COMMERCE AND ECONOMIC DEVELOPMENT BUREAU

THE GOVERNMENT OF

THE HONG KONG SPECIAL ADMINISTRATIVE REGION

SPECIFICATION FOR DATA INHERITANCE FROM ROCARS TO TDEC OF GOVERNMENT ELECTRONIC TRADING SERVICES (GETS) SYSTEM

Version 1.0

December 2009

Amendment History

Change Request Ref.	Description	Section / Page	Version no. / Date	Effective Date

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Objective

This specification is written to enable the Government Electronic Trading Services (GETS) Service Providers (SPs) to implement data inheritance functions such that common data from Road Cargo System (ROCARS) could be inherited to Import/Export Declarations (TDEC) of the GETS System. The GETS SPs are required to provide user-friendly functions to facilitate data inheritance in accordance with relevant GETS contracts. However, any third party value-added service providers in the market interested in the provision of the data inheritance functions will also find this document a valuable reference.

1.2 Reference Documents

1. Implementation Instructions of Government Electronic Trading Services (GETS) System
2. Implementation Instructions of the Road Cargo System (ROCARS) System-to-System Interface for Bulk Submission

1.3 Assumptions

1. For ease understand of the flow of data inheritance, the term “Shipper” is used throughout this specification to refer to the party who submits the electronic cargo information of ROCARS. This party can be the Importer/Exporter or his/her authorized agent. To avoid confusion, Shipper is the party who submits electronic cargo information of ROCARS and Importer/Exporter is the party who prepares the TDEC submission. In this specification, these terms cannot be used interchangeably.

2. If Importer/Exporter has proper access right to the electronic cargo information of ROCARS, the Importer/Exporter can download his/her own electronic cargo information from ROCARS for re-use and preparing TDEC submission.

3. If the electronic cargo information of ROCARS is submitted by his/her authorized agent, the Importer/Exporter does not have proper access right to the electronic cargo information. The authorized agent of the Importer/Exporter can download electronic cargo information from ROCARS and pass it to the Importer/Exporter for preparation of TDEC submission.

4. GETS SPs will validate all TDEC messages, regardless of whether they are prepared via data inheritance functions or not, according to the validation requirement as stipulated in the Implementation Instructions of GETS System before sending the messages to the Government.

5. This specification is by no mean exhaustive due to the complexity of different business scenarios for data inheritance from ROCARS to TDEC. GETS SPs shall endeavor to provide user-friendly data inheritance functions to meet various business scenarios so as to minimize data re-entry efforts by the Trading Community and to facilitate data re-use across trade-related documents to be submitted to the Government.

1.4 Background

ROCARS submission
By the end of 2009, Customs and Excise Department (C&ED) will launch a new system called ROCARS for clearance of cargo entering or leaving Hong Kong by trucks at Land Boundary Control Points (LBCPs).

Under the new system, a Shipper is required to submit electronic cargo information (ROCARS data) to ROCARS not more than 14 days in advance for road mode cargo imported into and exported out of Hong Kong at LBCPs. In return, ROCARS will generate a unique Customs Cargo Reference Number (CCRN) to the Shipper. Shipper will acknowledge this CCRN and pass the cargo concerned with the CCRN to the truck driver (or his/her authorized agents) who will physically convey the cargo across the LBCP. Subsequently, the truck driver is required to bundle his/her vehicle registration number (VRN) with the CCRN concerned and submit this bundling information to ROCARS at least 30 minutes before arriving the LBCP.

Shipper can submit ROCARS data via web interface or bulk submission channel of the Portal & XML Gateway Subsystem of ROCARS. Truck driver can submit bundling information via a dedicated interactive voice response system (IVRS) or the ROCARS web portal.

TDEC submission
Under the Import and Export (Registration) Regulations, an Importer/Exporter is required to lodge the import or export/re-export declaration to the Government within 14 days after the importation or exportation of any articles other than exempted articles.

Importer/Exporter is required to submit TDEC to the Government via the GETS SPs' systems.

Purpose for Data Inheritance (DI)
In order to provide convenience for Trading Community to re-use ROCARS data for preparing TDEC submission and enhance data accuracy on trade-related documents submitted to the Government, GETS SPs shall provide functions to facilitate proper inheritance of common data from ROCARS to TDEC under various business scenarios.

The common data between ROCARS and TDEC for the sake of creating temporary trade declarations include the following. In addition, the Transport Mode for TDEC shall be pre-set to "Road".

1. Consignee Name and Address (for exportation only)
2. Container Number
3. Goods Description
4. Type of Packages
5. Number of Packages
6. Net Weight or Gross Volume
7. Net Weight Unit or Gross Volume Unit
8. Vehicle Registration Number (VRN)
9. Customs Cargo Reference Number (CCRN)
10. Unique Consignment Reference (UCR)

2. DATA INHERITANCE OPERATION

2.1 Flow of Message

In ROCARS, a ROCARS message can be an electronic cargo information namely Import Consignment (AIM) for importation or Export Consignment (AEX) for exportation; or can be a bundling information namely Import Bundling (ACRID) for importation or Export Bundling (ACRED) for exportation. A Shipper is responsible to submit electronic cargo information using the message AIM or AEX whereas the truck driver is responsible to submit bundling information with VRN using the message ACRID or ACRED.

ROCARS will provide functions for the Shipper to extract his or her own data residing in ROCARS into bundle(s) of ROCARS messages. Then the Importer/Exporter may upload to GETS such extracted bundle(s) of ROCARS messages via GETS SP's trading community interface[1]. A bundle of ROCARS messages can be "one AIM + one ACRID" for importation; or "one AEX + one ACRED" for exportation. Details are described in Section 5.

2.2 Scenario of Data Inheritance

In some circumstances, multiple bundle(s) of ROCARS messages should be consolidated into one TDEC message. In other circumstances, a bundle of ROCARS messages should be split to prepare multiple TDEC messages. The following table illustrates the different scenarios for data inheritance.

#	Scenario	Conditions
1.	One bundle of ROCARS messages corresponding to a CCRN is mapped to one TDEC message	The bundle of ROCARS messages is having: - the goods belong to one Importer/Exporter for TDEC submission; - the total number of line items of corresponding HS code does not exceed the limit of a TDEC message; - only one VRN is specified; - all line items with HS codes categorized under the same Form Type of TDEC; and - the same consignee for exportation.
2.	Multiple bundles of ROCARS messages corresponding to multiple CCRNs are mapped to one TDEC message	The multiple bundles of ROCARS messages are having: - the goods belong to one Importer/Exporter for TDEC submission; - the total number of line items of corresponding HS code does not exceed the limit of a TDEC message; - the same voyage (same VRN, same arrival/departure date and time) of goods transportation; - all line items with HS codes categorized under the same Form Type of TDEC; and - the same consignee for exportation.
3.	One bundle of ROCARS messages	The bundle of ROCARS messages is having: - the goods belong to more than one Importer/Exporter for TDEC submission; or

[1] The trading community interface refers to the software and related procedures/specifications developed by GETS SP to facilitate electronic message preparation or receipt of messages sent/broadcasted by the Government at the Trading Community's end.

#	Scenario	Conditions
	corresponding to a CCRN is mapped to multiple TDEC messages	- the total number of line items of corresponding HS code exceeds the limit of a TDEC message after splitting of line items; or - any line item with HS code categorized under different Form Types of TDEC; or - the goods belong to more than one consignee for exportation.

