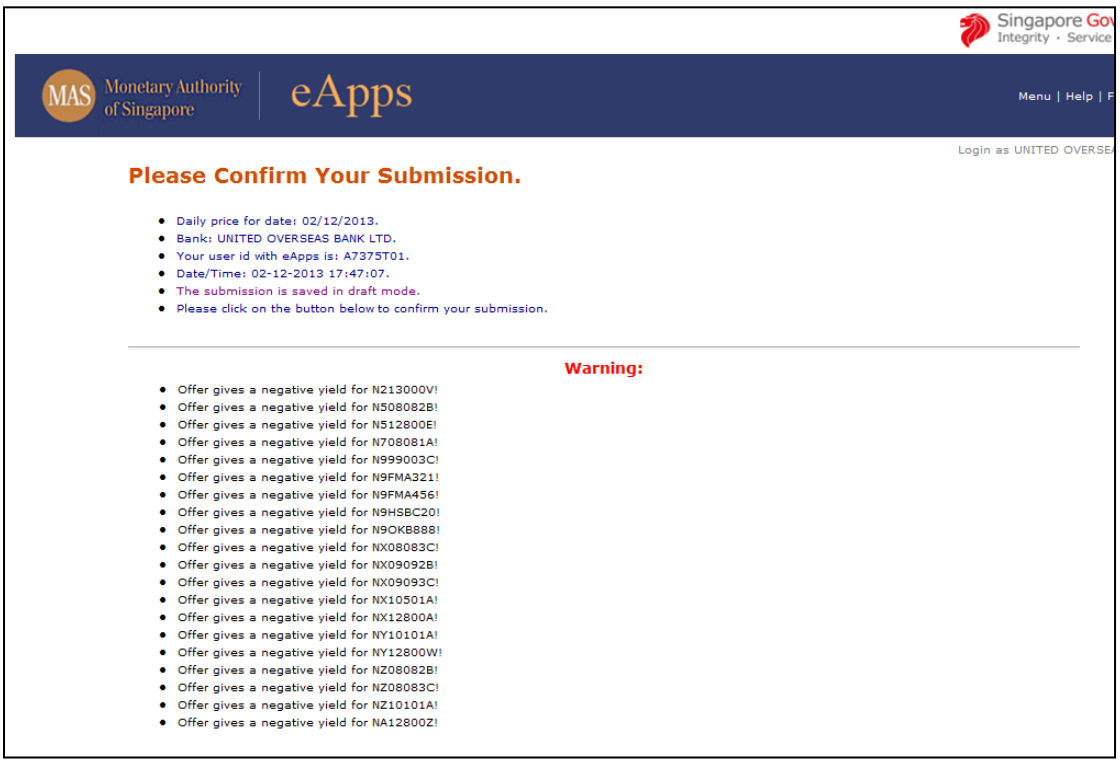


4.1.31 Re-enter daily closing price



	Issue Code	Maturity Date	Bid	Offer	Highest Done	Lowest Done
	MA13100W	30/10/2013		0.0		
8-wk	MH13000X	11/12/2013	-0.08	-0.13	0.0	0.0
12-wk	ML13000T	08/01/2014	0.15	0.1	0.0	0.0
	MZ13000A	16/04/2014	0.25	0.2		
	BQ13000F	10/01/2014	-0.05	-0.1		
	BS13000X	11/04/2014	0.06	0.01		
1-yr	BY13000A	10/10/2014	0.4	0.35	0.0	0.0
	N213000V	12/10/2015	99.4	99.45		
	NY01100F	01/09/2016	99.98	100.03		
	NY05100N	01/09/2020	99.98	100.03		
	NY10101A	01/11/2025	100.0	100.05		
	NY12800W	15/09/2027	100.0	100.05		
15-yr	NY13000W	11/10/2028	100.0	100.05	-	-
	NZ08082B	15/08/2028	100.0	100.05		
	NZ08083C	31/12/2020	100.0	100.05		
	NZ10101A	15/01/2031	100.0	100.05		
20-yr	NZ13000H	11/10/2033	100.0	100.05	-	-
	NA12800Z	15/09/2042	100.0	100.05		
30-yr	NA13000Z	11/10/2043	100.0	100.05	-	-

4.1.32 Confirm daily closing price



4.1.33 Submission succeeded page for daily closing price file

Singapore Government
Integrity · Service · Excellence

Monetary Authority of Singapore | eApps

Menu | Help | FAQ | Logoff

Login as UNITED OVERSEAS BANK LTD

Transaction Succeeded.

- Your submission has been accepted. Please print a copy for reference.
- Daily price for date: 02/12/2013.
- Bank: UNITED OVERSEAS BANK LTD.
- Token User Id: A7375T01.
- Date/Time: 02-12-2013 17:51:11.

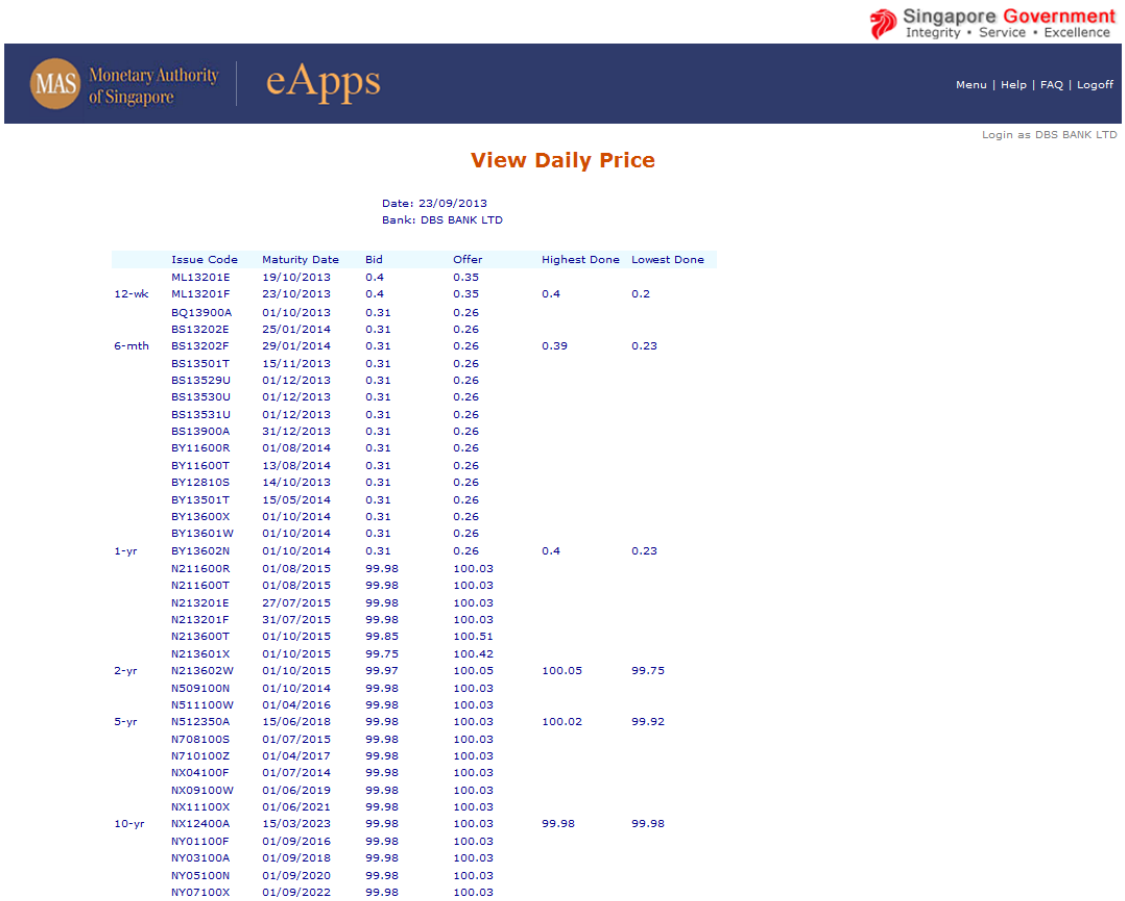
	Issue Code	Maturity Date	Bid	Offer	Highest Done	Lowest Done
8-wk	MH13000X	11/12/2013	0.1	0.05	0.0	0.0
12-wk	ML13000T	08/01/2014	0.15	0.1	0.0	0.0
	MZ13000A	16/04/2014	0.25	0.2		
	MZ13000B	16/04/2014	0.25	0.2		
6-mth	BS13203U	26/05/2014	0.3	0.25	0.35	0.2
2-yr	N212000N	15/11/2015	99.4	99.45	0.0	0.0
	N213000V	12/10/2015	99.4	99.45		
	N508082B	01/02/2014	99.5	99.55		
	N512800E	15/09/2017	99.5	99.55		
	N513000E	11/10/2018	99.98	100.03		
	N708081A	15/11/2015	99.5	99.55		
	N999003C	01/02/2024	99.5	99.55		
	N9FMA321	09/03/2014	99.5	99.55		
	N9FMA456	09/03/2014	99.5	99.55		
	N9HSBC20	20/04/2014	99.5	99.55		
	N9OKB888	23/09/2015	99.5	99.55		
	NX04100F	01/07/2014	101.5	101.55		
5-yr	NX08083C	15/10/2018	99.5	99.55	0.0	0.0
	NX09092B	01/10/2019	99.5	99.55		
	NX09093C	28/09/2019	99.5	99.55		
	NX10101A	15/10/2020	98.5	98.55		
	NX10501A	15/06/2020	99.5	99.55		
	NX11500A	15/03/2021	100.0	100.05		
	NX12800A	15/09/2022	99.5	99.55		
10-yr	NX13000A	11/10/2023	100.0	100.05	0.0	0.0
	NY01100F	01/09/2016	110.0	110.05		
	NY05100N	01/09/2020	107.1	107.15		
	NY10101A	01/11/2025	99.5	99.55		
	NY12800W	15/09/2027	99.0	99.05		
15-yr	NY13000W	11/10/2028	100.0	100.05	0.0	0.0
	NZ08082B	15/08/2028	99.5	99.55		
	NZ08083C	31/12/2020	99.5	99.55		
	NZ10101A	15/01/2031	99.5	99.55		
20-yr	NZ13000H	11/10/2033	100.0	100.05	0.0	0.0
	NA12800Z	15/09/2042	99.5	99.55		
30-yr	NA13000Z	11/10/2043	100.0	100.05	0.0	0.0

Return Print

Rate This Site
Copyright © 2013 Monetary Authority of Singapore

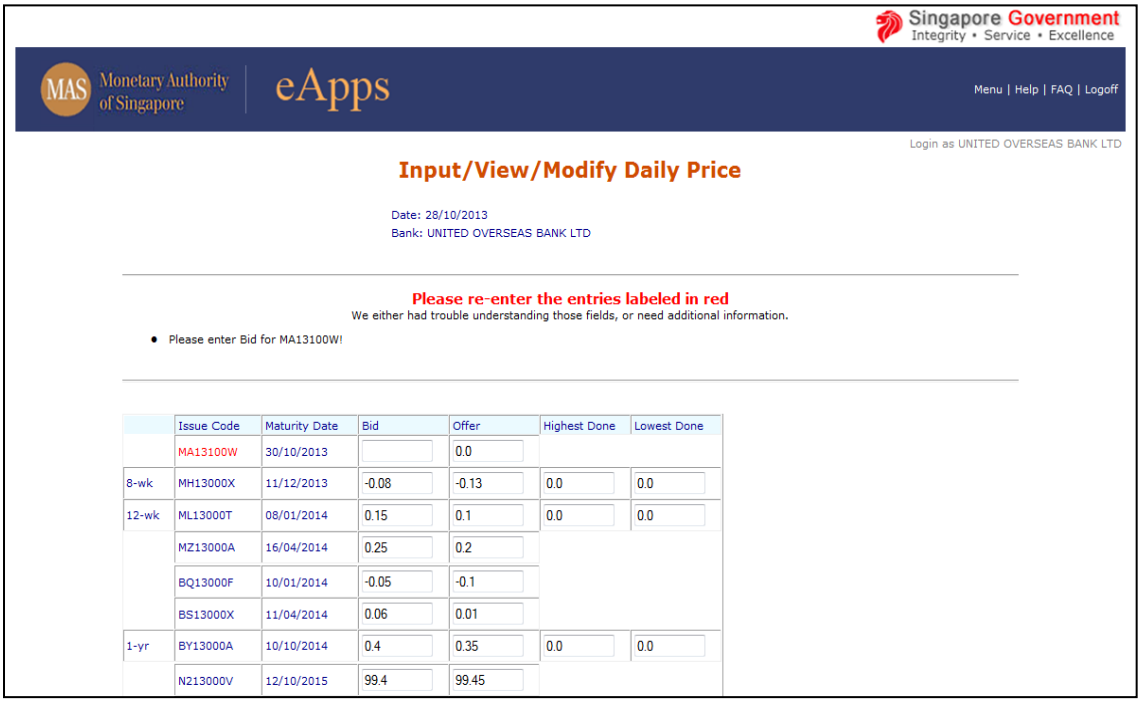
Best viewed using IE 6.0+ at 1024x768 resolution

4.1.34 Inquire daily closing price



	Issue Code	Maturity Date	Bid	Offer	Highest Done	Lowest Done
	ML13201E	19/10/2013	0.4	0.35		
12-wk	ML13201F	23/10/2013	0.4	0.35	0.4	0.2
	BQ13900A	01/10/2013	0.31	0.26		
	BS13202E	25/01/2014	0.31	0.26		
6-mth	BS13202F	29/01/2014	0.31	0.26	0.39	0.23
	BS13501T	15/11/2013	0.31	0.26		
	BS13529U	01/12/2013	0.31	0.26		
	BS13530U	01/12/2013	0.31	0.26		
	BS13531U	01/12/2013	0.31	0.26		
	BS13900A	31/12/2013	0.31	0.26		
	BY11600R	01/08/2014	0.31	0.26		
	BY11600T	13/08/2014	0.31	0.26		
	BY12810S	14/10/2013	0.31	0.26		
	BY13501T	15/05/2014	0.31	0.26		
	BY13600X	01/10/2014	0.31	0.26		
	BY13601W	01/10/2014	0.31	0.26		
1-yr	BY13602N	01/10/2014	0.31	0.26	0.4	0.23
	N211600R	01/08/2015	99.98	100.03		
	N211600T	01/08/2015	99.98	100.03		
	N213201E	27/07/2015	99.98	100.03		
	N213201F	31/07/2015	99.98	100.03		
	N213600T	01/10/2015	99.85	100.51		
	N213601X	01/10/2015	99.75	100.42		
2-yr	N213602W	01/10/2015	99.97	100.05	100.05	99.75
	N509100N	01/10/2014	99.98	100.03		
	N511100W	01/04/2016	99.98	100.03		
5-yr	N512350A	15/06/2018	99.98	100.03	100.02	99.92
	N708100S	01/07/2015	99.98	100.03		
	N710100Z	01/04/2017	99.98	100.03		
	NX04100F	01/07/2014	99.98	100.03		
	NX09100W	01/06/2019	99.98	100.03		
	NX11100X	01/06/2021	99.98	100.03		
10-yr	NX12400A	15/03/2023	99.98	100.03	99.98	99.98
	NY01100F	01/09/2016	99.98	100.03		
	NY03100A	01/09/2018	99.98	100.03		
	NY05100N	01/09/2020	99.98	100.03		
	NY07100X	01/09/2022	99.98	100.03		

4.1.35 Modify daily closing price



Singapore Government
Integrity • Service • Excellence

Monetary Authority of Singapore | eApps

Menu | Help | FAQ | Logoff

Login as UNITED OVERSEAS BANK LTD

Input/View/Modify Daily Price

Date: 28/10/2013
Bank: UNITED OVERSEAS BANK LTD

Please re-enter the entries labeled in red
We either had trouble understanding those fields, or need additional information.

- Please enter Bid for MA13100W!

	Issue Code	Maturity Date	Bid	Offer	Highest Done	Lowest Done
	MA13100W	30/10/2013		0.0		
8-wk	MH13000X	11/12/2013	-0.08	-0.13	0.0	0.0
12-wk	ML13000T	08/01/2014	0.15	0.1	0.0	0.0
	MZ13000A	16/04/2014	0.25	0.2		
	BQ13000F	10/01/2014	-0.05	-0.1		
	BS13000X	11/04/2014	0.06	0.01		
1-yr	BY13000A	10/10/2014	0.4	0.35	0.0	0.0
	N213000V	12/10/2015	99.4	99.45		

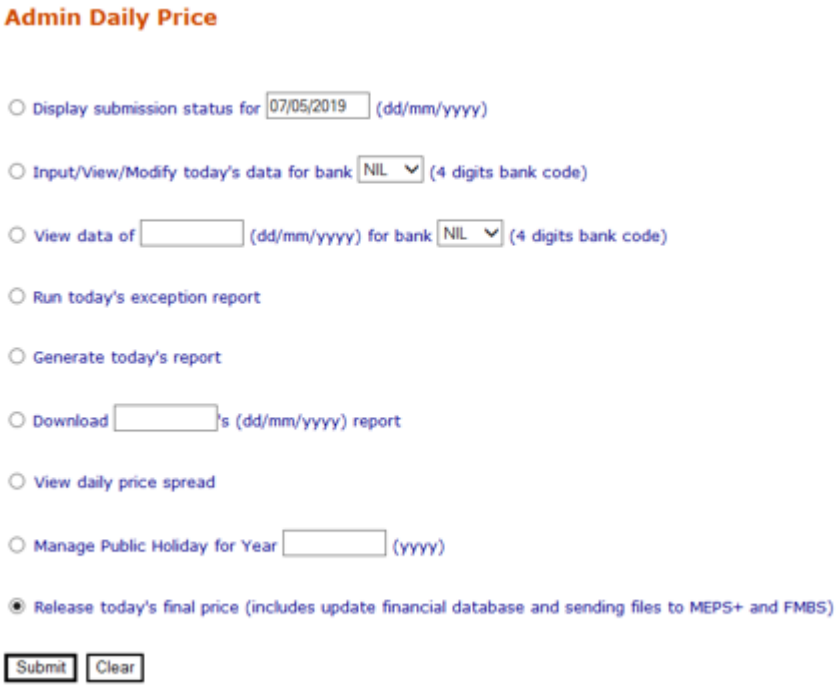
4.1.36 Daily closing price menu for MAS

Admin Daily Price

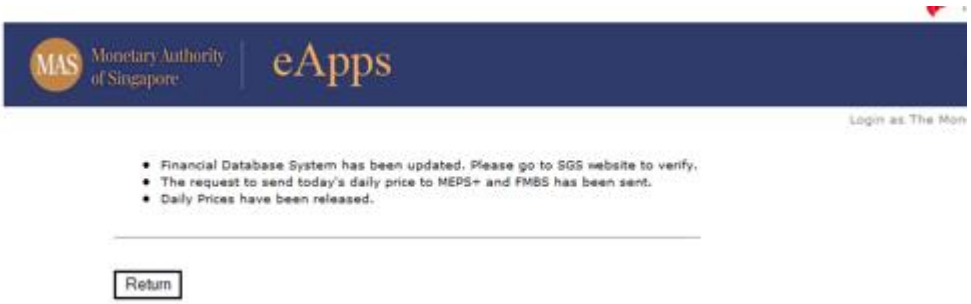
- (●) Display submission status for 09/04/2020 (dd/mm/yyyy)
- ( ) Input/View/Modify today's data for bank NIL (4 digits bank code)
- ( ) View data of (dd/mm/yyyy) for bank NIL (4 digits bank code)
- ( ) Run today's exception report
- ( ) Generate today's report
- ( ) Download 's (dd/mm/yyyy) report
- ( ) View daily price spread
- ( ) Manage Public Holiday for Year (yyyy) of country NIL (Country Code)
- ( ) Release today's final price (includes update financial database and sending files to MEPS+ and FMBS)

Submit Clear

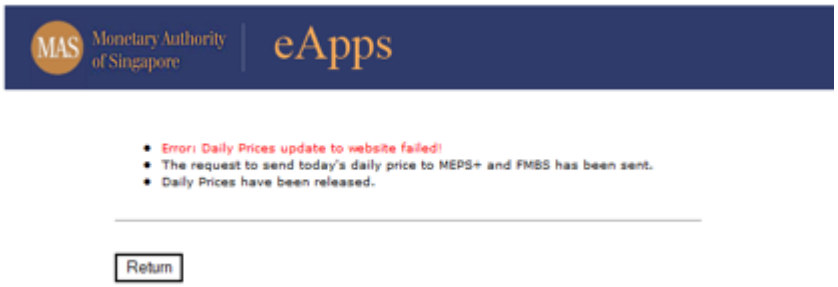
4.1.36.1 Release today daily price.



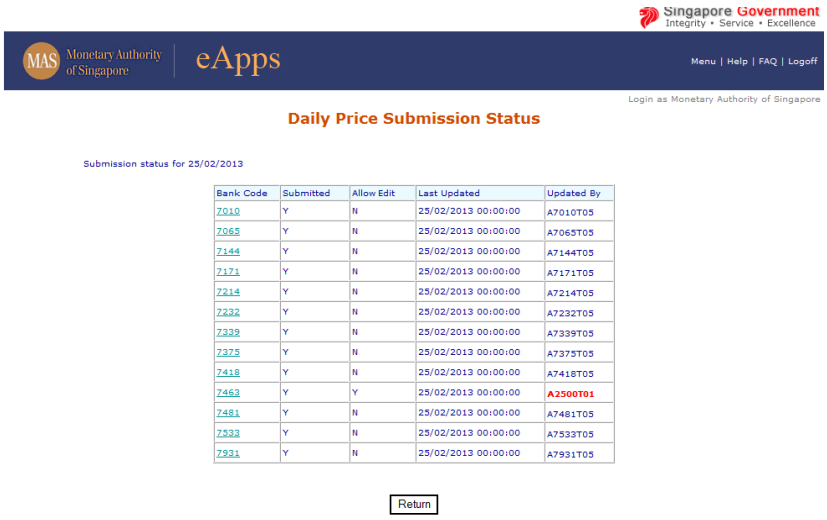
1. Success message of release daily price.



2. Failure message of release daily price



4.1.37 Display today's submission status



Bank Code	Submitted	Allow Edit	Last Updated	Updated By
7010	Y	N	25/02/2013 00:00:00	A7010T05
7065	Y	N	25/02/2013 00:00:00	A7065T05
7144	Y	N	25/02/2013 00:00:00	A7144T05
7171	Y	N	25/02/2013 00:00:00	A7171T05
7214	Y	N	25/02/2013 00:00:00	A7214T05
7232	Y	N	25/02/2013 00:00:00	A7232T05
7339	Y	N	25/02/2013 00:00:00	A7339T05
7375	Y	N	25/02/2013 00:00:00	A7375T05
7418	Y	N	25/02/2013 00:00:00	A7418T05
7463	Y	Y	25/02/2013 00:00:00	A2500T01
7481	Y	N	25/02/2013 00:00:00	A7481T05
7533	Y	N	25/02/2013 00:00:00	A7533T05
7931	Y	N	25/02/2013 00:00:00	A7931T05

4.1.38 View submission

Singapore Government
Integrity · Service · Excellence

MAS Monetary Authority of Singapore | eApps

Menu | Help | FAQ | Logoff

Login as Monetary Authority of Singapore

View Daily Price

Date: 06/12/2013
Bank: BANK OF AMERICA NA

	Issue Code	Maturity Date	Bid	Offer	Highest Done	Lowest Done
8-wk	MH13000X	11/12/2013	-0.08	-0.13	0.0	0.0
12-wk	ML13000T	08/01/2014	0.15	0.1	0.0	0.0
24-wk	MX13500Z	21/05/2014				
	MZ13000A	16/04/2014	0.25	0.2		
	MZ13000B	16/04/2014	0.25	0.2		
6-mth	BS13203U	26/05/2014				
1-yr	BY13500A	04/12/2014				
2-yr	N212000N	15/11/2015	100.0	100.05	-	-
	N213000V	12/10/2015	99.4	99.45		
	N508082B	01/02/2014	100.0	100.05		
	N512800E	15/09/2017	100.0	100.05		
	N513000E	11/10/2018	99.98	100.03		
5-yr	N513500E	01/12/2018				
	N708081A	15/11/2015	100.0	100.05		
	N999003C	01/02/2024	100.0	100.05		
	NX04100F	01/07/2014	102.48	102.53		
	NX08083C	15/10/2018	100.0	100.05		
	NX09092B	01/10/2019	100.0	100.05		
	NX09093C	28/09/2019	100.0	100.05		
	NX10101A	15/10/2020	98.5	98.55		
	NX10501A	15/06/2020	100.0	100.05		
	NX11500A	15/03/2021	100.0	100.05		
	NX12800A	15/09/2022	100.0	100.05		
10-yr	NX13000A	11/10/2023	100.0	100.05	0.0	0.0
	NY01100F	01/09/2016	110.0	110.05		
	NY05100N	01/09/2020	107.1	107.15		
	NY10101A	01/11/2025	100.0	100.05		
	NY12800W	15/09/2027	99.0	99.05		
15-yr	NY13000W	11/10/2028	100.0	100.05	0.0	0.0
	NZ08082B	15/08/2028	100.0	100.05		
	NZ08083C	31/12/2020	100.0	100.05		
	NZ10101A	15/01/2031	100.0	100.05		
20-yr	NZ13000H	11/10/2033	100.0	100.05	0.0	0.0
	NA12800Z	15/09/2042	100.0	100.05		
30-yr	NA13000Z	11/10/2043	100.0	100.05	0.0	0.0
			Sell (Offer)	Buy (Bid)		

Set draft Modify data

Return | Submit | Clear

Rate This Site
Copyright © 2013 Monetary Authority of Singapore
Best viewed using IE 6.0+ at 1024x768 resolution

4.1.39 Edit submission

Singapore Government
Integrity • Service • Excellence

Monetary Authority of Singapore | eApps

Menu | Help | FAQ | Logoff

Login as DBS BANK LTD

Input/View/Modify Daily Price

Date: 23/09/2013
Bank: DBS BANK LTD

	Issue Code	Maturity Date	Bid	Offer	Highest Done	Lowest Done
	ML13201E	19/10/2013	0.4	0.35		
12-wk	ML13201F	23/10/2013	0.4	0.35	0.4	0.2
	BQ13900A	01/10/2013	0.31	0.26		
	BS13202E	25/01/2014	0.31	0.26		
6-mth	BS13202F	29/01/2014	0.31	0.26	0.39	0.23
	BS13501T	15/11/2013	0.31	0.26		
	BS13529U	01/12/2013	0.31	0.26		
	BS13530U	01/12/2013	0.31	0.26		
	BS13531U	01/12/2013	0.31	0.26		
	BS13900A	31/12/2013	0.31	0.26		
	BY11600R	01/08/2014	0.31	0.26		
	BY11600T	13/08/2014	0.31	0.26		
	BY12810S	14/10/2013	0.31	0.26		
	BY13501T	15/05/2014	0.31	0.26		
	BY13600X	01/10/2014	0.31	0.26		
	BY13601W	01/10/2014	0.31	0.26		
1-yr	BY13602N	01/10/2014	0.31	0.26	0.4	0.23
	N211600R	01/08/2015	99.98	100.03		
15-yr	NY12810E	15/10/2027	99.98	100.03	-	-
	NZ07100S	01/03/2027	99.98	100.03		
	NZ10100F	01/09/2030	99.98	100.03		
20-yr	NZ12810N	15/10/2032	99.98	100.03	100.05	99.95
	NA12100N	01/04/2042	99.98	100.03		
30-yr	NA12811W	15/10/2042	99.98	100.03	101.0	99.0
			Sell (Offer)	Buy (Bid)		
	O/N Repo Rate		0.1	0.2		
	3-Mth Comm Bill		0.3	0.4		

☐ Save as draft

Submit Clear

Local intranet | Protected Mode: On

4.1.40 Confirm submission

Singapore Government
Integrity · Service · Excellence

Monetary Authority of Singapore | eApps

Menu | Help | FAQ | Logoff

Login as Monetary Authority of Singapore

Please Confirm Your Submission.

- Daily price for date: 06/12/2013.
- Bank: BANK OF AMERICA NA.
- Your user id with eApps is: A2500T01.
- Date/Time: 06-12-2013 15:49:41.
- Please click on the button below to confirm your submission.

Warning:

- Note that Bid for MH13000X is <= 0.
- Note that Offer for MH13000X is <= 0.
- Offer gives a negative yield for N508082B!
- Offer gives a negative yield for N512800E!
- Offer gives a negative yield for N708081A!
- Offer gives a negative yield for N999003C!
- Bid gives a negative yield for NX04100F!
- Offer gives a negative yield for NX04100F!
- Offer gives a negative yield for NX08083C!
- Offer gives a negative yield for NX09092B!
- Offer gives a negative yield for NX09093C!
- Offer gives a negative yield for NX10501A!
- Offer gives a negative yield for NX12800A!
- Offer gives a negative yield for NY10101A!
- Offer gives a negative yield for NZ08082B!
- Offer gives a negative yield for NZ08083C!
- Offer gives a negative yield for NZ10101A!
- Offer gives a negative yield for NA12800Z!

	Issue Code	Maturity Date	Bid	Offer	Highest Done	Lowest Done
8-wk	MH13000X	11/12/2013	-0.08	-0.13	0.0	0.0
12-wk	ML13000T	08/01/2014	0.15	0.1	0.0	0.0
24-wk	MX13500Z	21/05/2014	0.15	.1	-	-
	MZ13000A	16/04/2014	0.25	0.2		
	MZ13000B	16/04/2014	0.25	0.2		
6-mth	BS13203U	26/05/2014	.4	.38	-	-
1-yr	BY13500A	04/12/2014	.38	.35	-	-
2-yr	N212000N	15/11/2015	100.0	100.05	-	-
	N213000V	12/10/2015	99.4	99.45		
	N508082B	01/02/2014	100.0	100.05		
	N512800E	15/09/2017	100.0	100.05		
	N513000E	11/10/2018	99.98	100.03		
5-yr	N513500E	01/12/2018	100	100.04	-	-
	N708081A	15/11/2015	100.0	100.05		
	N999003C	01/02/2024	100.0	100.05		
	NX04100F	01/07/2014	102.48	102.53		
	NX08083C	15/10/2018	100.0	100.05		
	NX09092B	01/10/2019	100.0	100.05		
	NX09093C	28/09/2019	100.0	100.05		
	NX10101A	15/10/2020	98.5	98.55		
	NX10501A	15/06/2020	100.0	100.05		
	NX11500A	15/03/2021	100.0	100.05		
	NX12800A	15/09/2022	100.0	100.05		
10-yr	NX13000A	11/10/2023	100.0	100.05	0.0	0.0
	NY01100F	01/09/2016	110.0	110.05		
	NY05100N	01/09/2020	107.1	107.15		
	NY10101A	01/11/2025	100.0	100.05		
	NY12800W	15/09/2027	99.0	99.05		
15-yr	NY13000W	11/10/2028	100.0	100.05	0.0	0.0
	NZ08082B	15/08/2028	100.0	100.05		
	NZ08083C	31/12/2020	100.0	100.05		
	NZ10101A	15/01/2031	100.0	100.05		
20-yr	NZ13000H	11/10/2033	100.0	100.05	0.0	0.0
	NA12800Z	15/09/2042	100.0	100.05		
30-yr	NA13000Z	11/10/2043	100.0	100.05	0.0	0.0

Confirm Cancel

10-yr	NX12400A	15/03/2023	99.98	100.03	99.98	99.98
	NY01100F	01/09/2016	99.98	100.03		
	NY03100A	01/09/2018	99.98	100.03		
	NY05100N	01/09/2020	99.98	100.03		
	NY07100X	01/09/2022	99.98	100.03		
	NY09100H	01/09/2024	99.98	100.03		
15-yr	NY12810E	15/10/2027	99.98	100.03	-	-
	NZ07100S	01/03/2027	99.98	100.03		
	NZ10100F	01/09/2030	99.98	100.03		
20-yr	NZ12810N	15/10/2032	99.98	100.03	100.05	99.95
	NA12100N	01/04/2042	99.98	100.03		
30-yr	NA12811W	15/10/2042	99.98	100.03	101.0	99.0
			Sell (Offer)	Buy (Bid)		
	O/N Repo Rate		0.1	0.2		
	3-Mth Comm Bill		0.3	0.4		

Confirm Cancel

Rate This Site
Copyright © 2013 Monetary Authority of Singapore

Best viewed using IE 6.0+ at 1024x768 resolution

Local intranet | Protected Mode: On

4.1.41 Submission succeed

Singapore Government
Integrity · Service · Excellence

Monetary Authority of Singapore | eApps

Menu | Help | FAQ | Logoff

Login as Monetary Authority of Singapore

Transaction Succeeded.

- Your submission has been accepted. Please print a copy for reference.
- Daily price for date: 06/12/2013.
- Bank: BANK OF AMERICA NA.
- Token User Id: A2500T01.
- Date/Time: 06-12-2013 15:49:41.

	Issue Code	Maturity Date	Bid	Offer	Highest Done	Lowest Done
8-wk	MH13000X	11/12/2013	-0.08	-0.13	0.0	0.0
12-wk	ML13000T	08/01/2014	0.15	0.1	0.0	0.0
24-wk	MX13500Z	21/05/2014	0.15	.1	-	-
	MZ13000A	16/04/2014	0.25	0.2		
	MZ13000B	16/04/2014	0.25	0.2		
6-mth	BS13203U	26/05/2014	.4	.38	-	-
1-yr	BY13500A	04/12/2014	.38	.35	-	-
2-yr	N212000N	15/11/2015	100.0	100.05	-	-
	N213000V	12/10/2015	99.4	99.45		
	N508082B	01/02/2014	100.0	100.05		
	N512800E	15/09/2017	100.0	100.05		
	N513000E	11/10/2018	99.98	100.03		
5-yr	N513500E	01/12/2018	100	100.04	-	-
	N708081A	15/11/2015	100.0	100.05		
	N999003C	01/02/2024	100.0	100.05		
	NX04100F	01/07/2014	102.48	102.53		
	NX08083C	15/10/2018	100.0	100.05		
	NX09092B	01/10/2019	100.0	100.05		
	NX09093C	28/09/2019	100.0	100.05		
	NX10101A	15/10/2020	98.5	98.55		
	NX10501A	15/06/2020	100.0	100.05		
	NX11500A	15/03/2021	100.0	100.05		
	NX12800A	15/09/2022	100.0	100.05		
10-yr	NX13000A	11/10/2023	100.0	100.05	0.0	0.0
	NY01100F	01/09/2016	110.0	110.05		
	NY05100N	01/09/2020	107.1	107.15		
	NY10101A	01/11/2025	100.0	100.05		
	NY12800W	15/09/2027	99.0	99.05		
15-yr	NY13000W	11/10/2028	100.0	100.05	0.0	0.0
	NZ08082B	15/08/2028	100.0	100.05		
	NZ08083C	31/12/2020	100.0	100.05		
	NZ10101A	15/01/2031	100.0	100.05		
20-yr	NZ13000H	11/10/2033	100.0	100.05	0.0	0.0
	NA12800Z	15/09/2042	100.0	100.05		
30-yr	NA13000Z	11/10/2043	100.0	100.05	0.0	0.0

Return Print

Rate This Site
Copyright © 2013 Monetary Authority of Singapore
Best viewed using IE 6.0+ at 1024x768 resolution

4.1.42 Run Exception report

Singapore Government
Integrity · Service · Excellence

MAS Monetary Authority of Singapore | eApps

Menu | Help | FAQ | Logoff

Login as Monetary Authority of Singapore

Exception report

Banks that have not submitted:

High/Low spread exception:

- For issue BS13203U, 7065's bid 0.4 appears large against today's ave 0.0.
- For issue BS13203U, 7065's bid 0.4 and 's bid 0.0 exceed highest/lowest spread of 25 cents.
- For issue BY13500A, 7065's bid 0.38 appears large against today's ave 0.0.
- For issue BY13500A, 7065's bid 0.38 and 's bid 0.0 exceed highest/lowest spread of 25 cents.
- For issue N213000V, 7144's high/ave spread appears wide!
- For issue N213000V, 7144's low/ave spread appears wide!
- For issue N213000V, 7533's high/ave spread appears wide!
- For issue N213000V, 7533's low/ave spread appears wide!
- For issue N213000V, 7931's high/ave spread appears wide!
- For issue N213000V, 7931's low/ave spread appears wide!
- For issue N213000V, 7171's high/ave spread appears wide!
- For issue N213000V, 7171's low/ave spread appears wide!
- For issue N213000V, 7418's high/ave spread appears wide!
- For issue N213000V, 7418's low/ave spread appears wide!
- For issue N213000V, 7463's high/ave spread appears wide!
- For issue N213000V, 7463's low/ave spread appears wide!
- For issue N213000V, 7375's high/ave spread appears wide!
- For issue N213000V, 7375's low/ave spread appears wide!
- For issue N213000V, 7481's high/ave spread appears wide!
- For issue N213000V, 7481's low/ave spread appears wide!
- For issue N213000V, 7214's high/ave spread appears wide!
- For issue N213000V, 7214's low/ave spread appears wide!
- For issue N213000V, 7232's high/ave spread appears wide!
- For issue N213000V, 7232's low/ave spread appears wide!
- For issue N213000V, 7339's high/ave spread appears wide!
- For issue N213000V, 7339's low/ave spread appears wide!
- For issue N213000V, 7302's high/ave spread appears wide!
- For issue N213000V, 7302's low/ave spread appears wide!
- For issue N513500F, 7065's bid 100.0 appears large against today's ave 0.0.

