aptean

LFS

LFS Vehicle Checks Export

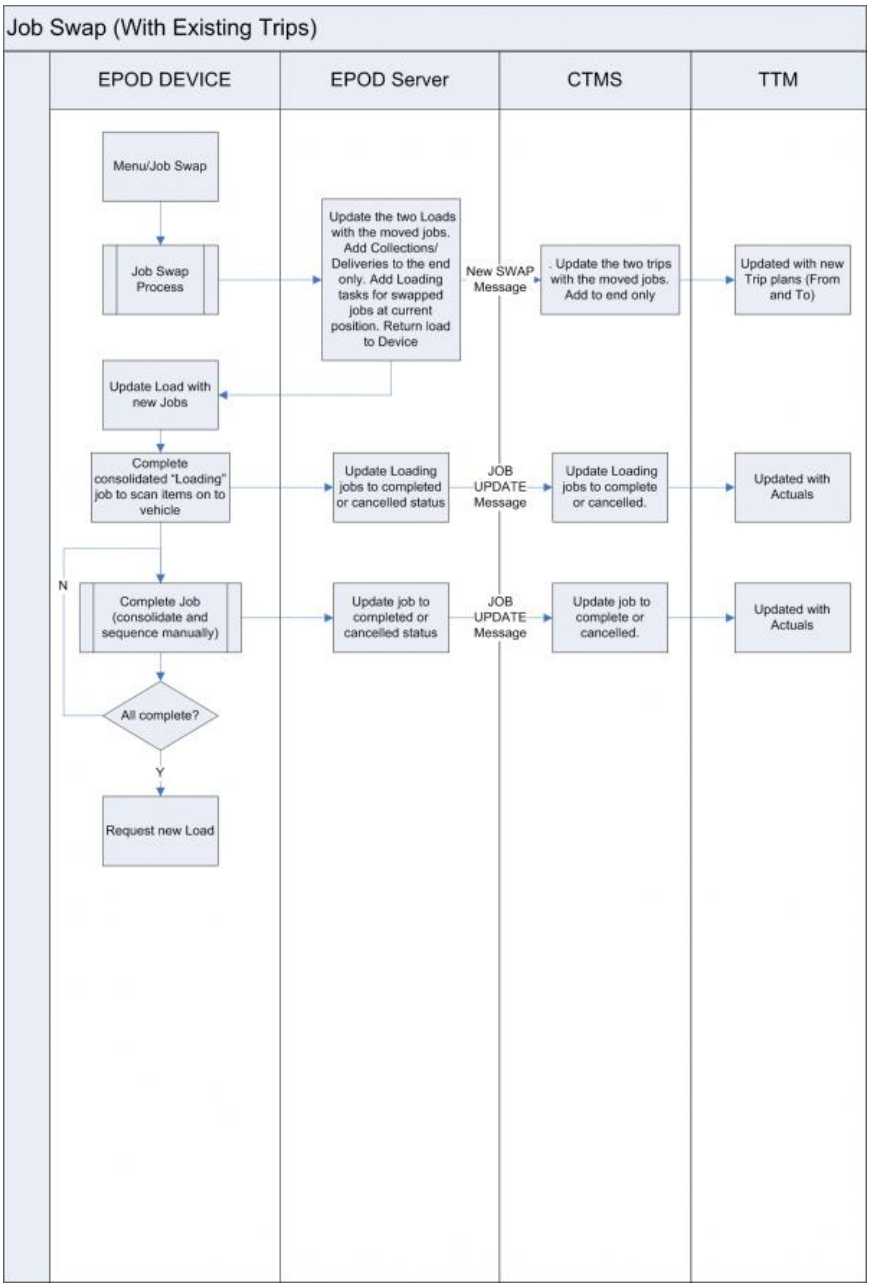
CALIDUS ePOD

24th February 2014 - 0.1
Reference: FS 315080 AUNZ-027

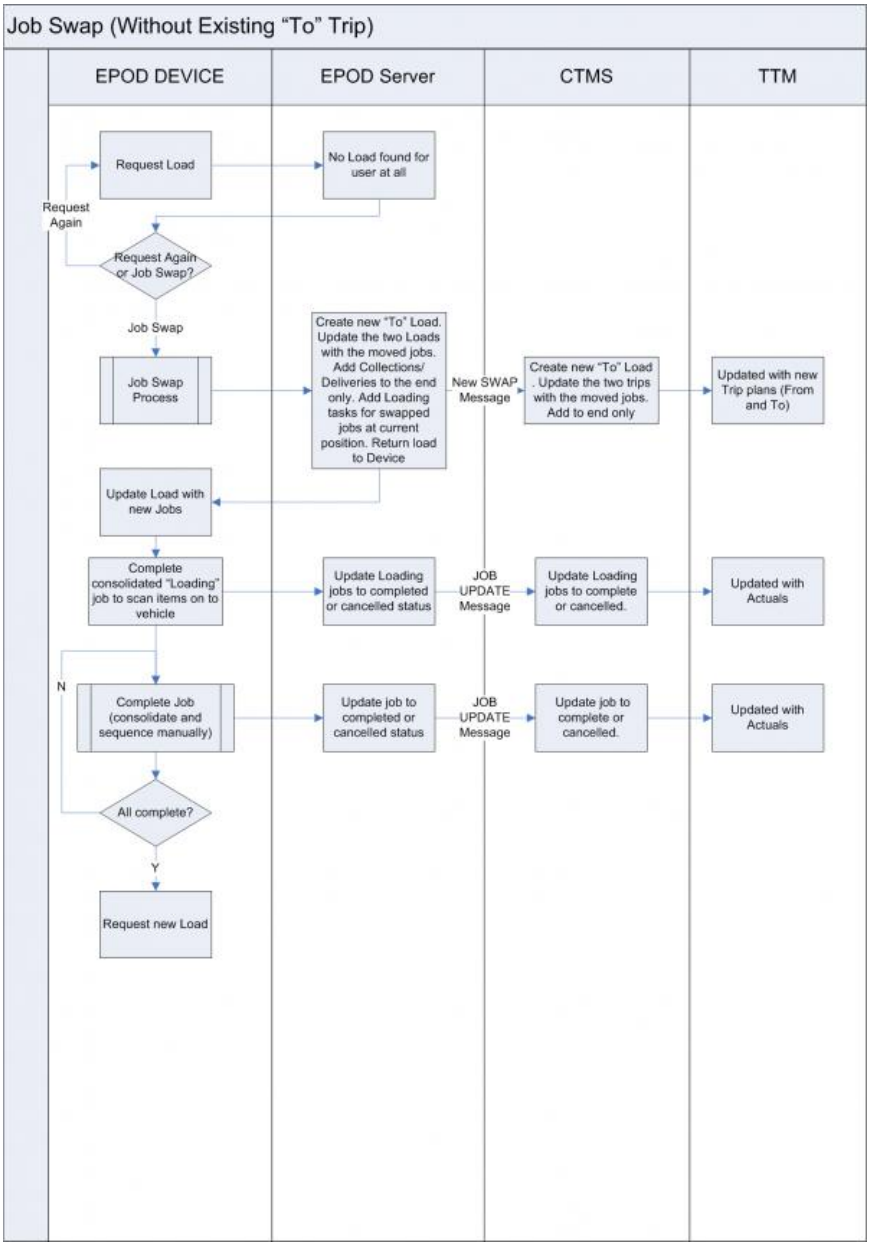
Contents

1 Appendix A: Process Flows

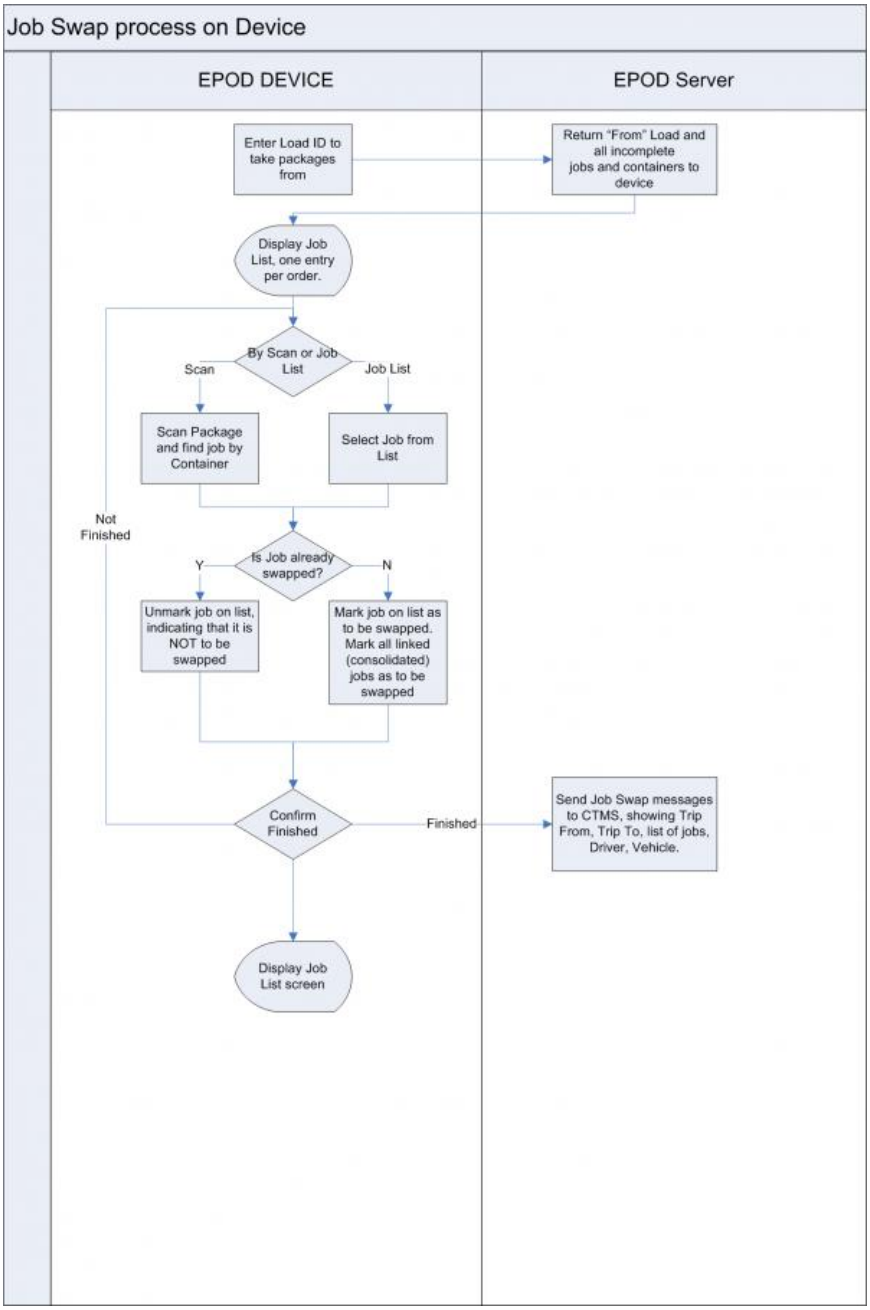
1.1 Transferring to an Existing Load



1.2 Transferring without an Existing Load



1.3 PDA Device Process



2 Appendix B: Quote & Document References

Cost Details				
Activity	**Estimate No. of Days**	**No. of Days**	**Rate per Day (?)**	**Cost (? Exc. VAT)**
Requirements	0.00	0.00	0	?0.00
Change Request Evaluation	0.00	0.00	0	?0.00
Functional Specification	0.00	6.00	0	?0.00
Technical Specification	0.00	0.00	0	?0.00
Development	0.00	12.00	0	?0.00
Testing and Release	0.00	1.50	0	?0.00
Implementation	0.00	0.25	0	?0.00
Project Management	First argument to "number_format" must be a number.	First argument to "number_format" must be a number.	0	?First argument to "number_format" must be a number.
TOTAL	First argument to "number_format" must be a number.	First argument to "number_format" must be a number.		?First argument to "number_format" must be a number.

Estimate excludes training, release to live and go live support.