3. ACTIVITY DIAGRAM

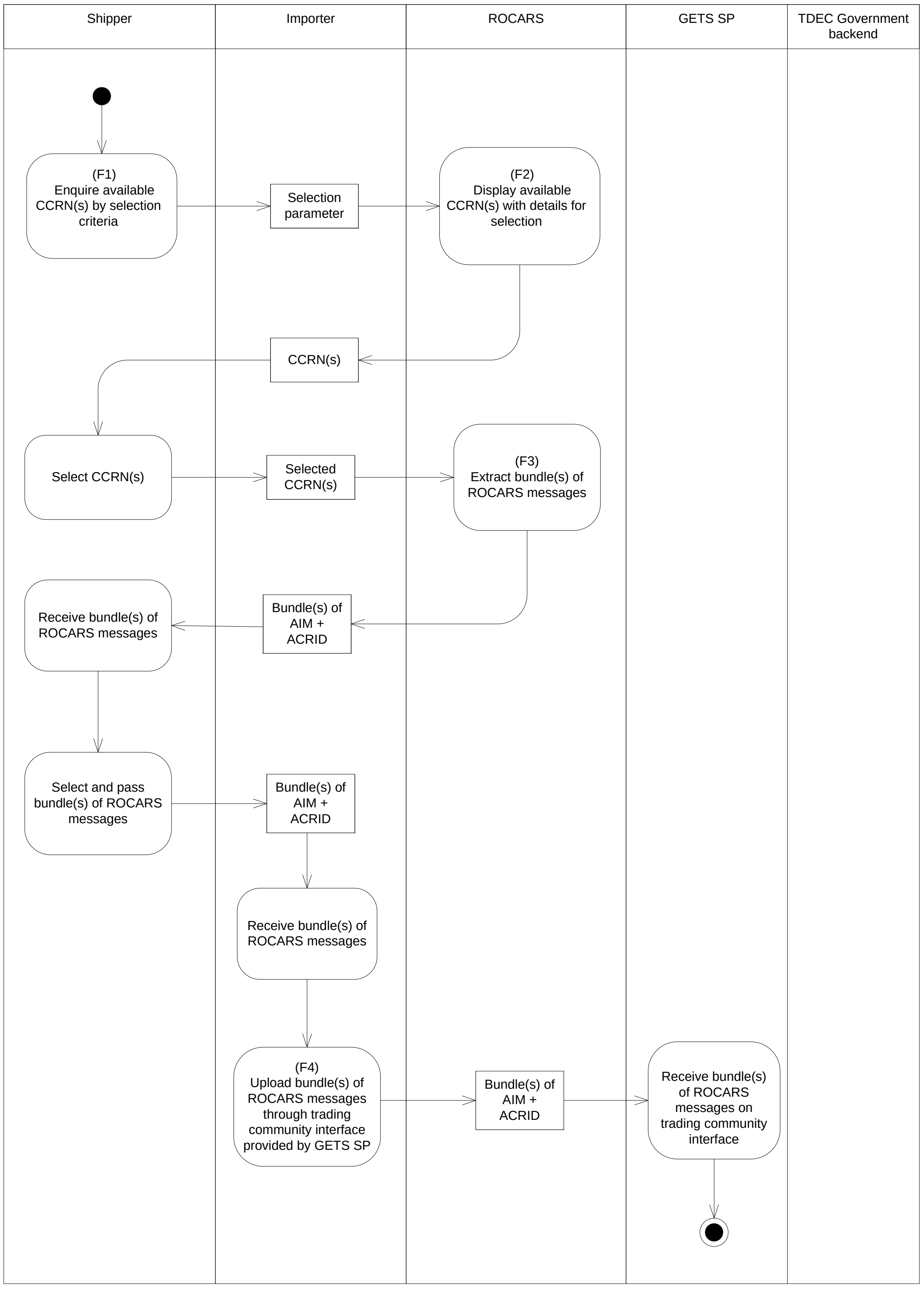
3.1 Importation

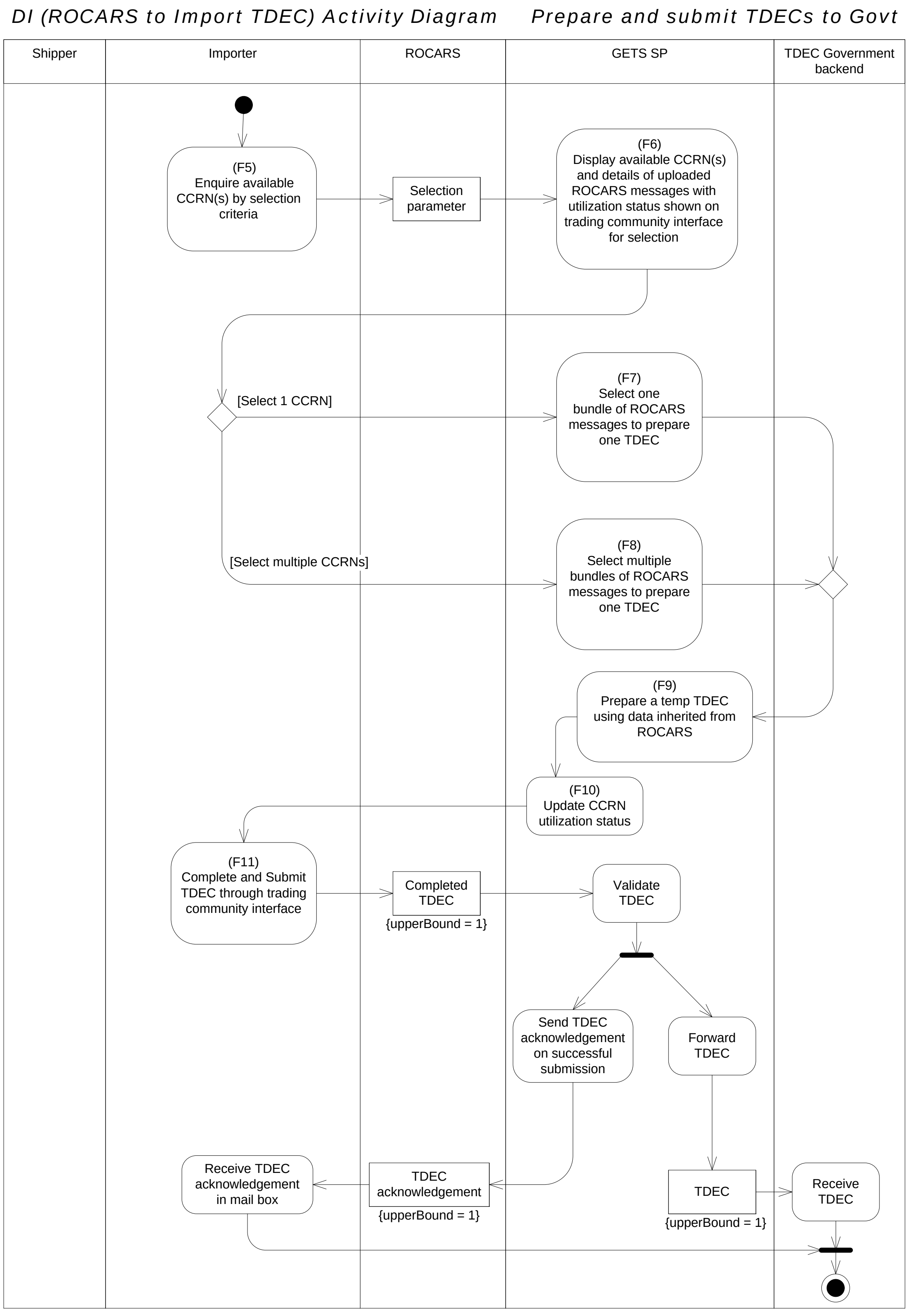
A high level overview of data inheritance activities for ROCARS to Import TDEC is illustrated in the following two activity diagrams. In the activity diagrams, the essential functions required for facilitating the data inheritance implementation are marked with “Fn” of which the detailed function descriptions can be referred to Section 4.2.

Firstly, in the diagram of “Upload ROCARS messages to SP”, a Shipper enters a list of selection parameters via a user-friendly interface of the ROCARS system to enquire the available CCRN(s) that the Shipper submitted earlier. Then the Shipper selects one or more CCRNs to extract the corresponding bundle(s) of ROCARS AIM and ACRID messages in a pre-defined XML file. After that, the Shipper passes the XML file to the corresponding importer by some means such as disk storage, CD/DVD etc with proper security protection. Finally, the importer uploads the XML file containing the bundle(s) of ROCARS messages to GETS via GETS SP’s trading community interface.

Secondly, in the diagram of “Prepare and submit TDECs to Govt”, the importer enters a list of selection parameters via a user-friendly interface in GETS SP’s trading community interface to enquire the available CCRN(s). Then the importer selects one or more CCRNs of the corresponding uploaded bundle(s) of ROCARS messages for preparing TDEC message(s). Then the importer prepares a temporary TDEC with common data inherited from the selected bundle(s) of ROCARS messages through GETS SP’s inheritance functions. The common data being inherited are listed in Section 5.5 with Consignee Name and Consignee Address excluded because they are applicable for inheritance in export TDEC only. Finally, the importer fills in other necessary information to complete a TDEC message and submit it to the Government via GETS SP.

DI (ROCARS to Import TDEC) Activity Diagram   Upload ROCARS messages to SP


Shipper
Importer
ROCARS
GETS SP
TDEC Government backend
(F1) Enquire available CCRN(s) by selection criteria
Selection parameter
(F2) Display available CCRN(s) with details for selection
CCRN(s)
Select CCRN(s)
Selected CCRN(s)
(F3) Extract bundle(s) of ROCARS messages
Receive bundle(s) of ROCARS messages
Bundle(s) of AIM + ACRID
Select and pass bundle(s) of ROCARS messages
Bundle(s) of AIM + ACRID
Receive bundle(s) of ROCARS messages
(F4) Upload bundle(s) of ROCARS messages through trading community interface provided by GETS SP
Bundle(s) of AIM + ACRID
Receive bundle(s) of ROCARS messages on trading community interface


DI (ROCARS to Import TDEC) Activity Diagram
Prepare and submit TDECs to Govt
Shipper
Importer
ROCARS
GETS SP
TDEC Government backend
(F5) Enquire available CCRN(s) by selection criteria
Selection parameter
(F6) Display available CCRN(s) and details of uploaded ROCARS messages with utilization status shown on trading community interface for selection
[Select 1 CCRN]
(F7) Select one bundle of ROCARS messages to prepare one TDEC
[Select multiple CCRNs]
(F8) Select multiple bundles of ROCARS messages to prepare one TDEC
(F9) Prepare a temp TDEC using data inherited from ROCARS
(F10) Update CCRN utilization status
(F11) Complete and Submit TDEC through trading community interface
Completed TDEC
{upperBound = 1}
Validate TDEC
Send TDEC acknowledgement on successful submission
Forward TDEC
Receive TDEC acknowledgement in mail box
TDEC acknowledgement
{upperBound = 1}
TDEC
{upperBound = 1}
Receive TDEC