4.1.43 View price spread

Singapore Government
Integrity • Service • Excellence

MAS Monetary Authority of Singapore | eApps

Menu | Help | FAQ | Logoff

Login as Monetary Authority of Singapore

Today's Daily Price Spread

Date: 25/02/2013

Tenure	Check - Front-end & Back-end				Check - Back-end	
	Against yesterday's	Bid/Ask spread	High/Low spread	Highest/Lowest & Ave	Highest/Lowest	High/Low & Ave
T-Bill/MAS-Bill						
<=3mths	0.15	0.25	0.15	0.15	0.25	0.9
>3mths	0.15	0.15	0.15	0.15	0.25	0.9
Bond						
<=1 yr	0.15	0.15	0.15	0.15	0.25	0.9
>1 to 2 yrs	0.15	0.15	0.15	0.15	0.25	0.9
>2 to 3 yrs	0.3	0.3	0.5	0.2	0.25	0.9
>3 to 4 yrs	0.3	0.3	0.5	0.2	0.5	0.9
>4 to 5 yrs	0.3	0.3	0.5	0.2	0.25	0.9
>5 to 6 yrs	0.75	0.35	1.65	0.25	0.4	1.4
>6 to 7 yrs	0.75	0.35	1.65	0.25	0.4	1.4
>7 to 8 yrs	0.75	0.35	1.65	0.25	0.4	1.4
>8 to 9 yrs	0.75	0.35	1.65	0.25	0.4	1.4
>9 to 10 yrs	0.75	0.35	1.65	0.25	0.4	1.4
>10 to 11 yrs	1.05	0.4	1.85	0.35	0.55	2.05
>11 to 12 yrs	1.05	0.4	1.85	0.35	0.55	2.05
>12 to 13 yrs	1.05	0.4	1.85	0.35	0.55	2.05
>13 to 14 yrs	1.2	0.6	1.9	0.5	0.7	2.2
>14 to 15 yrs	1.2	0.6	1.9	0.5	0.7	2.2
>15 to 16 yrs	1.45	0.5	2.05	0.4	0.65	2.1
>16 to 17 yrs	1.45	0.5	2.05	0.4	0.65	2.1

4.1.44 Input price spread succeed

No longer available

4.1.45 Manage public holiday

Manage Public Holiday for Year [ ] (yyyy) of country NIL (Country Code)

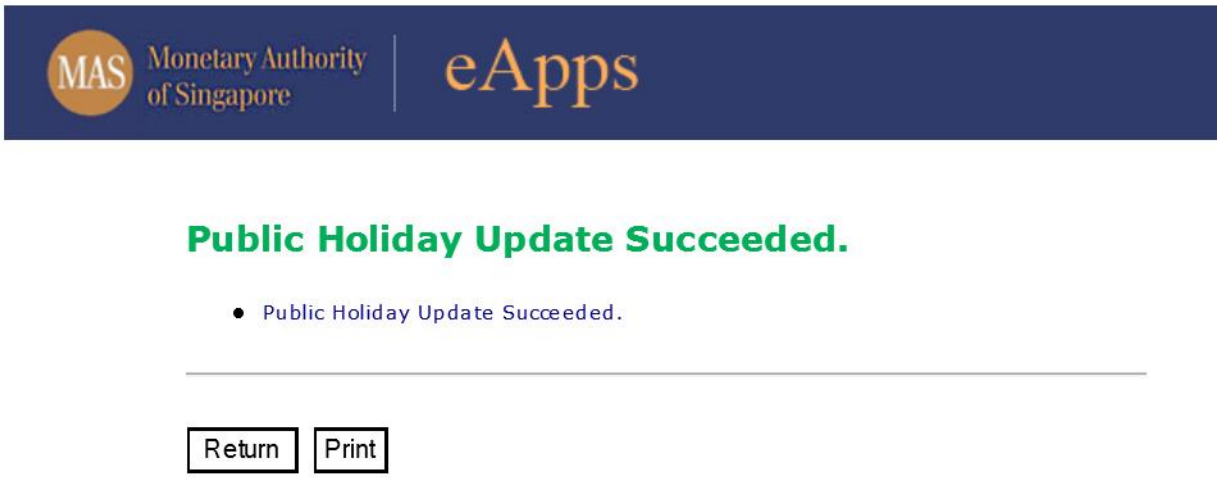
NIL
SG
US

SG Public Holidays for Year 2020

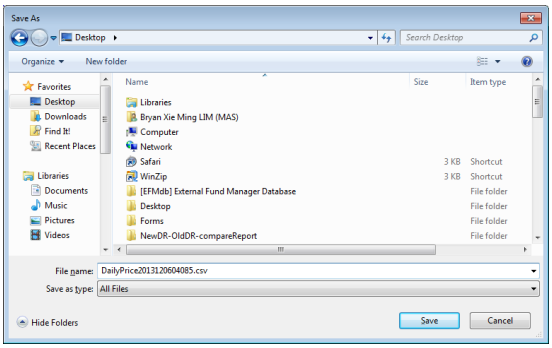
	Holiday Date (dd/mm/yyyy)	Description
1	01/01/2020	New Year
2	25/01/2020	Chinese New Year
3	26/01/2020	Chinese New Year Holiday
4	27/01/2020	Chinese New Year Holiday
5	10/04/2020	Good Friday
6	01/05/2020	Labour Day
7	07/05/2020	Vesak Day
8	24/05/2020	Hari Raya Puasa
9	25/05/2020	Hari Raya Puasa Holiday
10	31/07/2020	Hari Raya Haji
11	09/08/2020	National Day
12	10/08/2020	National Day Holiday
13	14/11/2020	Deepavali
14	25/12/2020	Christmas Day
15		

Return Submit Clear

4.1.46 Manage public holiday succeed



4.1.47 Run daily closing price report



Run today's exception report

Generate today's report

Download 's (dd/mm/yyyy) report

View daily price spread

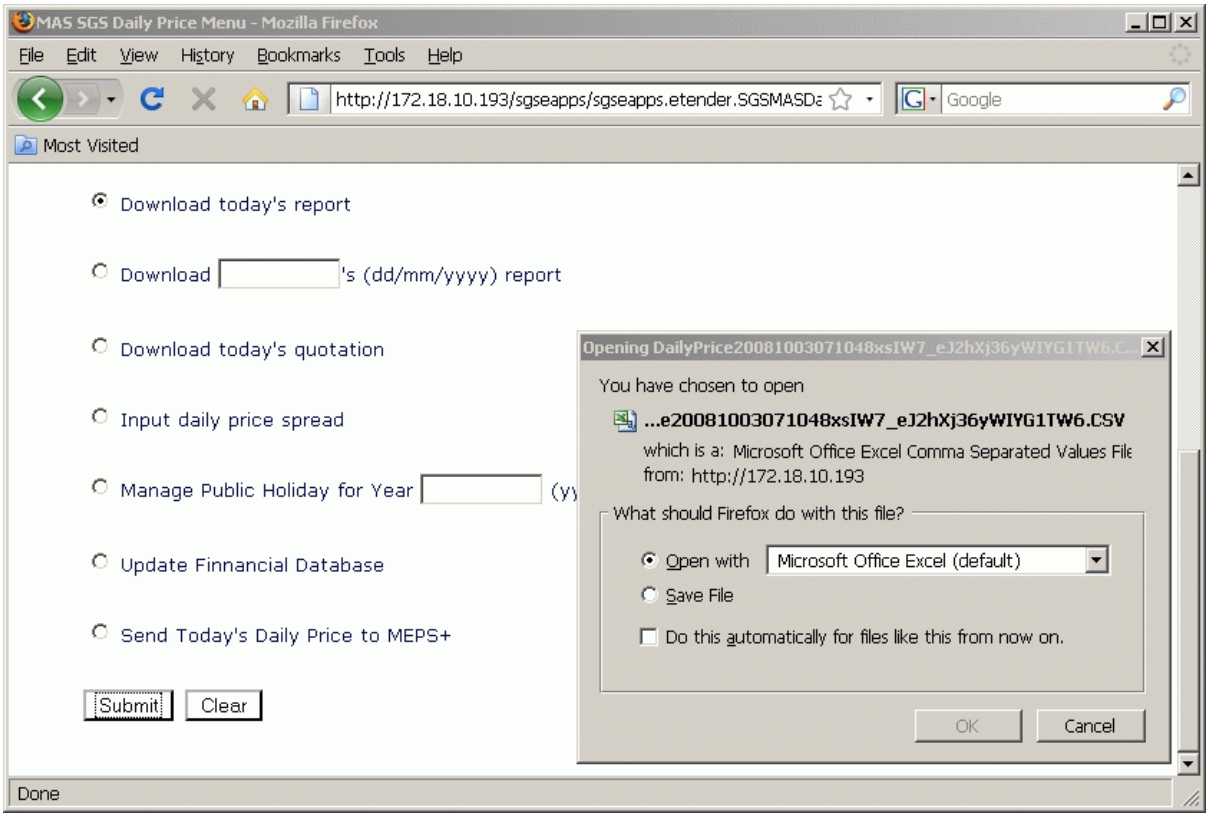
Manage Public Holiday for Year (yyyy)

Release today's final price (includes update financial database and sending files to MEPS+ and FMBS)

Submit Clear

Rate This Site
Copyright © 2013 Monetary Authority of Singapore

Best viewed using IE 6.0+ at 1024x768 resolution


MAS SGS Daily Price Menu - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://172.18.10.193/sgseapps/sgseapps.etender.SGSMASDa
Google
Most Visited
Download today's report
Download 's (dd/mm/yyyy) report
Download today's quotation
Input daily price spread
Manage Public Holiday for Year
Update Finnancial Database
Send Today's Daily Price to MEPS+
Submit
Clear
Opening DailyPrice20081003071048xsIW7_eJ2hXj36yWIYG1TW6.C...
You have chosen to open
...e20081003071048xsIW7_eJ2hXj36yWIYG1TW6.CSV
which is a: Microsoft Office Excel Comma Separated Values File
from: http://172.18.10.193
What should Firefox do with this file?
Open with
Microsoft Office Excel (default)
Save File
Do this automatically for files like this from now on.
OK
Cancel
Done

4.1.48 New issue announcement list

Announcement of Auctions for Securities

05 Apr 11

Latest

- **MAS Bills**

Announcement Date	Issue Code	Description	Issue Date	Auction Date
09/03/2011	MD11503D	4-WEEK MAS BILL DUE 080411	11/03/2011	10/03/2011

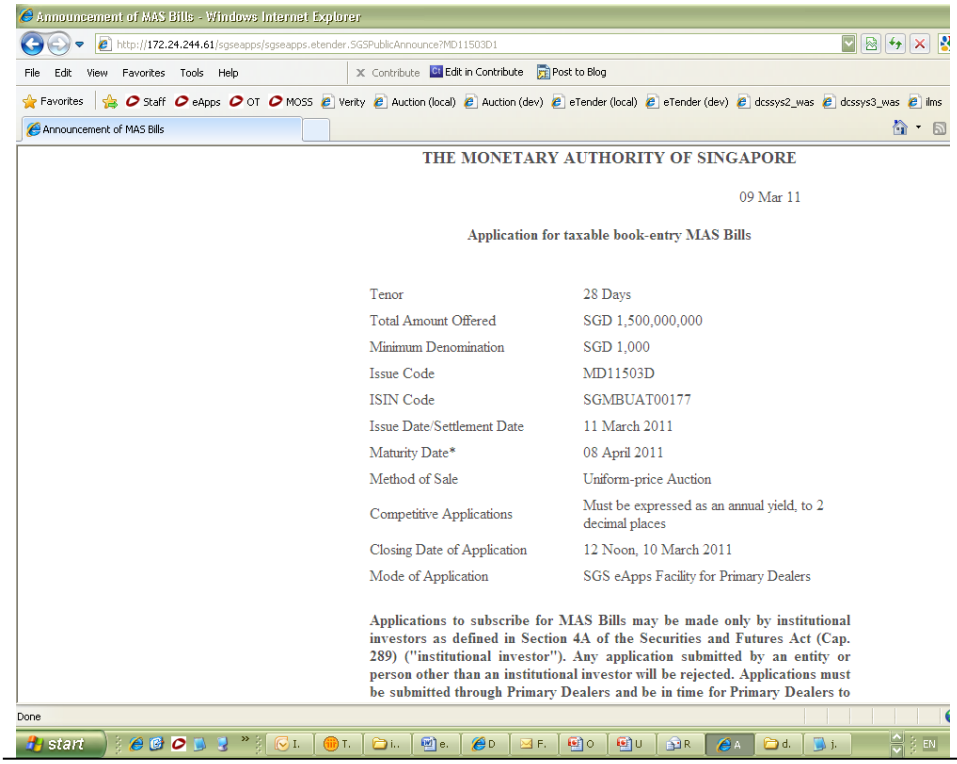
- **Treasury Bill (3-Mth)**

Announcement Date	Issue Code	Description	Issue Date	Auction Date
09/03/2011	BQ11503D	91-D T-BILL DUE 100611	11/03/2011	10/03/2011
09/03/2011	BQ11504E	91-D T-BILL DUE 100611	18/03/2011	17/03/2011
09/03/2011	BQ11505F	91-D T-BILL DUE 100611	09/04/2011	08/04/2011

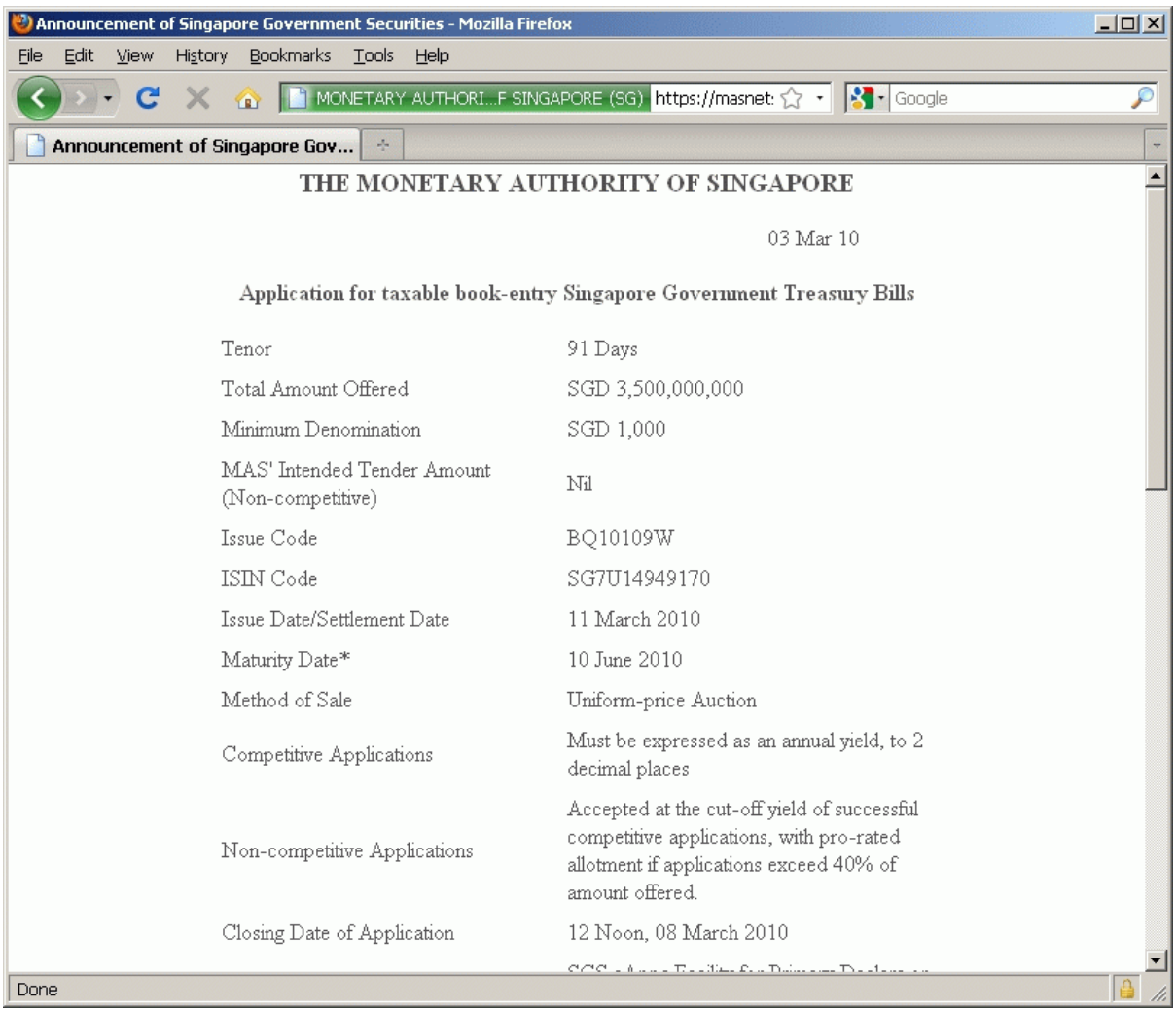
- **Treasury Bill (1-Yr)/SGS Bond**

Announcement Date	Issue Code	Description	Issue Date	Auction Date
09/03/2011	NX11500A	10-YR BOND DUE 150321	11/03/2011	10/03/2011
09/03/2011	NX11501A	10-YR BOND DUE 150321	18/03/2011	17/03/2011

4.1.49 MAS Bill new issue announcement



4.1.50 T-bill new issue announcement



Announcement of Singapore Government Securities - Mozilla Firefox

File Edit View History Bookmarks Tools Help

MONETARY AUTHORI...F SINGAPORE (SG) https://masnet:

Google

Announcement of Singapore Gov...

THE MONETARY AUTHORITY OF SINGAPORE

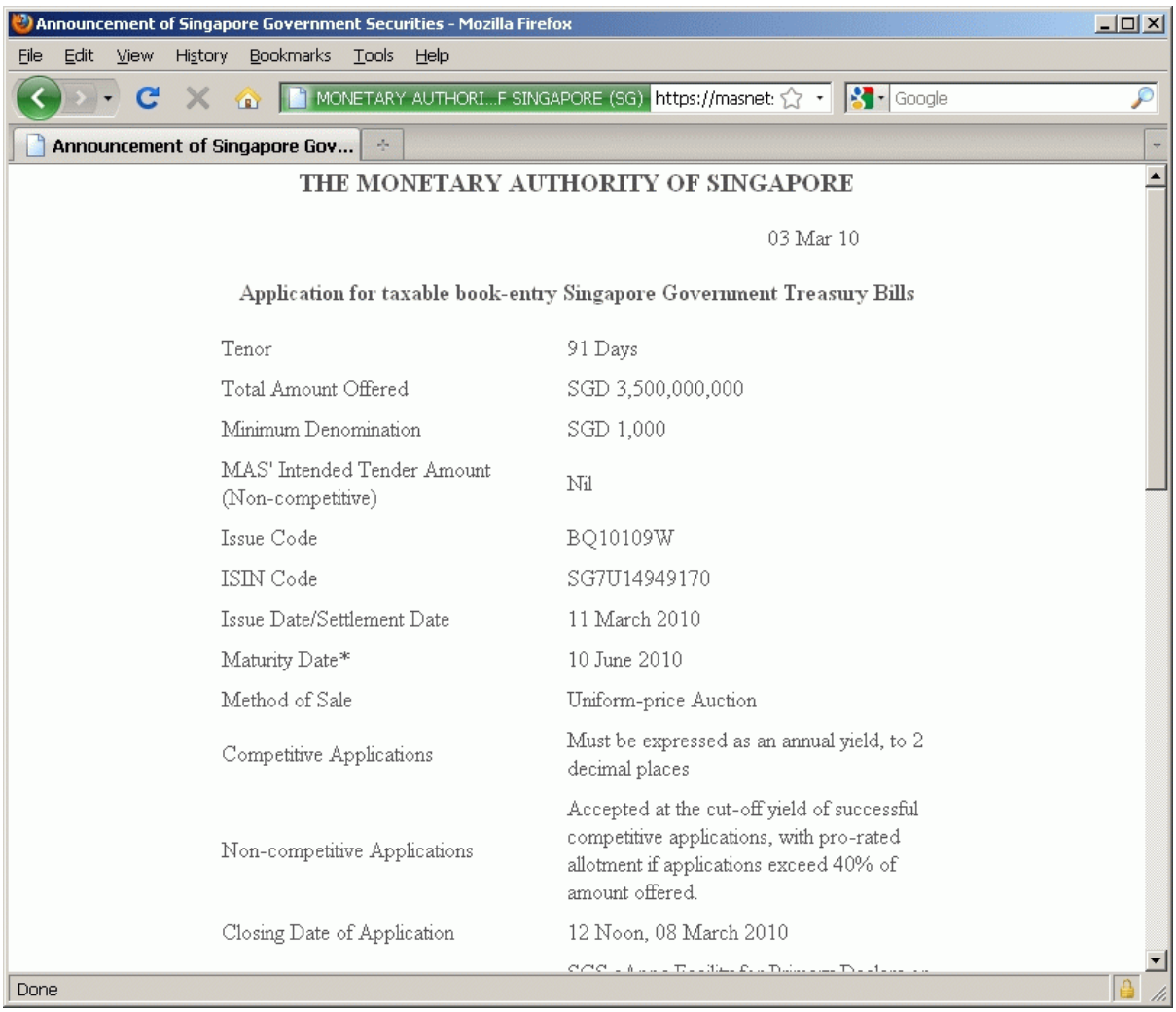
03 Mar 10

Application for taxable book-entry Singapore Government Treasury Bills

Tenor	91 Days
Total Amount Offered	SGD 3,500,000,000
Minimum Denomination	SGD 1,000
MAS' Intended Tender Amount (Non-competitive)	Nil
Issue Code	BQ10109W
ISIN Code	SG7U14949170
Issue Date/Settlement Date	11 March 2010
Maturity Date*	10 June 2010
Method of Sale	Uniform-price Auction
Competitive Applications	Must be expressed as an annual yield, to 2 decimal places
Non-competitive Applications	Accepted at the cut-off yield of successful competitive applications, with pro-rated allotment if applications exceed 40% of amount offered.
Closing Date of Application	12 Noon, 08 March 2010

Done

4.1.51 Bond new issue announcement



Announcement of Singapore Government Securities - Mozilla Firefox

File Edit View History Bookmarks Tools Help

MONETARY AUTHORI...F SINGAPORE (SG) https://masnet:

Google

Announcement of Singapore Gov...

THE MONETARY AUTHORITY OF SINGAPORE

03 Mar 10

Application for taxable book-entry Singapore Government Treasury Bills

Tenor	91 Days
Total Amount Offered	SGD 3,500,000,000
Minimum Denomination	SGD 1,000
MAS' Intended Tender Amount (Non-competitive)	Nil
Issue Code	BQ10109W
ISIN Code	SG7U14949170
Issue Date/Settlement Date	11 March 2010
Maturity Date*	10 June 2010
Method of Sale	Uniform-price Auction
Competitive Applications	Must be expressed as an annual yield, to 2 decimal places
Non-competitive Applications	Accepted at the cut-off yield of successful competitive applications, with pro-rated allotment if applications exceed 40% of amount offered.
Closing Date of Application	12 Noon, 08 March 2010

Done

4.1.52 Auction result list

Result of Auctions for Securities

29 May 14

Latest

Publish To publish result of auctions conducted on 29/05/2014.

- **MAS Bills**

Issue Code	Description	Issue Date	Auction Date
MD14800T	Mbill 2014Q1 UAT	10/02/2014	06/02/2014

- **Treasury Bill (6-Mth)**

Issue Code	Description	Issue Date	Auction Date
BS14100H	6 Mth test Tbill	03/02/2014	30/01/2014

- **Treasury Bill (1-Yr)/SGS Bond**

Issue Code	Description	Issue Date	Auction Date
BY14101H	1 Yr test Tbill	01/04/2014	21/03/2014
N514800V	5-yr Bond	05/06/2014	29/05/2014

Message display for when the Signal file creation is success or failure for Publishing Auction Results

3. Success

Result of Auctions for Securities

- Publishing of auction results is successful.

29 May 14

Latest

Publish To publish result of auctions conducted on 29/05/2014.

- **MAS Bills**

Issue Code	Description	Issue Date	Auction Date
MD14800T	Mbill 2014Q1 UAT	10/02/2014	06/02/2014

- **Treasury Bill (6-Mth)**

Issue Code	Description	Issue Date	Auction Date
BS14100H	6 Mth test Tbill	03/02/2014	30/01/2014

- **Treasury Bill (1-Yr)/SGS Bond**

Issue Code	Description	Issue Date	Auction Date
BY14101H	1 Yr test Tbill	01/04/2014	21/03/2014
N514800V	5-yr Bond	05/06/2014	29/05/2014

4. Failure

Result of Auctions for Securities

- Error: Publishing of auction results to website is unsuccessful.

29 May 14

Latest

Publish **To publish result of auctions conducted on 29/05/2014.**

- **MAS Bills**

Issue Code	Description	Issue Date	Auction Date
MD14800T	Mbill 2014Q1 UAT	10/02/2014	06/02/2014

- **Treasury Bill (6-Mth)**

Issue Code	Description	Issue Date	Auction Date
BS14100H	6 Mth test Tbill	03/02/2014	30/01/2014

- **Treasury Bill (1-Yr)/SGS Bond**

Issue Code	Description	Issue Date	Auction Date
BY14101H	1 Yr test Tbill	01/04/2014	21/03/2014
N514800V	5-yr Bond	05/06/2014	29/05/2014

MAS Bill auction result

THE MONETARY AUTHORITY OF SINGAPORE

10 Mar 11

Results of auction of taxable book-entry MAS Bills to be issued on 11 March 2011

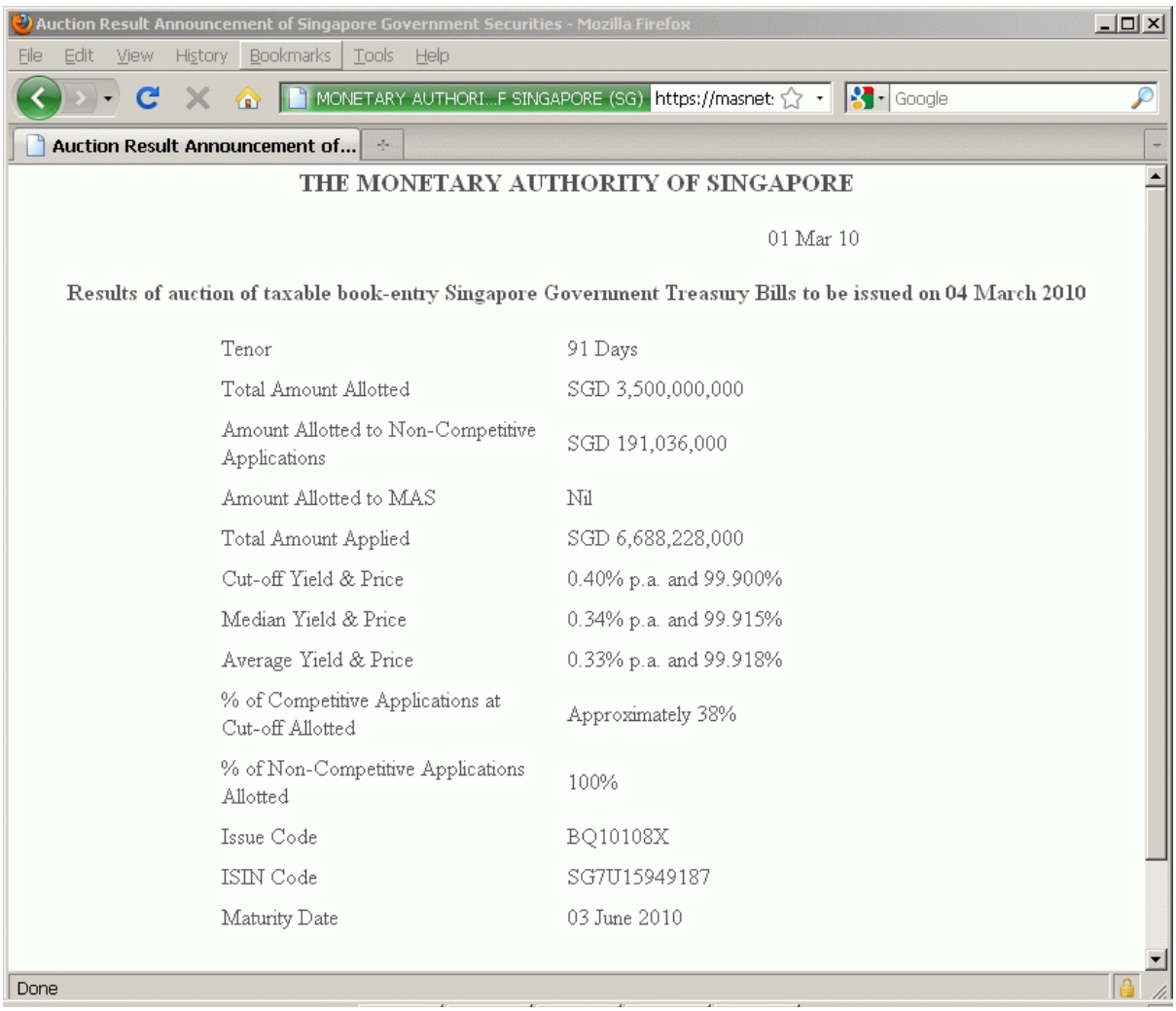
Tenor	28 Days
Total Amount Allotted	SGD 1,500,000,000
Total Amount Applied	SGD 2,180,000,000
Cut-off Yield & Price	0.28% p.a. and 99.979%
Median Yield & Price	0.23% p.a. and 99.982%
Average Yield & Price	0.24% p.a. and 99.982%
% of Competitive Applications at Cut-off Allotted	Approximately 19%
Issue Code	MD11503D
ISIN Code	SGMBUAT00177
Maturity Date	08 April 2011

Rate this e-service

Please contact the MASNET helpdesk at 1800-225-8247 if you encounter any problem.

Done Internet 100%

4.1.53 T-bill auction result



Auction Result Announcement of Singapore Government Securities - Mozilla Firefox

File Edit View History Bookmarks Tools Help

MONETARY AUTHORI...F SINGAPORE (SG) https://masnet: Google

Auction Result Announcement of...

THE MONETARY AUTHORITY OF SINGAPORE

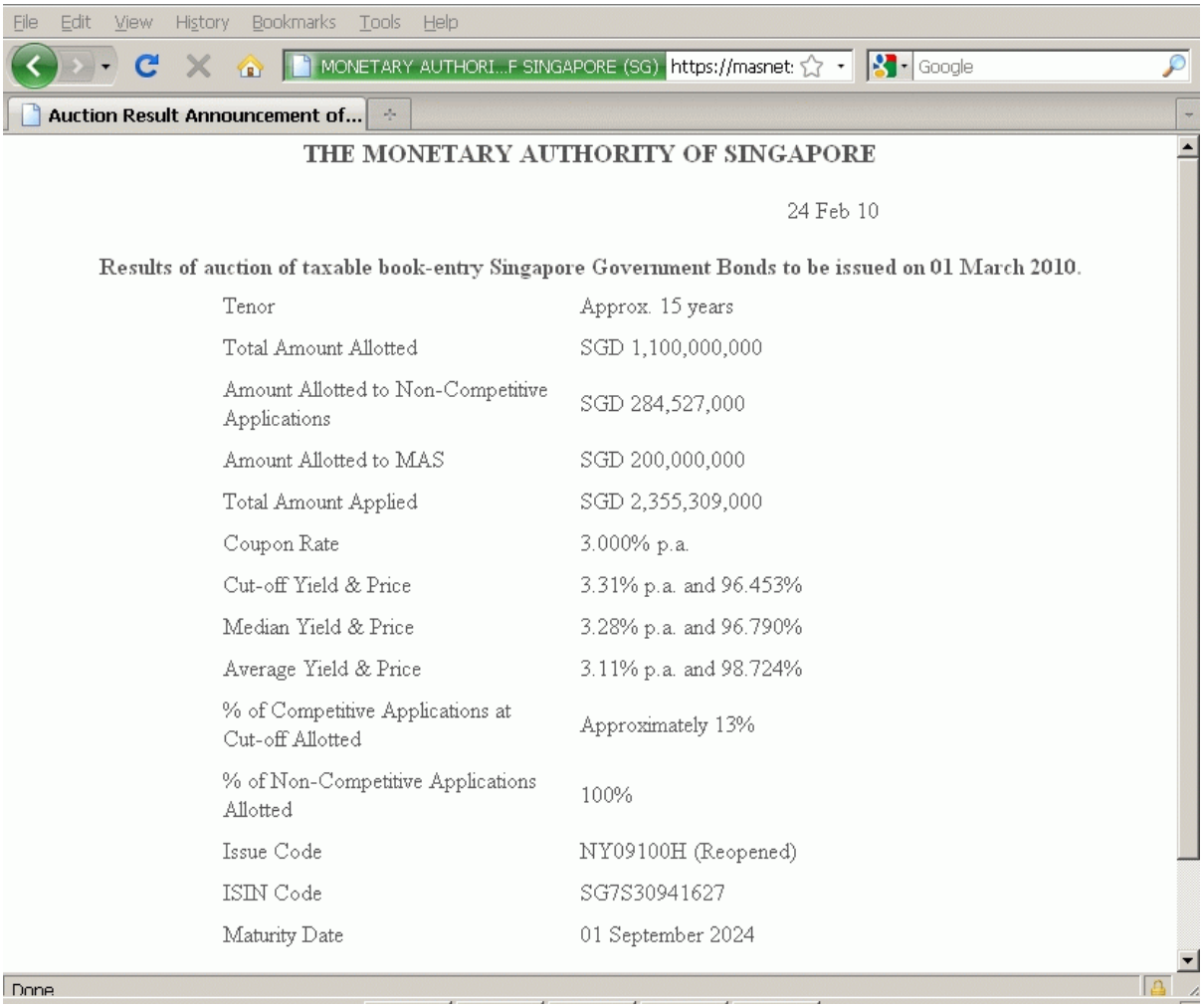
01 Mar 10

Results of auction of taxable book-entry Singapore Government Treasury Bills to be issued on 04 March 2010

Tenor	91 Days
Total Amount Allotted	SGD 3,500,000,000
Amount Allotted to Non-Competitive Applications	SGD 191,036,000
Amount Allotted to MAS	Nil
Total Amount Applied	SGD 6,688,228,000
Cut-off Yield & Price	0.40% p.a. and 99.900%
Median Yield & Price	0.34% p.a. and 99.915%
Average Yield & Price	0.33% p.a. and 99.918%
% of Competitive Applications at Cut-off Allotted	Approximately 38%
% of Non-Competitive Applications Allotted	100%
Issue Code	BQ10108X
ISIN Code	SG7U15949187
Maturity Date	03 June 2010

Done

4.1.54 Bond auction result



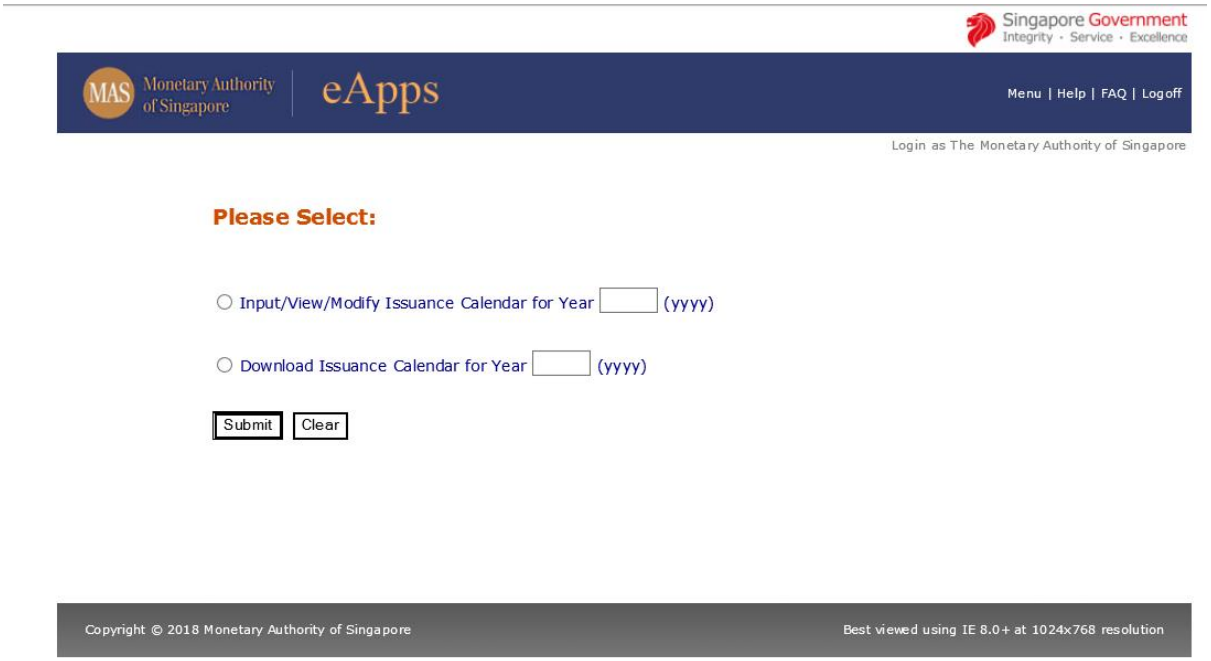
THE MONETARY AUTHORITY OF SINGAPORE

24 Feb 10

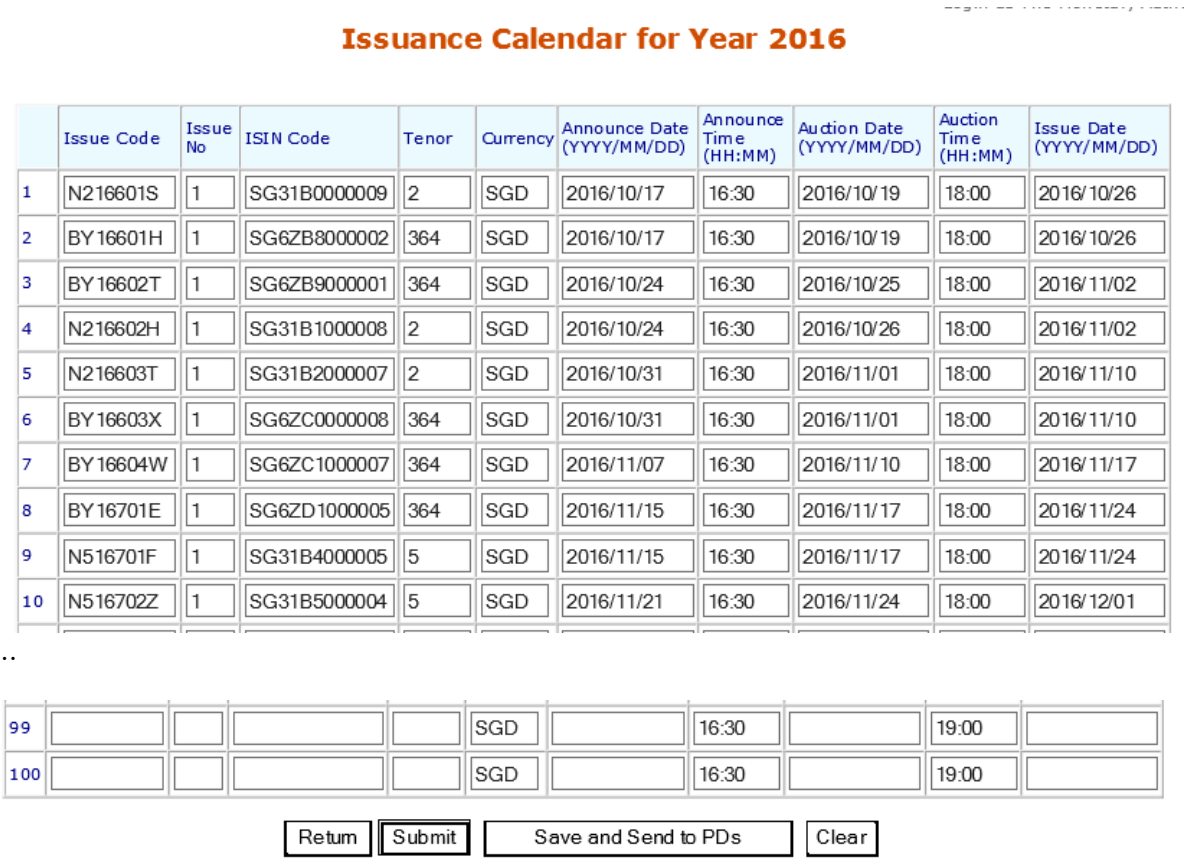
Results of auction of taxable book-entry Singapore Government Bonds to be issued on 01 March 2010.

Tenor	Approx. 15 years
Total Amount Allotted	SGD 1,100,000,000
Amount Allotted to Non-Competitive Applications	SGD 284,527,000
Amount Allotted to MAS	SGD 200,000,000
Total Amount Applied	SGD 2,355,309,000
Coupon Rate	3.000% p.a.
Cut-off Yield & Price	3.31% p.a. and 96.453%
Median Yield & Price	3.28% p.a. and 96.790%
Average Yield & Price	3.11% p.a. and 98.724%
% of Competitive Applications at Cut-off Allotted	Approximately 13%
% of Non-Competitive Applications Allotted	100%
Issue Code	NY09100H (Reopened)
ISIN Code	SG7S30941627
Maturity Date	01 September 2024

4.1.55 Issuance Calendar



Select the "Input/View/Modify Issuance Calendar for Year" == "2016"
Click Submit



Issuance Calendar for Year 2016

	Issue Code	Issue No	ISIN Code	Tenor	Currency	Announce Date (YYYY/MM/DD)	Announce Time (HH:MM)	Auction Date (YYYY/MM/DD)	Auction Time (HH:MM)	Issue Date (YYYY/MM/DD)
1	N216601S	1	SG31B0000009	2	SGD	2016/10/17	16:30	2016/10/19	18:00	2016/10/26
2	BY16601H	1	SG6ZB8000002	364	SGD	2016/10/17	16:30	2016/10/19	18:00	2016/10/26
3	BY16602T	1	SG6ZB9000001	364	SGD	2016/10/24	16:30	2016/10/25	18:00	2016/11/02
4	N216602H	1	SG31B1000008	2	SGD	2016/10/24	16:30	2016/10/26	18:00	2016/11/02
5	N216603T	1	SG31B2000007	2	SGD	2016/10/31	16:30	2016/11/01	18:00	2016/11/10
6	BY16603X	1	SG6ZC0000008	364	SGD	2016/10/31	16:30	2016/11/01	18:00	2016/11/10
7	BY16604W	1	SG6ZC1000007	364	SGD	2016/11/07	16:30	2016/11/10	18:00	2016/11/17
8	BY16701E	1	SG6ZD1000005	364	SGD	2016/11/15	16:30	2016/11/17	18:00	2016/11/24
9	N516701F	1	SG31B4000005	5	SGD	2016/11/15	16:30	2016/11/17	18:00	2016/11/24
10	N516702Z	1	SG31B5000004	5	SGD	2016/11/21	16:30	2016/11/24	18:00	2016/12/01

….