B.1 References

Ref No	Document Title & ID	Version	Date
1			

B.2 Glossary

Term	Definition
EPOD	Electronic Proof of Delivery. The OBS EPOD system is *CALIDUS* ePOD.
CALIDUS eSERV	The OBS mobile system to complete Service functionality in the field. This is part of the *CALIDUS* ePOD system.
PDA	The mobile device on which the C-ePOD system will run in the field. This can be a Phone, EDA or industrial PDA, running Android.
DAL	Data Access Layer. A mechanism for accessing data by the system that is removed from the application, allowing for simplified access and providing protection to the data, as only approved DAL methods can be used to modify it.
GPS	Global Positioning System. A mechanism of retrieving accurate positioning information in the form of Latitude and Longitude (Lat-Long) co-ordinates from a device.
GPRS, 3G, HSDPA, Data Service	All terms referring to mobile device network connectivity, and the speed at which the device connects to the internet.

B.3 Authorised By

Matt Tipping Project Manager _____

_____(PRINT) Client Representative _____

Note: *CALIDUS* TMS functionality is mentioned only by reference in this solution - see elsewhere for specification of the changes to that system.

3 Set-up

3.1 Pre-requisites

- A working *CALIDUS* ePOD system.
- A working *CALIDUS* TMS system, accessible to the *CALIDUS* ePOD system.

3.1.1 *CALIDUS* TMS Interface Changes

- EPL_UDF_JOBDETS to be populated with PACKAGING, GEL and DRY ICE counts, in UDF format, only on collections from supplier. Format example:

```
<FORM NAME="Job Details" REQUIRED="Y">
  <FIELD ID="DU1_COUNT">
     <TEXT>No. of DU1: X</TEXT>
     <FORMAT>L</FORMAT>
  </FIELD>
  <FIELD ID="GEL_PACK_COUNT">
     <TEXT>No. of Gel Packs: X</TEXT>
     <FORMAT>L</FORMAT>
  </FIELD>
  <FIELD ID="DRY_ICE_QTY">
     <TEXT>Qty of Dry Ice: X.X Kgs</TEXT>
     <FORMAT>L</FORMAT>
  </FIELD>
</FORM>
```

- EPL_LOAD_INFORMATION to be populated with rolled-up values of PACKAGING, GEL and DRY ICE counts, as follows:

  No. of DU1: X
  No. of Gel Packs: X
  Qty of Dry Ice: X.X Kgs

- New Job Address information will be passed in through a new tag EPOD_JOB_ADDRESS in the EPOD_JOB tag, before the existing tag EPL_OWNER_NAME. This tag can occur zero to many times, is nillable, and will have the following content:
  - EPL_ADDRESS_TYPE - 1-character field indicating the address type (C = Collection, D = Delivery). This will be stored in EPL_JOB_TYPE of EPOD_JOB_ADDRESS, as follows:
    - For Collections from the customer only (i.e. not loading tasks), the ultimate destination address of the order, sent with type "D". The Job Address (i.e the address of the collection or loading job) will be stored with type "C".
    - For Deliveries to the final destination only (i.e. not unloading tasks), the original pickup location, set with type "C". The Job Address (i.e the address of the delivery or unloading job) will be stored with type "D".
  - EPL_ADDRESS_1 - 40-characters, occurs 1 time only, not nillable.
  - EPL_ADDRESS_2 - 40-characters, occurs 0 to 1 time, nillable.
  - EPL_ADDRESS_3 - 40-characters, occurs 0 to 1 time, nillable.
  - EPL_ADDRESS_4 - 40-characters, occurs 0 to 1 time, nillable.
  - EPL_ADDRESS_5 - 40-characters, occurs 0 to 1 time, nillable.
  - EPL_POSTCODE - 10-characters, occurs 0 to 1 time, nillable.
  - EPL_CONTACT - 40-characters, occurs 0 to 1 time, nillable.
  - EPL_TELEPHONE - 50-characters, occurs 0 to 1 time, nillable.
  - EPL_EMAIL - 255-character, occurs 0 to 1 time, nillable.

3.2 Menu Structure

None

3.3 Data

None

4 Functional Description

4.1 Database/DAL

Table EPOD_SITE requires the following modification:

- EPL_PDA_FULL_ADDRESSES - nvarchar(1), NOT NULL DEFAULT 'N', values 'Y' or 'N'.

Existing packages will be modified to allow the creating, editing and selecting of the new flag, including but not limited to:

- EPOD_SITE_INSERT
- EPOD_SITE_SELECT
- EPOD_SITE_SELECT_UPDATED_DATA
- EPOD_SITE_UPDATE

The existing EPOD_SITE DAL object will be changed to:

- Read the new flag

Note: It is not necessary to add this flag as a searchable item. However, if allowing this keeps the packages and DAL objects standard in design, then this can also be done, within the DAL and the packages.