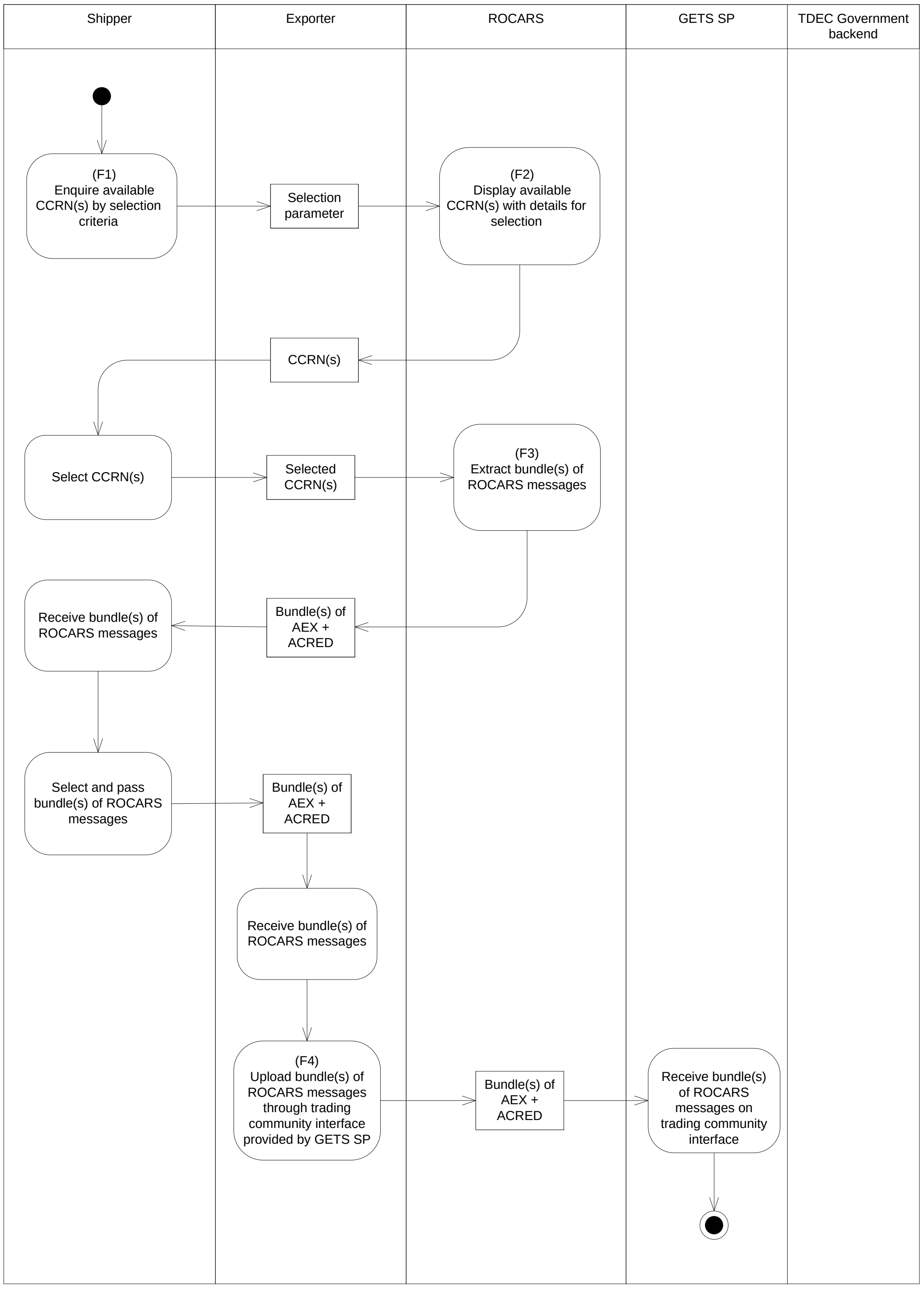
3.2 Exportation

A high level overview of data inheritance activities for ROCARS to Export TDEC is illustrated in the following two activity diagrams. In the activity diagrams, the essential functions required for facilitating the data inheritance implementation are marked with “Fn” of which the detailed function descriptions can be referred to Section 4.2.

Firstly, in the diagram of “Upload ROCARS messages to SP”, a Shipper enters a list of selection parameters via a user-friendly interface of the ROCARS system to enquire the available CCRN(s) that the Shipper submitted earlier. Then the Shipper selects one or more CCRNs to extract the corresponding bundle(s) of ROCARS AEX and ACRED messages in a pre-defined XML file. After that, the Shipper passes the XML file to the corresponding exporter by some means such as disk storage, CD/DVD etc with proper security protection. Finally, the exporter uploads the XML file containing the bundle(s) of ROCARS messages to GETS via GETS SP’s trading community interface.

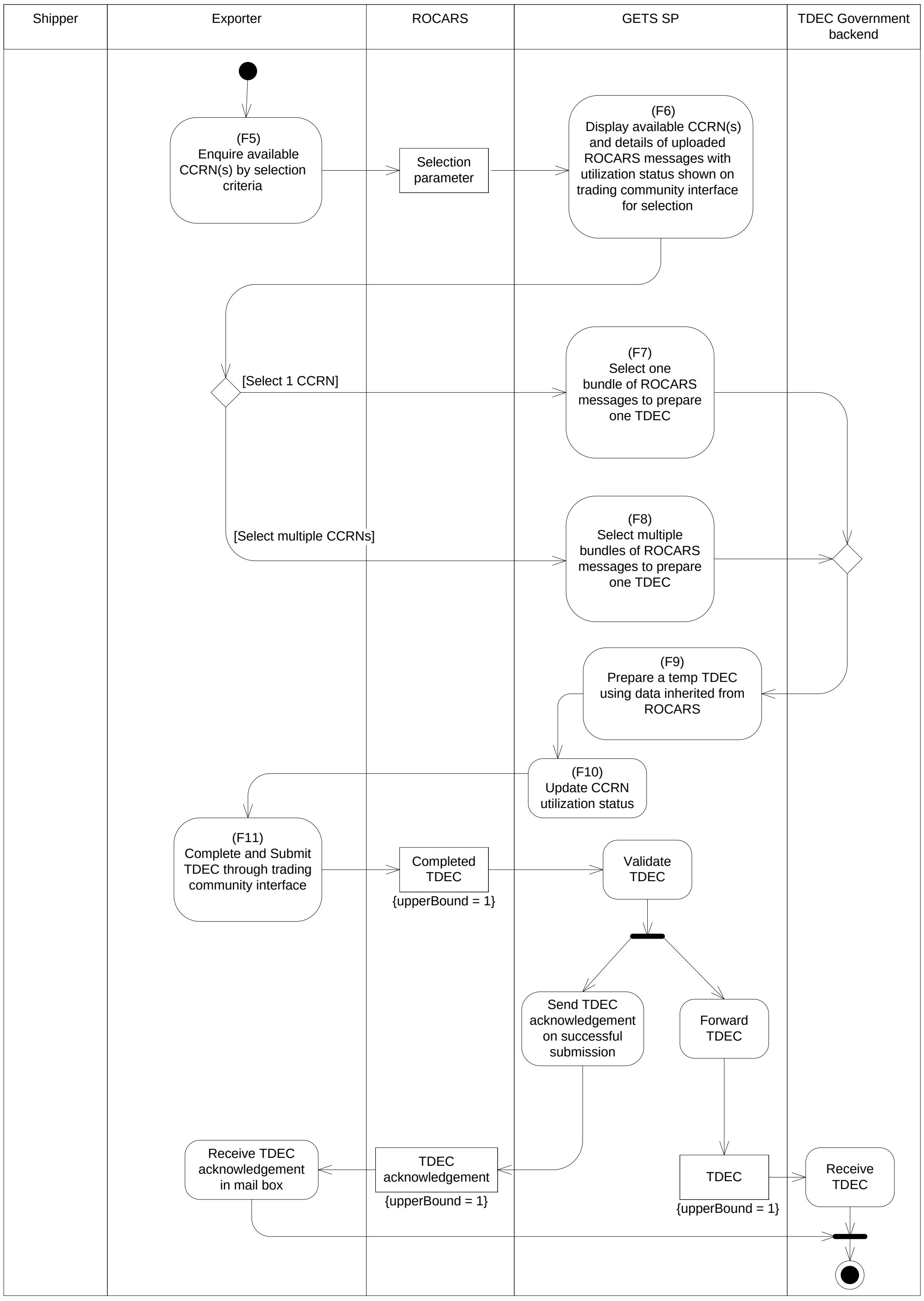
Secondly, in the diagram of “Prepare and submit TDECs to Govt”, the exporter enters a list of selection parameters via a user-friendly interface in GETS SP’s trading community interface to enquire the available CCRN(s). Then the exporter selects one or more CCRNs of the corresponding uploaded bundle(s) of ROCARS messages for preparing TDEC message(s). Then the exporter prepares a temporary TDEC with common data inherited from the selected bundle(s) of ROCARS messages through GETS SP’s inheritance functions. The common data being inherited, including the Consignee Name and Consignee Address which are required in export TDEC, are listed in Section 5.5. Finally, the exporter fills in other necessary information to complete a TDEC message and submit it to the Government via GETS SP.

DI (ROCARS to Export TDEC) Activity Diagram Upload ROCARS messages to SP



Shipper
Exporter
ROCARS
GETS SP
TDEC Government backend
(F1) Enquire available CCRN(s) by selection criteria
Selection parameter
(F2) Display available CCRN(s) with details for selection
CCRN(s)
Select CCRN(s)
Selected CCRN(s)
(F3) Extract bundle(s) of ROCARS messages
Receive bundle(s) of ROCARS messages
Bundle(s) of AEX + ACRED
Select and pass bundle(s) of ROCARS messages
Bundle(s) of AEX + ACRED
Receive bundle(s) of ROCARS messages
(F4) Upload bundle(s) of ROCARS messages through trading community interface provided by GETS SP
Bundle(s) of AEX + ACRED
Receive bundle(s) of ROCARS messages on trading community interface

DI (ROCARS to Export TDEC) Activity Diagram *Prepare and submit TDECs to Govt*


Shipper
Exporter
ROCARS
GETS SP
TDEC Government backend
(F5) Enquire available CCRN(s) by selection criteria
Selection parameter
(F6) Display available CCRN(s) and details of uploaded ROCARS messages with utilization status shown on trading community interface for selection
[Select 1 CCRN]
(F7) Select one bundle of ROCARS messages to prepare one TDEC
[Select multiple CCRNs]
(F8) Select multiple bundles of ROCARS messages to prepare one TDEC
(F9) Prepare a temp TDEC using data inherited from ROCARS
(F10) Update CCRN utilization status
(F11) Complete and Submit TDEC through trading community interface
Completed TDEC
{upperBound = 1}
Validate TDEC
Send TDEC acknowledgement on successful submission
Forward TDEC
Receive TDEC acknowledgement in mail box
TDEC acknowledgement
{upperBound = 1}
TDEC
{upperBound = 1}
Receive TDEC

4. FUNCTIONS FOR DATA INHERITANCE

4.1 Function Description

After the Shipper has extracted bundle(s) of ROCARS messages from ROCARS, GETS SPs shall provide upload function in GETS SP’s trading community interface for the Importer/Exporter to upload the extracted bundle(s) of ROCARS messages of corresponding CCRN(s) for preparation of TDEC submission.