99					SGD		16:30		19:00	
100					SGD		16:30		19:00	

Return | Submit | Save and Send to PDs | Clear

1. When "Save and Send to PD's" clicked, if signal file for "T-Bills and SGS Bonds" does not exists then Update the Flag to 'P' in DB and generate signal file

2. When "Save and Send to PD's" clicked, if signal file for "T-Bills and SGS Bonds" already exists "Another Job is in Progress" message would be shown to user

4.1.56 Announcement

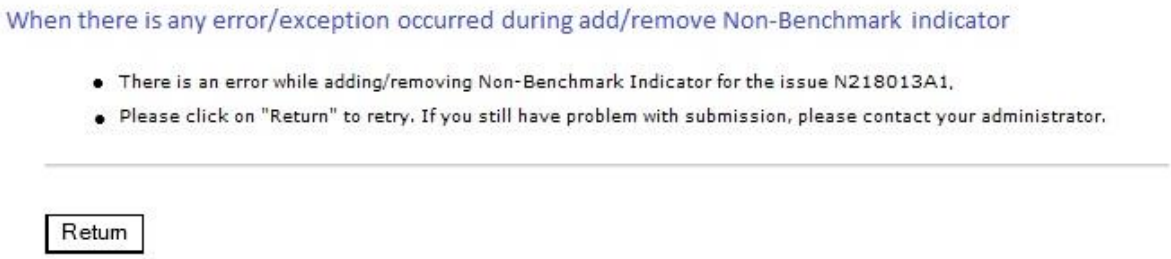
1. Non-Benchmark Indicator
   Message displayed for the different scenarios:

2. Issue Code does not exist

3. Issue Code successfully added to the table ABA0022_NON_BENCHMARK

4. Issue Code successfully removed from the table ABA0022_NON_BENCHMARK

5. Error encountered during add/remove of Issue Code.

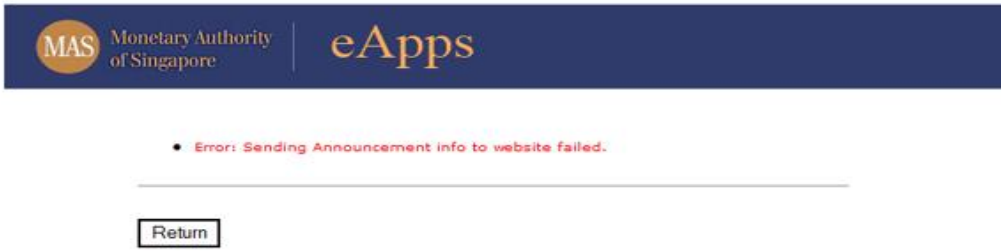


6. Release Announcement

Message display for when the Signal file creation is success or failure for releasing of MAS / T Bills announcement.

7. Success

8. Failure



4.1.57 Footer

Privacy Statement

Privacy Statement

General

This is a Government of Singapore digital service.

We may use "cookies", where a small data file is sent to your browser to store and track information about you when you enter our digital service. The cookie is used to track information such as the number of users and their frequency of use, profiles of users and their preferred digital services. While this cookie can tell us when you enter our digital services and which pages you visit, it cannot read data off your hard disk.

You can choose to accept or decline cookies. Most web browsers automatically accept cookies, but you can usually modify your browser setting to decline cookies if you prefer. This may prevent you from taking full advantage of the digital service.

Use of Data

If you provide us with personal data:

i) Where appropriate, we may share necessary data with other Government agencies, so as to improve the discharge of public functions, and to serve you in the most efficient and effective way unless such sharing is prohibited by law.

ii) We may share your personal data with non-Government entities that have been authorised to carry out specific Government services. We will NOT share your personal data with other non-Government entities without your consent except where such sharing complies with the law.

iii) For your convenience, we may also display to you data you had previously supplied us or other Government agencies. This will speed up the transaction and save you the trouble of repeating previous submissions. Should the data be out-of-date, please supply us the latest data.

Terms of Use

Singapore Government
Integrity · Service · Excellence

Monetary Authority of Singapore | eApps

Terms of Use

Thank you for using the eApps ("this Web Site").

By accessing and using any part of this Web Site, you shall be deemed to have accepted to be legally bound by these Terms & Conditions of Use. If you do not agree to these Terms & Conditions of Use, please do not use this Web Site.

General

These Terms & Conditions of Use may be changed from time to time. All changes will be posted on this page, and your use of this Web Site after such changes have been posted will constitute your agreement to the modified Terms & Conditions of Use and all of the changes. You should therefore read this page carefully each time you visit this Web Site.

Proprietary Rights

This Web Site is owned and operated by the Monetary Authority of Singapore (hereinafter "MAS").

The materials located on this Web Site, including the information as well as any software programs available on or through this Web Site ("the Contents"), are protected by copyright, trademark and other forms of proprietary rights. All rights, title and interest in the Contents are owned by, licensed to or controlled by MAS.

Restrictions on use of Materials

Except as otherwise provided, the Contents of this Web Site shall not be reproduced, republished, uploaded, posted, transmitted or otherwise distributed in any way, without the prior written permission of MAS.

You may download one copy of any information provided in this Web Site onto a single computer for your own personal non-commercial use provided that you keep intact all accompanying copyright and other proprietary notices.

Modification of any of the Contents or use of the Contents for any other purpose will be a violation of MAS's copyright and other intellectual property rights. Graphics and images on this Web Site are protected by copyright and may not be reproduced or appropriated in any manner without the written permission of their respective copyright owners.

5 Data format

5.1 Load new security issues into Oracle table

The format of "secmast.txt" is:

Field Name	Length	e.g.	Comment
ISSUE_CODE	8	NX00100H	1-8
ISSUE_NO	1		9-9
ISSUE_TYPE	1		10-10
CURR	3	SGD	11-13
ISSUE_DESC	30		14-43
ISSUE_DATE	8	YYYYMMDD	44-51
TENDER_DATE	8	YYYYMMDD	52-59
QTY_OFFERED	13		60-72
QTY_APPLIED	13		73-85
AVE_YIELD	5	999.99	86-90
CUT_OFF_YIELD	5	999.99	91-95
MATURITY_DATE	8	YYYYMMDD	96-103
PERCENT_COY	5	999.99	104-108
PERCENT_SUB	5	999.99	109-113
NC_PERCENT	5	999.99	114-118
NC_QTY_ALLOT	13		119-131
INT_RATE	7		132-138
TAX_STATUS	1		139-139
CUT_OFF_YIELD PRICE	7	999.9999	140-146
AVE_YIELD PRICE	7	999.9999	147-153
CLOSING_PRICE	7	999.9999	154-160
ISIN_CODE	12		161-172
TENOR	3		173-175
ETENDER_IND	1	'Y' if issue_date>14012002	176-176
MAS_APPLIED	13		177-189
MAS_ALLOTTED	13		190-202
INT_PAID_IND	1		203-203
LAST_INT_DATE	8	YYYYMMDD	204-211
NEXT_INT_DATE	8	YYYYMMDD	212-219
ACCRUED_INT_DAYS	3		220-222
INT_DATE1	4	MMDD	223-226
INT_DATE2	4	MMDD	227-230
EX_INT_DATE	8	YYYYMMDD	231-238

5.2 Load security daily updates (secint.txt) into Oracle table

The format of "secint.txt" is:

Field Name	Length	e.g.	Comment
ISSUE_CODE	8	NX00100H	1-8
ISSUE_NO	1		9-9
ISSUE_TYPE	1		10-10
CURR	3	SGD	11-13
ISSUE_DESC	30		14-43
ISSUE_DATE	8	YYYYMMDD	44-51
TENDER_DATE	8	YYYYMMDD	52-59
QTY_OFFERED	13		60-72
QTY_APPLIED	13		73-85
AVE_YIELD	5	999.99	86-90
CUT_OFF_YIELD	5	999.99	91-95
MATURITY_DATE	8	YYYYMMDD	96-103
PERCENT_COY	5	999.99	104-108
PERCENT_SUB	5	999.99	109-113
NC_PERCENT	5	999.99	114-118
NC_QTY_ALLOT	13		119-131
INT_RATE	7		132-138
TAX_STATUS	1		139-139
CUT_OFF_YIELD PRICE	7	999.9999	140-146
AVE_YIELD PRICE	7	999.9999	147-153
CLOSING_PRICE	7	999.9999	154-160
ISIN_CODE	12		161-172
TENOR	3		173-175
ETENDER_IND	1	'Y' if issue_date>14012002	176-176
MAS_APPLIED	13		177-189
MAS_ALLOTTED	13		190-202
INT_PAID_IND	1		203-203
LAST_INT_DATE	8	YYYYMMDD	204-211
NEXT_INT_DATE	8	YYYYMMDD	212-219
ACCRUED_INT_DAYS	3		220-222
INT_DATE1	4	MMDD	223-226
INT_DATE2	4	MMDD	227-230
EX_INT_DATE	8	YYYYMMDD	231-238
CPN_PAYM_IND	1	This field is not applicable for T-Bills, MAS-Bills and SGS Bonds. For these ISIN Types, it will be defaulted to blank value regardless of any coupon payment on that day. For MAS FRN Bonds, it will be either Y or blank value depending whether there is coupon payment on that day.	239-239

		Field Name: Coupon Payment Indicator Value: Y or Blank Y – if there is coupon payment on that day Blank – if there is no coupon payment on that day	

5.3 Load bank list into Oracle table

The format of "bank.txt" is:

Field Name	Length	Comments
Bank_Code	4	
Bank_Name	40	
Bank_Shortname	15	
Autodebit_Indicator	1	
Participant_Indicator	1	

5.4 File format of application data file for bank upload file

5.4.1 Overview

Each PD will submit the application data file daily through MEFT-access to eApps Application. And the application data files will be placed in the MEFT incoming directories through SFTP.

The eApps Application will retrieve the application data files (CASH, CPFIS and SRS) for each PD from MEFT-access incoming folder for processing.

Once the application data file is processed, an acknowledgement file will be generated. The generated file will be renamed to MEFT File format and will be moved to the MEFT outgoing directory. The moved files from the MEFT outgoing directories will be picked up by the SFTP job and will be pushed them to MEFT-access and take them as backups.

The acknowledgement files (CASH, CPFIS and SRS) will contain the status of the processed file and the status of each individual records.

The PDs will retrieve the acknowledgement files (CASH, CPFIS and SRS) via MEFT-access.

The process will be scheduled by Autosys to run on working days.

5.4.2 Assumption

i) Each application data file will contain subscription requests for more than 1 issuances.
ii) In the event there are no subscription records, the banks will not send in any file.
iii) The application data file deposited by PDs in MEFT-access will be overwritten when an updated file is uploaded.
iv) For Input empty file from PD's – Acknowledgement file will be generated as Invalid File Format
v) For Input file with No Details records from PD's – Acknowledgment file will be generated with Header (File status as 'P') and Trailer alone.
vi) The upstream files from the PDs that are processed will be taken as backups.
vii) The acknowledgement files are renamed to MEFT file format, are down streamed to MEFT-Access, and downstream files are taken as backups.

The filename of the application data file uploaded will not have any date-time stamp.

1. The filename format of the Application Data file submitted

1. ea_<PD Name>_ap1.txt – CASH

Example
ea_dbs_ap1.txt
ea_ocbc_ap1.txt
ea_uob_ap1.txt

2. ea_<PD Name>_ap3.txt – SRS

Example

ea_dbs_ap3.txt
ea_ocbc_ap3.txt
ea_uob_ap3.txt

3. ea_<PD Name>_ap2.txt – CPFIS
Example
ea_dbs_ap2.txt
ea_ocbc_ap2.txt

4. The filename format of the Acknowledgement file that is generated by the application.

1. RT_MASSG.<PD Name>.ABA.ea_<PD NAME>_aa1.txt – CASH
Example
RT_MASSG.DBS.ABA.ea_dbs_aa1.txt
RT_MASSG.OCBCSSB.ABA.ea_ocbc_aa1.txt
RT_MASSG.UOBSFTP.ABA.ea_uob_aa1.txt

2. RT_MASSG.<PD Name>.ABA.ea_<PD NAME>_aa2.txt – CPFIS
Example
RT_MASSG.DBS.ABA.ea_dbs_aa2.txt
RT_MASSG.OCBCSSB.ABA.ea_ocbc_aa2.txt

3. RT_MASSG.<PD Name>.ABA.ea_<PD NAME>_aa3.txt – SRS
Example
RT_MASSG.DBS.ABA.ea_dbs_aa3.txt
RT_MASSG.OCBCSSB.ABA.ea_ocbc_aa3.txt
RT_MASSG.UOBSFTP.ABA.ea_uob_aa3.txt

5. The file name of the Acknowledgement file that is down streamed to MEFT-Access.

1. CASH
EAPPSDBSZZ00RT_MASSG.DBS.ABA.ea_dbs_aa1.txt
EAPPSOCBCZ00RT_MASSG.OCBCSSB.ABA.ea_ocbc_aa1.txt
EAPPSUOBZZ00RT_MASSG.UOBSFTP.ABA.acoeaapplack.txt

2. CPFIS
EAPPSDBSZZ00RT_MASSG.DBS.ABA.ea_dbs_aa2.txt
EAPPSOCBCZ00RT_MASSG.OCBCSSB.ABA.ea_ocbc_aa2.txt

3. SRS
EAPPSDBSZZ00RT_MASSG.DBS.ABA.ea_dbs_aa3.txt
EAPPSOCBCZ00RT_MASSG.OCBCSSB.ABA.ea_ocbc_aa3.txt
EAPPSUOBZZ00RT_MASSG.UOBSFTP.ABA.srseaapplack.txt

6. MEFT-access Incoming directories and their backup directories
Incoming directories
/mas_apps/aba/data/MEFT/DBS/Incoming
/mas_apps/aba/data/MEFT/OCBC/Incoming
/mas_apps/aba/data/MEFT/UOB/Incoming
Backup directories
/mas_apps/aba/data/MEFT/Backup/DBS/Incoming
/mas_apps/aba/data/MEFT/Backup/OCBC/Incoming
/mas_apps/aba/data/MEFT/Backup/UOB/Incoming

7. MEFT-access Outgoing directories and their backup directories

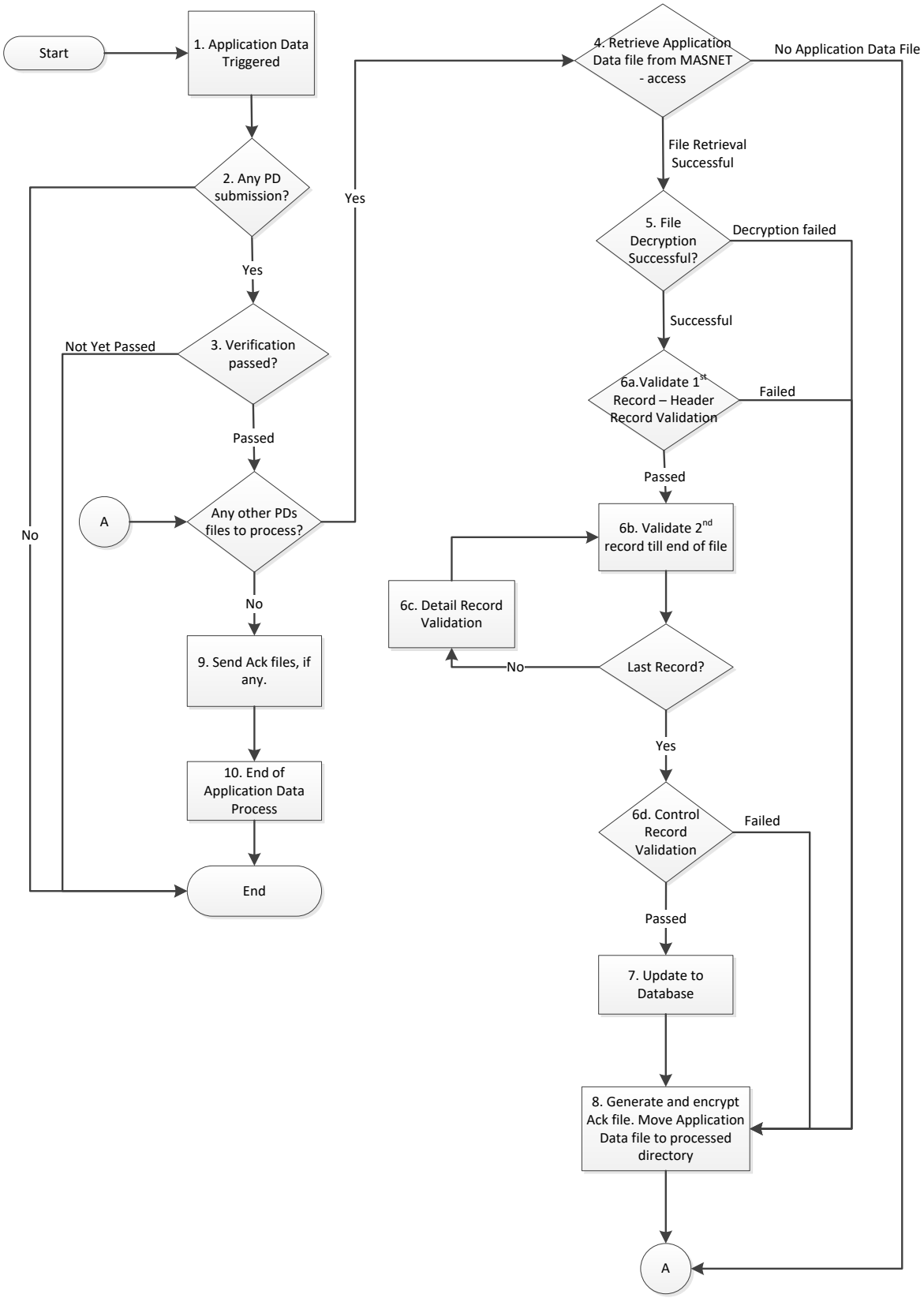
Outgoing directories

/mas_apps/aba/data/MEFT/DBS/Outgoing
/mas_apps/aba/data/MEFT/OCBC/Outgoing
/mas_apps/aba/data/MEFT/UOB/Outgoing

Backup directories

/mas_apps/aba/data/MEFT/Backup/DBS/Outgoing
/mas_apps/aba/data/MEFT/Backup/OCBC/Outgoing
/mas_apps/aba/data/MEFT/Backup/UOB/Outgoing

5.4.3 Process Flow



5.4.4 Process

1) Application Data process is triggered.

   Update Job Status=" ***Application Data submission - Process started [YYYYMMDD HH:MM]***"

   Proceed to **(2)**

2) To check for any PD submission.

   If there are no records found,
   i) update job status =" ***Application Data submission - Process ended – No active issuance [YYYYMMDD HH:MM]***"

   ii) Acknowledgement file will **NOT** be generated for any PD.

   iii) Error code "No Application Data Submission" will be sent to Operator Console.

   iv) The entire process will end.

3) To check if the cut-off timings for Submission have reached.

   a) If Processing_time < SUBMISSION_CUTOFF_TIMING,
      i) update job status =**" *Application Data – Process ended – Cut-off time for submission not yet reached [YYYYMMDD HH:MM]*"**

      ii) Acknowledgement file will **NOT** be generated for any PD.

      iii) Error code "Application Data cut-off timing has not reached" will be sent to Operator Console.

      iv) The entire process will end.

   b) If the cut-off timing has reached proceed to **(4)**

4) Retrieval of Application Data files from MEFT-access.

   For each PD, SFTP to MEFT-access to retrieve the Application Data file.
   If there are no more PD submission files to be processed, proceed to **(10)**

   a) If the Application Data file is retrieved successfully, update the job status=" ***Application Data Submission – <Bank Code> file received [YYYYMMDD HH:MM]***"
      Proceed to **(5)**

   b) If any PD's Application Data file not found in MEFT-access,
      i) update job status = "***Application Data Submission – <Bank Code> file not found [YYYYMMDD HH:MM]***"

      ii) Acknowledgement file for this PD will **NOT** be generated.

~~iii) Error code "Application Data Submission file for <Bank Code> not found" will be sent to Operator Console.~~

iv) Proceed to **(4)** to retrieve the next PD submission file.

5) Decryption of PD Application Data Submission file.

a) If the decryption is successful, the job status will be updated="***Application Data Submission – <Bank Code> file decrypted [YYYYMMDD HH:MM]***"
Proceed to **(6)**.

b) If the decryption failed,

i) Update job status will be = "***Application Data Submission – <Bank Code> file decryption failed [YYYYMMDD HH:MM]***"

ii) Error code "Decryption for <Bank Code> Application Data Submission file failed" will be sent to Operator Console.

iii) Acknowledgement file generated in the format in Annex 5.8.1 for the PD will only contain:

Header record – indicate file status=" F" with file error message to indicate File decryption failed.
Control Record – Record count will be 0 and overall checksum as 0.

The Acknowledgement file will be encrypted using the public key of the bank code after generation.

iv) Update job status =" ***Application Data Submission – <Bank Code> acknowledgement file generated [YYYYMMDD HH:MM]***"

v) The processed Subscription file will be moved and stored in the respective PD Application Data processed directory.

vi) Proceed to **(4)** to retrieve the next PD submission file.

6) Application Data file validation check.

Please refer to section 5.4.5 on the details of the valid value validation

a) Validate 1st record in the file (section 5.4.5)
- Header Record validation check – Record format is correct.
- Valid value for the fields

If there is validation error detected,

i) The job status will be updated=" ***Application Data Submission – <Bank Code> file Header validation failed [YYYYMMDD HH:MM]***"

ii) Error code "Application Data Submission file for <Bank Code> File header validation error" will be sent to Operator Console.

iii) Acknowledgement file generated in the format in Section 5.9.1 for the PD will only contain:

Header record – indicate file status="F" with file error message to indicate File header validation failed.

Control Record – Record count will be 0 and overall checksum as 0.

The Acknowledgement file will be encrypted using the public key of the bank code after generation.

iv) The job status will be updated=" ***Application Data Submission – <Bank Code> acknowledgement file generated [YYYYMMDD HH:MM]***"

v) The processed Application Data file will be moved and stored in the respective PD Application Data processed directory.

vi) The file processing for this PD's submission will end.
Proceed to **(4)** to retrieve the next PD submission file.

If no validation error for Header record, proceed to **(6b)**

b) Validate 2nd record in the file (Section 5.4.5)
- Detail Record-Header validation check – Record format is correct.
- Valid value for the fields.

If there is validation error detected,

i) The job status will be updated=" ***Application Data Submission – <Bank Code> file Detail Header validation failed [YYYYMMDD HH:MM]***"

ii) Error code "Application Data Submission file for <Bank Code> File detail header validation error" will be sent to Operator Console.

iii) Acknowledgement file generated in the format in Section 5.9.1 for the PD will only contain:
Header record: indicate file status=" F" with file error message to indicate File detail header validation failed.
Control Record: Record count will be 0 and overall checksum as 0.

The Acknowledgement file will be encrypted using the public key of the bank code after generation.

iv) The job status will be updated=" ***Application Data Submission – <Bank Code> acknowledgement file generated [YYYYMMDD HH:MM]***"

v) The processed Application Data file will be moved and stored in the respective PD Application Data processed directory.

vi) The file processing for this PD's submission will end.
Proceed to **(4)** to retrieve the next PD submission file.

If no validation error for Header record, proceed to **(6c)**

If record is not the last record,
Proceed for Detail Record validation **(6c)**.
Else
Proceed for Control Record validation **(6d)**.

c) Detail Record validation: (Section 5.4.5)
- Detail record validation check – Record format is correct.
- Valid value for the fields.
- Valid record checksum
- Perform duplication record check with the following fields with the existing database records
  - Record is not "Rejected"
  - ISIN code
  - Bank code
  - Transaction Reference number

i) If any of the above error is identified, the record will be "Rejected" with the respective record error message and proceed to **(6b)**

ii) If there is no validation error, the record will be considered "Pass" and proceed to **(6b)**

d) Control Record validation: (Section 5.4.5)
- Control record validation check – Record format is correct.
- Valid value for the fields.
- Correct Record count
- Valid file checksum

If there is validation error detected,

i) Update job status =" ***Application Data Submission – <Bank Code> file Control validation failed [YYYYMMDD HH:MM]***"

ii) Acknowledgement file generated for the PD will only contain:
Header record – indicate file status=" F" with the respective file error message.

Control Record – Record count will be 0 and overall checksum as 0.

iii) The Acknowledgement file will be encrypted using the public key of the bank code after generation.

iv) Update job status =" ***Application Data Submission – <Bank Code> acknowledgement file generated [YYYYMMDD HH:MM]***"

v) The processed Application Data file will be moved and stored in the respective PD Application Data processed directory.

vi) The file processing for this PD's submission will end.

vii) Proceed to **(4)** to retrieve the next PD submission file.

If there is no validation error detected,

i) job status will be updated =" ***Application Data Submission – <Bank Code> file validated [YYYYMMDD HH:MM]***"

ii) Proceed to **(7)**

7) Update to the database.

The processed Application Data records ("Pass") will be inserted into the database in clear format.

Note:
- This is to follow the existing eAuction Tender submission logic to insert the retail bids record received into AQA0002_TRANSACTION@PRI1 table in clear format.
- To store the meta-data details in ABA0004_RETAIL_BID_TRANS@MNET table for the PDs/MDD to view the details in Submission Inquiry page

Proceed to **(8)**

8) For successful validated Application Data file, generate Acknowledgement file in the format in Section 5.9.1

a) Acknowledgement file generated for the PD will contained:
Header record – indicate file status=" P"
Detail Records – to generate the content and status of each record processed
Control Record – to provide the Record count and overall checksum value.

The Acknowledgement file will be encrypted using the public key of the bank code after generation.

b) Update job status =" ***Application Data Submission – <Bank Code> Acknowledgement file generated [YYYYMMDD HH:MM]***"

c) The processed Application Data file will be moved and stored in the respective PD Application Data processed directory.

d) The file processing for this PD's submission will end.
e) Proceed to **(4)** to retrieve the next PD submission file.

9) Sending of Application Data Acknowledgement files

For each PD's Application Data acknowledgement file,
a) SFTP the acknowledgement files to MEFT-access in the designated folder for the respective PDs.

If the SFTP is unsuccessful, the job status will be updated=" ***Application Data Submission – <Bank Code> acknowledgement file sent failed [YYYYMMDD HH:MM]***"

If the SFTP is successful, the job status will be updated=" ***Application Data Submission – <Bank Code> acknowledgement file sent [YYYYMMDD HH:MM]***"

Proceed to **(11)**

10) End of Subscription Submission process

a) Job status will be updated=" ***Application Data Submission – Process completed [YYYYMMDD HH:MM]***"

b) Success code will be sent to Operator Console.

Entire process end.

5.4.5 Format

	Field Name	Length	e.g.	Comment
Header Record				
1	RECORD_TYPE	X(10)	HHHHHHHHHH	
2	FILE_TYPE	X(3)	AP1/AP2/AP3 – Application (CASH/CPFIS/SRS) i.e. AP1 – Application for CASH, AP2 – Application for CPFIS, AP3 – Application for SRS	
3	PRI_DLR_CODE	9(4)	7171	Numeric bank code
4	BANK_NAME	X(20)		
5	PROCESSING_DATE	9(8)	YYYYMMDD	Date of upload
Detail Record- Header				
6	RECORD_TYPE	X(10)	BBBBBBBBBB	
7	ISSUE_CODE	X(8)	NX00100H	
8	ISIN_CODE	X(12)		
9	TENDER_DATE	9(8)	YYYYMMDD	Auction date
10	CURR	X(3)	SGD	
11	ISSUE_DESC	X(30)		
Detail Record – Detail (repetitive)				
12	RECORD_TYPE	X(10)	DDDDDDDDDD	
13	ISSUE_CODE	X(8)	NX00100H	
14	TRANS_REF	X(8)	Reference No	
15	RECEIVED_DATE	9(8)	YYYYMMDD	
16	RECEIVED_TIME	9(6)	HHMMSS	
17	TRANS_TYPE	X(3)	TE_	
18	NOMINAL_AMT	9(11)	00500000000	
19	COMP_NOCOMP	X(1)		
20	BID_YIELD_SIGN	X(1)	+ or -	either '+' or '-'. If bid yield is 0 would be a '+' sign

21	BID_YIELD	9(3)v9(2)	00305 00205 00000	3.05 2.05 0
22	NAME_OF_APPLN	X(30)		
23	NATIONALITY	X(1)		
24	IC_PASSPORT	X(14)		
25	CUST_BANK_CODE	9(4)		Settlement bank code
26	CUST_BANK_BC	X(1)		Bank or custodian
27	TYPE_OF_APPLN	X(3)		
28	REC_STATUS	X(1)		
29	CDP_ACCT_NO	X(16)		
30	SUB_METHOD	X(1)		'A', 'B', 'C'
31	CPF_SRS_ACCT_NO	X(16)		
32	CHECK_SUM	9(14)		
Control Record				
33	RECORD_TYPE	X(10)	TTTTTTTTTT	
34	RECORD_COUNT	9(9)		
35	OVERALL_CHK_SUM	9(14)		

1. Field 1 to 5 will be on line 1.
2. Field 6 to 11 will be for each issue and be on the same line.
3. Field 12 to 32 will be repetitive for each application record.
4. Each application record must start on a new line.
5. Field 33 to 35 will be the last line.
6. The unused area will be padded with leading zeroes or spaces accordingly.
7. '9' denotes field having a numeric value(s).
8. 'X' denotes field having alphanumeric value(s).
9. The received file will be encrypted using PGP.

5.4.6 Field Definitions

1. RECORD_TYPE
   i) Record type for header record is always 'HHHHHHHHHH'.

2. FILE_TYPE
   i) File type for this file.
   ii) AP1 – CASH, AP3 - SRS

3. PRI_DLR_CODE
   i) Primary dealer code. 4 digits.

4. BANK_NAME
   i) Bank's name.

5. PROCESSING_DATE
   i) Date of upload. The format is YYYYMMDD.

6. RECORD_TYPE
   i) Record type for the header of detail record is always 'BBBBBBBBBB'.

7. ISSUE_CODE
   i) Example of the code: NX00100H.
   ii) MAS Bills, FRN and CMTB are invalid Issue Code for Retail Bid file received from PDs
      a) MAS Bills – Issue Code starts with 'M' and not with 'M1'
      b) FRN – Issue Code starts with 'M1'
      c) CMTB – Issue Code starts with 'BA'

8. ISIN_CODE
   i) ISIN code for the issue, which is 12 alphanumeric characters.

9. TENDER_DATE
   i) Should be 8 digits. The format is YYYYMMDD. This is also known as auction date.

10. CURR
    i) The value will always be 'SGD'.

11. ISSUE_DESC
    i) The description of the issue.

12. RECORD_TYPE
    i) Record type for detail of detail record is always 'DDDDDDDDDD'.

13. ISSUE_CODE
    i) Example of the code: NX00100H.
    ii) MAS Bills, FRN and CMTB are invalid Issue Code for Retail Bid file received from PDs
       a) MAS Bills – Issue Code starts with 'M' and not with 'M1'
       b) FRN – Issue Code starts with 'M1'
       c) CMTB – Issue Code starts with 'BA'

14. TRANS_REF
    i) Bank's transaction reference number, which is alphanumeric and must be unique, e.g. 00000001.

15. RECEIVED_DATE
    i) The date that ATM/iBanking receives the transaction. It should be 8 digits. The format is YYYYMMDD.

16. RECEIVED_TIME
    i) The time that ATM/iBanking receives the transaction. It should be 6 digits. The format is HHMMSS.

17. TRANS_TYPE
    i) Always be TE_.

18. NOMINAL_AMT
    i) The applied amount of this application. Unused areas will be padded with leading zeroes.

19. COMP_NOCOMP
    i) 'C' for competitive bid. 'N' for non-competitive bid.

20. BID_YIELD_SIGN
    i) This refers to the sign forbid yield. It can be "+" or "-". For zero yield, it will be "+".

21. BID_YIELD
    i) The bid yield for this application. '00000' for non-competitive bid. The value in this field is that the actual bid yield multiplies 100, e.g. 00305 for actual yield 3.05.
    ii) Yield range accepted differs based on the "**mnet.acceptNegativeYield**" property value defined in the below properties file
        For upload file through eApps application - sgs.properties
        For files to be processed through eAuction Batch (Retail Bid file process) - eauction_batch.properties

        a) If the property value is true, then Yield range accepted is -99.99 to 99.99 (0 is a valid yield)
        b) If the property value is empty or false, then Yield range accepted is 0.01 to 99.99 (0 is not a valid yield)

22. NAME_OF_APPLN
    i) Name of the applicant.

23. NATIONALITY
    i) 'S' for Singapore citizen and PR. 'F' for a foreigner.

24. IC_PASSPORT
    i) IC or passport number. Singapore IC[3] no. will be validated. Records that fail in validation will be discarded.

25. CUST_BANK_CODE
    i) Settlement bank code. It should be same as primary dealer code for individual applicants.

26. CUST_BANK_BC
    i) Bank or custodian indicator. 'B' or 'C'. It should always be 'C' for individual applicants.

27. TYPE_OF_APPLN
    i) Type of applicant. It should always be 'IND' for an individual applicant.

28. REC_STATUS
    i) Always is space.

29. CDP_ACCT_NO
    i) Applicant's CDP account no. SGS eApps will not validate this field.

30. SUB_METHOD
    i) Submission method indicator. 'A' for ATM applications, 'B' for iBanking applications, and 'C' for mobile app applications.

31. CPF_SRS_ACCT_NO
    i) Applicant's CPF or SRS account no. SGS eApps will not validate SRS account no.
    ii) For upload bids and eAuction Batch (Retail Bid file process), CPF No Validation will be done only for the file type 'AP2'

32. CHECKSUM

[3] Both T Series and S Series NRIC numbers will be validated

i) The sum of all numeric fields of this application, i.e. RECEIVED_DATE, RECEIVED_TIME, NOMINAL_AMT, BID_YIELD, and CUST_BANK_CODE. BID_YIELD uses the integer value in field 21 (to ignore the sign field 20).

33. RECORD_TYPE
    i) Record type for control record is always 'TTTTTTTTTT'.

34. RECORD_COUNT
    i) A total number of applications.

35. OVERALL_CHK_SUM
    i) The total sum of each record's checksum (field 31). If the total sum is bigger than 99,999,999,999,999, use the last 14 digits of the number.

5.5 The format of auction result file for banks

5.5.1 Overview

The Auction process will be performed by an eAuction module and transferred to the eApps module.

A batch job will be triggered to retrieve the Auction processed info and update to the format in section 5.5.4.

The filename format of the Result file.

1. RT_MASSG.<PD Name>.ABA.ea_<PD NAME>_ra1.txt – CASH

Example
RT_MASSG.DBS.ABA.ea_dbs_ra1.txt
RT_MASSG.OCBCSSB.ABA.ea_ocbc_ra1.txt
RT_MASSG.UOBSFTP.ABA.ea_uob_ra1.txt

2. RT_MASSG.<PD Name>.ABA.ea_<PD NAME>_ra2.txt – CPFIS

Example
RT_MASSG.DBS.ABA.ea_dbs_ra2.txt
RT_MASSG.OCBCSSB.ABA.ea_ocbc_ra2.txt

3. RT_MASSG.<PD Name>.ABA.ea_<PD NAME>_ra3.TXT – SRS

Example
RT_MASSG.DBS.ABA.ea_dbs_ra3.txt
RT_MASSG.OCBCSSB.ABA.ea_ocbc_ra3.txt
RT_MASSG.UOBSFTP.ABA.ea_uob_ra3.txt

The downstream file names of the Auction Result file that are transferred to MEFT-Access.

CASH
EAPPSDBSZZ00RT_MASSG.DBS.ABA.ea_dbs_ra1.txt
EAPPSOCBCZ00RT_MASSG.OCBCSSB.ABA.ea_ocbc_ra1.txt
EAPPSUOBZZ00RT_MASSG.UOBSFTP.ABA.acoeaapplres.txt

CPF
EAPPSDBSZZ00RT_MASSG.DBS.ABA.ea_dbs_ra2.txt
EAPPSOCBCZ00RT_MASSG.OCBCSSB.ABA.ea_ocbc_ra2.txt

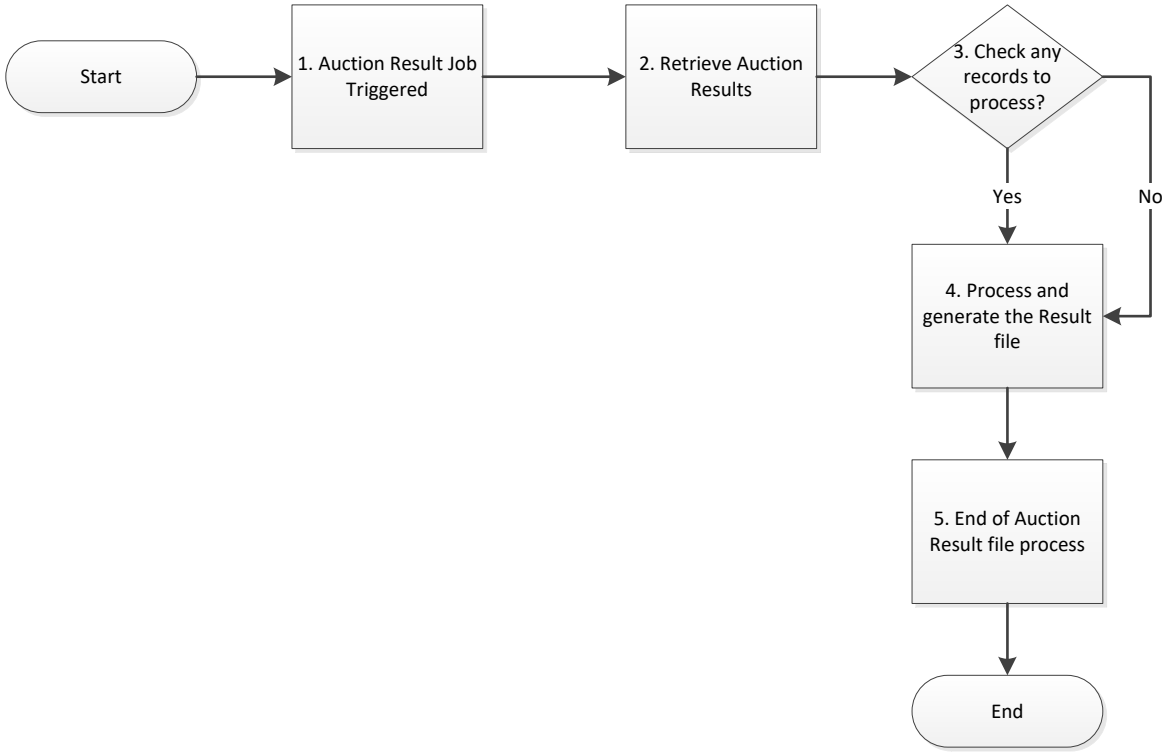
SRS
EAPPSDBSZZ00RT_MASSG.DBS.ABA.ea_dbs_ra3.txt
EAPPSOCBCZ00RT_MASSG.OCBCSSB.ABA.ea_ocbc_ra3.txt
EAPPSUOBZZ00RT_MASSG.UOBSFTP.ABA.srseaapplres.txt

MEFT-access Outgoing directories and their backup directories

Outgoing directories
/mas_apps/aba/data/MEFT/DBS/Outgoing
/mas_apps/aba/data/MEFT/OCBC/Outgoing
/mas_apps/aba/data/MEFT/UOB/Outgoing

Backup directories
/mas_apps/aba/data/MEFT/Backup/DBS/Outgoing
/mas_apps/aba/data/MEFT/Backup/OCBC/Outgoing
/mas_apps/aba/data/MEFT/Backup/UOB/Outgoing

5.5.2 Process Flow



5.5.3 Process

1. The auction process is triggered.
   Update Job Status="***Auction - Process started - <PD Code> - [YYYYMMDD HH:MM]***"

   Proceed to **(2)**

2. Retrieve records for generating result file.
   Update job status = **"*Auction – Generate Result file format Started - <PD Code> - [YYYYMMDD HH:MM]*"**

3. To check of any records to process.

1. If record to process found

1. Proceed to (4)

2. Update Job status=" ***Auction – Processed <PD Code> – [xxx] records [YYYYMMDD HH:MM]***"

3. Else If record to process not found
   1. Update Job status=" ***Auction - Process completed <PD Code> – 0 records [YYYYMMDD HH:MM]***"

   2. Proceed to (4) to generate the header info.
      Header record – Header info for the PD
      Control record – Record count will be 0 and overall checksum as 0

4. Generation of Result file format in Section 5.5.4.

5. End of Auction Result Process

1. Job status will be updated=" ***Auction – Process Completed <PD Code>- [YYYYMMDD HH:MM]***"

5.5.4 Format

	Field Name	Length	e.g.	Comment
Header Record				
1	RECORD_TYPE	X(10)	HHHHHHHHHH	
2	FILE_TYPE	X(3)	RA1/RA2/RA3 - Result File for Application (CASH/CPFIS/SRS)	RA1 – CASH RA3 - SRS
3	PRI_DLR_CODE	9(4)	7171	Numeric bank code
4	BANK_NAME	X(20)		
5	PROCESSING_DATE	9(8)	YYYYMMDD	Date of auction
Detail Record- Header				
6	RECORD_TYPE	X(10)	BBBBBBBBBB	
7	ISSUE_CODE	X(8)	NX00100H	
8	ISIN_CODE	X(12)		
9	TENDER_DATE	9(8)	YYYYMMDD	Auction date
10	CURR	X(3)	SGD	
11	ISSUE_DESC	X(30)		
12	CUTOFF_YIELD_SIGN	X(1)	+ or -	either '+' or '-'. If bid yield is 0 would be a '+' sign
13	CUTOFF_YIELD	9(3)v9(2)	00305 00205 00000	3.05 2.05 0
14	CUTOFF_PRICE	9(3)v9(4)	1022230	102.223
Detail Record – Detail (repetitive)				
15	RECORD_TYPE	X(10)	DDDDDDDDDD	
16	ISSUE_CODE	X(8)	NX00100H	
17	TRANS_REF	X(8)	Reference No	
18	NAME_OF_APPLN	X(30)		
19	NATIONALITY	X(1)		
20	IC_PASSPORT	X(14)		
21	NOMINAL_AMT	9(11)	00500000000	
22	COMP_NOCOMP	X(1)		
23	BID_YIELD_SIGN	X(1)	+ or -	either '+' or '-'. If bid yield is 0 would be a '+' sign
24	BID_YIELD	9(3)v9(2)	00305 00205 00000	3.05 2.05 0

25	ALLOTTED_AMT	9(11)	00500000000	
26	SETTLEMENT_AMT	9(13)v9(2)		
27	CUST_BANK_CODE	9(4)		Settlement bank code
28	CUST_BANK_BC	X(1)		Bank or custodian
29	TYPE_OF_APPLN	X(3)		
30	CDP_ACCT_NO	X(16)		
31	SUB_METHOD	X(1)		'A', 'B', 'C', 'E' or 'M'
32	CPF_SRS_ACCT_NO	X(16)		
33	CHECK_SUM	9(14)		
Control Record				
34	RECORD_TYPE	X(10)	TTTTTTTTTT	
35	RECORD_COUNT	9(9)		
36	OVERALL_CHK_SUM	9(14)		

1. Field 1 to 5 will be on line 1.
2. Field 6 to 14 will be for each issue, and be on the same line.
3. Field 15 to 33 will be repetitive for each result record.
4. Each result record must start on a new line.
5. Field 34 to 36 will be the last line.
6. The unused area will be padded with leading zeroes or spaces accordingly.
7. '9' denotes field having a numeric value(s).
8. 'X' denotes field having alphanumeric value(s).
9. The generated file will be PGP encrypted before sending to banks.

5.5.5 Field Definitions

1. RECORD_TYPE
   a. Record type for header record is always 'HHHHHHHHHH'.

2. FILE_TYPE
   a. File type for this file.
   b. RA1 – CASH, RA2 – CPFIS, RA3 - SRS

3. PRI_DLR_CODE
   a. Primary dealer code. 4 digits.

4. BANK_NAME
   a. Bank's name.

5. PROCESSING_DATE
   a. Date of the auction. The format is YYYYMMDD.

6. RECORD_TYPE
   a. Record type for the header of detail record is always 'BBBBBBBBBB'.

7. ISSUE_CODE
   a. Example of the code: NX00100H.

8. ISIN_CODE
    a. ISIN code for the issue, which is 12 alphanumeric characters.

9. TENDER_DATE
    a. Should be 8 digits. The format is YYYYMMDD. This is also known as auction date.

10. CURR
    a. The value will always be 'SGD'.

11. ISSUE_DESC

    a. The description of the issue.

12. CUTOFF_YIELD_SIGN

    a. This refers to the sign for cutoff yield. It can be "+" or "-". For zero yield, it will be "+".

13. CUTOFF_YIELD
    a. The cutoff yield of the auction. The value in this field is that the actual cutoff yield multiplies 100, e.g. 00305 for actual cutoff yield 3.05.

14. CUTOFF_PRICE
    a. The cutoff price of the auction. The value in this field is that the actual cutoff price multiplies 10,000, e.g. 1022230 for actual cutoff price 102.223.

15. RECORD_TYPE
    a. Record type for detail of detail record is always 'DDDDDDDDDD'.

16. ISSUE_CODE
    a. Example of the code: NX00100H.

17. TRANS_REF
    a. Bank's transaction reference number, which is alphanumeric and must be unique, e.g. 00000001.

18. NAME_OF_APPLN
    a. Name of the applicant.

19. NATIONALITY
    a. 'S' for Singapore citizen and PR. 'F' for a foreigner.

20. IC_PASSPORT
    a. IC or passport number.

21. NOMINAL_AMT
    a. The applied amount of this application. Unused areas will be padded with leading zeros.

22. COMP_NOCOMP
    a. 'C' for competitive bid. 'N' for non-competitive bid.

23. BID_YIELD_SIGN
    a. This refers to the sign for bid yield. It can be "+" or "-". For zero yield, it will be "+".

24. BID_YIELD
    a. The bid yield for this application. '00000' for non-competitive bid. The value in this field is that the actual bid yield multiplies 100, e.g. 00305 for actual yield 3.05.

25. ALLOTTED_AMT
    a. The allotted amount of an application. Unused areas will be padded with leading zeros.

26. SETTLEMENT_AMT
    a. The settlement amount for the application. The value in the result file equals the allotted amount multiplies cutoff price. (However, the actual settlement amount equals the amount in the result file divided by 100.)

27. CUST_BANK_CODE
    a. Settlement bank code. It should be same as primary dealer code for individual applicants.

28. CUST_BANK_BC
    a. Bank or custodian indicator. 'B' or 'C'. It should always be 'C' for individual applicants.

29. TYPE_OF_APPLN
    a. Type of applicant. It should always be 'IND' for an individual applicant.

30. CDP_ACCT_NO
    a. Applicant's CDP account no.

31. SUB_METHOD
    a. Submission method indicator. 'A' for ATM applications, 'B' for iBanking applications and 'C' for mobile applications. 'E' for applications which banks submit to SGS eApps via web application form. 'M' for paper form submission under contingency situations.

32. CPF_SRS_ACCT_NO
    a. Applicant's CPF or SRS account no.

33. CHECKSUM
    a. The sum of all numeric fields of this application, i.e. NOMINAL_AMT, BID_YIELD, ALLOTTED_AMT, SETTLEMENT_AMT and CUST_BANK_CODE. BID_YIELD uses the integer value in field 24 (to ignore the sign field 23).

34. RECORD_TYPE
    a. Record type for control record is always 'TTTTTTTTTT'.

35. RECORD_COUNT
    a. A total number of applications.

36. OVERALL_CHK_SUM
    a. The total sum of each record's checksum (field 32). If the total sum is bigger than 99,999,999,999,999, use the last 14 digits of the number.

5.6 The format of issuance calendar file

The filename format of the Issuance Calendar file

RT_MASSG.<PD Name>.ABA.ea_cal.txt

for example.

RT_MASSG.DBS.ABA.ea_cal.txt
RT_MASSG.OCBCSSB.ABA.ea_cal.txt
RT_MASSG.UOBSFTP.ABA.ea_cal.txt

The downstream filenames of the Issuance Calendar file that are transferred to MEFT-Access

EAPPSDBSZZ00RT_MASSG.DBS.ABA.ea_cal.txt
EAPPSOCBCZ00RT_MASSG.OCBCSSB.ABA.ea_cal.txt
EAPPSUOBZZ00RT_MASSG.UOBSFTP.ABA.acoeaicfbt.txt

MEFT-access Outgoing directories and their backup directories

Outgoing directories

/mas_apps/aba/data/MEFT/DBS/Outgoing
/mas_apps/aba/data/MEFT/OCBC/Outgoing
/mas_apps/aba/data/MEFT/UOB/Outgoing

Backup directories

/mas_apps/aba/data/MEFT/Backup/DBS/Outgoing
/mas_apps/aba/data/MEFT/Backup/OCBC/Outgoing
/mas_apps/aba/data/MEFT/Backup/UOB/Outgoing

5.6.1 Format

	Field Name	Length	e.g.	Comment
Header Record				
1	RECORD_TYPE	X(10)	HHHHHHHHHH	
2	FILE_TYPE	X(3)	CAL	
3	PROCESSING_DATE	9(8)	YYYYMMDD	Date of download
Detail Record – Detail (repetitive)				
4	RECORD_TYPE	X(10)	DDDDDDDDDD	
5	TBILL_BOND_IND	X(1)		T or B
6	ISSUE_CODE	X(8)	NX00100H	
7	ISIN_CODE	X(12)		
8	TENOR	9(3)		
9	TENOR_MEASURE	X(1)		D or Y
10	NEW_REOPEN_IND	X(1)		N or R
11	CURR	X(3)	SGD	
12	ANNOUNCE_DATE	9(8)	YYYYMMDD	Announcement date
13	ANNOUNCE_TIME	9(4)	HHMM	Usually 1630
14	AUCTION_DATE	9(8)	YYYYMMDD	Auction date
15	AUCTION_TIME	9(4)	HHMM	Usually 1200
16	ISSUE_DATE	9(8)	YYYYMMDD	Issuance date
17	CHECK_SUM	9(14)		
Control Record				
18	RECORD_TYPE	X(10)	TTTTTTTTTT	
19	RECORD_COUNT	9(9)		
20	OVERALL_CHK_SUM	9(14)		

10. Field 1 to 3 will be on line 1.
11. Field 4 to 17 will be repetitive for each result record.
12. Each record must start on a new line.
13. Field 18 to 20 will be the last line.
14. Unused area will be padded with leading zeroes or spaces accordingly.
15. '9' denotes field having numeric value(s).
16. 'X' denotes field having alphanumeric value(s).

5.6.2 Field Definitions

1. RECORD_TYPE
    a. Record type for header record is always 'HHHHHHHHHHH'.

2. FILE_TYPE
    a. File type for this file. Always is 'CAL' for issuance calendar.

3. PROCESSING_DATE
    a. Date when banks download the file. The format is YYYYMMDD.

4. RECORD_TYPE
    a. Record type for header of detail record is always 'DDDDDDDDDD'.

5. TBILL_BOND_IND
    a. T-bill or bond indicator. 'T' for T-bill, and 'B' for bond.

6. ISSUE_CODE
    a. Example of the code: NX00100H.

7. ISIN_CODE
    a. ISIN code for the issue, which is 12 alphanumeric characters.

8. TENOR
    a. Tenor of the issue. It's the number of days for T-bill, and number of years for bond.

9. TENOR_MEASURE
    a. 'D' for days, which is for T-bill. 'Y' for years, which is for bond.

10. NEW_REOPEN_IND
    a. New issue or reopen old issue indicator. 'N' for new issue. 'R' for reopen.

11. CURR
    a. Value will always be 'SGD'.

12. ANNOUNCE_DATE
    a. The date that the issue is open for subscription. The format is YYYYMMDD. This is usually same as the issue's announcement date.

13. ANNOUNCE_TIME

a. The time that the issue is open for subscription on OPEN_DATE. The format is HHMM. The value is usually '1630'.

14. AUCTION_DATE
    a. The date that the issue is closed for subscription. The format is YYYYMMDD. This is usually same as the issue's auction date.

15. AUCTION_TIME
    a. The time that the issue is closed for subscription on CLOSE_DATE. The format is HHMM. The value is usually '1200'.

16. ISSUE_DATE
    a. The date that the issue is issued. The format is YYYYMMDD. This is usually same as the issue's issuance date.

17. CHECKSUM
    a. The sum of all numeric fields of this application, i.e. TENOR, OPEN_DATE, OPEN_TIME, CLOSE_DATE, CLOSE_TIME, and ISSUE_DATE.

18. RECORD_TYPE
    a. Record type for control record is always 'TTTTTTTTTT'.

19. RECORD_COUNT
    a. Total number of issues.

20. OVERALL_CHK_SUM
    a. The total sum of each record's checksum (field 16). If the total sum is bigger than 99,999,999,999,999, use the last 14 digits of the number.

5.7 Closing Price file for MEPS+ and FMBS

The format of "closingprice.txt" and "closingprice_fmbs.txt" are:

Field Name	Length	Comments
Submission Date	8	From ABA0017_SUBMISSION_DATE, in yyyymmdd format
Security ISIN Code	12	From ABA0001_ISIN_CODE
Security Code	8	From ABA0017_SECURITY_CODE
MLA Price	8	For T-Bill/MAS-Bill: From ABA0017_YIELD For Bond/FRN: From ABA0017_MLA_PRICE Examples: 99.938 => 09993800 100.101 => 10010100
T1 Dirty Price	8	From ABA0017_T1_DIRTY_PRICE Examples: 99.938 => 09993800 100.101 => 10010100

Yield	9	For T-Bill/MAS-Bill: From ABA0017_AVE_PRICE For Bond/FRN: From ABA0017_YIELD Include sign, "+" for positive numbers and "-" for negative values. Examples: -3.200 => -00320000 0.000 => +00000000 3.200 => +00320000

The fields are delimited by ";"
Fields ABA0001_XXX are taken from ABA0001_SECURITY_MASTER table.
Fields ABA0017_XXX are taken from ABA0017_FINAL_DAILY_PRICE table.

```
Examples:
20140113;SGUATJAN1422;N214001T;09940000;09940000;+00032000
20140113;SGUATJAN1401;MX13502T;09988300;09988400;+00030000
20140113;SGUATJAN1402;BS14002T;09985000;09985000;+00030000
```

5.8 Closing Price file for AMMO

The format of "closingprice_ammo.txt" are:

Field Name	Length	Comments
Submission Date	8	From ABA0017_SUBMISSION_DATE, in yyyymmdd format
Security ISIN Code	12	From ABA0001_ISIN_CODE
Security Code	8	From ABA0017_SECURITY_CODE
MLA Price	8	For T-Bill/MAS-Bill: From ABA0017_YIELD For Bond/FRN: From ABA0017_MLA_PRICE Examples: 99.938 => 09993800 100.101 => 10010100
T1 Dirty Price	8	From ABA0017_T1_DIRTY_PRICE Examples: 99.938 => 09993800 100.101 => 10010100
Yield	9	For T-Bill/MAS-Bill: From ABA0017_AVE_PRICE For Bond/FRN: From ABA0017_YIELD Include sign, "+" for positive numbers and "-" for negative values. Examples: -3.200 => -00320000 0.000 => +00000000 3.200 => +00320000
Clean Price	8	For T-Bill/MAS-Bill:

		From ABA0017_YIELD For Bond/FRN: From ABA0017_AVE_PRICE Examples: 99.938 => 09993800 100.101 => 10010100

The fields are delimited by ";"
Fields ABA0001_XXX are taken from ABA0001_SECURITY_MASTER table.
Fields ABA0017_XXX are taken from ABA0017_FINAL_DAILY_PRICE table.

This file should contain only the issue codes which are issued. (i.e., aba0001_issue_date <= SYSDATE)

This file to be generated on the PCSSYS90/dcssys2 for the previous working day closing price details and to be sent via SFTP to AMMO system. (i.e., aba0017_submission_date = (select max(aba0017_submission_date) from aba0017_final_daily_price)

```
Examples:
20140113;SGUATJAN1422;N214001T;09940000;09940000;+00032000;09930000
20140113;SGUATJAN1401;MX13502T;09988300;09988400;+00030000;09988300
20140113;SGUATJAN1402;BS14002T;09985000;09985000;+00030000;09985000
```

5.9 File format of application data Acknowledgement file from bank upload file

5.9.1 Format

	Field Name	Length	e.g.	Comment
Header Record				
1	RECORD_TYPE	X(10)	HHHHHHHHHH	
2	FILE_TYPE	X(3)	AA1 – Application for CASH, AA2 – Application for CPFIS, AA3 – Application for SRS	AA1/AA2/AA3 – Application (CASH/CPFIS/SRS) File type provided from application data file
3	PRI_DLR_CODE	9(4)	7171	Numeric bank code provided from application data file
4	BANK_NAME	X(20)		Bank_Name provided from application data file
5	PROCESSING_DATE	9(8)	YYYYMMDD	Date of processing
6	PROCESSING_TIME	9(6)	HHMMSS	Time of processing
7	FILE_STATUS	X(1)	Status: 'P'ass or 'F'ail	Status of processing
8	FILE_ERROR_DESC	X(20)		File level error description
Detail Record- Header				
9	RECORD_TYPE	X(10)	BBBBBBBBBB	
10	ISSUE_CODE	X(8)	NX00100H	Issue_Code provided from application data file
11	ISIN_CODE	X(12)		ISIN_CODE provided from application data file

12	TENDER_DATE	9(8)	YYYYMMDD	Date provided from application data file
13	CURR	X(3)	SGD	
14	ISSUE_DESC	X(30)		
Detail Record – Detail (repetitive)				
15	RECORD_TYPE	X(10)	DDDDDDDDDD	
16	ISSUE_CODE	X(8)	NX00100H	Issue_Code provided from application data file
17	TRANS_REF	X(8)	Reference No	Trans_Ref provided from application data file
18	RECEIVED_DATE	9(8)	YYYYMMDD	YYYYMMDD as in record submitted
19	RECEIVED_TIME	9(6)	HHMMSS	HHMMSS as in record submitted
20	TRANS_TYPE	X(3)	TE_	
21	NOMINAL_AMT	9(11)	00500000000	Quantity (Value as in record submitted)
22	COMP_NOCOMP	X(1)		provided from application data file
23	BID_YIELD_SIGN	X(1)	+ or -	provided from application data file
24	BID_YIELD	9(3)v9(2)	00305 00205 00000	provided from application data file
25	NAME_OF_APPLN	X(30)		provided from application data file
26	NATIONALITY	X(1)		provided from application data file
27	IC_PASSPORT	X(14)		provided from application data file
28	CUST_BANK_CODE	9(4)		Settlement bank code
29	CUST_BANK_BC	X(1)		Bank or custodian
30	TYPE_OF_APPLN	X(3)		provided from application data file
31	REC_STATUS	X(1)		provided from application data file
32	CDP_ACCT_NO	X(16)		provided from application data file
33	SUB_METHOD	X(1)	‘A’, ‘B’, ‘C’	provided from application data file
34	SRS_ACCT_NO	X(16)	Mandatory field for SRS	provided from application data file
35	REC_STATUS	X(1)	Status of processing at record level: ‘P’ or ‘F’	Status of processing at record level: ‘P’ or ‘F’
36	REC_ERROR_DESC	X(20)	Record Error description	Record Error description
37	CHECK_SUM	9(14)	Computed record check sum value	Computed record check sum value
Control Record				
38	RECORD_TYPE	X(10)	TTTTTTTTTT	
39	RECORD_COUNT	9(9)		
40	OVERALL_CHK_SUM	9(14)		

1. Field 1 to 8 will be on line 1.
2. Field 9 to 14 will be for each issue, and be on the same line.
3. Field 15 to 37 will be repetitive for each application record.
4. Each application record must start on a new line.
5. Field 38 to 40 will be the last line.
6. Unused area will be padded with leading zeroes or spaces accordingly.
7. '9' denotes field having numeric value(s).
8. 'X' denotes field having alphanumeric value(s).
9. The generate file will be PGP encrypted before sending to the banks.

5.9.2 Field Definitions

1) RECORD_TYPE
   a. Record type for header record is always 'HHHHHHHHHH'.

2) FILE_TYPE
   a) AP1 – Application for CASH,
   b) AP2 – Application for CPFIS,
   c) AP3 – Application for SRS

3) PRI_DLR_CODE
   a. Primary dealer code. 4 digits.

4) BANK_NAME
   a. Bank's name.

5) PROCESSING_DATE
   a. Date of processing. The format is YYYYMMDD.

6) PROCESSING_TIME
   a. Time of processing. The format is HHMMSS.

7) FILE_STATUS
   a. File processing status. 'P' for pass or 'F' for fail. If the entire file failed processing, all the records in the file will be rejected.

8) FILE_ERROR_DESC
   a. File processing error description.

9) RECORD_TYPE
   a. Record type for header of detail record is always 'BBBBBBBBBB'.

10) ISSUE_CODE
   a. Example of the code: NX00100H.

11) ISIN_CODE
   a. ISIN code for the issue, which is 12 alphanumeric characters.

12) TENDER_DATE

a. Should be 8 digits. The format is YYYYMMDD. This is also known as auction date.

13) CURR
   a. Value will always be 'SGD'.

14) ISSUE_DESC
   a. The description of the issue.

15) RECORD_TYPE
   a. Record type for detail of detail record is always 'DDDDDDDDDD'.

16) ISSUE_CODE
   a. Example of the code: NX00100H.

17) TRANS_REF
   a. Bank's transaction reference number, which is alphanumeric and must be unique, e.g. 00000001.

18) RECEIVED_DATE
   a. The date that ATM/iBanking receives the transaction. It should be 8 digits. The format is YYYYMMDD.

19) RECEIVED_TIME
   a. The time that ATM/iBanking receives the transaction. It should be 6 digits. The format is HHMMSS.

20) TRANS_TYPE
   a. Always be TE_.

21) NOMINAL_AMT
   a. The applied amount of this application. Unused areas will be padded with leading zeroes.

22) COMP_NOCOMP
   a. 'C' for competitive bid. 'N' for non-competitive bid.

23) BID_YIELD_SIGN
   a. This refers to the sign for bid yield. It can be "+" or "-". For zero yield, it will be "+".

24) BID_YIELD
   a. The bid yield for this application. '00000' for non-competitive bid. The value in this field is that the actual bid yield multiplies 100, e.g. 00305 for actual yield 3.05.

25) NAME_OF_APPLN
   a. Name of the applicant.

26) NATIONALITY
   a. 'S' for Singapore citizen and PR. 'F' for foreigner.

27) IC_PASSPORT
   a. IC or passport number. Singapore IC[4] no. will be validated. Records that fail in validation will be discarded.

28) CUST_BANK_CODE

[4] Both T Series and S Series NRIC numbers will be validated

a. Settlement bank code. It should be same as primary dealer code for individual applicants.

29) CUST_BANK_BC
a. Bank or custodian indicator. 'B' or 'C'. It should always be 'C' for individual applicants.

30) TYPE_OF_APPLN
a. Type of applicant. It should always be 'IND' for individual applicant.

31) REC_STATUS
a. Always is space.

32) CDP_ACC_NO
a. Applicant's CDP account no. SGS eApps will not validate this field.

33) SUB_METHOD
a. Submission method indicator. 'A' for ATM applications, 'B' for iBanking applications, and 'C' for mobile app applications.

34) SRS ACCT_NO
a. This field is used for SRS Account number.

35) REC_STATUS
a. Record processing status. 'P' for pass or 'F' for fail. If the entire file failed processing, all the records in the file will be rejected.
b. There will not be partial pass at record level. All applied amount will be rejected.

36) REC_ERROR_DESC
a. Record processing error description.

37) CHECKSUM
a. The sum of all numeric fields of this application, i.e. RECEIVED_DATE, RECEIVED_TIME, NOMINAL_AMT, BID_YIELD, and CUST_BANK_CODE. BID_YIELD uses the integer value in field 21 (to ignore the sign field 20).

38) RECORD_TYPE
a. Record type for control record is always 'TTTTTTTTTT'.

39) RECORD_COUNT
a. Total number of applications.

40) OVERALL_CHK_SUM
a. The total sum of each record's checksum (field 31). If the total sum is bigger than 99,999,999,999,999, use the last 14 digits of the number.

Error messages displayed in UI for upload bids and eAuction Batch (Retail Bid file process) (file level/record level):

eApps_Batch_Error Messages.xlsx

6 Deployment

The deployment of SGS eApps assumes all necessary hardware, operating systems, web servers, database server, MAF Authentication server and all network connections are in place already.

The deployment involves deploying .ear file using WAS Admin Console, copying UNIX scheduled job files, and creating Oracle tables.

6.1 Deploy .ear file using WAS Admin Console

Developer uses RAD to generate .ear file, and passes it to WAS admin. WAS admin will do the deployment using WAS Admin Console.

6.1.1 Configuring interface with RSA

Change the below property value in the sgs.properties file to the following:
authentication.mechanism=RSA

Ensure that the following files exist in /mas_appsw/was8/sgs:

a. sample.properties
b. sdconf.rec
c. securid

Ensure that the following files exist in /var/ace/api:

d. sdconf.rec
e. secured

Ensure that the following classes exist in /was/AppServer/profiles/AppSrv01/installedApps/PRDSYSCell01/SGSeApps.ear/SGSWeb.war/WEB-INF/classes/sgseapps:

f. SGSAuthFactory.class
g. SGSAuthIntf.class
h. SGSSWAuthImpl.class
i. SGSRSAAuthImpl.class
j. SGSNoAuthImpl.class

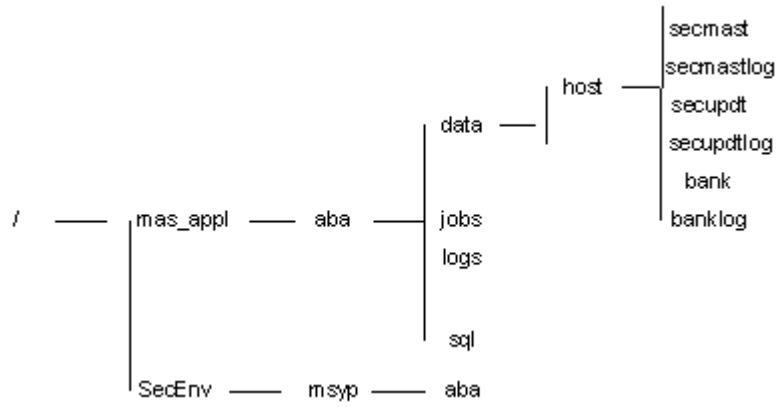
Ensure that the login to application using the RSA token is successful.

6.2 Copying UNIX scheduled job and Oracle job files to PRDSYS

6.2.1 Prerequisites:

- Oracle client
- Configure Oracle connection string
- Java run-time environment

6.2.2 Directory structure on PRDSYS:



The project id "ms1ab" should have full access right to the directories "/mas_appl/aba/", "/SecEnv/msyp/aba/" and all their sub-directories. The project group "ms1aba" should have read and write access to the data directory "/mas_appl/aba/data/".

6.2.3 Copy files

The following files should be copied to directory "/mas_appl/aba/jobs":

- oracleprofile.job – the program to set Oracle parameters
- secmast.job – the job to load secmast.txt.
- secint.job – the job to load secint.txt.
- bank.job – the job to load bank.txt.
- ClosingPrice.job – the job to generate closingprice.txt.

The following files should be copied to directory "/mas_appl/aba/sql":

- bank.ctl – control file for bank list loading
- security.ctl – control file for new security loading
- delete.sql – SQL script to delete old records in the staging table
- secint.ctl – control file for loading security update into staging table
- secint.sql – SQL script for loading security update

6.2.4 Scramble Oracle id and password file

Oracle DBA should give security administrator Oracle id, password and Oracle connection string in the format of "id/password@ConnectionString". Then security administrator will create a scramble file named "abascramble.txt"

and put it under "/SecEnv/msyp/aba". The program used to unscramble it should be put under directory "/SecEnv/msyp".

The following file should be put under directory "/SecEnv/msyp":
- msypdpwd – program to unscramble file "abascramble.txt"

The following file should be put under directory "/proc/msyp/aba":
- abascramble.txt – the scramble file

6.2.5 UNIX run-time setting

UNIX system administrator should create a file named "Set_AppEnv" and put the file under directory "/AppEnv". The file specifies the working directory for SGS eApps jobs.

The following file should be put under directory "/AppEnv":
- Set_AppEnv – program to set working directory

6.2.6 Schedule jobs

After copying the above files, AutoSys system administrator should schedule the job using AutoSys admin console.

Note:

Because the working directories for development and production environment are different, the paths in the following file need to be changed before being copied to production environment.
- oracleprofile.job
- bank.ctl
- secint.ctl
- security.ctl

6.3 Creating Oracle tables

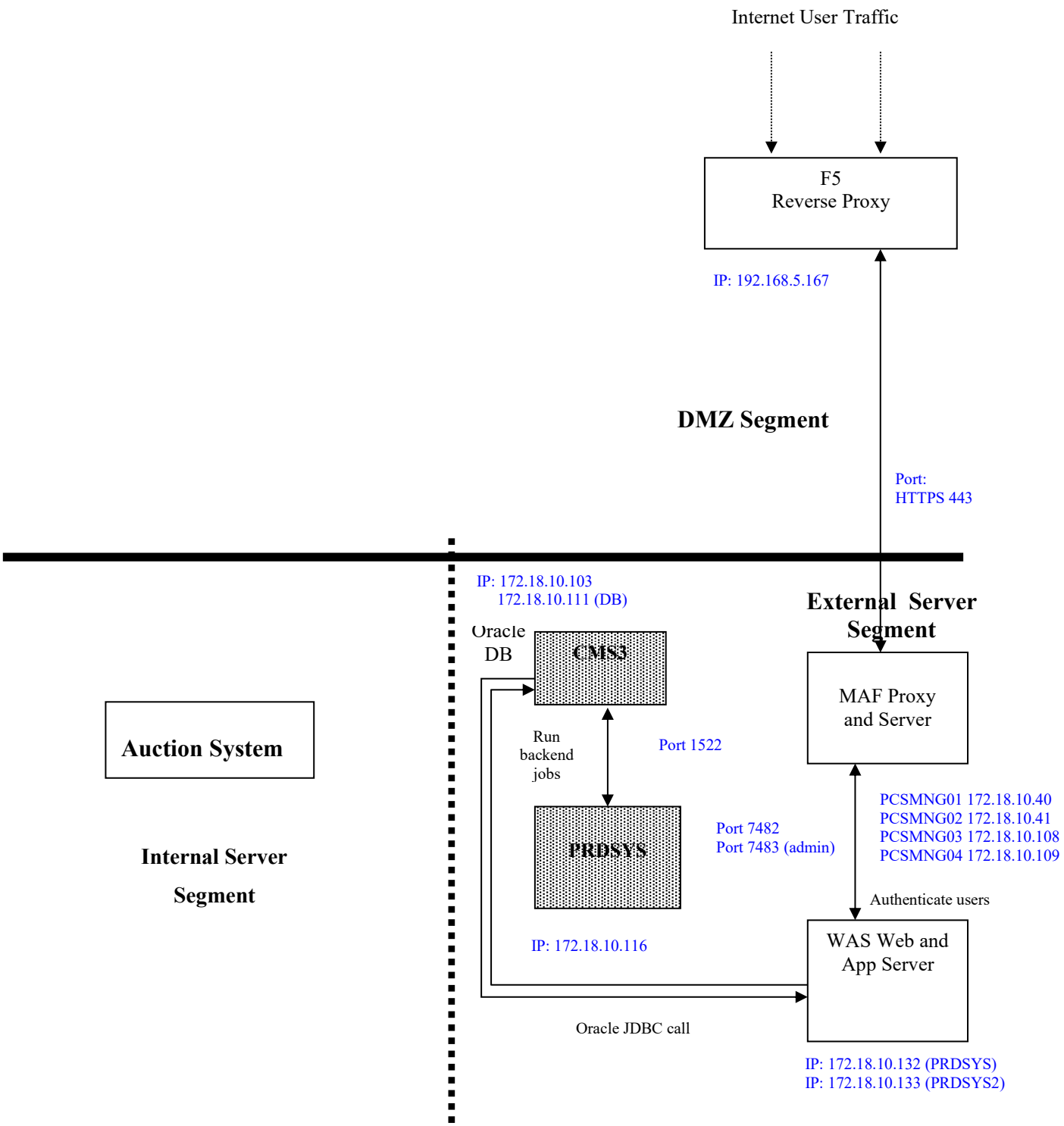
Before creating Oracle tables, Oracle DBA should create necessary user ids and assign table space.

It involves the following steps:

- Run the DDL script file “CreateTable.sql” to create tables.
- Run a SQL script file “grant.sql” to grant access right to WAS users.

7 Appendix

7.1 eTender Networked Components



7.2 NRIC No. Validation Algorithm

Formula for validation of NRCI (for Singaporeans)

	IC1	IC2	IC3	IC4	IC5	IC6	IC7	CHK
NRIC								

Weight	2	7	6	5	4	3	2

	CCK 1	CCK 2	CCK 3	CCK 4	CCK 5	CCK 6	CCK 7	CCK 8	CCK 9	CCK 10	CCK 11
Check Digit Table (S series)	A	B	C	D	E	F	G	H	I	Z	J
Check Digit Table (T series)	H	I	Z	J	A	B	C	D	E	F	G

Calculate:-

NRIC- Total = (IC1 * 2) + (IC2 * 7) + (IC3 * 6) + (IC4 * 5) + (IC5 * 4) + (IC6 * 3) + (IC7 * 2)

Remainder = #NRIC-Total mod 11

i = 11 – #Remainder

*** For a valid NRIC, CKi = CHK

where CKi = (Respective location of the alphabet in the Check Digit Table)

CHK = Last digit in NRIC

Employee NRIC No. Check Digit Routine

Let the NRIC number be 1234567_

i) Multiply the numeric digit by the corresponding factors (Weight) –

2, 7, 6, 5, 4, 3, 2,

Hence,

1	2	3	4	5	6	7
x	x	x	x	x	x	x
2	7	6	5	4	3	2
2	**14**	**18**	**20**	**20**	**18**	**14**

ii) Compute the total of (1)

Total = 2 + 14 + 18 + 20 + 20 + 18 + 14
= 106

iii) Divide total by 11

106 / 11 = 9 remainder 7

iv) Subtract Remainder from 11

11 – 7 = 4

v) The valid check digit should be as follows –

Remainder	Valid Check Digit (S Series)	Valid Check Digit (T Series)	Valid Check Digit (G Series)
1	A	H	K
2	B	I	L
3	C	Z	M
4	D	J	N
5	E	A	P
6	F	B	Q
7	G	C	R
8	H	D	T
9	I	E	U
10	Z	F	W
11	J	G	X

Hence, the valid check digit should be "D" (S series), "J" (T Series), "N" (G series)
And the valid NRIC number would be **S1234567D** or **T1234567J** or **G1234567N**

Formula for validation of NRIC (for Singaporeans) (obtained from payroll system)

	IC1	IC2	IC3	IC4	IC5	IC6	IC7	CHK
NRIC								

	CK1	CK2	CK3	CK4	CK5	CK6	CK7	CK8	CK9	CK10	CK11
Check Digit Table	A	B	C	D	E	F	G	H	I	Z	J

Calculate :-

#NRIC-total = (IC1 * 2) + (IC2 * 7) + (IC3 * 6) + (IC4 * 5) + (IC5 * 4) + (IC6 * 3) + (IC7 * 2)

#Remainder = Mod (#NRIC-total ÷ 11)

i = 11 - #Remainder

*** For a valid NRIC, CKi = CHK

where CKi = (respective location of the alphabet in the Check Digit Table)

CHK = Last digit in NRIC

```
EMPLOYEE NRIC NO. CHECK DIGIT ROUTINE
=====================================

LET THE NRIC NO. BE 1234567_

1) MULTIPLY THE NUMERIC DIGIT BY THE CORRESPONDING FACTORS -

            2, 7, 6, 5, 4, 3, 2,

   HENCE,  1   2   3   4   5   6   7
           *   *   *   *   *   *   *
           2   7   6   5   4   3   2
          -------------------------
           2  14  18  20  20  18  14
          -------------------------

2) COMPUTE THE TOTAL OF (1)

   TOTAL = 2 + 14 + 18 + 20 + 20 + 18 + 14

         = 106

3) DIVIDE TOTAL BY 11

   106 / 11 = 9 REMAINDER 7

4) SUBTRACT REMAINDER FROM  11

   11 - 7 = 4

5) THE VALID CHECK DIGIT SHOULD BE AS FOLLOWS -

   REMAINDER     VALID CHECK DIGIT
   ---------     -----------------
       1                A
       2                B
       3                C
       4                D
       5                E
       6                F
       7                G
       8                H
       9                I
      10                Z
      11                J

   HENCE, THE VALID CHECK DIGIT SHOULD BE "D"
          AND THE VALID NRIC NO. WOULD BE 1234567D
```

7.3 CPF No. Validation Algorithm

CONFIDENTIAL B

Compute check digit of CPF non-numeric a/c

Input Format

1st	2nd	3rd	4th	5th	6th	7th	8th	9th
S	X1	n	n	n	n	n	n	X2
T	X1	n	n	n	n	n	n	X2

Valid values :
1st char is either 'S' or 'T'.
2nd char is an alpha character.
3rd to 8th char is numeric character.
9th char is an alpha character.

Formula

1. Mulitply 3rd to 8th char with 7, 6, 5, 4, 3, 2 respectively
2. Sum the result of each mulitplication
3. **If 1st char is 'T', add 4 to the sum**
4. Add a weightage to the computed total. Please refer to Table 1.
5. Divide by 11 to obtain the remainder
6. Subtract the remainder from 11
   If the final result is 11, consider it as 0.
7. Check the final result(@) against the table found in Table 2 for 2nd char = 'A', 'B', 'C' or 'Y' and Table 3 for 2nd char ='D' or 'F' to determine the alpha character check digit for the 9th char.