GETS SPs shall provide function to allow Importer/Exporter to prepare a TDEC by specifying one bundle of ROCARS messages (AIM and ACRID for importation; or AEX and ACRED for exportation) of corresponding CCRN. On the other hand, as multiple bundles of ROCARS messages may be required for preparing one TDEC, GETS SPs shall also provide function to allow Importer/Exporter to prepare a TDEC by specifying multiple bundles of ROCARS messages. The function shall incorporate an alert mechanism to remind the Importer/Exporter if the selection of ROCARS messages violates the TDEC validation requirements.

Due to different requirements in reporting goods description in ROCARS and TDEC services, GETS SPs shall also provide function to allow Importer/Exporter to split one line item of a selected bundle of ROCARS messages into multiple line items of a TDEC message. If the total number of line items in the bundle(s) of ROCARS messages exceeds the limit of a TDEC message after splitting, the function shall create new TDEC(s) by copying all data fields from the first TDEC and the remaining line items that have not been inherited for TDEC preparation. The function shall also allow the Importer/Exporter to provide detailed goods description and manually supplement the corresponding HS code in the line items. GETS SPs shall prepare a reminder message on screen to remind Importer/Exporter to submit detailed goods description as required by TDEC service. Subject to the data input behavior of Shipper, the Government reserves the right requesting GETS SPs to do checking on goods description in order to ensure the data quality inherited from ROCARS. Under all circumstances, the function shall incorporate an alert mechanism to remind the Importer/Exporter to correct those inherited data fields that do not comply with the requirements specified in the corresponding Implementation Instructions of GETS System for TDEC service.

GETS SPs shall provide function to show the Importer/Exporter which CCRNs of the uploaded ROCARS messages have been utilized for preparation of TDEC messages and which have not. This information will facilitate the Importer/Exporter to select the appropriate ROCARS messages for preparing TDEC submissions.

4.2 Function List

The following is a function list to describe the above requirements.

#	Function Name	Function Description
Functions F1 – F3 are provided at ROCARS system		
F1	Enquire available CCRN(s) by selection criteria	This function enables the Shipper to specify multiple levels of selection criteria to enquire the available CCRN(s) submitted by his/her own at the ROCARS system. The selection criteria shall include, but not be limited to the following: 1. ROCARS type (importation or exportation) 2. Customs Cargo Reference Number (CCRN) 3. Departure/Arrival Date range (the expected date) 4. Consignee/Consignor Name 5. Importer/Exporter ID/Name 6. Agent ID (if available) 7. Vehicle Registration Number (VRN) 8. Container Number (if available) 9. Message Sender's Reference[2] (if available) 10. Null (to show all CCRNs that has been bundled and effected at the Land Boundary Control Point capped to a maximum of 1 month on-line retrieval period)
F2	Display available CCRN(s) with details for selection	This function displays the available CCRN(s) with details for selection by Shipper. Only those records having (i) been assigned with CCRN(s) and (ii) bundling information and (iii) the vehicle(s) been crossed the LBCP inclusive will be displayed.
F3	Extract bundle(s) of ROCARS messages	The system shall allow the Shipper to download the selected records in F2 for all the ROCARS mandatory and optional information submitted by his/her own, the CCRN, and the VRN performed in the bundling act. This function extracts bundle(s) of ROCARS messages (AIM + ACRID for importation; or AEX + ACRED for exportation) of corresponding CCRN(s) as a pre-defined XML file with handling of following:- 1. Remove other CCRN(s) and details which are not authorized for the corresponding Importer/Exporter to know. 2. A single XML file can contain one or multiple bundle(s) of "AIM + ACRID" and/or "AEX + ACRED" of the same Importer/Exporter only. Details can be referred to Section 5.1. 3. The XML file will be assigned with a default filename convention. The Shipper can store the file under a location for data inheritance designated by the function mentioned in F4. The extracted data shall include, but not be limited to the following: 1. ROCARS type (importation or exportation) 2. Consignee Name and Address (for exportation only) 3. Consignor Name 4. Container Number (if available) 5. Goods Description 6. Type of Packages (available for cargo in package) 7. Number of Packages (available for cargo in package) 8. Net Weight or Gross Volume (available for cargo in bulk, Net Weight is optional

[2] Message Sender's Reference refers to the information for XML element tag <AdditionalInformation><Content> of ROCARS messages AIM and AEX. Corresponding I.M. Indexes are AIM1410 and AEX1810 in Implementation Instructions of the Road Cargo System (ROCARS) System-to-System Interface for Bulk Submission.

#	Function Name	Function Description
		when Shipper reporting weight while Gross Volume is available when Shipper reporting volume) 9. Net Weight Unit or Gross Volume Unit (available for cargo in bulk, Net Weight Unit is optional when Shipper reporting weight while Gross Volume Unit is available when Shipper reporting volume) 10. Vehicle Registration Number (VRN) 11. Customs Cargo Reference Number (CCRN) 12. Unique Consignment Reference (UCR, if available) 13. Importer/Exporter ID and Name 14. Agent ID (if available) 15. Expected Date of Departure/Arrival 16. Message Sender's Reference (if available) The extracted bundle(s) of ROCARS messages will only include those having (i) been assigned with CCRN(s) and (ii) bundling information and (iii) the vehicle(s) been crossed the LBCP inclusive.
Functions F4 – F11 are provided by the GETS SPs		
F4	Upload bundle(s) of ROCARS messages through trading community interface provided by GETS SP	This function enables Importer/Exporter to upload bundle(s) of ROCARS messages in a pre-defined XML message format to GETS via GETS SP's trading community interface. The uploaded ROCARS messages shall be allowed for retrieval in GETS SP's trading community interface by Importer/Exporter when preparing TDEC submission. The function shall retrieve the XML files with default filename convention and location but allow the Importer/Exporter to change if he/she wishes. The changed filename convention and location will be the default for subsequent retrieval of ROCARS XML files.
F5	Enquire available CCRN(s) by selection criteria	This function enables Importer/Exporter to specify multiple levels of selection criteria to enquire the available CCRN(s) in GETS SP's trading community interface. The selection criteria shall include, but not be limited to the following: 1. ROCARS type (importation or exportation) 2. Customs Cargo Reference Number (CCRN) 3. Departure/Arrival Date range (the expected date) 4. Consignee/Consignor Name 5. Importer/Exporter ID/Name 6. Agent ID (if available) 7. Vehicle Registration Number (VRN) 8. Container Number (if available) 9. Message Sender's Reference (if available) 10. Null (to show all available CCRNs)
F6	Display available CCRN(s) and details of uploaded ROCARS messages with utilization status shown on trading community interface for selection	This function displays the available CCRN(s) with details on the trading community interface for selection by Importer/Exporter. It is suggested to enable the function to display the utilization status of the CCRN(s) for Importer/Exporter to determine whether to select it or not. The details shall include, but not be limited to the fields stated in Section 5.3.
F7	Select one bundle of ROCARS messages to prepare one TDEC	This function enables Importer/Exporter to select one bundle of ROCARS messages for preparing one TDEC message.

#	Function Name	Function Description
F8	Select multiple bundles of ROCARS messages to prepare one TDEC	This function enables Importer/Exporter to select multiple bundles of ROCARS messages for preparing one TDEC message. Through displaying reminder on screen, the function shall alert Importer/Exporter to select ROCARS messages that belong to the same voyage (same VRN, same shipment date and time), the same Importer/Exporter, and the same consignee for Export TDEC when preparing one TDEC submission. The function should validate the selected ROCARS messages as follows: 1. The VRNs in the selected ROCARS messages must be the same. Otherwise, the function should display error message and tell the Importer/Exporter to de-select some ROCARS messages. 2. The expected Departure/Arrival Dates in the selected ROCARS messages must be the same. Otherwise the function should display warning message and tell the Importer/Exporter either to confirm the deviations or de-select some ROCARS messages. The expected Departure/Arrival Dates are used for selection and validation only and will not be inherited from ROCARS to TDEC. 3. The Importer/Exporter names (if available) in the selected ROCARS messages must be the same. Otherwise, the function should display warning messages and tell Importer/Exporter either to confirm the deviations or de-select some ROCARS messages. Nevertheless, if the Importer/Exporter considers that those different names refer to the same Importer/Exporter (e.g. both “ABC Co. Ltd.” and “ABC Company Limited” refer to the same company), the Importer/Exporter may ignore the warning. In any case, the Importer/Exporter name and address are to facilitate selection only and will not be inherited from ROCARS to TDEC. 4. The consignee name in all export ROCARS messages selected must be the same. Otherwise, the function should display warning messages together with a list-box of available consignee names for the Exporter to either confirm the deviations or de-select ROCARS messages. If the Exporter considers that those different names refer to the same consignee (e.g. both “ABC Co. Ltd.” and “ABC Company Limited” refer to the same company), the Exporter may simply ignore the warning, select the correct consignee name from the list-box and confirm. In addition, the function shall copy the container number of the corresponding CCRN to line items of that CCRN when the line items are selected for inheriting to TDEC.
F9	Prepare a temp TDEC using data inherited from ROCARS	This function enables Importer/Exporter to prepare a temporary TDEC with the inherited data from the selected ROCARS message(s). The data being inherited shall include, but not be limited to the fields stated in Section 5.5. In addition, set the Transport Mode for TDEC to the default value of “Road”. For ROCARS messages of importation, the function enables Importer to select one of the form types of import TDEC (1, 1A, 1B) for preparation. On the other hand, for ROCARS messages of exportation, the function enables Exporter to select one of the form types of export TDEC (2, 2A, 2B) for preparation. The function shall enable the Importer/Exporter to handle the following two situations. Depends on the system design of GETS SPs, the function can be deferred to F11 below when Importer/Exporter editing the temporary TDEC before submitting the completed TDEC to GETS. <u>Split one line item of a bundle of ROCARS messages into multiple line items of a TDEC</u>:- In TDEC, HS code is required for each commodity. However, in ROCARS, HS code is not an input field and goods description needs not align with HS codes. If a goods description in ROCARS corresponds to multiple TDEC line items with different HS codes, this function enables Importer/Exporter to split one line item of a bundle of