Table 1 : Weightage

2nd Char value	Weightage
A	-1
B	1
C	4
D	0
F	0
Y	3

Table 2 : Reference Table for 9th Char value if 2nd char is = 'A', 'B', 'C' or 'Y'

@	9th char value
0	J
1	A
2	B
3	C
4	D
5	E
6	F
7	G
8	H

9	I
10	Z

Table 3 : Reference Table for 9th Char value if 2nd char is = 'D' or 'F'

@	9th char value
0	A
1	B
2	C
3	E
4	J
5	K
6	L
7	N
8	P
9	X
10	V

Examples

1a. 1st char is 'S' and 2nd char is 'A'

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th
	S	A	0	8	4	1	0	9	???
Multiply			7	6	5	4	3	2	
Sum			0	48	20	4	0	18	= 90
Add weightage (2nd char = 'A')									-1
Total									**89**
Divide by 11 to get Remainder									1
Final Result (11 - Remainder)								**= 11 - 1 = 10**	
Based on Table 2, 9th Char								= Z	

Result : **SA084109Z**

1b. 1st char is 'T' and 2nd char is 'A'

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th
	T	A	0	8	4	1	0	9	???
Multiply			7	6	5	4	3	2	
Sum			0	48	20	4	0	18	= 90

Add weightage 4 (1st char = 'T')	4
Add weightage (2nd char = 'A')	-1
Total	**93**
Divide by 11 to get Remainder	5
Final Result (11 - Remainder)	**= 11 - 5 = 6**
Based on Table 2, 9th Char	= F

Result :**TA084109F**

2A. 1st char is 'S' and 2nd char is with 'B'

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th
	S	B	0	0	0	0	0	1	???
Multiply			7	6	5	4	3	2	
Sum			0	0	0	0	0	2	= 2

Add weightage (2nd char = 'B')	1
Total	**3**
Divide by 11 to get Remainder	3
Final Result (11 - Remainder)	**= 11 - 3 = 8**
Based on Table 2, 9th Char	= H

Result : **SB000001H**

2B. 1st char is 'T' and 2nd char is with 'B'

	1st	2nd	3rd	4th	5th	6th	7th	8th		9th
	T	B	0	0	0	0	0	1		???
Multiply			7	6	5	4	3	2		
Sum			0	0	0	0	0	2	=	2

Add weightage 4 (1st char = 'T') 4

Add weightage (2nd char = 'B') 1

Total **7**

Divide by 11 to get Remainder 7

Final Result (11 - Remainder) **= 11 - 7**
= 4

Based on Table 2, 9th Char = D

Result : **TB000001D**

3A. 1st char is 'S' and 2nd char is with 'D'

	1st	2nd	3rd	4th	5th	6th	7th	8th		9th
	S	D	0	0	0	6	1	8		???
Multiply			7	6	5	4	3	2		
Sum			0	0	0	24	3	16	=	43

Add weightage (2nd char = 'D') 0

Total **43**

Divide by 11 to get Remainder 10

Final Result (11 - Remainder) **= 11 - 10**
= 1

Based on Table 2, 9th Char = B

Result : **SD000618B**

3B. 1st char is 'T' and 2nd char is with 'D'

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th
	S	D	0	0	0	6	1	8	???
Multiply			7	6	5	4	3	2	
Sum			0	0	0	24	3	16	= 43

Add weightage 4 (1st char = 'T')	4
Add weightage (2nd char = 'D')	0
Total	**47**
Divide by 11 to get Remainder	3
Final Result (11 - Remainder)	**= 11 - 3 = 8**
Based on Table 2, 9th Char	= P

Result : **TD000618P**

7.4 Bid Checksum Algorithm

- Algorithm for servlet SGSSubResult to calculate checksum:

Checksum	=	bid amount 1 + yield 1
	+	bid amount 2 + yield 2
	+	bid amount 3 + yield 3
	+	bid amount 4 + yield 4
	+	bid amount 5 + yield 5
	+	PD code + bank code
	+	transaction reference no.

7.5 Warning messages of backend scheduled jobs for operators

- APPLSGSE DECRYPT PASSWORD NOT PROPERLY EXECUTED for loading new security
- APPLSGSE Error occurred for SGS eApps loading of new security
- APPLSGSE DECRYPT PASSWORD NOT PROPERLY EXECUTED for loading security update
- APPLSGSE Error occurred for SGS eApps loading of security update
- APPLSGSE DECRYPT PASSWORD NOT PROPERLY EXECUTED for loading bank
- APPLSGSE Error occurred for SGS eApps loading of bank

7.6 Formulae used in SGS Daily Price calculation

7.6.1 Formulae for T-Bill

The formulae for price and yield calculations are as follows : -

$$\mathbf{D} = \frac{M}{365} \times R$$

$$\mathbf{P} = S\$100 - \left[\frac{M}{365} \times R \right]$$

Where

M	=	Days to maturity **(Settlement date* to maturity date)**
R	=	Annual rate of discount (yield), expressed as a percentage
D	=	Full discount per S$100 face value
P	=	Dollar price per S$100 face value

*** For New Issue of T-Bill, if settlement date is before issue date, then Days to maturity = Issue date to maturity date**

7.6.2 Formulae for Bond

7.6.2.1 Irregular Interest Period for New Issue

Normal

$$P = \left\{ \frac{RV}{\left(1 + \frac{Y}{200}\right)^{N - 1 + \frac{DSC}{E}}} \right\} + \left(\left\{ \sum_{K=2}^{N} \frac{\frac{CPN}{2}}{\left(1 + \frac{Y}{200}\right)^{K - 1 + \frac{DSC}{E}}} \right\} + \frac{FC}{\left(1 + \frac{Y}{200}\right)^{\frac{DSC}{E}}} \right) - AI$$

$$- \quad AI$$

$$AI = \left(\frac{CPN}{2} \times \frac{DIS}{E} \right) \qquad FC = \left(\frac{CPN}{2} \times \frac{DIF}{E} \right)$$

Ex-Interest

$$P = \left\{ \frac{RV}{\left(1 + \frac{Y}{200}\right)^{N - 1 + \frac{DSC}{E}}} \right\} + \left\{ \sum_{K=2}^{N} \frac{\frac{CPN}{2}}{\left(1 + \frac{Y}{200}\right)^{K - 1 + \frac{DSC}{E}}} \right\} - AI$$

$$AI = - \left(\frac{CPN}{2} \times \frac{DSC}{E} \right)$$

where

FC	=	Amount of first interest (coupon) payment per S$100 face value
AI	=	Accrued interest per S$100 face value
DSC	=	Days between settlement date and first interest payment date
DIF	=	Days between issue date and first coupon date
DIS	=	Days between issue date and settlement date
E	=	Number of days in the first coupon period (6 months)
N	=	Number of semi-annual coupons payable between settlement date and maturity date
CPN	=	Annual coupon rate (as a percentage)
Y	=	Annual yield (as a percentage)
P	=	Dollar price per S$100 face value
RV	=	Redemption value

For New Issue of Bond only (not applicable to reopened issues)
If Settlement Date is before Issue date, then settlement date should be equal to issue date.

7.6.2.2 With more than six months to maturity

Normal

$$P = \left\{ \frac{RV}{\left(1 + \frac{Y}{200}\right)^{N - 1 + \frac{DSC}{E}}} \right\} + \left\{ \sum_{K=1}^{N} \frac{\frac{CPN}{2}}{\left(1 + \frac{Y}{200}\right)^{K - 1 + \frac{DSC}{E}}} \right\} - AI$$

$$AI = \left[\frac{CPN}{2} \times \frac{DCS}{E} \right]$$

Ex-Interest

$$P = \left\{ \frac{RV}{\left(1 + \frac{Y}{200}\right)^{N - 1 + \frac{DSC}{E}}} \right\} + \left\{ \sum_{K=2}^{N} \frac{\frac{CPN}{2}}{\left(1 + \frac{Y}{200}\right)^{K - 1 + \frac{DSC}{E}}} \right\} - AI$$

$$AI = - \left[\frac{CPN}{2} \times \frac{DSC}{E} \right]$$

Where

AI	= Accrued interest per S$100 face value
DSM	= Days between settlement date and maturity date
DCS	= Days between beginning of current coupon period and settlement date
DSC	= E-DCS = days from settlement date to next 6-month coupon date
E	= Number of days in coupon period where settlement occurs
N	= number of semi-annual coupons payable between settlement date and maturity date
CPN	= Annual coupon rate (as a percentage)
Y	= Annual yield (as a percentage)
P	= Dollar price per S$100 face value
RV	= Redemption value

7.6.2.3 With six months or less to maturity

Normal

$$\mathbf{P} = \left\{ \frac{100 \left\{ RV + \frac{CPN}{2} \right\}}{100 + \left\{ \frac{DSM}{E} \times \frac{Y}{2} \right\}} \right\} - \mathbf{AI}$$

Where

$$\mathbf{AI} = \left[\frac{CPN}{2} \times \frac{DCS}{E} \right]$$

Ex-Interest

$$\mathbf{P} = \left\{ \frac{100 \times RV}{100 + \left\{ \frac{DSM}{E} \times \frac{Y}{2} \right\}} \right\} - \mathbf{AI}$$

Where

$$\mathbf{AI} = - \left[\frac{CPN}{2} \times \frac{DSC}{E} \right]$$

Where

AI	= Accrued interest per S$100 face value
DSM	= Days between settlement date and maturity date
DCS	= Days between beginning of current coupon period and settlement date
DSC	= E-DCS = days from settlement date to next 6-month coupon date
E	= Number of days in coupon period where settlement occurs
N	= number of semi-annual coupons payable between settlement date and maturity date
CPN	= Annual coupon rate (as a percentage)
Y	= Annual yield (as a percentage)
P	= Dollar price per S$100 face value
RV	= Redemption value

7.7 Calculation of T+1, T+2, T+3 and Repo T+1 Prices, T+3 Prices

	Ave Price	High	Low	Dirty Price	MLA	Yield	T+1 Price (ILF)	Repo T+1 Price (MRF)	Repo T+2 Price	T+3 Price (USDF)
T-Bill / MAS Bills	Price	Highest bid yield	Lowest bid yield	= Price	= Price	Yield	Price on current day's next working day	Price on current day's Repo's next working day	Price on current day's Repo's next next working day	Price of Repo T+2 Price on next working day
Bond	Ave Price	High	Low	Ave+ AI on current day	= Dirty Price	Yield	Ave + AI up to current day's next working day	Ave + AI up to current day's Repo's next working day	Ave + AI up to current day's Repo's next next working day	Ave + AI up to current day's Repo T+2 Price on next working day

Definition:

T+1 Day: Next working day. Monday Friday, excluding SG public holiday
T+2 Day: Next Next working day. (i.e. Next working day of T+1 day) Monday Friday, excluding SG public holiday
Repo T+1 Day: Repo's next working day. Monday – Friday, excluding SG public holiday
Repo T+2 Day: Repo's next next working day. Repo T+1 Day's Repo T+1 Day
T+3 Day: Next working day of T+2 day. Monday Friday, excluding US and SG public holiday. USD Facility settles on T+3 from the current date, hence it cannot be a US or SG holiday on T+3.

e.g.: Current Day – 13th April 2020

	Scenario 1 No Public Holiday in April for both SG and US	Scenario 2 SG Public Holiday = 15th April 2020 No US Public Holiday	Scenario 3 No SG Public Holiday US Public Holiday = 15th April 2020	Scenario 4 No SG Public Holiday US Public Holiday = 15th and 16th April 2020	Scenario 5 SG Public Holiday = 16th April 2020 US Public Holiday = 15th April 2020 and 17th April 2020
T+1 Day	14th April 2020	14th April 2020	14th April 2020	14th April 2020	14th April 2020
T+2 Day	15th April 2020	16th April 2020	15th April 2020	15th April 2020	15th April 2020
T+3 Day	16th April 2020	17th April 2020	16th April 2020	17th April 2020	20th April 2020

Public Holiday information is input by MDD staff and stored in database.

AI: Accrued Interest
MLA price: MLA in the table
ILF price: T+1 price in the table

USDF price: T+3 price in the table

7.8 Others

As discussed, OPS will alert us (May Ling, Hui Ling) as well as FD and MDD users (FD: Jean Tsen, Nicholas, Goh Chung Hwa; MDD: Puey Siew Pheng) when any of the following production servers or services are down. In addition, the email for problem reporting will indicate that 'SGS eApps' will be affected.

Critical Servers/Services:
1. PRDSYS, PRDSYS2
2. MAF Authentication Servers
3. WAS App Server
4. WAS HTTP Server
5. MAS Web Server
6. CM5 and CM3 are both down
7. Internet Service provider (Starhub) for MAS

7.9 Data scramble/unscramble algorithm for bank files (this is no longer in used)

7.9.1 Requirement

To protect data integrity and security, Banks will scramble SGS application data file and upload the scrambled file to SGS eApps. MAS will scramble auction result file and send the scrambled file to banks.

In the original clear file, each line is basically a text string. When processing the file, the encryption program will encrypt the text string line by line. Likewise, when processing the encrypted file, the decryption program will decrypt the encrypted string line by line.

Banks should only use ASCII code for character to hexadecimal number and hexadecimal number to character conversion, i.e. for encryption, banks should convert characters to their ASCII codes equivalent; while for decryption, banks should convert ASCII codes to their characters equivalent. Banks should not use EBCDIC code in the conversion.

7.9.2 Encryption

7.9.2.1 Convert the Character String (Char) into its Hexadecimal (Hex) equivalent

Example.

Input Char String	a		Z		-		$		0		.		9	
Convert to Hex equivalent	6	1	5	A	2	D	2	4	3	0	2	E	3	9

7.9.2.2 Scramble the data using the following algorithm

Example.

	Offset														
Input Hex String & Initial offset = 7	7	6	1	5	A	2	D	2	4	3	0	2	E	3	9
Starting from the beginning of the input string, for each pair of chars,															
Convert the 1st Hex char to Integer value (e.g. 6 (Hex) = 6 (Dec))		6													
Add the integer value to the offset (e.g. 6 + 7 = 13)		13													
Mod the value by 16 (e.g. 13 % 16 = 13)		13													
Convert the integer value to Hex char (e.g. 13 (Dec) = d (Hex))		**d**													
Convert the 2nd Hex char to Integer value (e.g. 1 (Hex) = 1 (Dec))			1												
Add the integer value to the offset			8												

(e.g. 1 + 7 = 8)															
Mod the value by 16 (e.g. 8 % 16 = 8)			8												
Convert the integer value to Hex char (e.g. 8 (Dec) = 8 (Hex))			**8**												
Use the integer value as the new offset for the next pair of chars (e.g. offset = 8)	**8**														
Repeat above for the next pairs of chars, until the end of the input string				5											
				13											
				13											
				d											
					10										
					18										
					2										
					2										
	2														
						2									
						4									
						4									
						4									
							13								
							15								
							15								
							f								
	15														
								2							
								17							
								1							
								1							
									4						
									19						
									3						
									3						
	3														
										3					
										6					
										6					
										6					
											0				
											3				
											3				
											3				
	3														
												2			
												5			
												5			
												5			
													14		

													17		
													1		
													1		
	1														
														3	
														4	
														4	
														4	
															9
															10
															10
															a
	10														
Convert result string to upper case		**D**	**8**	**D**	**2**	**4**	**F**	**1**	**3**	**6**	**3**	**5**	**1**	**4**	**A**

Hence, an input string of "**aZ-$0.9**" will give an encrypted output string of "**D8D24F1363514A**".

7.9.3 Decryption

7.9.3.1 Unscramble the encrypted data using the following algorithm

Example.

	Offset														
Input encrypted Hex String & Initial offset = 7	**7**	**D**	**8**	**D**	**2**	**4**	**F**	**1**	**3**	**6**	**3**	**5**	**1**	**4**	**A**
Starting from the end of the input string, for each pair of chars,															
Convert the 2nd Hex char, from the pair in front, to Integer value (e.g. 1 (Hex) = 1 (Dec))													1		
Offset = integer value of the 2nd char from the pair in front (e.g. offset = 1)	**1**														
Convert the 2nd Hex char to Integer value (e.g. A (Hex) = 10 (Dec))															10
Add 16 if value is less than offset (e.g. 10 > 1 => do nothing)															-
Minus offset from value (e.g. 10 - 1 = 9)															9
Mod the value by 16 (e.g. 9 % 16 = 9)															9

Convert the integer value to Hex (e.g. 9 (Dec) = 9 (Hex))															**9**
Convert the 1st Hex char to Integer value (e.g. 4 (Hex) = 4 (Dec))														4	
Add 16 if value is less than offset (e.g. 4 > 1 => do nothing)														-	
Minus offset from value (e.g. 4 - 1 = 3)														3	
Mod the value by 16 (e.g. 3 % 16 = 3)														3	
Convert the integer value to Hex (e.g. 3 (Dec) = 3 (Hex))														**3**	
Repeat above for the next pairs of chars, until the end of the input string											3				
	3														
													1		
													17		
													14		
													14		
													e		
												5			
												-			
												2			
												2			
												2			
									3						
	3														
											3				
											-				
											0				
											0				
											0				
										6					
										-					
										3					
										3					
										3					
							15								
	15														
									3						
									19						
									4						
									4						
									4						
								1							
								17							
								2							
								2							

								2							
					2										
	2														
							15								
							-								
							13								
							13								
							d								
						4									
						-									
						2									
						2									
						2									
			8												
	8														
					2										
					18										
					10										
					10										
					a										
				13											
				-											
				5											
				5											
				5											
(Note: Use the initial offset for the 1st pair of chars)	**7**														
			8												
			-												
			1												
			1												
			1												
		13													
		-													
		6													
		6													
		6													
Convert result string to upper case		**6**	**1**	**5**	**A**	**2**	**D**	**2**	**4**	**3**	**0**	**2**	**E**	**3**	**9**

7.9.4 Convert the Hexadecimal (Hex) String into its Character equivalent

Example.

Input encrypted Hex string	**6**	**1**	**5**	**A**	**2**	**D**	**2**	**4**	**3**	**0**	**2**	**E**	**3**	**9**
Convert to Char equivalent	**a**		**Z**		**-**		**$**		**0**		**.**		**9**	

Hence, an input string of “D8D24F1363514A” will give a decrypted output string of “aZ-$0.9”.

7.10 Validation of application form

7.10.1 Required fields

	Field Name	Max length	Comment
1	Nominal amount	11	The applied amount
2	Comp or Non-comp	1	The 'Y'/'N' indicator for competitive/non competitive
3	Bid yield	3.2	The bid yield
4	Name of applicant	30	
5	NRIC/passport no.	14	
6	Nationality	1	'S' or 'F'

7.10.2 Validation rules

1) The maximum length of each field is specified in the above table.

2) The **Nominal Amount** must be in multiple of one thousand with a minimum **Nominal Amount** of 1,000.

3) **Bid yield** must be numeric up to 2 decimal places.

1. Yield range accepted differs based on the "**mnet.acceptNegativeYield**" property value defined in sgs.properties file
   1. If the property value is true, then Yield range accepted is -99.99 to 99.99 (0 is a valid yield)
   2. If the property value is empty or false, then Yield range accepted is 0.01 to 99.99 (0 is not a valid yield)

4) If **Competitive** Bid selected, then **Bid yield** must not be empty.

5) If **Bid yield** entered, the **Competitive** must be 'Y'.

6) **Nationality** must be S/F.

For Singapore citizen and PR (**Nationality** 'S'), the **NRIC** no. must be valid.

7.11 Configuration and deployment of RSA authentication

7.11.1 Configuration of RSA authentication agent

a) IT Security team will have to setup the authentication agent for prdsys and pass the following files to application team.
   a. AM_Config.zip (sdconf.rec)
   b. NodeSecret.zip (securid)
   c. Encryption password for NodeSecret
b) Copy the 2 files (sdconf.rec and nodesecret.rec) to the folder /mas_appl/aba/jobs/rsatest/examples/sample
c) Execute the command
   export VAR_ACE=/mas_appl/aba/jobs/rsatest/examples/sample
   export USR_ACE=/mas_appl/aba/jobs/rsatest/examples/sample
d) /was/AppServer/bin/setupCmdLine.sh (or was bin folder equivalent)
e) export PATH=/was/AppServer/java/bin:$PATH
f) export AAA = "maseapps(((", if maseapps(((is the password.
g) Execute the command *agent_nsload –f nodesecret.rec –p $AAA* where $AAA is the encryption password in 1c.
h) The securid file should be created in the same folder.
i) Copy securid and sdconf.rec to the folders listed below
   a. /mas_appsw/was8/sgs
   b. /var/ace/api
j) Change the ownership of the 2 files in /mas_appsw/was8/sgs to be ms1ab:web.
k) Execute the command runRSA.job in /mas_appl/aba/jobs/rsatest/examples/sample. (This does not seems to work)
l) Test login using the RSA token.

7.11.2 Deployment of RSA authentication

a) Change the below property value in sgs.properties file to the following
authentication.mechanism=RSA
b) Ensure that the following files exist in /mas_appsw/was8/sgs
   k. sample.properties
   l. sdconf.rec
   m. securid
c) Ensure that the following files exist in /var/ace/api
   n. sdconf.rec
   o. securid
d) Ensure that the following classes exist in /was/AppServer/profiles/AppSrv01/installedApps/PRDSYSCell01/SGSeApps.ear/SGSWeb.war/WEB-INF/classes/sgseapps
   p. SGSAuthFactory.class
   q. SGSAuthIntf.class
   r. SGSSWAuthImpl.class
   s. SGSRSAAuthImpl.class
   t. SGSNoAuthImpl.class
e) Ensure that the login to application using the RSA token is successful.

7.11.3 Possible failure

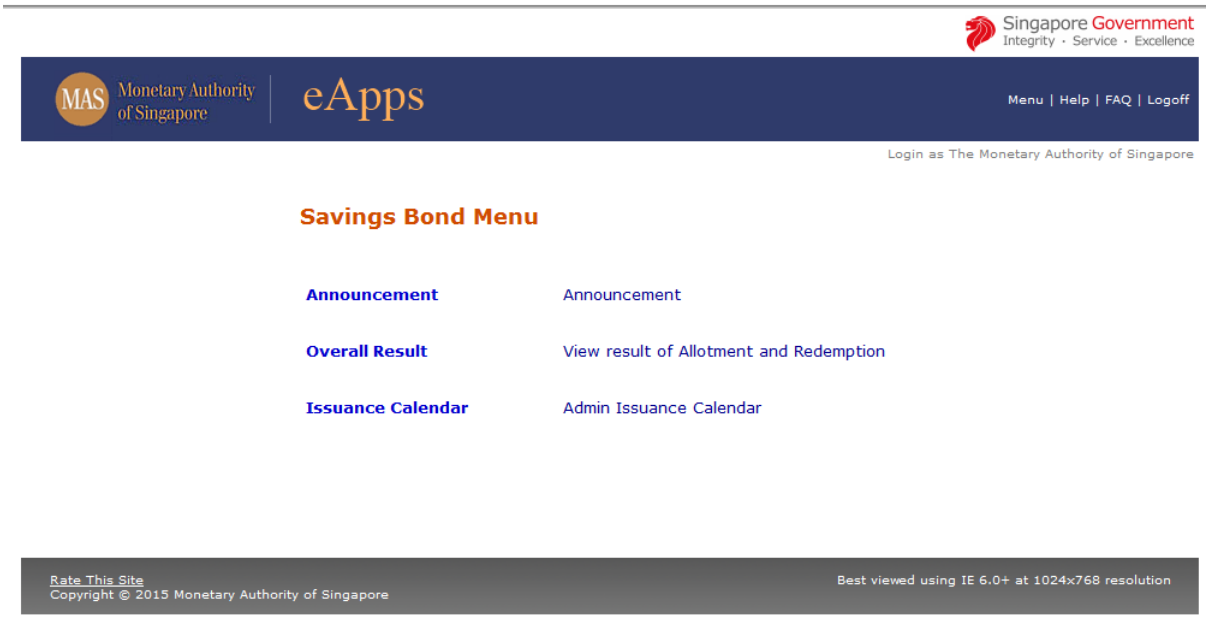
i) The tokens are out of sync. To sync again.
ii) The node secret is not extracted correctly. To extract again.
iii) JAStatus1. file contains the RSA server configuration. To check this file if it is connecting to the wrong server.
iv) Token number is changed every 1 min.

7.12 Savings Bond Module

The following modules are added in for Savings Bond:

i. Announcement
ii. Overall result
iii. Issuance Calendar

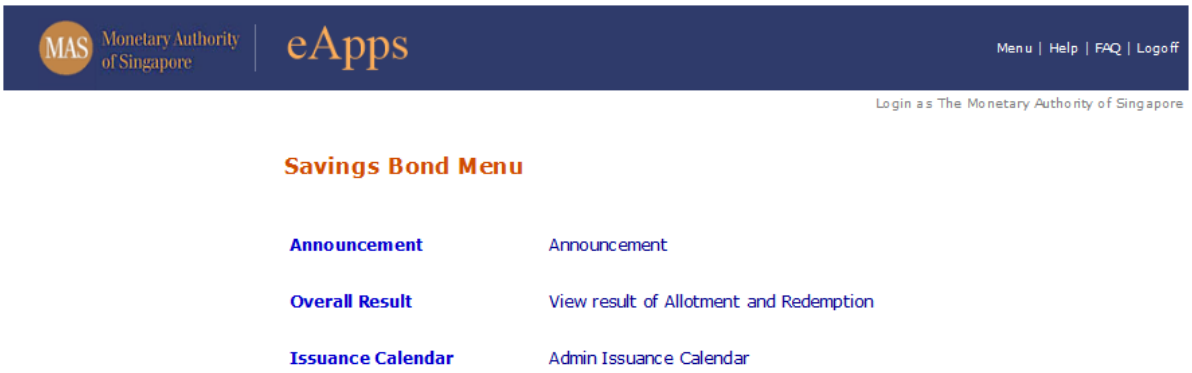
Refer to link below for the details:
https://home.dms.mas.gov.sg/_layouts/15/MASGlobalID/DocAveRedirect.aspx?DocId=38c30eb5-3055-466c-a68c-2d557ff1bf16&SiteID=7eb45e75-9990-4507-ab2f-8359778cd302_a9c7a2b1-d10e-4a88-9525-9cbbaa864def



7.12.1 Announcement

Message display for when the Signal file creation is success or failure for releasing of announcement to website for:

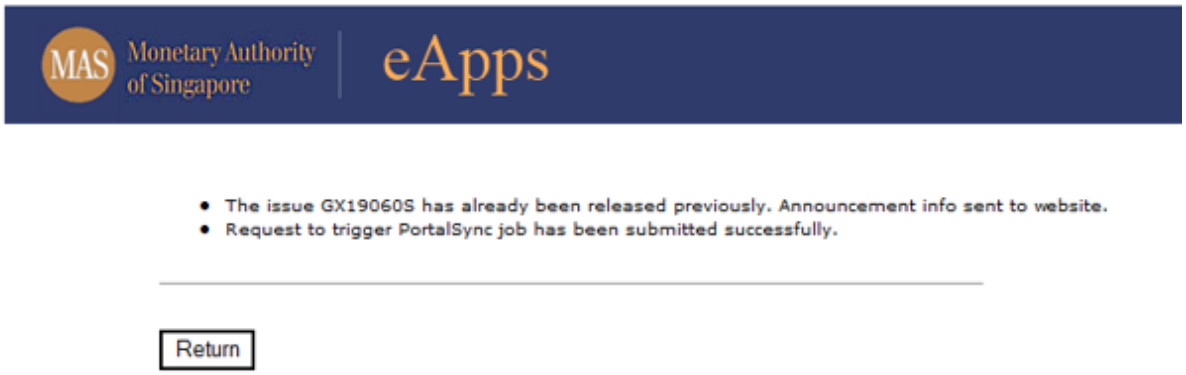
1. Portal sync
2. Execute Adhoc job



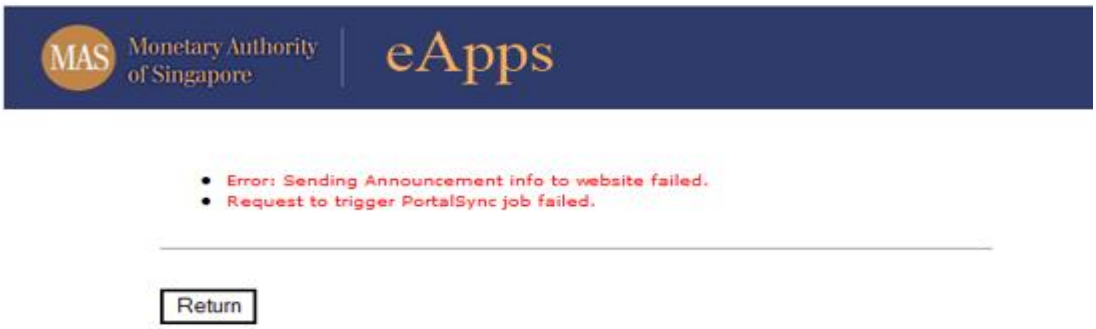
Select Announcement

The error messages return from Signal File activation.

3. Success message



4. Failure message

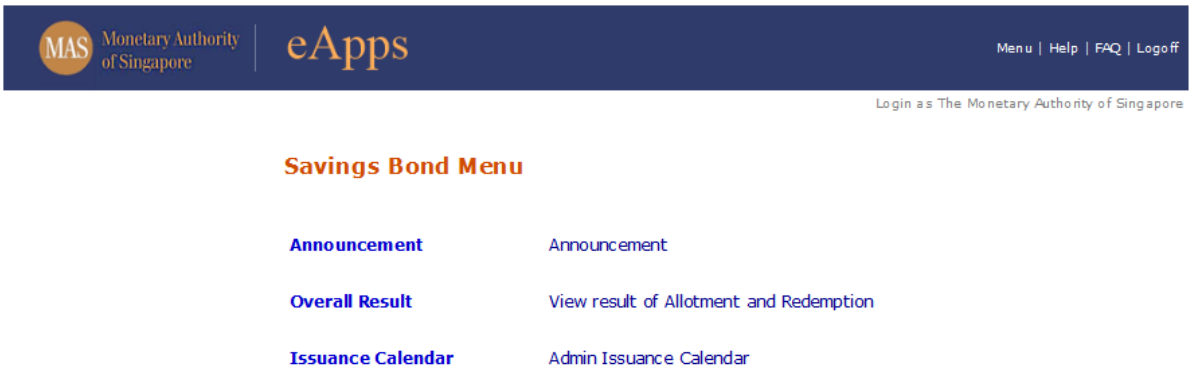


Refer to section 7.14 for signal files and messages.

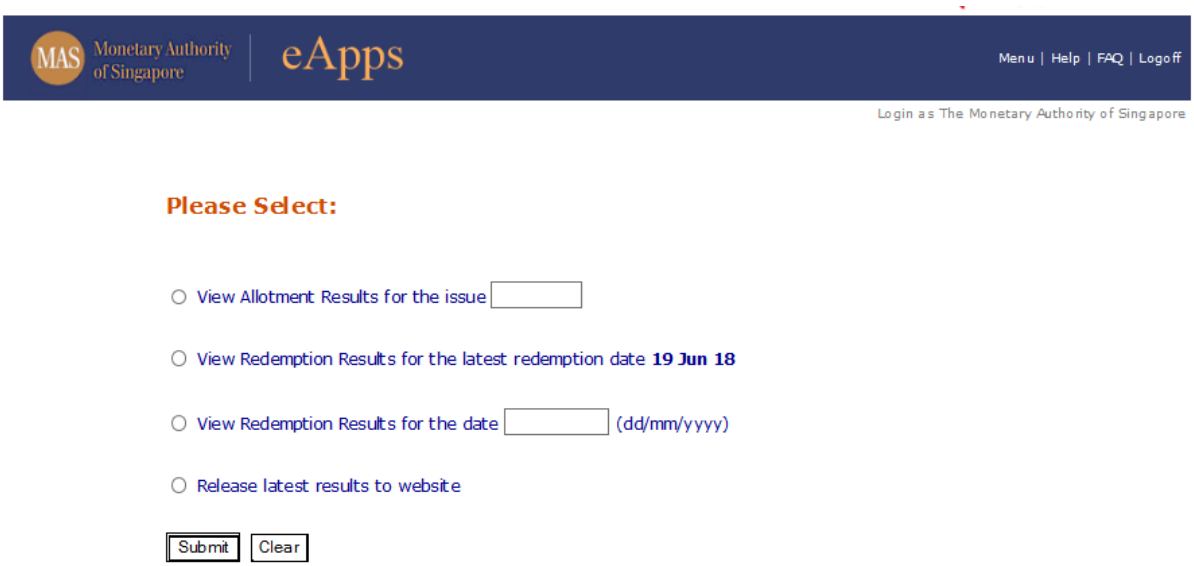
7.12.2 Overall Results

Message display for when the Signal file creation is success or failure for releasing of results to website for:

5. Portal sync
6. Execute Adhoc job

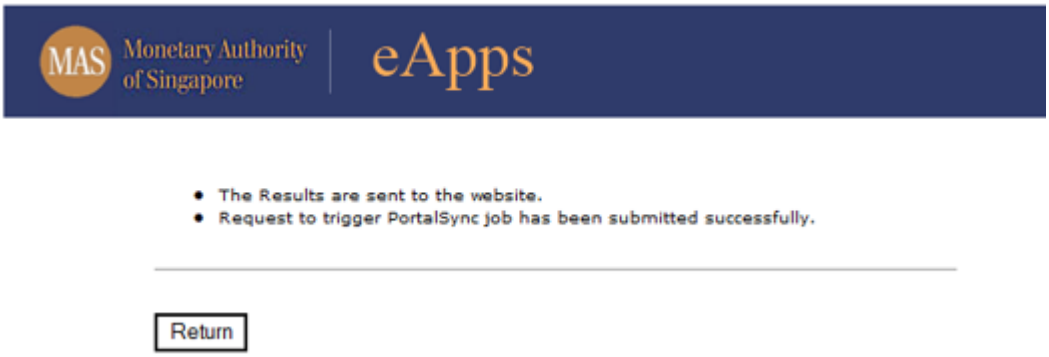


7.12.2.1 Select Overall Result

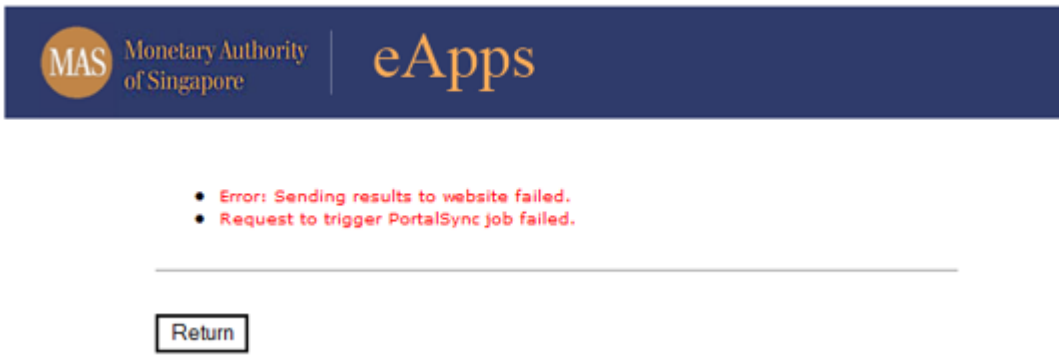


The error messages return from Signal File activation from “Select latest result to website”.

7. Success message



8. Failure message



Refer to section 7.14 for signal files and messages.

7.13 Announcement file.

The update to the Announcement file during 08-Nov-2018 and 03-Jul-2019.

2018-11-08
Announcement Noti

7.14 Signal File and Messages

Function Name	Filename	Remarks	Current Error Message displayed	New Error message displayed
After finalise Closing prices (SGS bonds and bills)	wcm_sgs_closingprices.dat	"call ms9gsw.aba0017_refresh" "call ms9gsw.aba0018_refresh" "call ms9gsw.abav001_refresh" "call ms9gsw.abav102_refresh(1)" "call ms9gsw.abav101_reload(1)"	Error: Financial Database System has not been updated!	Error: Daily Prices update to website failed!
After SGS Auction	wcm_sgs_auctionresults.dat	"call ms9gsw.aba0001_refresh" "call ms9gsw.abav001_refresh"	Publishing of auction results is unsuccessful.	Error: Publishing of auction results to website is unsuccessful
After SSB results	wcm_ssb_results.dat	"call ms9aba. USPCURSORALLOTMENTRESULTS" "call ms9aba. USPCURSORAMOUNTOUTSTANDING"	There is an error sending results to website.	Error: Sending results to website failed.
After SSB Announcement	wcm_ssb_announcement.dat	"call ms9aba. USPCURSORSTEPUPINTEREST"	There is an error sending Announcement info to website.	Error: Sending Announcement info to website failed.
After SGS Announcement	wcm_sgs_announcement.dat	No Stored Procedure	The issue XXXXXXX does not exist or is not an new issue	Error: Sending Announcement info to website failed.
Issue Calendar - T-Bills and SGS Bonds	adhoc_sgs.dat			
Issue Calendar - Savings Bond	adhoc_sb.dat			
Savings Bond - Release Announcement and Release results to Website	adhoc_portalsync.dat			

7.15 WOGAA Changes

WOGAA Onboarding Guide_

Refer to the below internet website for the implementation instructions

https://gist.github.com/wogaa/ac375e1d4cfad3fdd6424b59e8c7682a

1. ADD the below script in <head> in all JSPs
   <script src="https://assets.wogaa.sg/scripts/wogaa.js"></script>

2. Transactional service

Tracking ID	Transactional Service Name	Remarks	StartTransaction	CompleteTransaction
eservicesmas-1342	eApps – SGS Subscription	Both Web form submission and Upload	SGSSubscribe.jsp - Onclick of Submit SGSMasBillsSubscribe.jsp - Onclick of Submit SGSSubmitFileAttach.jsp - Onclick of Submit SGSFrnSubscribe.jsp - Onclick of Submit	SGSSubConfirm.jsp - Onclick of Confirm SGSMasBillsSubConfirm.jsp - Onclick of Confirm SGSUploadConfirm.jsp - Onclick of Submit SGSFrnSubConfirm.jsp - Onclick of Submit
eservicesmas-1660	eApps – Closing Price	Upload - For PDs and Modify data - For PDs and MAS User	For PDs: SGSUserDailyPriceFileAttach.jsp - OnClick of Submit SGSUserDailyPriceInput.jsp - OnClick of Submit For MAS user: SGSMASDailyPriceInput.jsp - OnClick of Submit	For PDs: SGSUserDailyPriceConfirm.jsp - OnClick of Confirm For MAS User: SGSMASDailyPriceConfirm.jsp - OnClick of Confirm
eservicesmas-1659	ERF – Create applications	Both Create applications and Upload	listBids.jsp - Onclick of Submit uploadBids.jsp - Onclick of Upload	confirmBids.jsp - Onclick of Submit

7.16 FRN and SORA

Please refer the document "FRN_SORA_SDS_v1.4.docx" for below details

- FRN – Announcement, Subscription, Auction Results, Closing Price
- SORA Rate Upload
- SORA sync to Website
- Computation of SORA Index and Compounded SORA
- SORA details file to MEPS+
- FRN Outstanding amount for each tenor by month and year

7.17 CorpPass and MAF Integration

7.17.1 Websphere configuration changes

- Place the ninja jars under $WAS_HOME/lib/ext (received from MAF team)

```
file:/was85/IBM/AppServer/lib/ext/
file:/was85/IBM/AppServer/lib/ext/jcan-commons.jar
file:/was85/IBM/AppServer/lib/ext/jcan-sectoken.jar
file:/was85/IBM/AppServer/lib/ext/nevissecwas.jar
file:/was85/IBM/AppServer/lib/ext/ninja-commons.jar
```

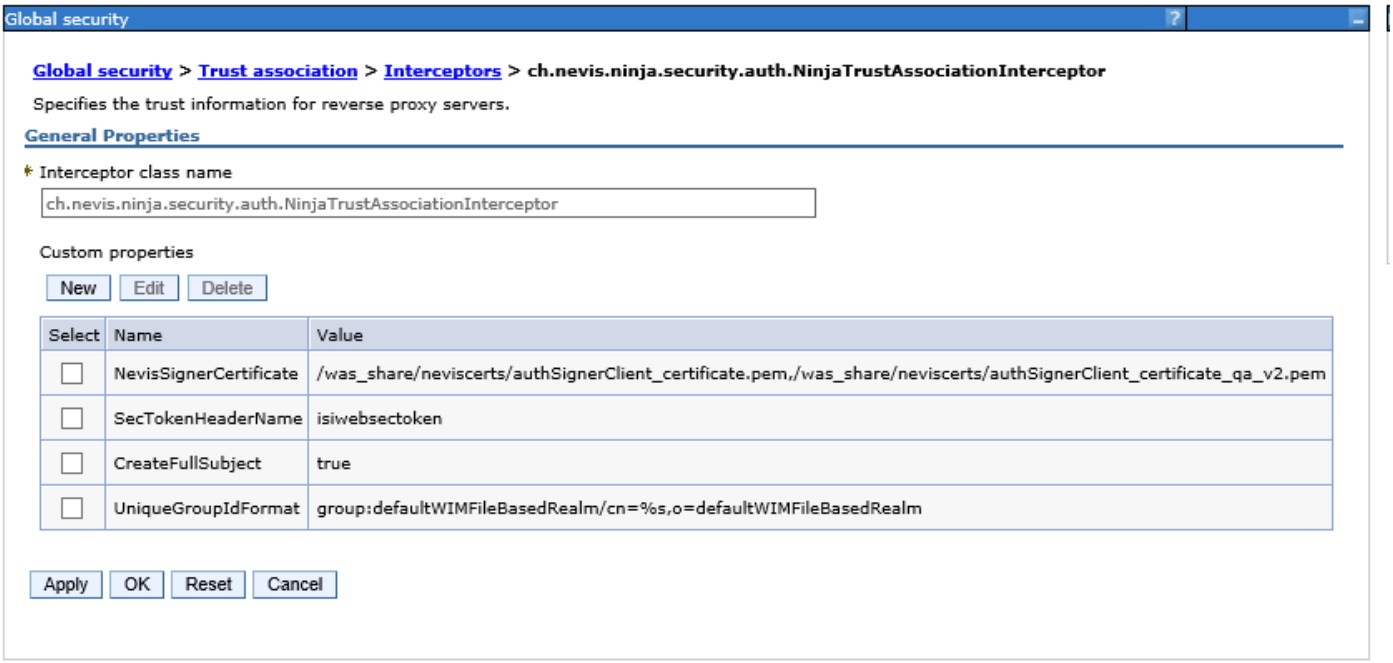
- Place the Ninja certs under /was_share/neviscerts (received from MAF team)
    - Certs file should be assigned "web" group as this is shared across all Applications in the Websphere server which are integrated with MAF

- Create Ninja Trust Association Interceptor as below in Websphere Console
    - Click on Security -> Global Security -> Web and SIP security -> Trust Association
    - Select "Enable trust associations" and click "Apply"
    - Click on "Interceptors"
    - Create a new interceptor and specify "ch.nevis.ninja.security.auth.NinjaTrust-AssociationInterceptor" as its class name
    - Click on the newly created interceptor
    - Click on "Custom Properties"
    - Set the following properties for a Ninja Trust Association Interceptor (TAI)
        1. NevisSignerCertificate –
           For UAT – Certs for MAF DEV and QA
           /was_share/neviscerts/authSignerClient_certificate.pem,/was_share/neviscerts/authSignerClient_certificate_qa_v2.pem
           For PROD:
           /was_share/neviscerts/authSignerClient_certificate_prod.pem
        2. SecTokenHeaderName - isiwebsectoken
        3. CreateFullSubject – true
        4. UniqueGroupIdFormat –
           group:defaultWIMFileBasedRealm/cn=%s,o=defaultWIMFileBasedRealm

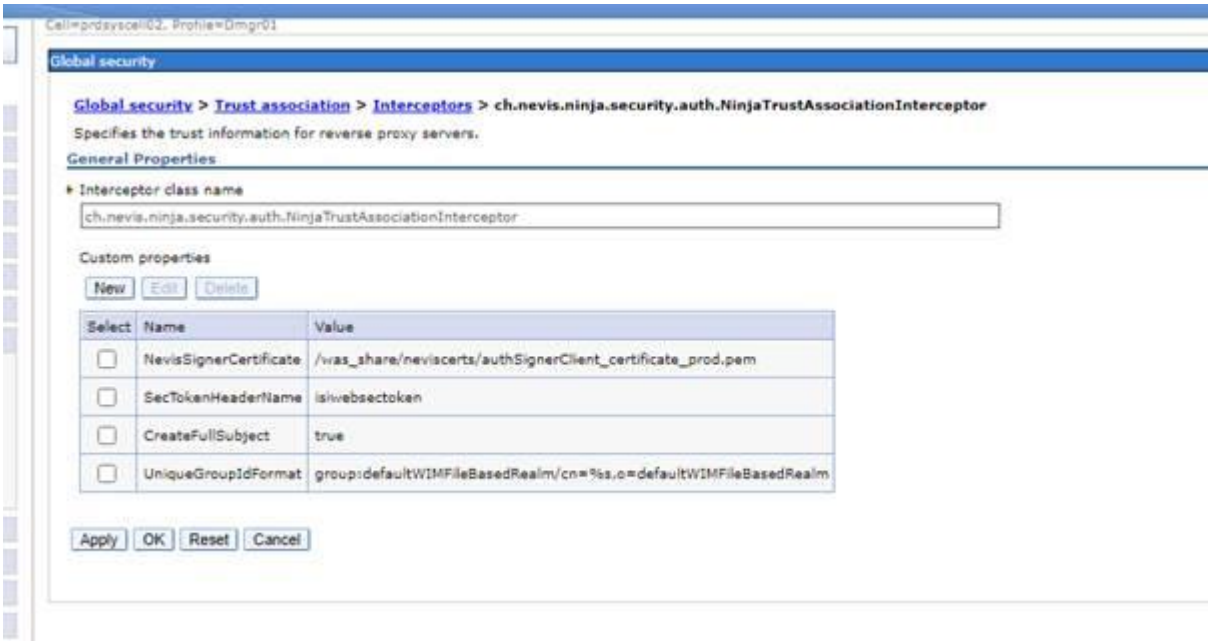
- Restart the WAS server after all config changes

Ninja Trust Association Interceptor Configuration in WebSphere Console

UAT:



PROD:



Please refer the document “masnet_maf_app_intg_spec_latest.pdf” for more details on MAF integration.