#	Function Name	Function Description
		ROCARS messages to multiple TDEC line items with different HS codes. 1. The function should allow any Importer/Exporter to split one ROCARS goods description into multiple TDEC line items and manually supplement each line item with the corresponding HS code. 2. The function should check if the Net Weight Unit or Gross Volume Unit captured in ROCARS matches with the unit of quantity of the corresponding HS code assigned by TDEC according to HKIECL. If they are different, the function should adopt the TDEC assigned unit of quantity and clear the corresponding Net Weight or Gross Volume value inherited from ROCARS and alert the Importer/Exporter to input the correct quantity. 3. The function should remind the Importer/Exporter to submit detailed goods description as required by TDEC service by a reminder message on screen and allow the Importer/Exporter to confirm goods descriptions at individual item level. More precisely, the function should have selected all check boxes of individual line items in ROCARS by default. The Importer/Exporter can de-select the line items which are not applicable for inheritance and manually amend goods descriptions and other inherited data elements for the selected line items. <u>Split data contained in bundle(s) of ROCARS messages into multiple TDEC messages</u>:- 1. If the total number of line items exceeds the limit of a TDEC message, the function should create new TDEC(s) by copying all data fields from the first TDEC and the remaining line items that have not been inherited for TDEC preparation. The function shall also allow the Importer/Exporter to provide detailed goods description and manually supplement the corresponding HS code in the line items. 2. If a ROCARS message corresponding to a CCRN covers line items of different categories, the function should allow the Importer/Exporter to create separate TDEC Form Types (1, 1A, 1B, 2, 2A or 2B but not a mix of import and export declaration form types) for different goods categories.
F10	Update CCRN utilization status	This function updates the CCRN utilization so as to enable the Importer/Exporter to distinguish whether the uploaded ROCARS message has been utilized for preparation of TDEC messages or not. The utilization information will facilitate the Importer/Exporter to decide which ROCARS messages should be selected for preparing TDEC submission.
F11	Complete and Submit TDEC through trading community interface	This function enables Importer/Exporter to supplement all the necessary information to a temporary TDEC and then submit the completed TDEC to GETS.

5. MESSAGE DESCRIPTION

5.1 XML file format for data inheritance

Authorized Shipper can extract his or her own bundle(s) of ROCARS messages in XML format from ROCARS which will then be uploaded to GETS via GETS SP's trading community interface. The XML file will consist of one or multiple bundle(s) of "AIM + ACRID"; and/or one or multiple bundle(s) of "AEX + ACRED".

In the XML file, <SingleBundle> and </SingleBundle> tags will embrace each bundle of ROCARS messages, while <BundleList> and </BundleList> tags will embrace all bundle(s) of ROCARS messages within the XML file. The XML schema and XML file sample are illustrated in Appendix I.

5.2 XML Message List

A list of XML messages involved in data inheritance from ROCARS to TDEC is provided as follows:

System	Message Name	XML Schema	Government Assigned Code	Remark
ROCARS	Import Consignment	AIM_1p0.xsd	AIM	A bundle of import ROCARS messages for a corresponding CCRN.
	Import Bundling	ACRID_1p0.xsd	ACRID	
	Export Consignment	AEX_1p0.xsd	AEX	A bundle of export ROCARS messages for a corresponding CCRN.
	Export Bundling	ACRED_1p0.xsd	ACRED	
TDEC	Import/Export Declaration	TdecTradeDeclaration.xsd	TDLD01	

5.3 ROCARS Message Structure for data inheritance

An extracted version of "AIM + ACRID" and "AEX + ACRED" will be used for data inheritance. The following two branching diagrams illustrate the extracted version of a bundle of AIM and ACRID for importation; and a bundle of AEX and ACRED for exportation, of which these extracted message structures are customized for data inheritance only. The extracted version message structures will include the available data elements and exclude those not authorized to be disclosed for data inheritance. In the diagrams, the data elements which are to be inherited from ROCARS message(s) for the creating of temporary TDEC are highlighted in yellow. The remaining data elements may be used in the intermediate process but are not for TDEC submission to the Government. For the complete version of the message structures and descriptions of the data elements for ROCARS, they can be referred to the Implementation Instructions of ROCARS System-to-System Interface for Bulk Submission.

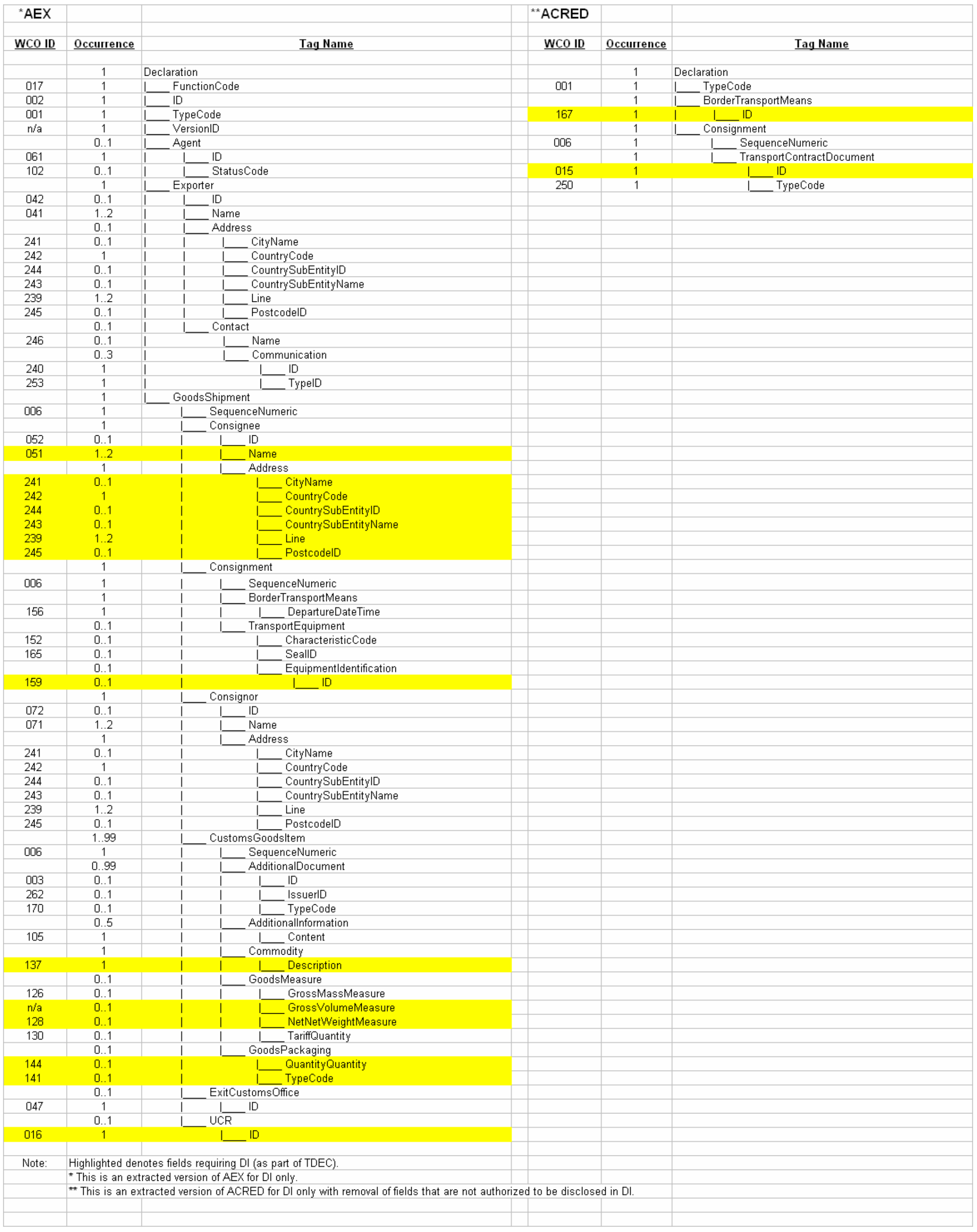
*AIM

WCO ID	Occurrence	Tag Name
	1	Declaration
017	1	|____ FunctionCode
002	1	|____ ID
001	1	|____ TypeCode
n/a	1	|____ VersionID
	0..1	|____ Agent
061	1	| |____ ID
102	0..1	| |____ StatusCode
	1	|____ GoodsShipment
006	1	| |____ SequenceNumeric
	1	| |____ Consignee
052	0..1	| | |____ ID
051	1..2	| | |____ Name
	1	| | |____ Address
241	0..1	| | |____ CityName
242	1	| | |____ CountryCode
244	0..1	| | |____ CountrySubEntityID
243	0..1	| | |____ CountrySubEntityName
239	1..2	| | |____ Line
245	0..1	| | |____ PostcodeID
	1	| |____ Consignment
006	1	| | |____ SequenceNumeric
	1	| | |____ BorderTransportMeans
172	1	| | | |____ ArrivalDateTime
	0..1	| | |____ TransportEquipment
152	0..1	| | |____ CharacteristicCode
165	0..1	| | |____ SealID
	0..1	| | |____ EquipmentIdentification
159	0..1	| | |____ ID
	1	| |____ Consignor
072	0..1	| | |____ ID
071	1..2	| | |____ Name
	1	| | |____ Address
241	0..1	| | |____ CityName
242	1	| | |____ CountryCode
244	0..1	| | |____ CountrySubEntityID
243	0..1	| | |____ CountrySubEntityName
239	1..2	| | |____ Line
245	0..1	| | |____ PostcodeID
	1..99	| |____ CustomsGoodsItem
006	1	| | |____ SequenceNumeric
	0..99	| | |____ AdditionalDocument
003	0..1	| | | |____ ID
262	0..1	| | | |____ IssuerID
170	0..1	| | | |____ TypeCode
	0..5	| | |____ AdditionalInformation
105	1	| | | |____ Content
	1	| | |____ Commodity
137	1	| | | |____ Description
	0..1	| | |____ GoodsMeasure
126	0..1	| | | |____ GrossMassMeasure
n/a	0..1	| | | |____ GrossVolumeMeasure
128	0..1	| | | |____ NetNetWeightMeasure
130	0..1	| | | |____ TariffQuantity
	0..1	| | |____ GoodsPackaging
144	0..1	| | |____ QuantityQuantity
141	0..1	| | |____ TypeCode
	0..1	| |____ EntryCustomsOffice
046	1	| | |____ ID
	0..1	| |____ UCR
016	1	| | |____ ID
	1	|____ Importer
040	0..1	|____ ID
039	1..2	|____ Name
	0..1	|____ Address
241	0..1	| |____ CityName
242	1	| |____ CountryCode
244	0..1	| |____ CountrySubEntityID
243	0..1	| |____ CountrySubEntityName
239	1..2	| |____ Line
245	0..1	| |____ PostcodeID
	0..1	|____ Contact
246	0..1	|____ Name
	0..3	|____ Communication
240	1	|____ ID
253	1	|____ TypeID