7.17.2 Java Code changes

7.17.2.1 Add the below in SGSWeb web.xml

```
<security-constraint>
        <display-name>protected-constraint</display-name>
        <web-resource-collection>
                <web-resource-name>Protected Area</web-resource-name>
                <url-pattern>/SGSAuthentication/*</url-pattern>
                <url-pattern>/sgseapps*</url-pattern>
                <url-pattern>/etender/*</url-pattern>
                <url-pattern>/savings/*</url-pattern>
        </web-resource-collection>
        <auth-constraint>
                <role-name>ninja-group</role-name>
        </auth-constraint>
</security-constraint>
<security-constraint>
        <display-name>Un-Protected constraint</display-name>
        <web-resource-collection>
                <web-resource-name>Public Area</web-resource-name>
                <url-pattern>/etender/common/css/*</url-pattern>
                <url-pattern>/etender/common/images/*</url-pattern>
                <url-pattern>/etender/jsp/images/*</url-pattern>
                <url-pattern>/public/*</url-pattern>
                <url-pattern>/sgseapps.SGSLogin</url-pattern>
                <url-pattern>/etender/jsp/SGSLogin.jsp</url-pattern>
                <url-pattern>/etender/jsp/AuthenticationFailed.jsp</url-pattern>
                <url-pattern>/etender/jsp/CorpPassAuthorizationFailed.jsp</url-pattern>
                <url-pattern>/etender/jsp/DebitCreditInstruction.jsp</url-pattern>
                <url-pattern>/etender/jsp/URLChange.jsp</url-pattern>
                <url-pattern>/etender/common/DebitCredit.pdf</url-pattern>
                <url-pattern>/sgseapps.etender.SGSPublicAnnounce</url-pattern>
                <url-pattern>/etender/common/guide.pdf</url-pattern>
                <url-pattern>/etender/common/validate.js</url-pattern>
                <url-pattern>/etender/common/utility.js</url-pattern>
                <url-pattern>/Mnet*</url-pattern>
                <url-pattern>/sgseapps.Timeout</url-pattern>
                <url-pattern>/etender/jsp/Timeout.jsp</url-pattern>
        </web-resource-collection>
</security-constraint>
<login-config>
        <auth-method>BASIC</auth-method>
        <realm-name>customRealm</realm-name>
</login-config>
<security-role>
        <role-name>ninja-group</role-name>
</security-role>
```

7.17.2.2 Add the below in sgserf web.xml

```
<security-constraint>
        <display-name>protected-constraint</display-name>
```

```
        <web-resource-collection>
                <web-resource-name>Protected Area</web-resource-name>
                <url-pattern>*.do</url-pattern>
        </web-resource-collection>
        <auth-constraint>
                <role-name>ninja-group</role-name>
        </auth-constraint>
</security-constraint>
<security-constraint>
        <display-name>Un-Protected constraint</display-name>
        <web-resource-collection>
                <web-resource-name>Public Area</web-resource-name>
                <url-pattern>/theme/*</url-pattern>
                <url-pattern>/scripts/*</url-pattern>
                <url-pattern>/images/*</url-pattern>
                <url-pattern>/common/images/*</url-pattern>
                <url-pattern>/common/css/*</url-pattern>
                <url-pattern>/public/*</url-pattern>
                <url-pattern>/files/UserGuide_PD.pdf</url-pattern>
                <url-pattern>/files/erf_guideline.pdf</url-pattern>
                <url-pattern>/success.jsp</url-pattern>
                <url-pattern>/timeOut.jsp</url-pattern>
        </web-resource-collection>
</security-constraint>
<login-config>
        <auth-method>BASIC</auth-method>
        <realm-name>customRealm</realm-name>
</login-config>
<security-role>
        <role-name>ninja-group</role-name>
</security-role>
```

7.17.2.3 Add the below in application.xml

```
<security-role id="SecurityRole_1596515317917">
        <role-name>ninja-group</role-name>
</security-role>
```

7.17.2.4 Map Special subject for ninja-group

In WebSphere Console, select J2EE Application SGSeApps, “Security role to user/group mapping”: map **special subject** to, “All Authenticated in Application’s Realm” to the eApps application roles

Below entry would be added automatically to **ibm-application-bnd.xmi**

```
<authorizationTable xmi:id="AuthorizationTable_1301483974481">
   <authorizations xmi:id="RoleAssignment_1596515317916">
     <specialSubjects xmi:type="com.ibm.ejs.models.base.bindings.applicationbnd:AllAuthenticatedUsers"
xmi:id="AllAuthenticatedUsers_1596515317916" name="AllAuthenticatedUsers"/>
```

```
    <role href="META-INF/application.xml#SecurityRole_1596515317917"/>
  </authorizations>
</authorizationTable>
```

7.17.2.5 Code to read Ninja token principal using Ninja plugin

Below is the sample code on how to read NinjaTokenPrincipal.
This code is implemented in eApps in **SGSCorpPassAuth.java** class

User login details CPEntID (UEN) and CPUID (NRIC) are retrieved from **userAttributes** Map object.

```java
import javax.security.auth.Subject;
import com.ibm.websphere.security.auth.WSSubject;
import ch.nevis.ninja.commons.token.NinjaTokenPrincipal;

void sample(HttpServletRequest req)
{
    // User ID
    String userId = req.getUserPrincipal().toString();

    // Get NinjaTokenPrincipal
    NinjaTokenPrincipal principal = getNinjaTokenPrincipal();

    if (principal != null) {
        // get userID
        String userId = principal.getName();

        // Roles contained in security token
        String[] roles = principal.getRoles();

        // Attributes contained in security token
        Map<String,String> userAttributes = principal.getAttributes();
    }
}

NinjaTokenPrincipal getNinjaTokenPrincipal()
{
    NinjaTokenPrincipal tokenPrincipal = null;

    // Get Subject via Websphere API
    Subject subject = WSSubject.getCallerSubject();
    if (subject != null) {  // if subject is null, probably user is unauthenticated
        Iterator principals = subject.getPrincipals().iterator();
        while (principals.hasNext()) {
            Principal principal = (Principal) principals.next();
            if (principal instanceof NinjaTokenPrincipal) {
                tokenPrincipal = (NinjaTokenPrincipal) principal;
                break;
            }
        }
    }

    return tokenPrincipal;
}
```

7.17.2.6 Sample CorpPass Secure Token

The attribute starts with CP* are all specific to CorpPass authentication


```
<secToken version="CSSO-1.0" signTime="20170119102357Z" ttl="7200">
  <attr>
    <esauthid>I1</esauthid>
    <field name="CPNonUEN_RegNo">NULL</field>
    <field name="loginId">CP192</field>
    <field
name="authnContextClassRef">urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</field
>
    <field name="CPNonUEN_Name">NULL</field>
    <field
name="AuthAccess">PEF1dGhBY2Nlc3M+PFJlc3VsdF9TZXQ+PEVTcnZjX1Jvd19Db3VudD4xPC9FU3J2
Y19Sb3dfQ291bnQ+PEVTcnZjX1Jlc3VsdD48Q1BFU3J2Y0lEPkJHRVNSVjE8L0NQRVNydmNJRD48QX
V0aF9SZXN1bHRfU2V0PjxSb3dfQ291bnQ+MjwvUm93X0NvdW50PjxSb3c+PENQRW50SURfU1VCPk5
VTEw8L0NQRW50SURfU1VCPjxDUFJvbGU+TlVMTDwvQ1BSb2xlPjxTdGFydERhdGU+MjAxNi0wMS
0xNTwvU3RhcnREYXRlPjxFbmREYXRlPjIwMTYtMDItMTU8L0VuZERhdGU+PC9Sb3c+PFJvdz48Q1B
FbnRJRF9TVUI+TlVMTDwvQ1BFbnRJRF9TVUI+PENQUm9sZT5OVUxMPC9DUFJvbGU+PFN0YXJ0R
GF0ZT4yMDE2LTAzLTE1PC9TdGFydERhdGU+PEVuZERhdGU+MjAxNy0wNC0xNTwvRW5kRGF0ZT
48UGFyYW1ldGVyIG5hbWU9IllllYXIgb2YgYXNzZXNzbWVudCI+MjAxNDwvUGFyYW1ldGVyPjxQY
XJhbWV0ZXIgbmFtZT0ib3RoZXIwMiI+dmFsdWUgMDI8L1BhcmFtZXRlcj48UGFyYW1ldGVyIG5hbW
U9Im90aGVyMDMiPnZhbHVlIDAzPC9QYXJhbWV0ZXI+PFBhcmFtZXRlciBuYW1lPSJvdGhlcjA0Ij52Y
Wx1ZSAwNDwvUGFyYW1ldGVyPjxQYXJhbWV0ZXIgbmFtZT0ib3RoZXIwNSI+dmFsdWUgMDU8L1B
hcmFtZXRlcj48UGFyYW1ldGVyIG5hbWU9Im90aGVyMDYiPnZhbHVlIDA2PC9QYXJhbWV0ZXI+PFB
hcmFtZXRlciBuYW1lPSJvdGhlcjA3Ij52YWx1ZSAwNzwvUGFyYW1ldGVyPjxQYXJhbWV0ZXIgbmFtZ
T0ib3RoZXIwOCI+dmFsdWUgMDg8L1BhcmFtZXRlcj48L1Jvdz48L0F1dGhfUmVzdWx0X1NldD48L0VT
cnZjX1Jlc3VsdD48L1Jlc3VsdF9TZXQ+PC9BdXRoQWNjZXNzPg==</field>
    <field name="CPUID_Country">SG</field>
    <field name="roles">auth.weak</field>
    <field name="CPEnt_Status">Registered</field>
    <field name="CPAccType">User</field>
    <userid>CP192</userid>
    <sessid>XZxlv-XCfpPc5JB9HxPEE0VWJkYndznFzGv0CFcUiG4</sessid>
    <entryid>SAML-Federation-CP-CORPPASS</entryid>
    <field name="CPUID_FullName">John Grisham</field>
    <field name="CPEnt_TYPE">UEN</field>
    <field name="CPUID">F1234567P</field>
    <field name="domain">MAF_MASNET_CORPPASS_OPERA</field>
    <field name="CPSystemUID">CP192</field>
    <field name="ISSPHOLDER">YES</field>
    <field name="CPEntID">R90SS0001A</field>
    <authLevel>auth.weak</authLevel>
    <field name="CPNonUEN_Country">NULL</field>
  </attr>
  <signature alg="SHA1withRSA"
fingerPrint="5A:DB:65:BC:C7:4A:F7:41:9B:41:DE:FC:B4:DA:2D:D5">LtlGOf+jH8cS1YTebHlhtuJg4vSJG
+6so85XFKMj/F25pE/8iuywh5WuuoAMq1rWl6qa287XQMfIHHVzKM+KVuwfTQI0QCcgkLiWT4/TTTQ
CDKPIB20kWQ0WyVzluBdTktu+N2/UyceVRr/XSvcZjlUXH1iv6SWk0PXrY5QtBc8DR4c10i8XS38nzHM
KDES29uMFkjKYsSFt/70UgdL2TnHypWR9gZpAO4fd/g469qRezdJ5MYqewG9cstiLQ1BM6zISNXXWa3j
ni+4Rv4/SWXZVGG1L2OqGCCrpZdPNCcJ9KfI2XAkB4/JBkXJtBY4d0qfKVnCbGYQAqto/s+WM2A==<
/signature>
```

</secToken>

Description about few fields
CPUID : NRIC, FIN or Foreign ID number of the user.
CPEntID : Entity ID (UEN or non-UEN ID) of the entity to which the user belongs in CorpPass
CPSystemUID : System defined ID of the user.
AuthAccess : base64 content of the below

<AuthAccess><Result_Set><ESrvc_Row_Count>1</ESrvc_Row_Count><ESrvc_Result><CPESrvcID>BGESRV1</CPESrvcID><Auth_Result_Set><Row_Count>2</Row_Count><Row><CPEntID_SUB>NULL</CPEntID_SUB><CPRole>NULL</CPRole><StartDate>2016-01-15</StartDate><EndDate>2016-02-15</EndDate></Row><Row><CPEntID_SUB>NULL</CPEntID_SUB><CPRole>NULL</CPRole><StartDate>2016-03-15</StartDate><EndDate>2017-04-15</EndDate><Parameter name="Year of assessment">2014</Parameter><Parameter name="other02">value 02</Parameter><Parameter name="other03">value 03</Parameter><Parameter name="other04">value 04</Parameter><Parameter name="other05">value 05</Parameter><Parameter name="other06">value 06</Parameter><Parameter name="other07">value 07</Parameter><Parameter name="other08">value 08</Parameter></Row></Auth_Result_Set></ESrvc_Result></Result_Set></AuthAccess>

CorpPass Interface Specification

Tag Name	Tag Description	Data Type	Field Length	Mandatory
UserInfo	This the starting tag of the User Info XML	Tag	-	Yes
CPAccType	CorpPass account type i.e. if the user is a CorpPass Administrator, CorpPass Sub-Administrator or a CorpPass User	String	30	Yes
CPUID	NRIC, FIN or Foreign ID number of the user.	String	20	Yes
CPUID_Country	Country where the Identity Card of the user that was used for verification was issued	String	2	Yes
CPUID_FullName	Full Name of User	String	100	Yes
CPSystemUID	System defined ID of the user.	String	20	Yes
ISSPHOLDER	Indicates if the user is a SingPass holder i.e. the user has activated their CorpPass ID using SingPass (YES or NO)	String	3	Yes
CPEntID	Entity ID (UEN or non-UEN ID) of the entity to which the user belongs in CorpPass	String	10	Yes
CPEnt_Status	Status of the entity in CorpPass as provided by issuance agencies. The possible values for entity status are: • Registered • De-Registered • Withdrawn • Null (Specific for Non-UEN scenario)	String	20	Yes
CPEnt_TYPE	Type of Entity that the user belongs to. E.g. UEN, Non-UEN	String	10	Yes
CPNonUEN_RegNo	Entity registration number of Non-UEN entity which is entered during CorpPass Admin registration. Value will be "NULL" if user belongs to UEN entity.	String	15	Yes
CPNonUEN_Country	Country of incorporation of Non-UEN entity which is provided during CorpPass Admin registration. Value will be "NULL" if user belongs to UEN entity.	String	2	Yes

7.17.2.7 Corp Pass Login URL

1. The Login url is configured in "mnet.corpPass.url.login" property in sgs.properties file
2. UAT URL is as below

https://stg-saml.corppass.gov.sg/FIM/sps/CorpIDPFed/saml20/logininitial?RequestBinding=HTTPArtifact&Respo

nseBinding=HTTPArtifact&PartnerId=https://ueservices.mas.gov.sg/maf/saml/corppass/eapps&Target=https://ueservices.mas.gov.sg/sgseapps/SGSAuthentication&NameIdFormat=Email&esrvcID=MAS-EAPPS

PartnerId – MAF URL
Target – eApps Authorization URL where Ninja Token Principal is read (There should not be "." dot in this URL after the context root)
esrvcID – Service ID registered with CorpPass for eApps application

7.17.2.8 Global Logout page

1. The global logout url is configured in "mnet.corpPass.url.logoff" property in sgs.properties file
2. eApps application will redirect to global logout url only in case CorpPass Login

7.17.3 MAF Configuration

1. Response timeout set to 7 minutes
2. RFI configuration as per attached excel

masnet
maf-application integ

3. Un-protected files/folders in MAF side and eApps application web.xml should be same

List of unprotected files:
/sgseapps/sgseapps.SGSLogin
/sgseapps/etender/jsp/SGSLogin.jsp
/sgseapps/etender/jsp/AuthenticationFailed.jsp
/sgseapps/etender/jsp/CorpPassAuthorizationFailed.jsp
/sgseapps/etender/jsp/DebitCreditInstruction.jsp
/sgseapps/etender/jsp/URLChange.jsp
/sgseapps/etender/common/DebitCredit.pdf
/sgseapps/sgseapps.etender.SGSPublicAnnounce*
/sgseapps/etender/common/guide.pdf
/sgseapps/etender/common/validate.js
/sgseapps/etender/common/utility.js
/sgseapps/Mnet*
/sgseapps/sgseapps.Timeout
/sgseapps/etender/jsp/Timeout.jsp
/sgserf/files/UserGuide_PD.pdf
/sgserf/files/erf_guideline.pdf
/sgserf/success.jsp
/sgserf/timeOut.jsp

List of Unprotected folders:
/sgseapps/etender/common/css/*
/sgseapps/etender/common/images/*

/sgseapps/etender/jsp/images/*
/sgseapps/public/*
/sgserf/theme/*
/sgserf/scripts/*
/sgserf/images/*
/sgserf/common/images/*
/sgserf/common/css/*
/sgserf/public/*

Note:

- Context root **/sgseapps** and **/sgserf** is required only in MAF side configuration and not in our applications web.xml
- Any new un-protected files except Servlets should be added under /public folder and so there will not be any change required in MAF side.
- When any new un-protected Servlets are added, need to inform MAF to add those servlets as un-protected and in eApps need to modify web.xml and Login Filter accordingly.

7.17.4 Network configuration

1. WAF has to be pointed to MAF instead of application server
2. eApps application can be accessed using ueservices.mas.gov.sg for CorpPass Login using Internet VDI

7.18 Code changes for Non-CorpPass login

1. Remove the configurations added in web.xml of SGSWeb and sgserf in Section 7.17.2.1 and 7.17.2.2 and then deploy the EAR to access eApps application using non CorpPass login
2. Change the "authentication.mechanism" property value in sgs.properties file as below for different login
    1. If the value is empty, then it is RSA simulator (No auth mechanism) login
    2. If the value is RSA, then it is RSA login
    3. If the value is CorpPassSimulator, then it is CorpPass Simulator login
    4. If the value is CorpPass, then it is CorpPass login
    5. If the value is Toppan, then it is CorpPass TOPPAN Integration login
3. Access the eApps application using VIP URL https://172.24.244.172/sgseapps

Note: Can't access eApps application using ueservices.mas.gov.sg because WAF is pointing to MAF

7.19 Antivirus Scan

All file upload in eApps application should perform antivirus scan and if the scan is successful, then the upload file will be saved in file system server

There are 4 upload files available in eApps application

1. T-Bills/Bonds Bid Submission Upload
2. Closing Price Upload
3. ERF Bid Submission Upload
4. SORA Rate Upload

Antivirus scan is done using an API call with the information in the below attached pdf

rufus API guide.pdf

All the antivirus configuration details are stored in ABA0032_EAPPS_CONFIG table in MNET DB.

ABA0032_KEY	ABA0032_VALUE
antivirusScan.enabled	true
antivirusScan.url	https://dtrendapi/
antivirusScan.key	30ec716ea61bf146fb67200ec0cd988e
antivirusScan.user	SGSeApps

Antivirus scan will be performed only if the "antivirusScan.enabled" is true in DB and the value "true" is not case sensitive.

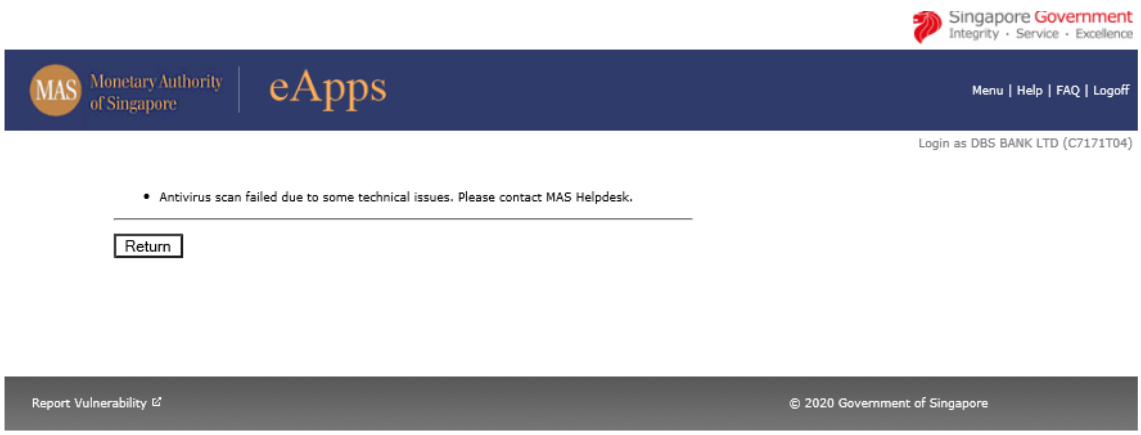
API Details:

Environment	API URL	API Key
UAT	https://dtrendapi/	30ec716ea61bf146fb67200ec0cd988e
PROD	https://trendapi/	1dca63c0af11ef9c560a14a3276a570a

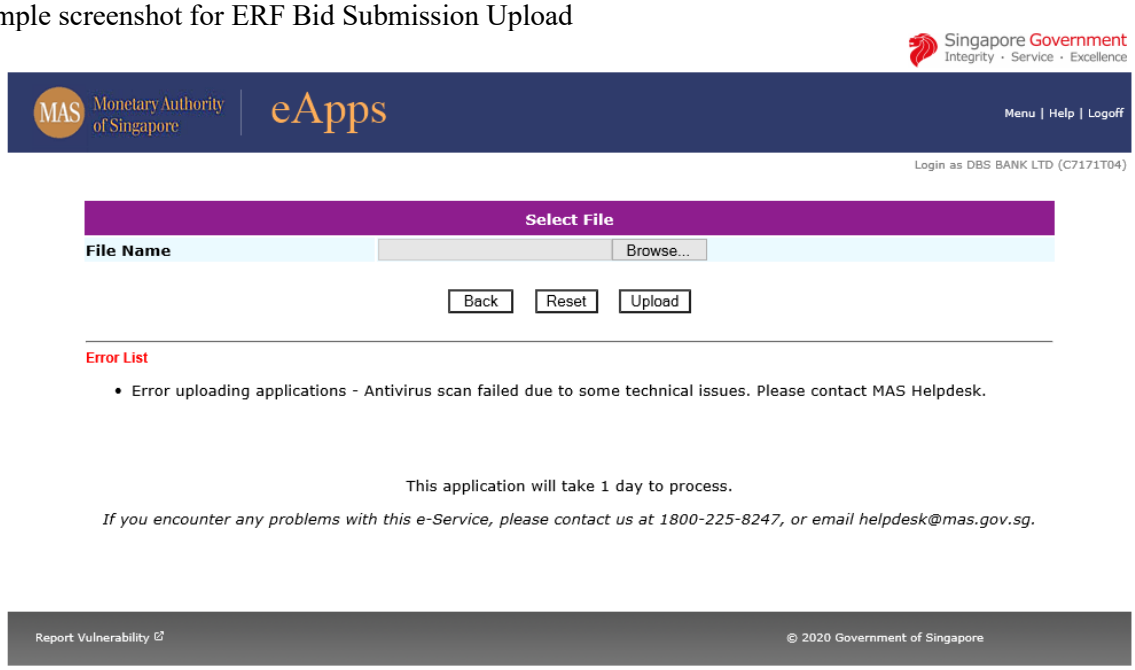
Error Messages:

1. General error message "**Antivirus scan failed due to some technical issues. Please contact MAS Helpdesk.**" will be shown if there is any error occurred during the API call.

Sample screenshot for
1. T-Bills/Bonds Bid Submission Upload
2. Closing Price Upload
3. SORA Rate Upload



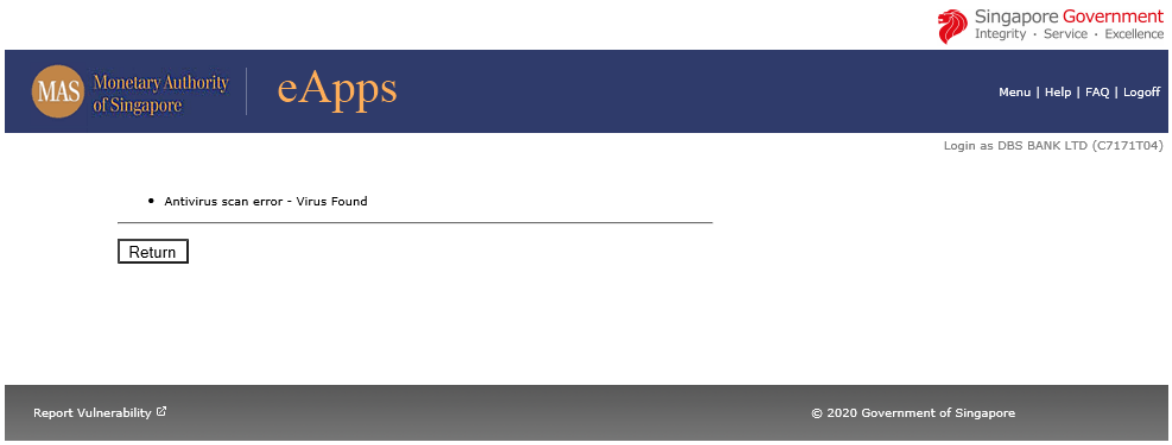
Sample screenshot for ERF Bid Submission Upload



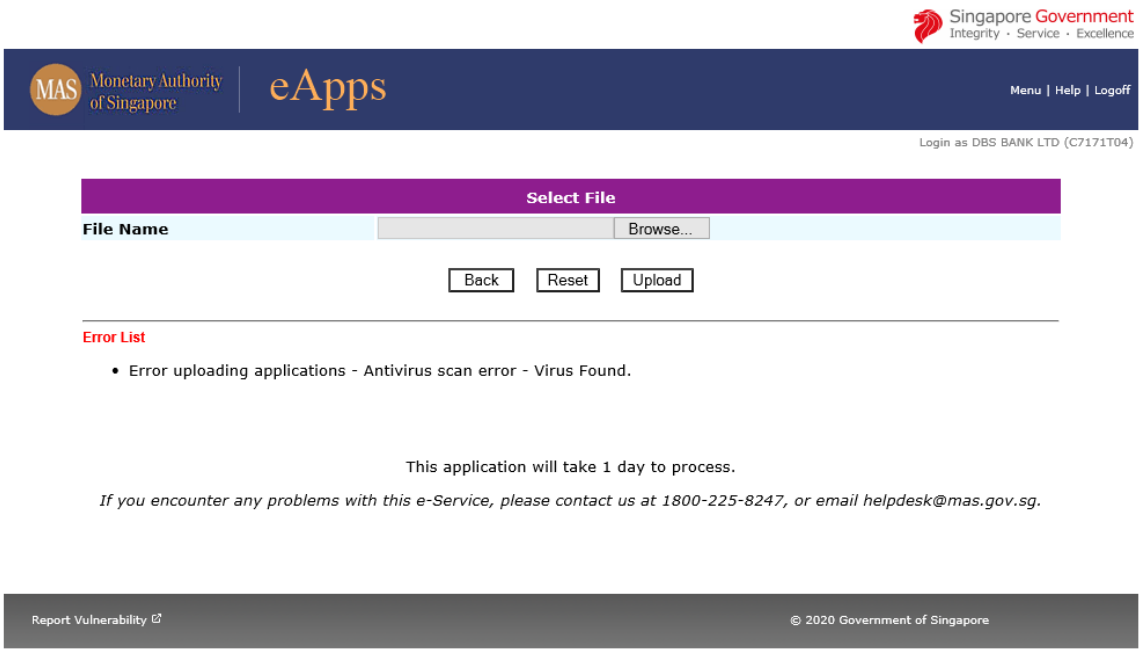
2. Error message "**Antivirus scan error - <Message from API response>.**" will be shown if unsuccessful response received from API.

Sample screenshot for

1. T-Bills/Bonds Bid Submission Upload
2. Closing Price Upload
3. SORA Rate Upload



Sample screenshot for ERF Bid Submission Upload



7.20 PDF Files used in eApps

PDF File	eApps Application Page	Referenced in JSP
SGSWeb\WebContent\etender\common\DebitCredit.pdf	eApps Login page -> Debit/Credit Authorisation Form link -> Debit/Credit Authorisation Form link	SGSWeb\WebContent\etender\jsp\DebitCreditInstruction.jsp
SGSWeb\WebContent\etender\common\guide.pdf	eApps Login page -> User Guide link Help link in the Header -> User Guide for eApps link	SGSWeb\WebContent\etender\jsp\SGSLogin.jsp SGSWeb\WebContent\etender\jsp\SGSHelp.jsp
SGSWeb\WebContent\etender\jsp\doc\bid_manual.pdf	Help link in the Header -> Bid Upload Help link	SGSWeb\WebContent\etender\jsp\SGSHelp.jsp
SGSWeb\WebContent\etender\jsp\doc\dp_manual.pdf	Help link in the Header -> Daily Price Help link	SGSWeb\WebContent\etender\jsp\SGSHelp.jsp
SGSWeb\WebContent\etender\jsp\doc\faq.pdf	FAQ link in the Header	SGSWeb\WebContent\etender\common\header.jsp
SGSWeb\WebContent\etender\jsp\doc\instruct_frn.pdf	Help link in FRN Subscription page	SGSWeb\WebContent\etender\jsp\SGSFrnSubscribe.jsp \SGSWeb\WebContent\etender\jsp\SGSFrnReSubmit.jsp
SGSWeb\WebContent\etender\jsp\doc\instruct_masbill.pdf	Help link in MAS Bills Subscription page	SGSWeb\WebContent\etender\jsp\SGSMasBillsSubscribe.jsp SGSWeb\WebContent\etender\jsp\SGSMasBillsReSubmit.jsp
SGSWeb\WebContent\etender\jsp\doc\instruct.pdf	Help link in T-Bills/Bonds Subscription page	SGSWeb\WebContent\etender\jsp\SGSSubscribe.jsp SGSWeb\WebContent\etender\jsp\SGSReSubmit.jsp
sgserf\WebContent\doc\erf_agreement.pdf	ERF Agreement page – On accessing “Enhanced Repo Facility” link in Main menu by PDs	sgserf\WebContent\erfAgreement.jsp
sgserf\WebContent\doc\erf_guideline.pdf	ERF Agreement page	sgserf\WebContent\erfAgreement.jsp

	– On accessing "Enhanced Repo Facility" link in Main menu by PDs	
sgserf\WebContent\doc\UserGuide_PD.pdf	Help link in the Header -> ERF Guide link Help link in the Header of all ERF pages	SGSWeb\WebContent\etender\jsp\SGSHelp.jsp sgserf\WebContent\common\header.jsp

7.21 Footer – Privacy Statement and Terms of use

For Privacy Statement and Terms of use HTML content data, refer the attached document.

privacy-statement and terms of use.docx

7.22 MQ Revamp – Steps to create keystore/renew certificate/CA certs change

Need to create keystore for each server in UAT and PROD.

UAT servers:

- dcssys2 (Internal)
- dcssys3 (External)

PROD servers:

- pcssys90 (Internal)
- prdsys (External)

All servers keystore can be created in one server dcssys2 and then can copy the generated key.jks file to the respective servers /aza/ssl folder.

Note:

- MQ jar is placed under /aza/mqprog/MQJob.jar in respective servers since this is being used by other teams/projects.
- Do not change the password "ms1aza" in the below steps.
- If the password is changed, need to encrypt the password using AES-GCM util and place the encrypted and IV files in /aza/aesgcm folder in the respective servers.

Steps to create keystore/renew certificate/CA certs change for dcssys2:

1. Log on to dcssys2 server using application id and cd to /mas_apps/aba/ssl folder.

2. Create Key Database Repository using runmqakm command. After successful execution, we can see key.p12 files generated
   runmqakm -keydb -create -db key.p12 -pw ms1aza -type p12 -expire 7300 -fips

3. Create new certificate request for dcssys2 server as shown below:
   runmqakm -certreq -create -db key.p12 -pw ms1aza -label dcssys2 -dn "CN=dcssys2.mas.gov.sg, OU=CBD, O=ITD, L=UAT, S=EAPPS, C=SG" -size 2048 -sig_alg SHA256WithRSA -file dcssys2_certreq.arm -fips

4. List the recently created certificate request in last step using list option.
   runmqakm -certreq -list -db key.p12 -pw ms1aza

5. Copy cert request file (arm file generated in Step 3) using ASCII mode and send it to MAS CA for signing and backup all files in the ssl folder to reuse if any issue in further steps.

6. After receiving Signed Certificate along with CA Certificate, transfer it into Server to same ssl folder. Please follow the steps in Section 7.22.1 to generate 3 certificates (root, intermediate, signed) before adding to keystore.

7. Add Intermediate CA Cert using below command into Signer certificate store.
   runmqakm -cert -add -db key.p12 -pw ms1aza -file INTERMEDIATECA.cer -label IntermediateCA -type p12 -format ascii -trust enable

8. Add Root CA Certificate using below command into Signer certificate store.
   runmqakm -cert -add -db key.p12 -pw ms1aza -file ROOTCA.cer -label RootCA -type p12 -format ascii -trust enable

9. Receive the Signed Certificate into Key Database using receive option.
   runmqakm -cert -receive -db key.p12 -pw ms1aza -file dcssys2_signed.cer -type p12 -format ascii -default_cert yes

10. Now list the entire certificate present in key database file for dcssys2.
    runmqckm -cert -list -db key.p12 -pw ms1aza -type p12

11. Convert p12 file into JKS format using below command.
    runmqckm -keydb -convert -db key.p12 -pw ms1aza -target key.jks -new_pw ms1aza -type p12 -new_format jks -expire 7300
    runmqckm -cert -list -db key.jks -pw ms1aza -type jks

Steps to create keystore/renew certificate/CA certs change for dcssys3:

1. Log on to dcssys2 server using application id and cd to /mas_apps/aba/ssl_sys3 folder.

2. Create Key Database Repository using runmqakm command. After successful execution, we can see key.p12 files generated
   runmqakm -keydb -create -db key.p12 -pw ms1aza -type p12 -expire 7300 -fips

3. Create new certificate request for dcssys3 server as shown below:
   runmqakm -certreq -create -db key.p12 -pw ms1aza -label dcssys3 -dn "CN=dcssys3.mas.gov.sg, OU=CBD, O=ITD, L=UAT, S=EAPPS, C=SG" -size 2048 -sig_alg SHA256WithRSA -file dcssys3_certreq.arm -fips

4. List the recently created certificate request in last step using list option.
   /runmqakm -certreq -list -db key.p12 -pw ms1aza

5. Copy cert request file (arm file generated in Step 3) using ASCII mode and send it to MAS CA for signing and backup all files in the ssl_sys3 folder to reuse if any issue in further steps.

6. After receiving Signed Certificate along with CA Certificate, transfer it into Server to same ssl_sys3 folder. Please follow the steps in Section 7.22.1 to generate 3 certificates (root, intermediate, signed) before adding to keystore.

7. Add Intermediate CA Cert using below command into Signer certificate store.
   runmqakm -cert -add -db key.p12 -pw ms1aza -file INTERMEDIATECA.cer -label IntermediateCA -type p12 -format ascii -trust enable

8. Add Root CA Certificate using below command into Signer certificate store.
   runmqakm -cert -add -db key.p12 -pw ms1aza -file ROOTCA.cer -label RootCA -type p12 -format ascii -trust enable

9. Receive the Signed Certificate into Key Database using receive option.
   runmqakm -cert -receive -db key.p12 -pw ms1aza -file dcssys3_signed.cer -type p12 -format ascii -default_cert yes

10. Now list the entire certificate present in key database file for dcssys3.
    runmqckm -cert -list -db key.p12 -pw ms1aza -type p12

11. Convert p12 file into JKS format using below command.
    runmqckm -keydb -convert -db key.p12 -pw ms1aza -target key.jks -new_pw ms1aza -type p12 -new_format jks -expire 7300
    runmqckm -cert -list -db key.jks -pw ms1aza -type jks

Steps to create keystore/renew certificate/CA certs change for pcssys90:

1. Log on to dcssys2 server using application id and cd to /mas_apps/aba/ssl_pcssys90 folder.

2. Create Key Database Repository using runmqakm command. After successful execution, we can see key.p12 files generated
   runmqakm -keydb -create -db key.p12 -pw ms1aza -type p12 -expire 7300 -fips

3. Create new certificate request for pcssys90 server as shown below:
   runmqakm -certreq -create -db key.p12 -pw ms1aza -label pcssys90 -dn "CN=pcssys90.mas.gov.sg, OU=CBD, O=ITD, L=PROD, S=EAPPS, C=SG" -size 2048 -sig_alg SHA256WithRSA -file pcssys90_certreq.arm -fips

4. List the recently created certificate request in last step using list option.
   runmqakm -certreq -list -db key.p12 -pw ms1aza

5. Copy cert request file (arm file generated in Step 3) using ASCII mode and send it to MAS CA for signing and backup all files in the ssl_pcssys90 folder to reuse if any issue in further steps.