**ACRID

WCO ID	Occurrence	Tag Name
	1	Declaration
001	1	|____ TypeCode
	1	|____ BorderTransportMeans
167	1	| |____ ID
	1	|____ Consignment
006	1	|____ SequenceNumeric
	1	|____ TransportContractDocument
015	1	|____ ID
250	1	|____ TypeCode

Note: Highlighted denotes fields requiring DI (as part of TDEC).
* This is an extracted version of AIM for DI only.
** This is an extracted version of ACRID for DI only with removal of fields that are not authorized to be disclosed in DI.



*AEX

WCO ID	Occurrence	Tag Name
	1	Declaration
017	1	|____ FunctionCode
002	1	|____ ID
001	1	|____ TypeCode
n/a	1	|____ VersionID
	0..1	|____ Agent
061	1	| |____ ID
102	0..1	| |____ StatusCode
	1	|____ Exporter
042	0..1	| |____ ID
041	1..2	| |____ Name
	0..1	| |____ Address
241	0..1	| | |____ CityName
242	1	| | |____ CountryCode
244	0..1	| | |____ CountrySubEntityID
243	0..1	| | |____ CountrySubEntityName
239	1..2	| | |____ Line
245	0..1	| | |____ PostcodeID
	0..1	| |____ Contact
246	0..1	| |____ Name
	0..3	| |____ Communication
240	1	| |____ ID
253	1	| |____ TypeID
	1	|____ GoodsShipment
006	1	|____ SequenceNumeric
	1	|____ Consignee
052	0..1	| |____ ID
051	1..2	| |____ Name
	1	| |____ Address
241	0..1	| |____ CityName
242	1	| |____ CountryCode
244	0..1	| |____ CountrySubEntityID
243	0..1	| |____ CountrySubEntityName
239	1..2	| |____ Line
245	0..1	| |____ PostcodeID
	1	|____ Consignment
006	1	| |____ SequenceNumeric
	1	| |____ BorderTransportMeans
156	1	| | |____ DepartureDateTime
	0..1	| |____ TransportEquipment
152	0..1	| |____ CharacteristicCode
165	0..1	| |____ SealID
	0..1	| |____ EquipmentIdentification
159	0..1	| |____ ID
	1	|____ Consignor
072	0..1	| |____ ID
071	1..2	| |____ Name
	1	| |____ Address
241	0..1	| |____ CityName
242	1	| |____ CountryCode
244	0..1	| |____ CountrySubEntityID
243	0..1	| |____ CountrySubEntityName
239	1..2	| |____ Line
245	0..1	| |____ PostcodeID
	1..99	|____ CustomsGoodsItem
006	1	| |____ SequenceNumeric
	0..99	| |____ AdditionalDocument
003	0..1	| | |____ ID
262	0..1	| | |____ IssuerID
170	0..1	| | |____ TypeCode
	0..5	| |____ AdditionalInformation
105	1	| | |____ Content
	1	| |____ Commodity
137	1	| | |____ Description
	0..1	| |____ GoodsMeasure
126	0..1	| | |____ GrossMassMeasure
n/a	0..1	| | |____ GrossVolumeMeasure
128	0..1	| | |____ NetNetWeightMeasure
130	0..1	| | |____ TariffQuantity
	0..1	| |____ GoodsPackaging
144	0..1	| |____ QuantityQuantity
141	0..1	| |____ TypeCode
	0..1	|____ ExitCustomsOffice
047	1	| |____ ID
	0..1	|____ UCR
016	1	|____ ID

**ACRED

WCO ID	Occurrence	Tag Name
	1	Declaration
001	1	|____ TypeCode
	1	|____ BorderTransportMeans
167	1	| |____ ID
	1	|____ Consignment
006	1	|____ SequenceNumeric
	1	|____ TransportContractDocument
015	1	|____ ID
250	1	|____ TypeCode

Note: Highlighted denotes fields requiring DI (as part of TDEC).
* This is an extracted version of AEX for DI only.
** This is an extracted version of ACRED for DI only with removal of fields that are not authorized to be disclosed in DI.

5.4 TDEC Message Structure

For data inheritance from ROCARS message(s) to TDEC message(s), Import/Export Declaration message (TDLD01) will be involved. It contains all the necessary declaration information for fresh TDEC submission to be submitted by Importer/Exporter to the Government.

The following branching diagram illustrates a TDLD01 structure. The data that may possibly be inherited from ROCARS message(s) to form part of TDEC message(s) in submission are highlighted in yellow.

TDEC			
WCO ID	**I.M. Index**	**Occurrence**	**Tag Name**
	TD0000	1	TradeDeclaration
	TD0004	1	|____ MessageType
	TD0008	1	|____ MessageName
	TD0016	1	|____ MessageFunction
	TD0017	0..1	|____ NoticeReferenceNo
	TD0019	0..1	|____ QueryNo
	TD0021	0..1	|____ TransferSeqNo
	TD0099	0..1	|____ AmendmentSeqNo
	TD0018	1	|____ DeclarantInfo
	TD0032	1..2	| |____ DeclarantName
	TD0095	1	| |____ DeclarantType
	TD0034	1	| |____ DeclarantBrIdNo
	TD0037	1	| |____ DeclarationDate
	TD0038	1	| |____ DeclarationTextCode
	TD0170	1	| |____ DeclarantAddress
241	TD0171	0..1	| |____ CityName
242	TD0172	0..1	| |____ CountryCode
244	TD0173	0..1	| |____ CountrySubEntityID
243	TD0174	0..1	| |____ CountrySubEntityName
239	TD0175	1..2	| |____ Line
245	TD0176	0..1	| |____ PostcodeID
	TD0020	0..1	|____ ContactPersonInfo
	TD0045	0..1	| |____ DeclarantDesignation
	TD0046	0..1	| |____ ContactTelNo
	TD0047	0..1	| |____ ContactFaxNo
	TD0033	0..1	|____ ConsigneeInfo
051	TD0048	1..2	| |____ ConsigneeName
	TD0177	0..1	| |____ ConsigneeAddress
241	TD0178	0..1	| |____ CityName
242	TD0179	0..1	| |____ CountryCode
244	TD0180	0..1	| |____ CountrySubEntityID
243	TD0181	0..1	| |____ CountrySubEntityName
239	TD0182	1..2	| |____ Line
245	TD0183	0..1	| |____ PostcodeID
	TD0035	1	|____ ValueAndPackageInfo
	TD0053	1	| |____ TotalCifFobValue
	TD0025	1	| |____ CifFobValueType
	TD0055	1	| |____ TotalNoOfPackages
	TD0054	1	|____ ShipmentInfo
	TD0056	1	| |____ DepartureArrivalDate
	TD0058	1	| |____ TransportMode
	TD0059	0..1	| |____ TransportId
	TD0060	0..1	| |____ TransportRefNo
	TD0093	0..1	| |____ CustomsDeclarationRef
	TD0094	0..1	| |____ ConsignmentNoteNo
	TD0057	1	|____ PortAndLocationInfo
	TD0063	0..1	| |____ PortOfLoadingDischargeName
	TD0026	1	| |____ PortType
	TD0065	1	| |____ ExportingDestinationCountry
	TD0068	0..1	| |____ DestinationPlace
	TD0069	0..2	|____ DeclarationRemark
	TD0096	0..99	|____ UCR
016	TD0097	1	| |____ ID
	TD0098	1..99	| |____ RefOfConsolidationItemNo
	TD0064	1..99	|____ AirWaybillBillOfLadingInfo
	TD0070	1	|____ ConsolidationItemNo
015	TD0071	0..1	|____ MasterAirWaybillNo
	TD0072	1	|____ ConsolidationIndicator
	TD0073	1	|____ HouseAirWaybillNo
	TD0092	0..1	|____ CustomsCargoRefNo
	TD0066	1..99	|____ LineItemDetails
	TD0074	1	|____ LineItemNo
144	TD0075	0..1	|____ NoOfPackages
	TD0076	1	|____ TypeOfPackage
	TD0077	0..1	|____ Quantity
	TD0079	1	|____ ItemCifFobValue
	TD0081	1	|____ OriginCountry
159	TD0082	1	|____ ContainerNo
	TD0083	1	|____ HsCode
137	TD0084	1	|____ DescriptionOfCommodity
	TD0085	1..10	|____ ShippingMark
	TD0086	0..2	|____ ItemRemark
	TD0091	0..1	|____ PreviousItemNo
	Note:	Highlighted denotes fields requiring DI.	

5.5 Data Element Mapping

The mapping of those common data elements which will be inherited from ROCARS message(s) to TDEC message(s) are listed below. For ease of reference of the details of the mapping arrangement in data inheritance, a sample Java program is provided in Appendix II for illustration purpose only.