6. After receiving Signed Certificate along with CA Certificate, transfer it into Server to same ssl_pcssys90 folder. Please follow the steps in Section 7.22.1 to generate 3 certificates (root, intermediate, signed) before adding to keystore.

7. Add Intermediate CA Cert using below command into Signer certificate store.
   runmqakm -cert -add -db key.p12 -pw ms1aza -file INTERMEDIATECA.cer -label IntermediateCA -type p12 -format ascii -trust enable

8. Add Root CA Certificate using below command into Signer certificate store.
   runmqakm -cert -add -db key.p12 -pw ms1aza -file ROOTCA.cer -label RootCA -type p12 -format ascii -trust enable

9. Receive the Signed Certificate into Key Database using receive option.
   runmqakm -cert -receive -db key.p12 -pw ms1aza -file pcssys90_signed.cer -type p12 -format ascii -default_cert yes

10. Now list the entire certificate present in key database file for pcssys90.
    runmqckm -cert -list -db key.p12 -pw ms1aza -type p12

11. Convert p12 file into JKS format using below command.
    runmqckm -keydb -convert -db key.p12 -pw ms1aza -target key.jks -new_pw ms1aza -type p12 -new_format jks -expire 7300
    runmqckm -cert -list -db key.jks -pw ms1aza -type jks

12. Infra Team to get root access and then su to ms1az account id
13. Go to folder: /mas_apps/aza/ssl and take backup of existing key.jks
14. Copy new key.jks file to /mas_apps/aza/ssl location

Steps to create keystore/renew certificate/CA certs change for prdsys:

1. Log on to dcssys2 server using application id and cd to /mas_apps/aba/ssl_prdsys folder.

2. Create Key Database Repository using runmqakm command. After successful execution, we can see key.p12 files generated
   runmqakm -keydb -create -db key.p12 -pw ms1aza -type p12 -expire 7300 -fips

3. Create new certificate request for prdsys server as shown below:
   runmqakm -certreq -create -db key.p12 -pw ms1aza -label prdsys -dn "CN=prdsys.mas.gov.sg, OU=CBD, O=ITD, L=PROD, S=EAPPS, C=SG" -size 2048 -sig_alg SHA256WithRSA -file prdsys_certreq.arm -fips

4. List the recently created certificate request in last step using list option.
   runmqakm -certreq -list -db key.p12 -pw ms1aza

5. Copy cert request file (arm file generated in Step 3) using ASCII mode and send it to MAS CA for signing and backup all files in the ssl_prdsys folder to reuse if any issue in further steps.

6. After receiving Signed Certificate along with CA Certificate, transfer it into Server to same ssl_prdsys folder. Please follow the steps in Section 7.22.1 to generate 3 certificates (root, intermediate, signed) before adding to keystore.

7. Add Intermediate CA Cert using below command into Signer certificate store.
   runmqakm -cert -add -db key.p12 -pw ms1aza -file INTERMEDIATECA.cer -label IntermediateCA -type p12 -format ascii -trust enable

8. Add Root CA Certificate using below command into Signer certificate store.
   runmqakm -cert -add -db key.p12 -pw ms1aza -file ROOTCA.cer -label RootCA -type p12 -format ascii -trust enable

9. Receive the Signed Certificate into Key Database using receive option.
   runmqakm -cert -receive -db key.p12 -pw ms1aza -file prdsys_signed.cer -type p12 -format ascii -default_cert yes

10. Now list the entire certificate present in key database file for prdsys.
    runmqckm -cert -list -db key.p12 -pw ms1aza -type p12

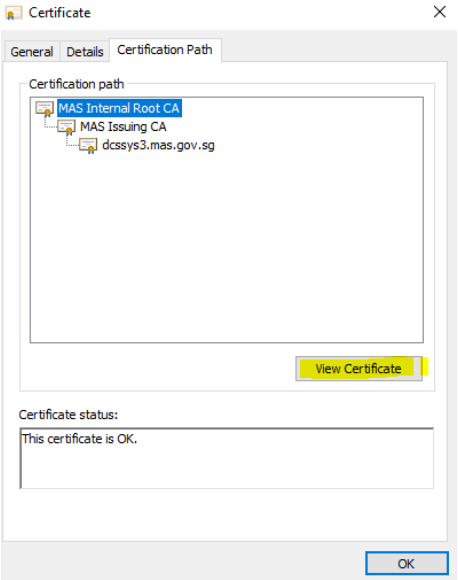
11. Convert p12 file into JKS format using below command.
    runmqckm -keydb -convert -db key.p12 -pw ms1aza -target key.jks -new_pw ms1aza -type p12 -new_format jks -expire 7300
    runmqckm -cert -list -db key.jks -pw ms1aza -type jks

12. Infra Team to get root access and then su to ms1az account id
13. Go to folder: /mas_appl/aza/ssl and take backup of existing key.jks
14. Copy new key.jks file to /mas_appl/aza/ssl location

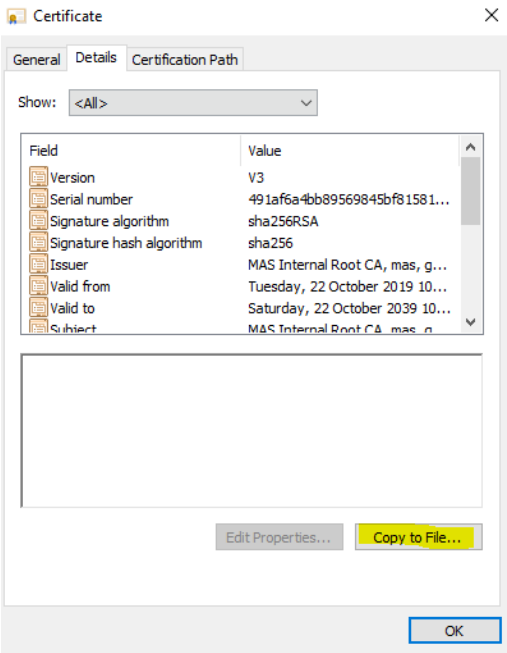
7.22.1 Steps to generate Root/Intermediate/Signed CA certs

7.22.1.1 Generate Root CA

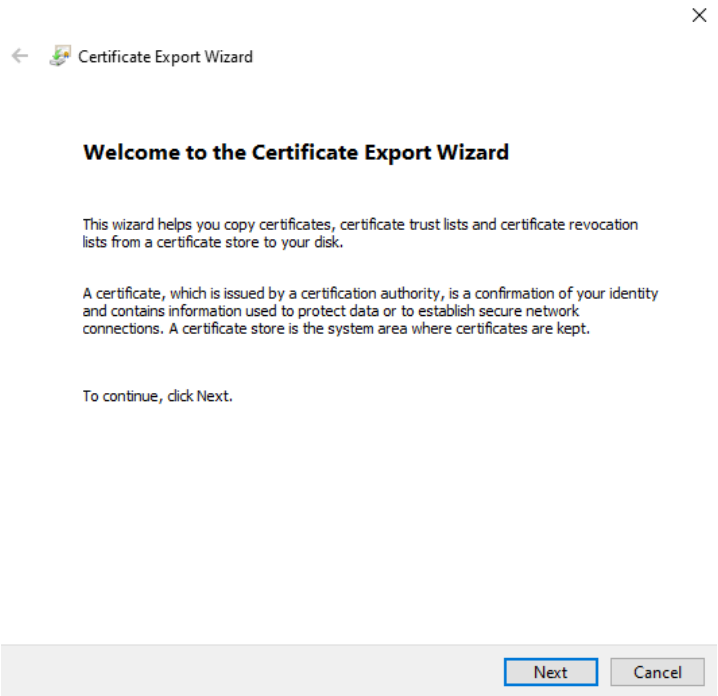
1. Double click on signed certificate (*_cetreq.arm.cer) received
2. Select MAS Internal Root CA on Certificate Path and Click View Certificate



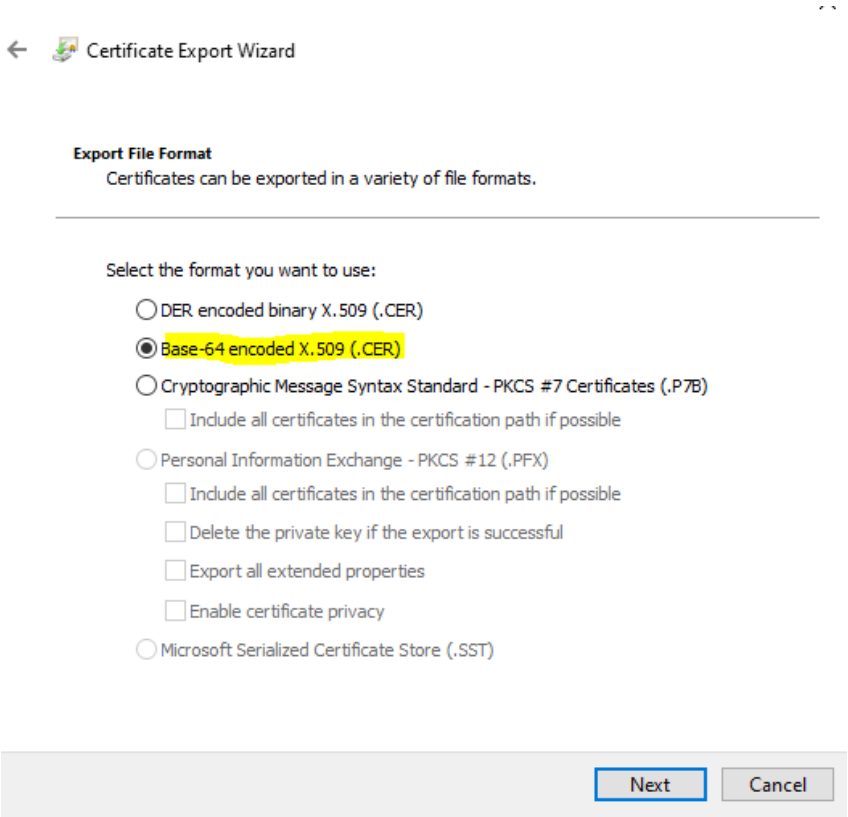
3. Click Copy to File on **Details** tab



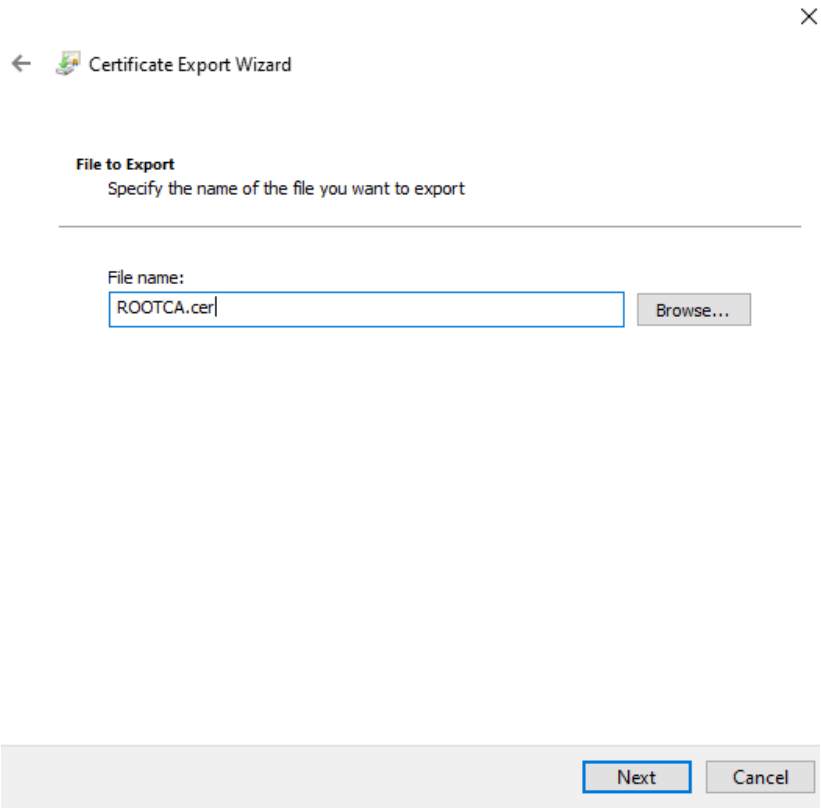
4. Click Next



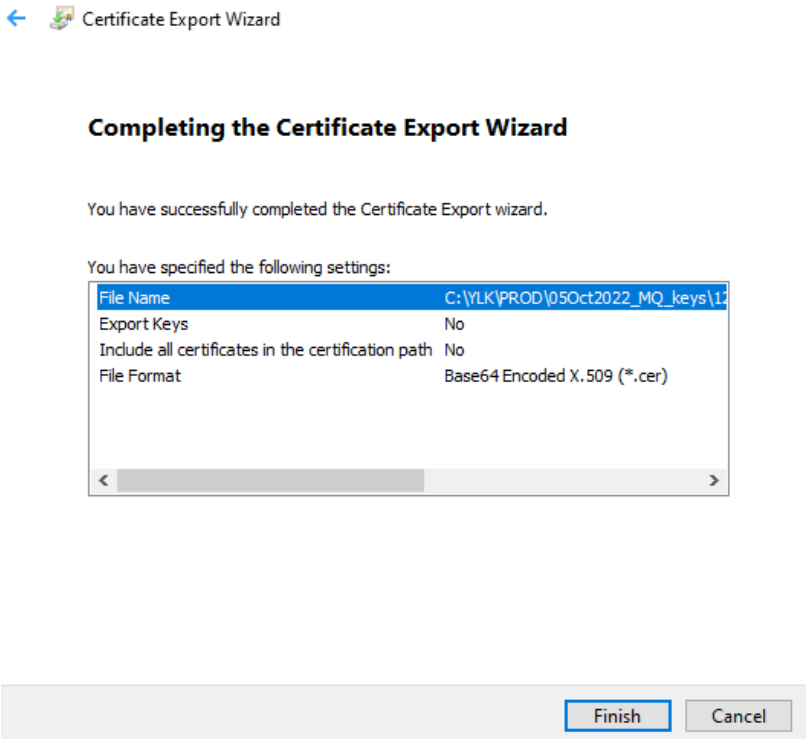
5. Select Base-64 encoded X.509 (.CER) and Click Next



6. Give File name: ROOTCA.cert and Click Next

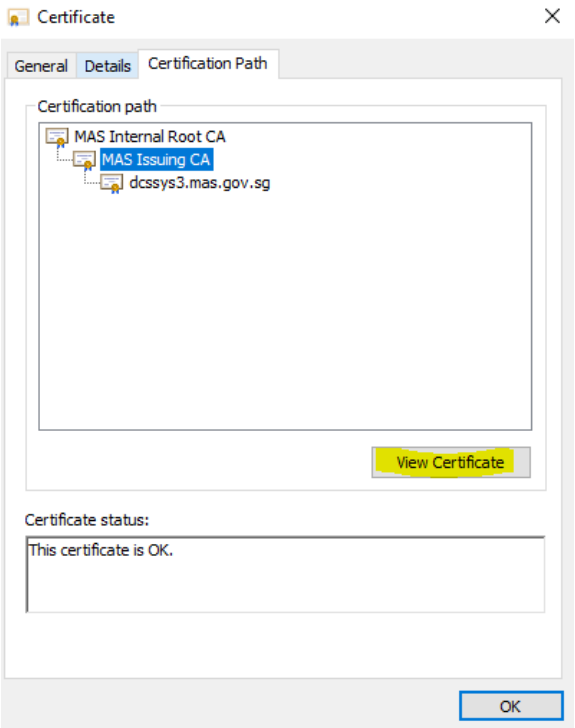


7. Click Finish.

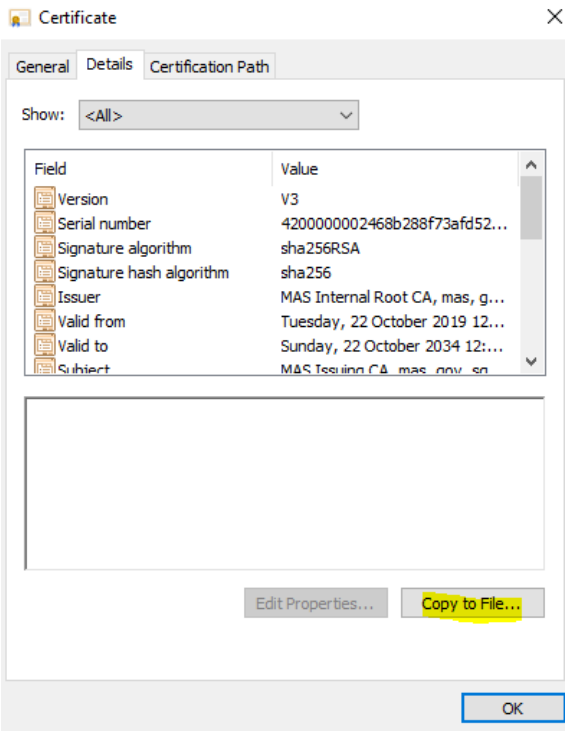


7.22.1.2 Generate Intermediate CA

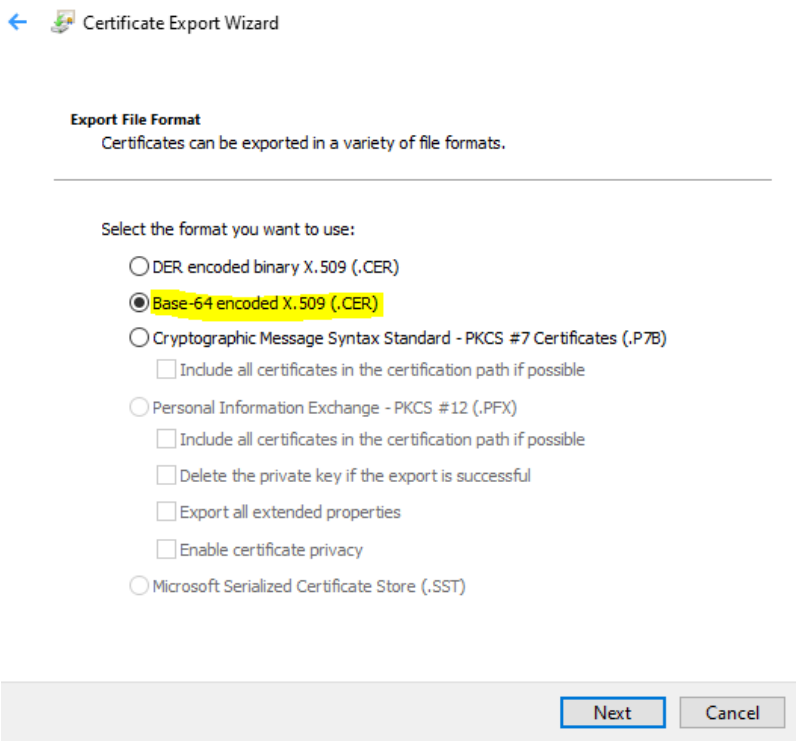
1. Double click on signed certificate (*_cetreq.arm.cer) received
2. Select MAS Issuing CA on Certificate Path and Click View Certificate



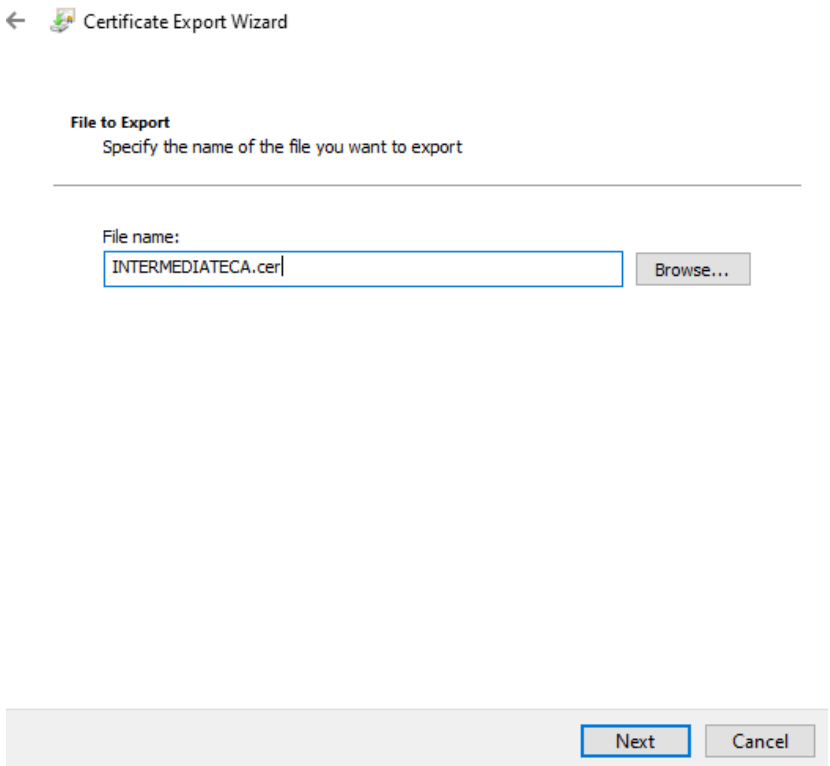
3. Click Copy to File on Details tab



4. Select Base-64 encoded X.509 (.CER) and Click Next

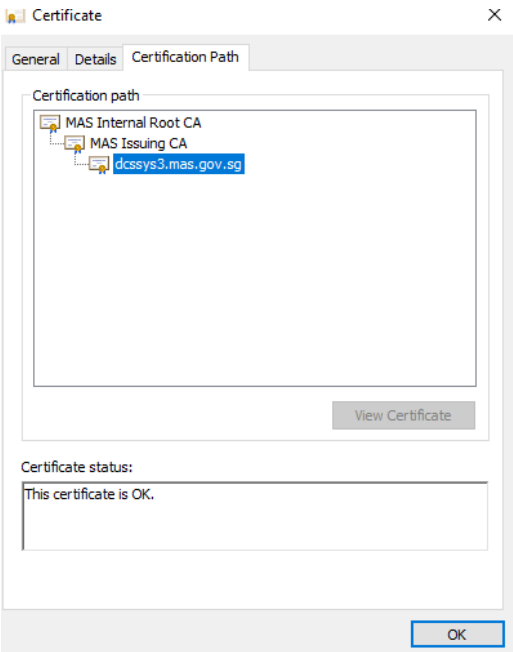


5. Give File name: INTERMEDIATECA.cer and Click Next and then Finish

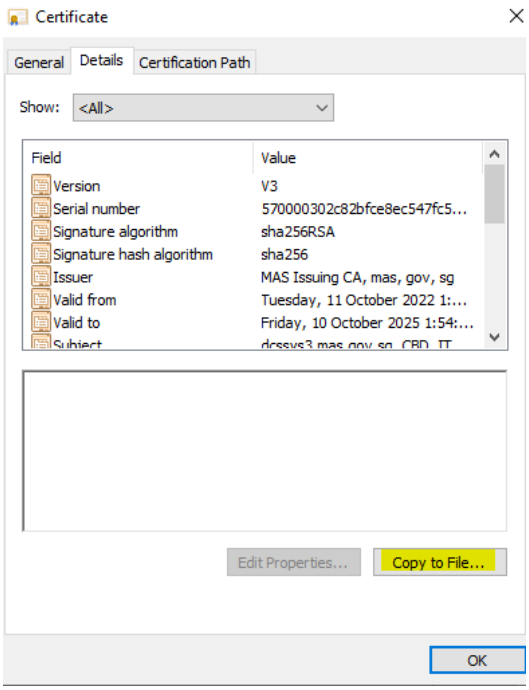


7.22.1.3 Generate signed CA

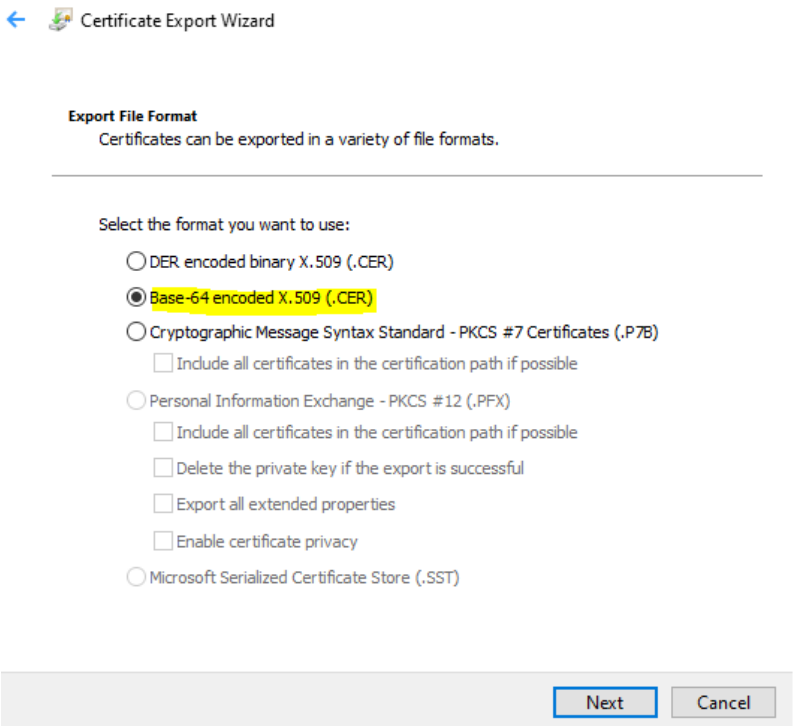
1. Double click on signed certificate (*_cetreq.arm.cer) received
2. Select MAS Issuing CA on Certificate Path and Click View Certificate



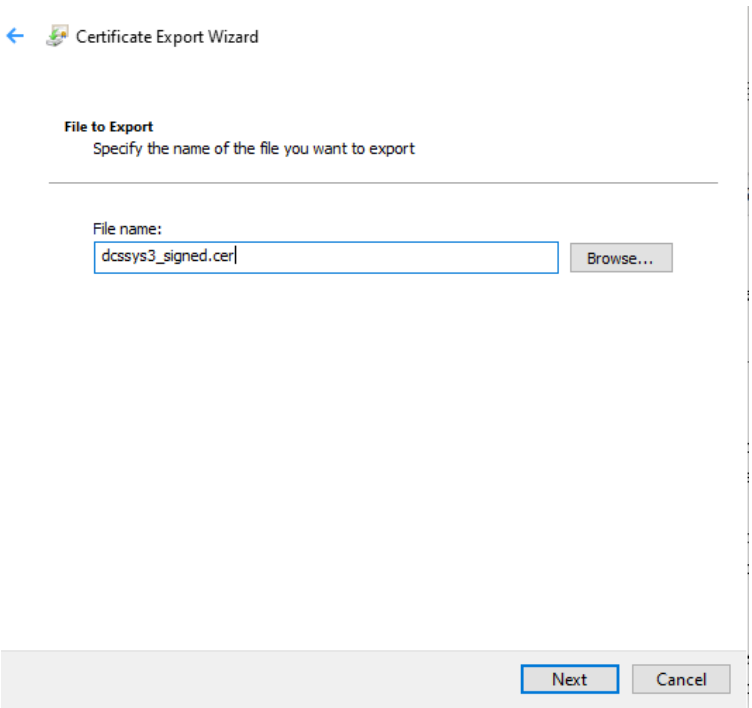
3. Click Copy to File on Details tab



4. Select Base-64 encoded X.509 (.CER) and Click Next



5. Give File name: servername_signed.cer and Click Next and then Finish



7.23 CMTB and SGS (Infra)

Please refer the document "CMTB_SGS(Infra)_SDS_v1.1.docx" for below details

- CMTB and SGS (Infra) – Issuance Calendar, Announcement, Subscription, Auction Results, Closing Price
- CMTB Outstanding amount
- SGS (Infra) and Green SGS (Infra) Outstanding amount by month and year
- SGS Issuance/Redemption amount for each Security Category by month and year

7.24 Syndication

Please refer the document "Syndication_SDS_v1.0.docx" for below details

- Closing Price

7.25 Upload File Size Restrictions in eApps

All file upload in eApps application should check for file size restrictions before performing antivirus scan.

There are 4 upload files available in eApps application

1. T-Bills/Bonds Bid Submission Upload
2. Closing Price Upload
3. ERF Bid Submission Upload
4. SORA Rate Upload

Maximum file size (MB) allowed is stored in ABA0032_EAPPS_CONFIG table in MNET DB.

ABA0032_KEY	ABA0032_VALUE
maxfile.sizemb	8

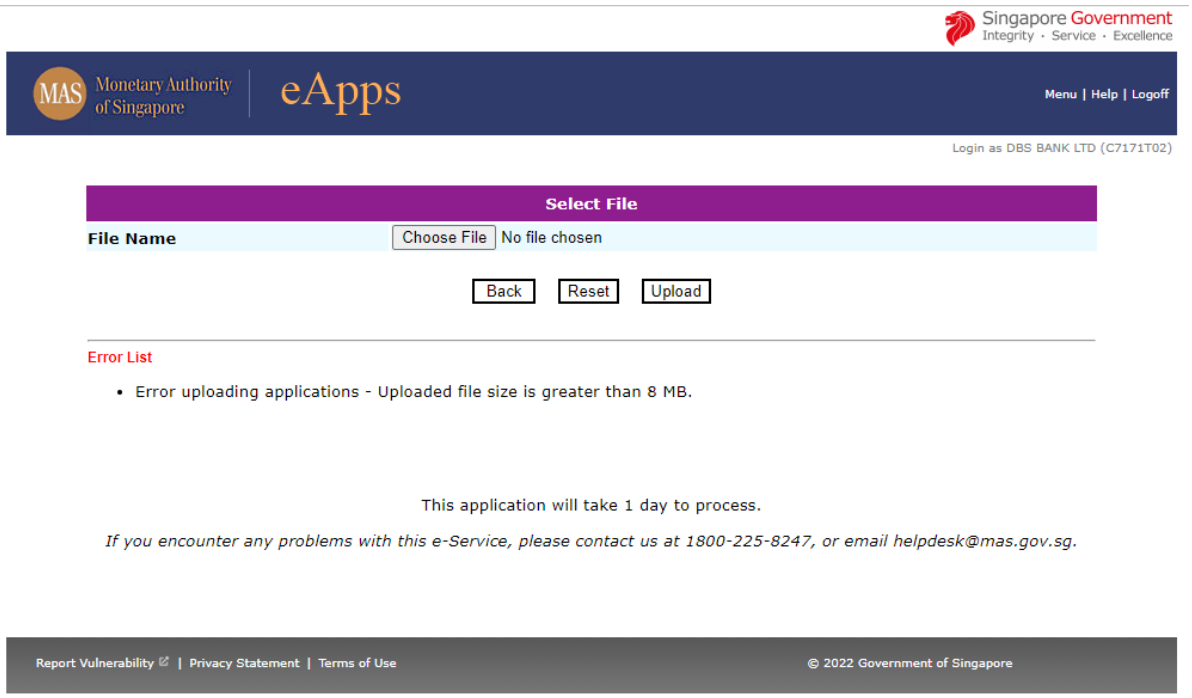
Error Message:

Error message "**Uploaded file size is greater than 8 MB.**" will be shown if the uploaded file size is greater than configured file size in DB. (value '8' in the error message is retrieved from the ABA0032_EAPPS_CONFIG table)

Sample screenshot for

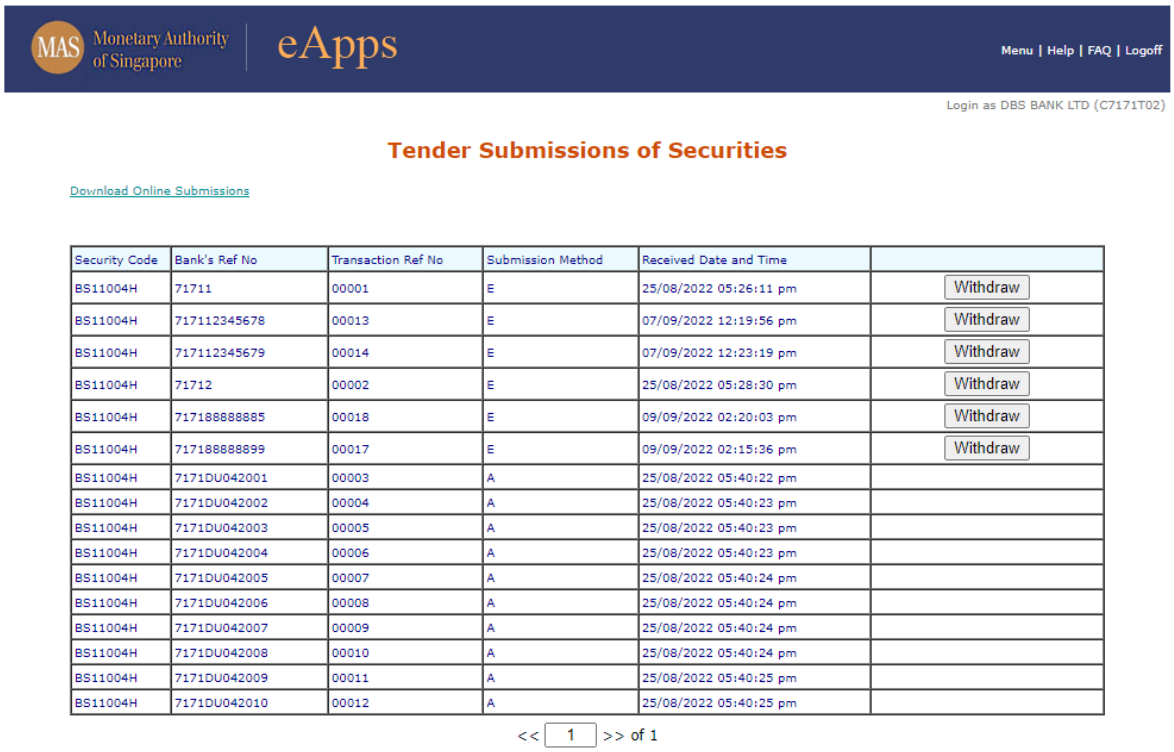
1. T-Bills/Bonds Bid Submission Upload
2. Closing Price Upload
3. SORA Rate Upload

Sample screenshot for ERF Bid Submission Upload



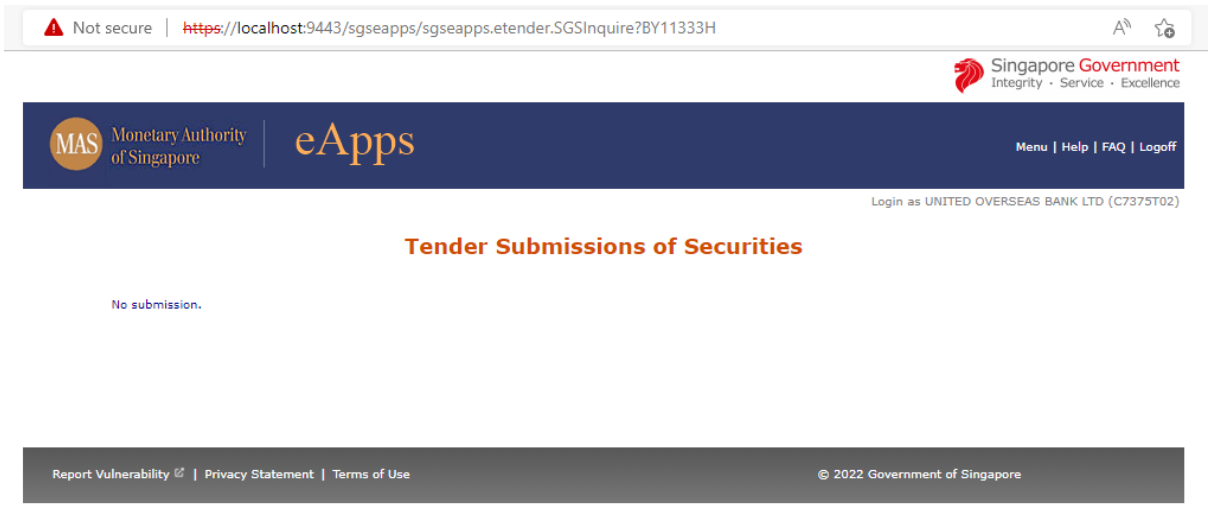
7.26 Submission Inquiry changes in eApps

1. When there are no Retail bids processed via new Retail Bid File Process Batch job and only there are bids submitted via eApps Online (Web submission/Upload File) the Submission Inquiry page will be displayed as below



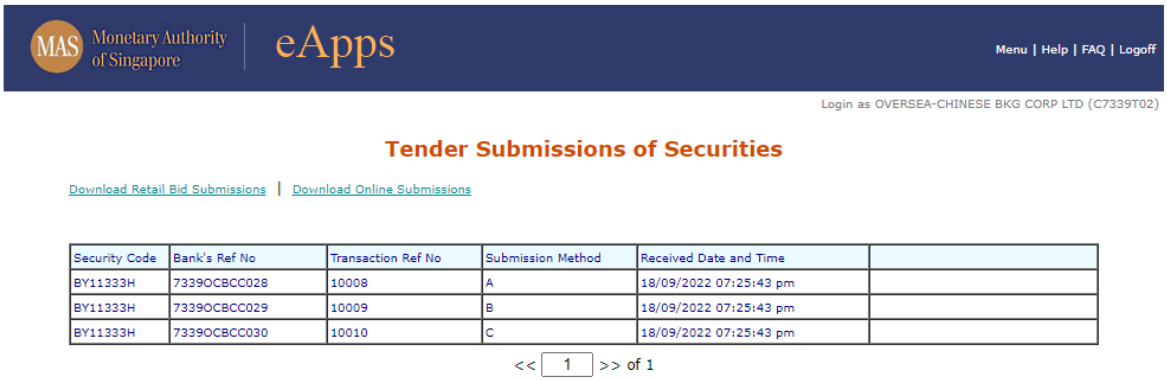
Security Code	Bank's Ref No	Transaction Ref No	Submission Method	Received Date and Time	
BS11004H	71711	00001	E	25/08/2022 05:26:11 pm	Withdraw
BS11004H	717112345678	00013	E	07/09/2022 12:19:56 pm	Withdraw
BS11004H	717112345679	00014	E	07/09/2022 12:23:19 pm	Withdraw
BS11004H	71712	00002	E	25/08/2022 05:28:30 pm	Withdraw
BS11004H	717188888885	00018	E	09/09/2022 02:20:03 pm	Withdraw
BS11004H	717188888899	00017	E	09/09/2022 02:15:36 pm	Withdraw
BS11004H	7171DU042001	00003	A	25/08/2022 05:40:22 pm	
BS11004H	7171DU042002	00004	A	25/08/2022 05:40:23 pm	
BS11004H	7171DU042003	00005	A	25/08/2022 05:40:23 pm	
BS11004H	7171DU042004	00006	A	25/08/2022 05:40:23 pm	
BS11004H	7171DU042005	00007	A	25/08/2022 05:40:24 pm	
BS11004H	7171DU042006	00008	A	25/08/2022 05:40:24 pm	
BS11004H	7171DU042007	00009	A	25/08/2022 05:40:24 pm	
BS11004H	7171DU042008	00010	A	25/08/2022 05:40:24 pm	
BS11004H	7171DU042009	00011	A	25/08/2022 05:40:25 pm	
BS11004H	7171DU042010	00012	A	25/08/2022 05:40:25 pm	

2. When there are no Retail bids processed via new Retail Bid File Process Batch job and there are no bids submitted via eApps Online (Web submission/Upload File) the Submission Inquiry page will be displayed as below



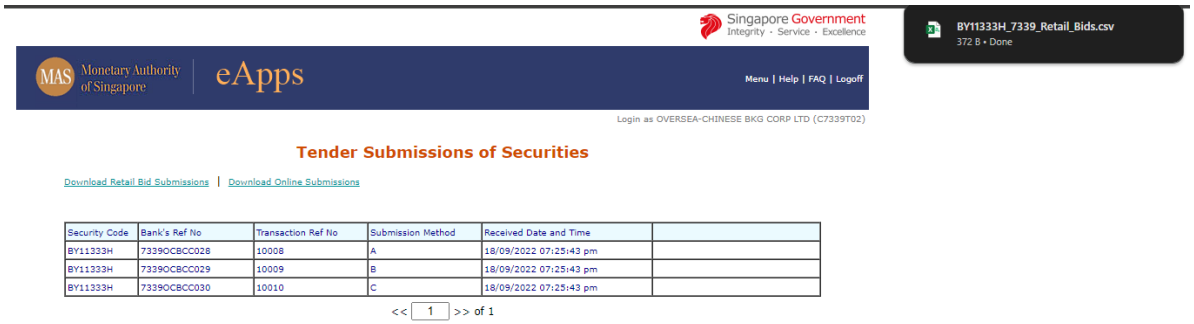
3. When there are Retail bids processed via new Retail Bid File Process Batch job and there are bids submitted via eApps Online (Web submission/Upload File), the Submission Inquiry page will be displayed as below only for the 3 local PD's (DBS, OCBC and UOB) and MDD login.