#	ROCARS Message (Input)		TDEC Message (Output)		Mapping Requirement
	WCO ID	Field Description	I.M. Index	Field Name	
1.	051	Consignee Name	TD0048	Consignee Name	For exportation only
2.	 241 242 244 243 239 245	Consignee Address - City Name - Country/Territory Code - Country Sub-Entity ID - Country Sub-Entity Name - Detailed Location within City - Postcode Identification	 TD0178 TD0179 TD0180 TD0181 TD0182 TD0183	Consignee Address - City Name - Country/Territory Code - Country/Territory Sub-Entity ID - Country/Territory Sub-Entity Name - Line - Postcode ID	For exportation only
3.	159	Container Number	TD0082	Container No.	If available, and the Container Number of each CCRN should be assigned to all the line items of that respective CCRN
4.	137	Commodity Description	TD0084	Description of Commodity	
5.	141	Type of Packages identification	TD0076	Type of Package	
6.	144	Number of Packages per commodity	TD0075	No. of Packages	
7.	128 or n/a	Net Weight or Gross Volume	TD0077	Quantity	If Net Weight is available, then inherit the Net Weight; else if Gross Volume is available, then inherit the Gross Volume.
8.		Net Weight Unit or Gross Volume Unit[3]	TD0078	Unit of Quantity[4]	If Net Weight Unit is available, then inherit the Net Weight Unit; else if Gross Volume Unit is available, then inherit the Gross Volume Unit.
9.	167	Vehicle Registration Number (VRN)	TD0059	Transport Identification / Customs Check Point	
10.	015	Customs Cargo Reference Number (CCRN)	TD0092	Customs Cargo Reference Number (CCRN)	
11.	016	Unique Consignment Reference (UCR)	TD0097	Unique Consignment Reference (UCR)	If available
12.			TD0058	Transport Mode	Pre-set the value to "Road"

[3] The value of this field will be stored as an XML attribute, i.e. unitCode, of WCO ID 128 or n/a e.g. <NetNetWeightMeasure unitCode="KGM"> or <GrossVolumeMeasure unitCode="LTR">.
[4] The value of this field will be stored as an XML attribute, i.e. unitCode, of TD0077, e.g. <Quantity unitCode="LTR">

5.6 Character set support

The characters used in ROCARS and TDEC are specified in Unicode values and the characters supported shall include:

<u>When data is specified in English</u>
All alphabetic characters with corresponding Unicode values that are within the following two ranges:

Unicode Values	Alphabetic characters
0020 – 007E	Basic Latin which covers uppercase and lowercase letters
00A0 – 00FF	Latin-1 supplement (for bilingual data fields only)

<u>When data is specified in Chinese</u>
All ideographic characters that fall into the intersections of the ISO/IEC 10646-1:2000 character set and any of the Big5, GB2312 or GBK character sets; and all characters included in the HKSCS-2001 character set.

5.7 Field length of bilingual field

For both ROCARS and TDEC, the same maximum number of Chinese ideographic characters, or English alphabetic characters, or a combination of both shall be allowed in all bilingual fields.

Appendix I - XML file format for data inheritance

(a) XML schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2006 sp2 U (http://www.altova.com) by OGCIO (Office of the
Government Chief Information Officer) -->
<!-- edited by the Office of the Government Chief Information Officer of the Government
of HKSAR -->
<!-- Last update: 04/08/2008

  Amendment History:
  ==================
  Version: 1.0
  Effective date: 01/01/2010
  Initial Version
-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ns1="http://www.gets.gov.hk/tdec" targetNamespace="http://www.gets.gov.hk/tdec"
elementFormDefault="qualified" attributeFormDefault="unqualified">
	<xs:element name="BundleList">
		<xs:complexType>
			<xs:sequence>
				<xs:element name="SingleBundle" maxOccurs="unbounded">
					<xs:complexType>
						<xs:sequence>
							<xs:any namespace="##any"
processContents="lax" minOccurs="2" maxOccurs="2"/>
						</xs:sequence>
					</xs:complexType>
				</xs:element>
			</xs:sequence>
		</xs:complexType>
	</xs:element>
</xs:schema>
```

(b) XML file sample

```
<?xml version="1.0" encoding="UTF-8"?>
<bd:BundleList xmlns:bd="http://www.gets.gov.hk/tdec"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.gets.gov.hk/tdec Bundle.xsd">
	<bd:SingleBundle>
		<AIM/>
		<ACRID/>
	</bd:SingleBundle>
	<bd:SingleBundle>
		<AIM/>
		<ACRID/>
	</bd:SingleBundle>
	<bd:SingleBundle>
		<AEX/>
		<ACRED/>
	</bd:SingleBundle>
	<bd:SingleBundle>
		<AIM/>
		<ACRID/>
	</bd:SingleBundle>
	<bd:SingleBundle>
		<AEX/>
		<ACRED/>
	</bd:SingleBundle>
</bd:BundleList>
```

Appendix II - Sample Java program for Data Inheritance mapping

Please note that this sample Java program is for illustration purpose only and by no means covers all possible business scenarios.

```
import org.w3c.dom.*;

/**
 * Main class for DOM Sample
 */
public class GenTDECXML
{
/**
  * @param args the command line arguments
*/
  public static void main(String[] args)
  {
    Document otdec = null;
    Document ihkacr = null;
    Document ihkaex = null;

    String name;
    String ID;
    String arrivalDateTime;
    String TransportContractDocument_ID;
    String UCR_ID;
    String arrivalDate;

    out("Input Parameter 1=" + args[0]);
    out("Input Parameter 2=" + args[1]);
    out("Input Parameter 3=" + args[2]);
    out("Input Parameter 4=" + args[3]);

// Read XML from file to DOM

    otdec = DOMUtil.parse(args[2]);
    ihkacr = DOMUtil.parse(args[0]);
    ihkaex = DOMUtil.parse(args[1]);

// Update TDEC MessageType with default value "TDLD01"

   DOMUtil.updateTDEC(otdec, "TDLD01", "MessageType", 0);

// Update TDEC MessageFunction with default value "9"

   DOMUtil.updateTDEC(otdec, "9", "MessageFunction", 0);

// Get BorderTransportMeans.ID from HKACR and update to ShipmentInfo.TransportId of
TDEC

    ID = DOMUtil.get2LevelText(ihkacr, "BorderTransportMeans", "ID");
    if (ID != null)
      {
        DOMUtil.updateTDEC(otdec, ID, "TransportId", 0);
      }

// Get ConsigneeName and Address and update ConsigneeInfo

    getConsignee(ihkaex, otdec);


// Set 1 to AirWaybillBillOfLadingInfo.ConsolidationItemNo

    DOMUtil.updateTDEC(otdec, "1", "ConsolidationItemNo", 0);

    TransportContractDocument_ID = DOMUtil.get2LevelText(ihkacr,
"TransportContractDocument", "ID");
    if (TransportContractDocument_ID != null)
      {
        DOMUtil.updateTDEC(otdec, TransportContractDocument_ID, "CustomsCargoRefNo",
0);
      }

// Get UCR.ID from HKAEX and update to ID of TDEC

    UCR_ID = DOMUtil.get2LevelText(ihkaex, "UCR", "ID");
    if (UCR_ID != null)
      {
        DOMUtil.updateTDEC(otdec, UCR_ID, "ID", 0);
      }

// Update TDEC TransportMode with default value "3"

   DOMUtil.updateTDEC(otdec, "3", "TransportMode", 0);

// Format Line Item Details

    formatLineItemDetails(ihkaex, otdec);

// Write output TDEC

    DOMUtil.writeXmlToFile(args[3],otdec);
    out("\n\nThe TDEC instant document is created to " + args[3]);
   }

/*
*/
  public static void getConsignee(Document ihkaex, Document otdec)
  {
    String name1 = null;
    String name2 = null;
    int    noOfName = 0;
    String line1 = null;
    String line2 = null;
    int    noOfLine = 0;

// Get the 2 lines Name from Consignee of HKAEX

//----    Element root = ihkaex.getDocumentElement();
    NodeList nl1 = ihkaex.getElementsByTagName("Consignee");
    if (nl1 != null && nl1.getLength() > 0)
      {

//get the Consignee element

        Element el1 = (Element)nl1.item(0);
        NodeList nl12 = el1.getElementsByTagName("Name");
       if (nl12 != null && nl12.getLength() > 0)
          {
           for (int y=0 ; y < nl12.getLength(); y++)
              {
                Element el12 = (Element)nl12.item(y);
                Node node = el12.getFirstChild();
                if (node != null)
                  {
                    noOfName = noOfName + 1;
                    if (noOfName == 1)
                      {
                        name1 = el12.getFirstChild().getNodeValue();
                      }
                    else
                      {
                        name2 = el12.getFirstChild().getNodeValue();
                      }
                  }
              }
          }
        }

// Get the 2 lines Line from Address of HKAEX

//----    Element root2 = ihkaex.getDocumentElement();
```

```
    NodeList nl2 = ihkaex.getElementsByTagName("Consignee");
    if (nl2 != null && nl2.getLength() > 0)
      {

//get the Consignee element

        Element el2 = (Element)nl2.item(0);
        NodeList nl21 = el2.getElementsByTagName("Address");
        if (nl21 != null && nl21.getLength() > 0)
          {

//get the Address element

            Element el21 = (Element)nl21.item(0);
            NodeList nl22 = el21.getElementsByTagName("Line");
            if (nl22 != null && nl22.getLength() > 0)
              {
               for (int y=0 ; y < nl22.getLength(); y++)
                  {
                    Element el22 = (Element)nl22.item(y);
                    Node node = el22.getFirstChild();
                    if (node != null)
                      {
                        noOfLine = noOfLine + 1;
                        if (noOfLine == 1)
                          {
                            line1 = el22.getFirstChild().getNodeValue();
                          }
                        else
                          {
                            line2 = el22.getFirstChild().getNodeValue();
                          }
                      }
                  }
              }
          }
       }

// Update 2 lines of Name to ConsigneeInfo.ConsigneeName of TDEC

   cloneElement(otdec, "ConsigneeInfo", "ConsigneeName", name1, name2);
   updateConsigneeInfo(otdec, "ConsigneeInfo", "ConsigneeName", name1, name2);