   (Note: Retail bids files are received via MEFT Access only from 3 local PDs for CASH, CPFIS and SRS)

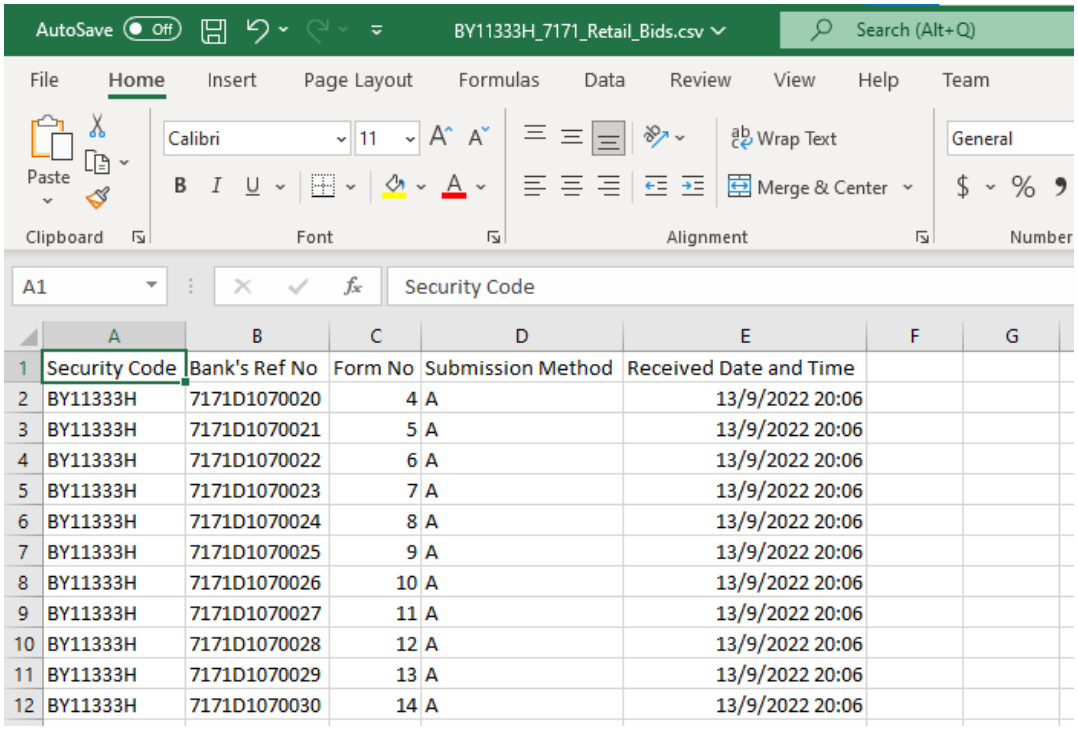


Security Code	Bank's Ref No	Transaction Ref No	Submission Method	Received Date and Time	
BY11333H	7339OCBCC028	10008	A	18/09/2022 07:25:43 pm	
BY11333H	7339OCBCC029	10009	B	18/09/2022 07:25:43 pm	
BY11333H	7339OCBCC030	10010	C	18/09/2022 07:25:43 pm	

"Download Retail Bid Submissions" and "Download Online Submissions" link will be displayed on top of the table.
On clicking the "Download Retail Bid Submissions" link, a csv file will be downloaded with file name as "<SecurityCode>_<BankCode>_Retail_Bids.csv".
This csv will contain only the retail bids received via the MEFT Access from PDs and processed via new Retail Bid File Process Batch job using the metadata stored in ABA0004_RETAIL_BID_TRANS@MNET table.



Security Code	Bank's Ref No	Transaction Ref No	Submission Method	Received Date and Time	
BY11333H	7339OCBCC028	10008	A	18/09/2022 07:25:43 pm	
BY11333H	7339OCBCC029	10009	B	18/09/2022 07:25:43 pm	
BY11333H	7339OCBCC030	10010	C	18/09/2022 07:25:43 pm	

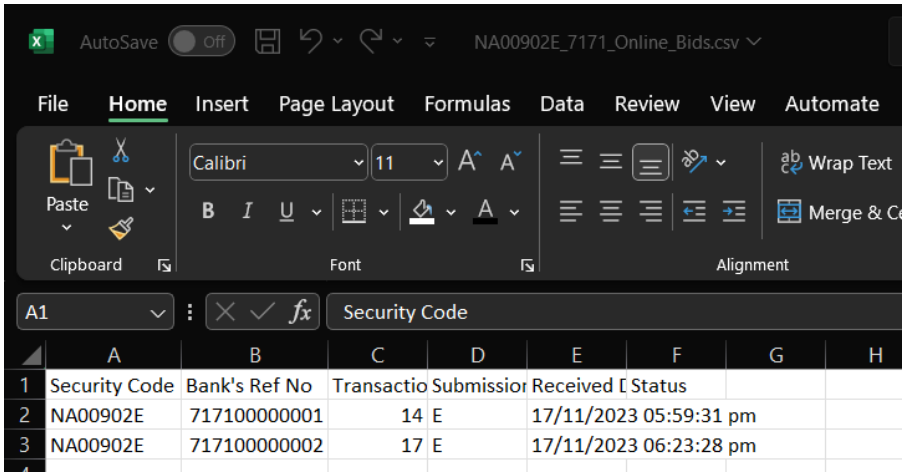


	A	B	C	D	E	F	G
1	Security Code	Bank's Ref No	Form No	Submission Method	Received Date and Time		
2	BY11333H	7171D1070020	4	A	13/9/2022 20:06		
3	BY11333H	7171D1070021	5	A	13/9/2022 20:06		
4	BY11333H	7171D1070022	6	A	13/9/2022 20:06		
5	BY11333H	7171D1070023	7	A	13/9/2022 20:06		
6	BY11333H	7171D1070024	8	A	13/9/2022 20:06		
7	BY11333H	7171D1070025	9	A	13/9/2022 20:06		
8	BY11333H	7171D1070026	10	A	13/9/2022 20:06		
9	BY11333H	7171D1070027	11	A	13/9/2022 20:06		
10	BY11333H	7171D1070028	12	A	13/9/2022 20:06		
11	BY11333H	7171D1070029	13	A	13/9/2022 20:06		
12	BY11333H	7171D1070030	14	A	13/9/2022 20:06		

BY11333H_7171_Retail_Bids.csv

On clicking the "Download Online Submissions" link, a csv file will be downloaded with file name as "<SecurityCode>_<BankCode>_Online_Bids.csv".
This csv will contain only the online bids submitted via eApps Online which stored in ABA0005_RSA_TEXT_ENC_TRANS@MNET table.

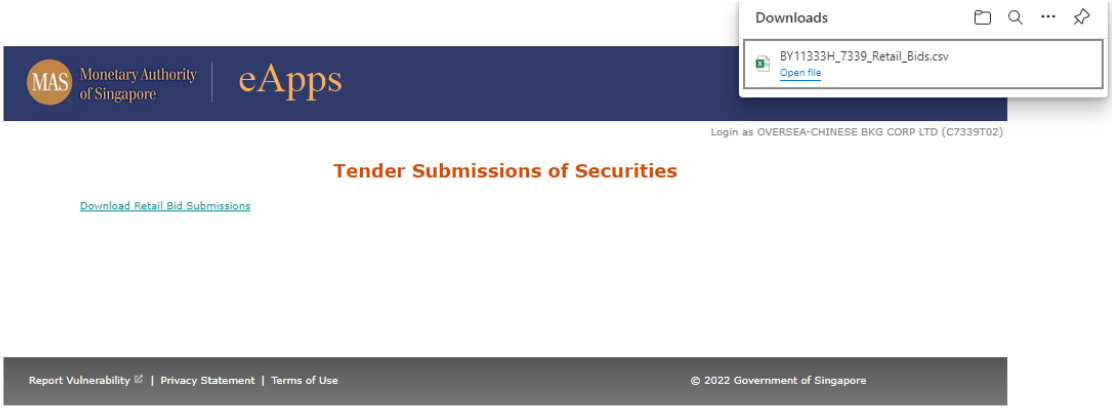
Security Code	Bank's Ref No	Transaction Ref No	Submission Method	Received Date and Time	
NA00902E	717100000001	00014	E	17/11/2023 05:59:31 pm	
NA00902E	717100000002	00017	E	17/11/2023 06:23:28 pm	

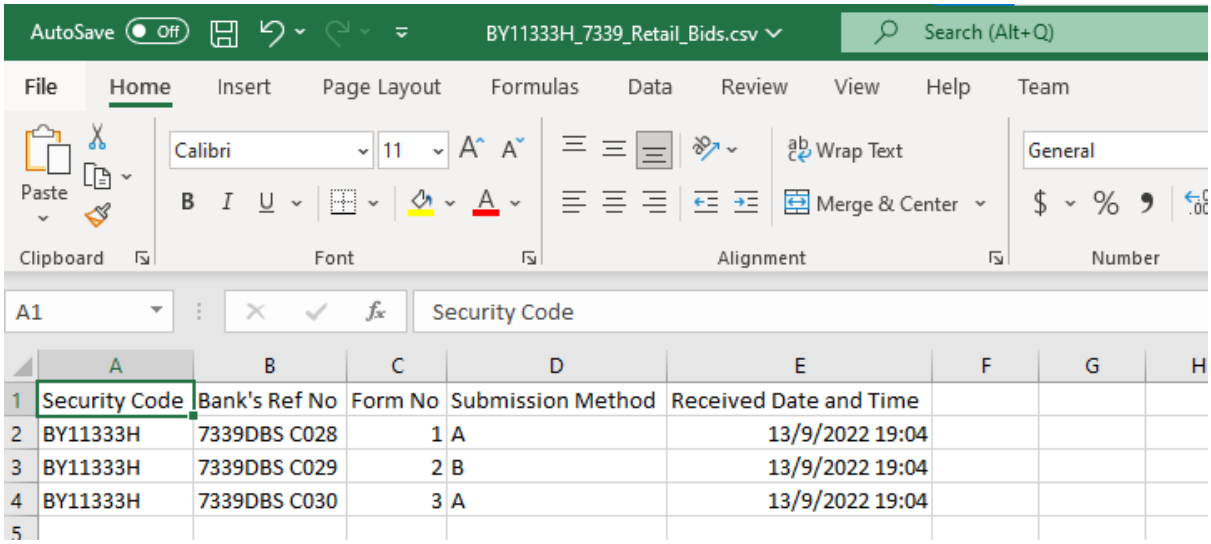


NA00902E_7171_Online_Bids.csv

4. When there are Retail bids processed via new Retail Bid File Process Batch job and there are no bids submitted via eApps Online (Web submission/Upload File), the Submission Inquiry page will be displayed as below only for the 3 local PD's (DBS, OCBC and UOB) and MDD login.

On clicking the link, a csv file will be downloaded with file name as "<SecurityCode>_<BankCode>_Retail_Bids.csv".
This csv will contain only the retail bids received via the MEFT Access from PDs and processed via new Retail Bid File Process Batch job using the metadata stored in ABA0004_RETAIL_BID_TRANS@MNET table.





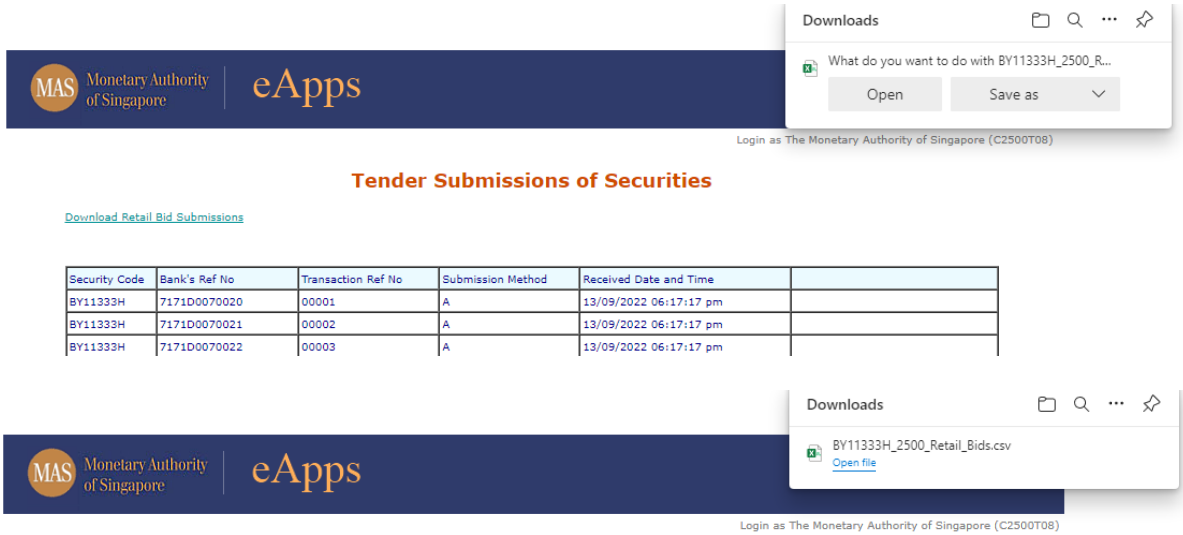
	A	B	C	D	E	F	G	H
1	Security Code	Bank's Ref No	Form No	Submission Method	Received Date and Time			
2	BY11333H	7339DBS C028	1	A	13/9/2022 19:04			
3	BY11333H	7339DBS C029	2	B	13/9/2022 19:04			
4	BY11333H	7339DBS C030	3	A	13/9/2022 19:04			
5								

BY11333H_7339_Retail_Bids.csv

5. MDD login - Submission Inquiry page will display all the Bids submitted by 13 PDs and MDD

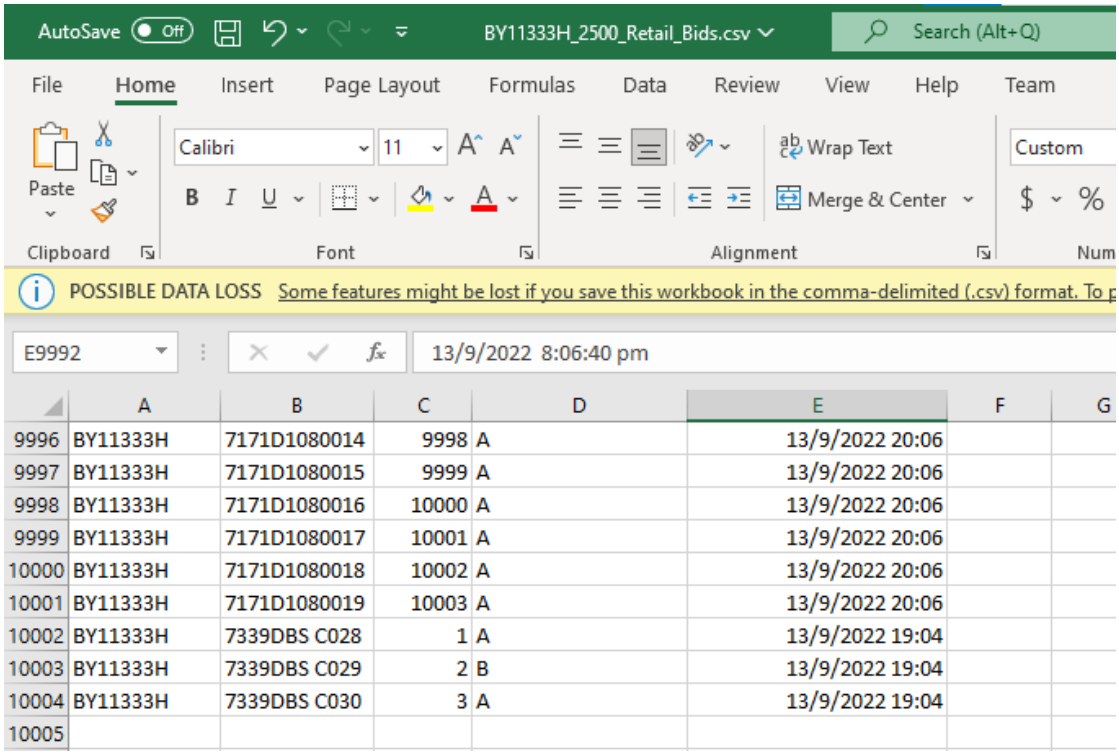
Security Code	Bank's Ref No	Transaction Ref No	Submission Method	Received Date and Time	
BY00902N	2500000001	00003	E	17/11/2023 05:46:55 pm	
BY00902N	7065000001	00004	E	17/11/2023 06:03:02 pm	
BY00902N	717100000001	00025	E	20/11/2023 02:56:58 pm	
BY00902N	7171000001	00005	E	17/11/2023 06:18:33 pm	Withdrawn at 20/11/2023 03:08:30 pm by C7171T02
BY00902N	721400000001	00012	E	20/11/2023 12:28:43 pm	
BY00902N	721400000002	00013	E	20/11/2023 12:29:35 pm	
BY00902N	721400000003	00014	E	20/11/2023 12:30:23 pm	
BY00902N	721400000004	00019	E	20/11/2023 01:09:11 pm	
BY00902N	721400000005	00015	E	20/11/2023 12:31:04 pm	
BY00902N	721400000007	00024	E	20/11/2023 02:55:39 pm	
BY00902N	723200000001	00001	E	17/11/2023 03:43:30 pm	
BY00902N	723200000002	00002	E	17/11/2023 03:44:24 pm	
BY00902N	733900000001	00016	E	20/11/2023 12:47:48 pm	
BY00902N	737500000001	00017	E	20/11/2023 12:55:15 pm	

On clicking the "Download Retail Bid Submissions" link, a csv file will be downloaded with file name as "<SecurityCode>_<BankCode>_Retail_Bids.csv".
This csv will contain only all the retail bids received via the MEFT Access from 3 PDs and processed via new Retail Bid File Process Batch job using the metadata stored in ABA0004_RETAIL_BID_TRANS@MNET table.



Security Code	Bank's Ref No	Transaction Ref No	Submission Method	Received Date and Time	
BY11333H	7171D0070020	00001	A	13/09/2022 06:17:17 pm	
BY11333H	7171D0070021	00002	A	13/09/2022 06:17:17 pm	
BY11333H	7171D0070022	00003	A	13/09/2022 06:17:17 pm	

Security Code	Bank's Ref No	Transaction Ref No	Submission Method	Received Date and Time	
BY11333H	7171D0070020	00001	A	13/09/2022 06:17:17 pm	
BY11333H	7171D0070021	00002	A	13/09/2022 06:17:17 pm	
BY11333H	7171D0070022	00003	A	13/09/2022 06:17:17 pm	

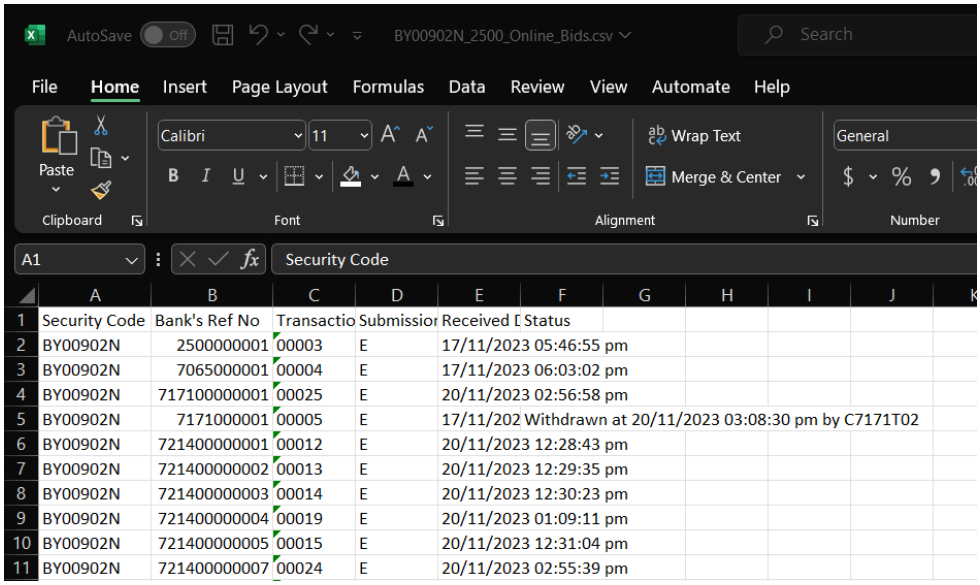


	A	B	C	D	E	F	G
9996	BY11333H	7171D1080014	9998	A	13/9/2022 20:06		
9997	BY11333H	7171D1080015	9999	A	13/9/2022 20:06		
9998	BY11333H	7171D1080016	10000	A	13/9/2022 20:06		
9999	BY11333H	7171D1080017	10001	A	13/9/2022 20:06		
10000	BY11333H	7171D1080018	10002	A	13/9/2022 20:06		
10001	BY11333H	7171D1080019	10003	A	13/9/2022 20:06		
10002	BY11333H	7339DBS C028	1	A	13/9/2022 19:04		
10003	BY11333H	7339DBS C029	2	B	13/9/2022 19:04		
10004	BY11333H	7339DBS C030	3	A	13/9/2022 19:04		
10005							

BY11333H_2500_Retail_Bids.csv

On clicking the "Download Online Submissions" link, a csv file will be downloaded with file name as "<SecurityCode>_<BankCode>_Online_Bids.csv".
This csv will contain only the online bids submitted via eApps Online which stored in ABA0005_RSA_TEXT_ENC_TRANS@MNET table.

Security Code	Bank's Ref No	Transaction Ref No	Submission Method	Received Date and Time	
BY00902N	2500000001	00003	E	17/11/2023 05:46:55 pm	
BY00902N	7065000001	00004	E	17/11/2023 06:03:02 pm	
BY00902N	717100000001	00025	E	20/11/2023 02:56:58 pm	
BY00902N	7171000001	00005	E	17/11/2023 06:18:33 pm	Withdrawn at 20/11/2023 03:08:30 pm by C7171T02
BY00902N	721400000001	00012	E	20/11/2023 12:28:43 pm	
BY00902N	721400000002	00013	E	20/11/2023 12:29:35 pm	
BY00902N	721400000003	00014	E	20/11/2023 12:30:23 pm	
BY00902N	721400000004	00019	E	20/11/2023 01:09:11 pm	



	A	B	C	D	E	F	G	H	I	J
1	Security Code	Bank's Ref No	Transactio	Submissior	Received [	Status				
2	BY00902N	2500000001	00003	E	17/11/2023 05:46:55 pm					
3	BY00902N	7065000001	00004	E	17/11/2023 06:03:02 pm					
4	BY00902N	717100000001	00025	E	20/11/2023 02:56:58 pm					
5	BY00902N	7171000001	00005	E	17/11/202 Withdrawn at 20/11/2023 03:08:30 pm by C7171T02					
6	BY00902N	721400000001	00012	E	20/11/2023 12:28:43 pm					
7	BY00902N	721400000002	00013	E	20/11/2023 12:29:35 pm					
8	BY00902N	721400000003	00014	E	20/11/2023 12:30:23 pm					
9	BY00902N	721400000004	00019	E	20/11/2023 01:09:11 pm					
10	BY00902N	721400000005	00015	E	20/11/2023 12:31:04 pm					
11	BY00902N	721400000007	00024	E	20/11/2023 02:55:39 pm					

BY00902N_2500_Online_Bids.csv

7.27 Unpublish results at MAS Website

- Overall results page in eApps, before Releasing results in eAuction

Result of Auctions for Securities

14 Sep 22

Latest

- **MAS Bills** No Issues Available
- **Floating Rate Notes** No Issues Available
- **Treasury Bill (6-Mth)** No Issues Available
- **Cash Management Treasury Bills** No Issues Available
- **Treasury Bill (1-Yr)/SGS Bond** No Issues Available

- Overall results page in eApps, after Releasing results in eAuction

Result of Auctions for Securities

14 Sep 22

Latest

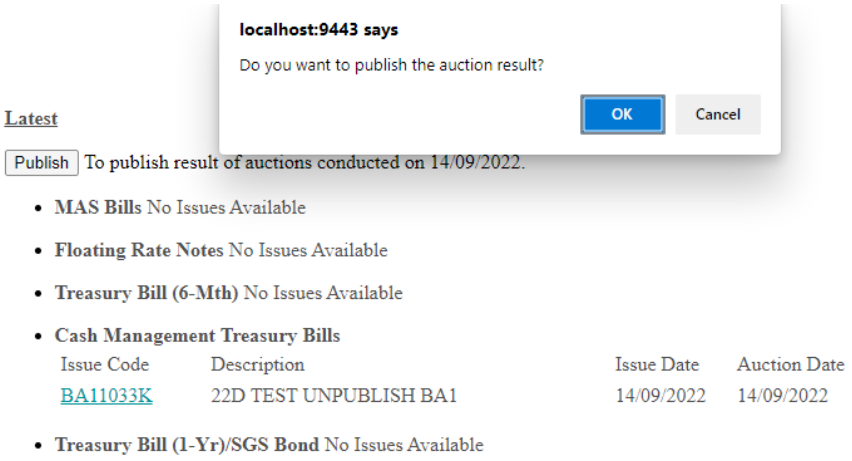
Publish To publish result of auctions conducted on 14/09/2022.

- **MAS Bills** No Issues Available
- **Floating Rate Notes** No Issues Available
- **Treasury Bill (6-Mth)** No Issues Available
- **Cash Management Treasury Bills**

Issue Code	Description	Issue Date	Auction Date
BA11033K	22D TEST UNPUBLISH BA1	14/09/2022	14/09/2022

- **Treasury Bill (1-Yr)/SGS Bond** No Issues Available

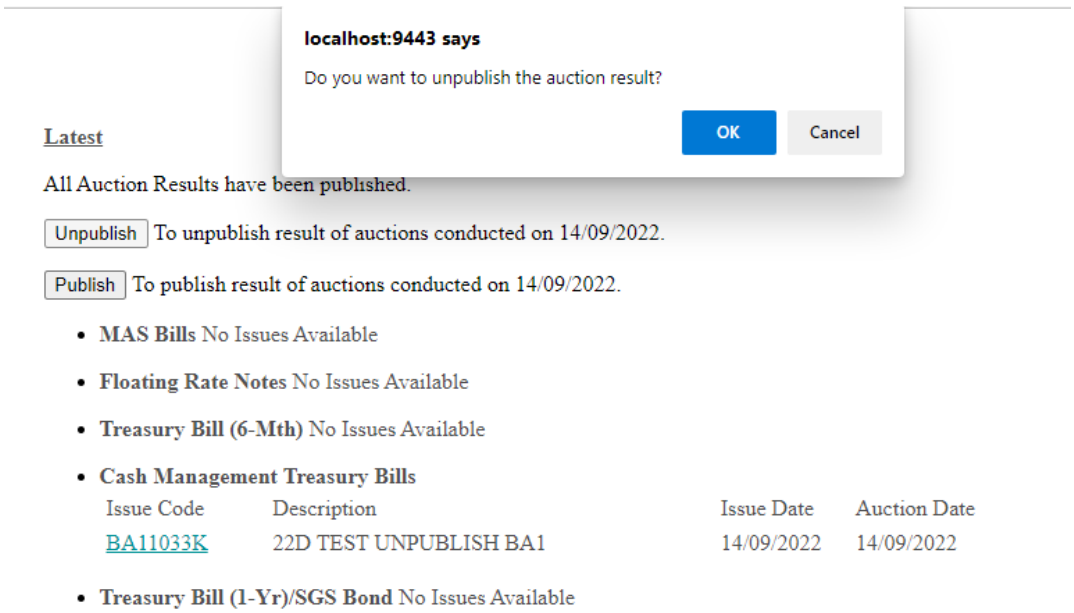
- After clicking on Publish button, a confirmation box will be shown as below



- After clicking Ok in the confirmation box, the below screen will be displayed along with **"Unpublish" button to unpublish the results**.

- When accessing the Overall results page again, the below page will be seen with both "Publish" and "Unpublish".

- After clicking the "Unpublish" button, a confirmation box will be shown as below



- After clicking Ok in the confirmation box, the below screen will be displayed

Result of Auctions for Securities

- Unpublishing of auction results is successful.

14 Sep 22

Latest

- **MAS Bills** No Issues Available
- **Floating Rate Notes** No Issues Available
- **Treasury Bill (6-Mth)** No Issues Available
- **Cash Management Treasury Bills** No Issues Available
- **Treasury Bill (1-Yr)/SGS Bond** No Issues Available

During Unpublish, below process will be performed

1. Below tables in the External Database will be updated

   a. **aba0001_security_master** – below fields will be updated to null for all securities with ABA0001_TENDER_DATE as today's date (system date)

      i. aba0001_cutoff_yield
      ii. aba0001_coy_price
      iii. aba0001_ave_yield
      iv. aba0001_ave_price
      v. aba0001_percent_coy
      vi. aba0001_percent_sub
      vii. aba0001_qty_applied
      viii. aba0001_qty_app_comp
      ix. aba0001_qty_app_noncomp
      x. aba0001_accrued_int
      xi. aba0001_mas_allotted
      xii. aba0001_nc_qty_allot
      xiii. aba0001_nc_percent
      xiv. aba0001_median_yield
      xv. aba0001_median_price
      xvi. aba0001_interest_rate (Only when the Issue Number = 1)
      xvii. aba0001_result_load_date

   b. **aqa0020_auction_results_report** – delete all the records for all the securities in aba0001_security_master table with ABA0001_TENDER_DATE as today's date (system date)

   c. **aba0007_detail_auction_result** – delete all the records for all the securities in aba0001_security_master table with ABA0001_TENDER_DATE as today's date (system date)

d. **aqa0008_sec_auction_para** – delete all the records for all the securities in aba0001_security_master table with ABA0001_TENDER_DATE as today's date (system date)

2. **wcm_sgs_auctionresults.dat** signal file will be created for MAS website team to sync the data again from eApps Database

3. Below stored procedures will be called

   a. call ms9gsw.aba0001_refresh@ocma
   b. call ms9gsw.abav001_refresh@ocma
   c. call ms9gsw.aba0001_refresh@ocmb
   d. call ms9gsw.abav001_refresh@ocmb

To Publish the results again in eApps, below is the SOP

1. Run Auction in eAuction (if required)
2. Release Results in eAuction
3. Publish results in eApps

7.28 WAF Changes

1. New servlets added to SGSWeb application in eApps, need to be whitelisted in WAF side in order to access those servlets via CorpPass login.

2. Attached the whitelisted list for eApps application for both UAT and PROD in WAF.

   Eerrvices_DEV_PROD.xlsx

3. Need to set the "Request Length" and "POST Data Length" to 10000000 for all new servlets added.

7.29 CorpPass TOPPAN Integration

7.29.1 Java Code changes

7.29.1.1 Set the below in sgs.properties

In sgs.properties, set the value for the following properties:

- mnet.toppan.eicProxy.auth.url= https://eservices.mas.gov.sg/eic/identity/auth?profile=MASPEAPP&client_id=SDVm6h6iOULfzqUym6iC&app_id=MAS-EAPPS&gateway=corppass&redirect_uri=https://eservices.mas.gov.sg/sgseapps/SGSAuthentication
- mnet.toppan.eicRP.API.url= https://eiceit.mas.gov.sg/identity/oidc/proxy/cp/userInfo/
- mnet.toppan.ecquaria-authToken=eyJhbGciOiJSUzI1NiIsImtpZCI6IjJBaldpOXJISTcvUDlMQUJFdU01SjYzRURBMXBEMEdESE0yNXd2OHRRTTBVTzZrcDE2Y2pVdkQwb21XTDhQOXQiLCJlbnRpdHlJZCI6ImR5ZzFlb283ZHdzMHVlTTFJN1p2RUEifQ.eyJpc3MiOiJFSVM6SldTIiwianRpIjoiZHlnMWVvbzdkd3MwdWVNMUk3WnZFQSIsImlhdCI6MTcxNTc1ODI3NywiZXhwIjoxNzc3ODc5ODc3LCJzdWIiOiJFQ1E6RUlDOlRPS0VOOkpXVCIsImNsYWltcyI6eyJ0ZW5hbnRJZCI6IlNEVm02aDZpT1VMZnpxVXltNmlDIiwiaWRwIjoiT0lEQyIsImdhdGV3YXkiOiJjcCIsInByb2ZpbGUiOiJNQVNQRUFQUCJ9fQ.e43FfTojORciTT70Zn9UP-xfasUdiIW-WA4aJB6WPncHiQPAhEkhfL7m5YB_u3qDgjhJ-Uq0CAPhbXiYHvsJMWAdrB7U5ye8OtKE0bdp1ttNmrKW2bUCSdq0UhX79iaYW1_1U83xWLTNt7PS51LFNebX3FrjE0YwpTd7ikVeVvfg7mG0bhP-Yt_0GZLIFVQC82xqqy_prxhHZ75RwQir0xWHfOHwNhVsimPYpEY6OzWWDRZduYK_2lSxliQ8Tn7gbSC8FwYqG9CCICFI0oFOioYXQ8Q_aYjeoGZJh9AG9pedEU53l9RxjN-0wcJ0Fbnkw7QtdyML3cYVZs7izmo8JA

mnet.toppan.eicProxy.auth.url – EIC Proxy Authentication URL
mnet.toppan.eicRP.API.url – EIC OIDC Relying Party (RP) REST API URL
mnet.toppan.ecquaria-authToken – EIC Security Token generated for eApps application to be included in the request header for API calls to EIC OIDC RP

7.29.1.2 TOPPAN EIC Proxy Authentication URL

1. The TOPPAN EIC Proxy Authentication URL is configured in "mnet.toppan.eicProxy.auth.url" property in sgs.properties file
2. UAT URL is as below
https://qeservices.mas.gov.sg/eic/identity/auth?profile=MASQEAPP&client_id=hiFUlcVDqEieVdMgJdPd&app_id=MAS-EAPPS&gateway=corppass&redirect_uri=https://ueservices.mas.gov.sg/sgseapps/SGSAuthentication&eic_state=bff9b178-f1e6-4d77-93ba-b629f59b0e0e&ecquaria_correlationId=3b29a595-8d58-47bb-ac08-448ec865721a
3. In eApps application, the URL will be constructed as below
https://eservices.mas.gov.sg/eic/identity/auth?profile=MASEAPP&client_id=SDVm6h6iOULfzqUym6iC&app_id=MAS-EAPPS&gateway=corppass&redirect_uri=https://eservices.mas.gov.sg/sgseapps/SGSAuthentication&eic_sta

te=bff9b178-f1e6-4d77-93ba-b629f59b0e0e&ecquaria_correlationId=3b29a595-8d58-47bb-ac08-448ec865721a

profile – Profile name as configured in EIC for eApps application
client_id – Client ID assigned to Agency during onboarding
app_id – The app ID as configured in EIC for eApps application
gateway – Gateway for which the application is trying to authenticate
redirect_uri – A callback URL that is configured in EIC for eApps application
eic_state – A random UUID will be generated by eApps
ecquaria_correlationId – A random UUID will be generated by eApps

7.29.1.3 TOPPAN EIC OIDC Relying Party (RP) API URL

1. The TOPPAN EIC OIDC Relying Party (RP) API URL is configured in "mnet.toppan.eicRP.API.url" property in sgs.properties file
2. UAT URL is as below
   https://qeiceit.mas.gov.sg/identity/oidc/proxy/cp/userInfo/{userToken}?authInfoScope=all
3. In eApps application, the API URL will be constructed as below
   https://eiceit.mas.gov.sg/identity/oidc/proxy/cp/userInfo/{userToken}?authInfoScope=all

   userToken – User token received upon successful authentication
   authInfoScope – The scope to retrieve the authorization info

7.29.1.4 IBM WebSphere Configuration Steps for TOPPAN Integration

1. Login into the IBM WebSphere Administrative Console.
2. Expand Security and click SSL certificate and key management. Under Configuration settings, click Manage endpoint security configurations.
3. Select the appropriate outbound configuration to get to the (cell): prdsyscell02 management scope.
4. Under Related Items, click Key stores and certificates and click the CellDefaultTrustStore key store.
5. Under Additional Properties, click Signer certificates and Retrieve from Port.
6. In the Host field, enter eiceit.mas.gov.sg in the host name field, enter 443 in the Port field, and eiceit.mas.gov.sg_cert in the Alias field.
7. Click Retrieve Signer Information.
8. Verify that the certificate information is for a certificate that you can trust.
9. Click Apply and Save.

7.29.1.5 TOPPAN EIC OIDC Relying Party (RP) API Response Specification

Body Parameter	Description	Type	Mandatory
statusCode	Indicates if the OOB call is success or failure. For success, the value is 200.	Int	Yes
authStatus	Indicates if the OOB call is success or failure. For success, the value is "success".	String	Yes
authMethod	Indicates if the user uses a second-factor authentication during login and the method used. one of the possible values is • 1FA • 2FA SMS OTP • QR Code • Facial Biometrics	String	Yes
userInfo.nricFin	The login user NRIC or FIN, if returned by Corppass.	String	Yes
userInfo.uuid	The unique UUID of the login user, if returned by Corppass.	String	No
userInfo.cpAccType	Corppass account type if administrator or sub-admin or user or enquiry user.	String	Yes
userInfo. cpUidFullName	Full name of the user.	String	Yes
userInfo.isSPHolder	Whether the user is a Corppass holder, the user has activated their Corppass ID using Corppass.	Boolean	Yes
userInfo.cpEntId	The Corppass Entity ID (UEN or NON-UEN ID) of the entity to which the user belongs in Corppass.	String	Yes
userInfo. cpEntStatus	Status of the entity in Corppass as provided by issuance agencies. Possible values are "Registered", "De-Registered", "Withdrawn", blank (for NON-UEN scenario).	String	Yes
userInfo.cpEntType	Type of the entity the user belongs to, UEN or NON-UEN.	String	Yes
userInfo. cpNonUENRegNo	Entity registration number of NON-UEN entity which is entered during Corppass admin registration. Empty if the user is a UEN entity.	String	No
userInfo. cpNonUENName	Name of the NON-UEN entity which is entered during Corppass admin registration. Empty if the user is a UEN entity userInfo.cpNonUENCountry Country of incorporation of a NON-UEN entity which is provided during Corppass admin registration. Empty if the user is a UEN entity.	String	No
userInfo. cpNonUENCountry	Country of incorporation of a NON-UEN entity which is provided during Corppass admin registration. Empty if the user is a UEN entity.	String	No
authInfo	Base64-encoded format of auth info JSON string	String	No
tpAuthInfo	Base64-encoded format of auth info JSON string	String	No
expiresIn	The expiry of the access token in seconds.	String	Yes
accessToken	The OIDC access token issued by Corppass. This is an optional field that can be made available upon request.	String	Yes