// Update 2 lines of Line to ConsigneeAddress.AddressLine of TDEC

    cloneElement(otdec, "ConsigneeAddress", "Line", line1, line2);
    updateConsigneeInfo(otdec, "ConsigneeAddress", "Line", line1, line2);

// Get other elements of Consignee.Address
    getConsigneeAddress(ihkaex, otdec);

  }
/*
*/
  public static void getConsigneeAddress(Document ihkaex, Document otdec)
  {
    String text1 = null;

//----    Element root = ihkaex.getDocumentElement();
    NodeList nl = ihkaex.getElementsByTagName("Consignee");
    if (nl != null && nl.getLength() > 0)
      {

//get the Consignee element

        Element el = (Element)nl.item(0);
        NodeList nl2 = el.getElementsByTagName("Address");
        if (nl2 != null && nl2.getLength() > 0)
          {

//get the Address element
```

```
            Element el2 = (Element)nl2.item(0);

//Get value of CityName and update CityName to ConsigneeAddress.CityName of TDEC

            text1 = getConsigneeAddressText(el2, "CityName");
            if (text1 != null)
              {
                updateConsigneeAddress(otdec, "ConsigneeAddress", "CityName", text1);
              }

//Get value of CountryCode and update CountryCode to ConsigneeAddress.CountryCode of
TDEC

            text1 = getConsigneeAddressText(el2, "CountryCode");
            if (text1 != null)
              {
                updateConsigneeAddress(otdec, "ConsigneeAddress", "CountryCode",
text1);
              }

//Get value of CountrySubEntityID and update CountrySubEntityID to
ConsigneeAddress.CountrySubEntityID of TDEC

            text1 = getConsigneeAddressText(el2, "CountrySubEntityID");
            if (text1 != null)
              {
                updateConsigneeAddress(otdec, "ConsigneeAddress", "CountrySubEntityID",
text1);
              }

//Get value of CountrySubEntityName and update CountrySubEntityName to
ConsigneeAddress.CountrySubEntityName of TDEC

            text1 = getConsigneeAddressText(el2, "CountrySubEntityName");
            if (text1 != null)
              {
                updateConsigneeAddress(otdec, "ConsigneeAddress",
"CountrySubEntityName", text1);
              }

//Get value of PostcodeID and update PostcodeID to ConsigneeAddress.PostcodeID of TDEC

            text1 = getConsigneeAddressText(el2, "PostcodeID");
            if (text1 != null)
              {
                updateConsigneeAddress(otdec, "ConsigneeAddress", "PostcodeID", text1);
              }

          }
      }
  }


/*
*/
  public static String getConsigneeAddressText(Element element, String tagName)
  {
    String text1 = null;

    NodeList nl = element.getElementsByTagName(tagName);
    if (nl != null && nl.getLength() > 0)
      {
        for (int y=0 ; y < nl.getLength(); y++)
           {
             Element el = (Element)nl.item(y);
             Node node = el.getFirstChild();
             if (node != null)
               {
                 text1 = el.getFirstChild().getNodeValue();
               }
           }
      }
```

```
     return text1;
  }
/*
*/
  public static void cloneElement(Document otdec, String tagName1, String tagName2,
String line1, String line2)
  {
//----    Element root = otdec.getDocumentElement();
    NodeList nl = otdec.getElementsByTagName(tagName1);

    if (nl != null && nl.getLength() > 0)
       {
              Element el = (Element)nl.item(0);


              NodeList nl21 = el.getElementsByTagName(tagName2);
              if (line2 != null)
                  {
                    Element element1 = (Element)nl21.item(0);
                    Element copyElement = (Element)element1.cloneNode(true);

element1.getParentNode().insertBefore(copyElement,element1.getNextSibling());
                  }
       }
  }
/*
*/
  public static void updateConsigneeInfo(Document otdec, String tagName1, String
tagName2, String line1, String line2)
  {

    int    noOfLine = 0;
    noOfLine = 0;
//----    Element root = otdec.getDocumentElement();
    NodeList nl = otdec.getElementsByTagName(tagName1);

    if (nl != null && nl.getLength() > 0)
       {
         for (int i = 0 ; i < nl.getLength();i++)
            {
              Element el = (Element)nl.item(i);
              NodeList nl2 = el.getElementsByTagName(tagName2);
              if (nl2 != null && nl2.getLength() > 0)
                {
               for (int y=0 ; y < nl2.getLength(); y++)
                     {
                       Element el2 = (Element)nl2.item(y);
                       noOfLine = noOfLine + 1;
                       if (noOfLine == 1)
                         {
                           if (line1 != null)
                             {
                               Node value = otdec.createTextNode(line1);
                               el2.appendChild(value);
                             }
                          }
                        else
                          {
                            if (line2 != null)
                              {
                                Node value = otdec.createTextNode(line2);
                                el2.appendChild(value);
                              }
                           }

                        }
                }

               }
       }
  }
/*
```

```
*/
  public static void updateConsigneeAddress(Document otdec, String tagName1, String
tagName2, String text1)
  {

//----    Element root = otdec.getDocumentElement();
    NodeList nl = otdec.getElementsByTagName(tagName1);

    if (nl != null && nl.getLength() > 0)
       {
         for (int i = 0 ; i < nl.getLength();i++)
            {
              Element el = (Element)nl.item(i);
              NodeList nl2 = el.getElementsByTagName(tagName2);
              if (nl2 != null && nl2.getLength() > 0)
                {
               for (int y=0 ; y < nl2.getLength(); y++)
                     {
                       Element el2 = (Element)nl2.item(y);
                       Node value = otdec.createTextNode(text1);
                       el2.appendChild(value);
                     }
             }

            }
       }
  }
/*
*/
  private static void formatLineItemDetails(Document ihkaex, Document otdec)
  {

      NodeList nlC = ihkaex.getElementsByTagName("CustomsGoodsItem");
      NodeList nlL = otdec.getElementsByTagName("LineItemDetails");

      /*Clone TDEC node <LineItemDetails> as many times as the (occurences - 1)
      of <CustomsGoodsItem> of HKAEX.*/

      if (nlC != null && nlC.getLength() > 0)
        {
         for (int i=1 ; i < nlC.getLength(); i++)
          {
               Element element1 = (Element)nlL.item(0);
               Element copyElement = (Element) element1.cloneNode(true);
               element1.getParentNode().insertBefore(copyElement,
                                          element1.getNextSibling());

          }
        }

      for (int i=0 ; i < nlC.getLength(); i++)
      {
         formatXthLineItem(ihkaex, otdec, i);
      }

  }

// Format the x+1 th Line Item Details record
/*
*/
  private static void formatXthLineItem(Document ihkaex, Document otdec, int x)
  {
      String textValue, textValueM, textValueV;
      String attributeValue, attributeValueM, attributeValueV;
      String id = null;

      NodeList nlC = ihkaex.getElementsByTagName("CustomsGoodsItem");

      Element elCustomsGoodsItem = (Element)nlC.item(x);

        // Update LineItemDetails.LineItemNo
```

```
                    DOMUtil.updateTDEC(otdec, String.valueOf(x+1), "LineItemNo", x);


         // Get GoodsPackaging.QuantityQuantity from HKAEX.xml
         // and update to TDEC LineItemDetails.NoOfPackages

          textValue = DOMUtil.getTextValue(elCustomsGoodsItem, "QuantityQuantity");
               if (textValue != null)
               {
                     DOMUtil.updateTDEC(otdec, textValue, "NoOfPackages", x);
               }


            // Get GoodsPackaging.TypeCode from HKAEX.xml
            // and update to TDEC LineItemDetails.TypeOfPackage

               NodeList nlC1 =
elCustomsGoodsItem.getElementsByTagName("GoodsPackaging");
               if (nlC1 != null && nlC1.getLength() > 0)
               {
                     Element elGoodsPackaging = (Element)nlC1.item(0);
                     textValue = DOMUtil.getTextValue(elGoodsPackaging, "TypeCode");
                     if (textValue != null)
                     {
                           DOMUtil.updateTDEC(otdec, textValue, "TypeOfPackage", x);
                     }
               }



                textValue = null;
                attributeValue = null;

                 textValueV = null;
                 textValueM = null;

                attributeValueV = null;
                attributeValueM = null;

                 textValueV = DOMUtil.getTextValue(elCustomsGoodsItem,
"NetNetWeightMeasure");
                attributeValueV = DOMUtil.getAttributeValue(elCustomsGoodsItem,
"NetNetWeightMeasure", "unitCode");
                textValueM = DOMUtil.getTextValue(elCustomsGoodsItem,
"GrossVolumeMeasure");
                 attributeValueM = DOMUtil.getAttributeValue(elCustomsGoodsItem,
"GrossVolumeMeasure", "unitCode");

              if (textValueV != null && attributeValueV != null)
              {
                textValue = textValueV;
                attributeValue = attributeValueV;
              }
              else if (textValueM != null && attributeValueM != null)
              {
                textValue = textValueM;
                attributeValue = attributeValueM;
              }

                if (textValue != null && attributeValue != null)
                     {
                           DOMUtil.updateTDEC(otdec, textValue, "Quantity", x);
                           DOMUtil.updateTDECAttribute(otdec, attributeValue,
"Quantity", x, "unitCode");
                     }


              // Get EquipmentIdentification.ID from HKAEX.xml
              // and TDEC LineItemDetails.ContainerNo

                id = DOMUtil.get2LevelText(ihkaex, "EquipmentIdentification", "ID");
                 if (id != null)
```

```
                   {
                     DOMUtil.updateTDEC(otdec, id, "ContainerNo", x);
                  }

             // Get Commodity.Description from HKAEX.xml
             // and update to TDEC LineItemDetails.DescriptionOfCommodity

              textValue = DOMUtil.getTextValue(elCustomsGoodsItem, "Description");
                    if (textValue != null)
                    {
                       DOMUtil.updateTDEC(otdec, textValue, "DescriptionOfCommodity",
x);
                    }
  }
/**
  * Prints to the System output a message
  * @param message String
*/

  public static void out(String message)
  {
    System.out.println(message);
  }
/*
*/
}
```