Note: It is not necessary to export the new flag in XML requests. However, if doing this keeps the functions standard in design, this can also be done. If so, the XML Export of EPOD_SITE records will look as follows:

```
<EPOD_SITE>
  <EPL_SITE_ID></EPL_SITE_ID>
  <EPL_DESCRIPTION></EPL_DESCRIPTION>
  <EPL_SERVICE_POD_FORMAT></EPL_SERVICE_POD_FORMAT>
  <EPL_DELIVERY_POD_FORMAT></EPL_DELIVERY_POD_FORMAT>
  <EPL_COLLECTION_POD_FORMAT></EPL_COLLECTION_POD_FORMAT>
  <EPL_SERVICE_ACTIVITIES></EPL_SERVICE_ACTIVITIES>
  <EPL_SERVICE_PREWORK></EPL_SERVICE_PREWORK>
  <EPL_SERVICE_INFO></EPL_SERVICE_INFO>
  <EPL_SERVICE_PRODUCTS></EPL_SERVICE_PRODUCTS>
  <EPL_SERVICE_MC_REF></EPL_SERVICE_MC_REF>
  <EPL_SERVICE_DIAGNOSIS></EPL_SERVICE_DIAGNOSIS>
  <EPL_SERVICE_POSTWORK></EPL_SERVICE_POSTWORK>
  <EPL_AUTO_COMPLETE_EMAIL></EPL_AUTO_COMPLETE_EMAIL>
  <EPL_AD_HOC_COLLECTION></EPL_AD_HOC_COLLECTION>
  <EPL_DEL_DRIVER_SIGN></EPL_DEL_DRIVER_SIGN>
  <EPL_COL_DRIVER_SIGN></EPL_COL_DRIVER_SIGN>
  <EPL_DELIVERY_PAYMENT></EPL_DELIVERY_PAYMENT>
  <EPL_DOCUMENT_PHOTO></EPL_DOCUMENT_PHOTO>
  <EPL_CONTAINER_ONLY></EPL_CONTAINER_ONLY>
  <EPL_PDA_DIS_JOB_CODE></EPL_PDA_DIS_JOB_CODE>
  <EPL_LINKED_C_D></EPL_LINKED_C_D>
  <EPL_METRICS_ENTRY></EPL_METRICS_ENTRY>
  <EPL_NOTES></EPL_NOTES>
  <EPL_FORCED_ENTRY></EPL_FORCED_ENTRY>
  <EPL_SYSTEM_TYPE></EPL_SYSTEM_TYPE>
  <EPL_UPDATE_FUNCTIONS></EPL_UPDATE_FUNCTIONS>
  <EPL_VEHICLE_CHECK_CONFIG></EPL_VEHICLE_CHECK_CONFIG>
  <EPL_JOB_LIST_CFG></EPL_JOB_LIST_CFG>
  <EPL_ARRIVAL_FLAG></EPL_ARRIVAL_FLAG>
  <EPL_SCAN_ERROR_FLAG></EPL_SCAN_ERROR_FLAG>
  <EPL_LAST_CHANGED_DATE></EPL_LAST_CHANGED_DATE>
  <EPL_LAST_CHANGED_TIME></EPL_LAST_CHANGED_TIME>
  <EPL_RESEQUENCE></EPL_RESEQUENCE>
  <EPL_CLAUSE_DELIVERY></EPL_CLAUSE_DELIVERY>
  <EPL_JOB_STATUS></EPL_JOB_STATUS>
  <EPL_CONSOLIDATION></EPL_CONSOLIDATION>
  <EPL_VEHICLE_STOCK_FLAG></EPL_VEHICLE_STOCK_FLAG>
  <EPL_SCAN_AT_VEHICLE></EPL_SCAN_AT_VEHICLE>
  <EPL_PDA_FULL_ADDRESSES></EPL_PDA_FULL_ADDRESSES>
</EPOD_SITE>
```

The existing database package EPOD_SETUP will be modified to ensure that the new flags are defaulted appropriately

The existing EPOD_LOAD table will be modified to add new fields as follows:

- EPL_LOAD_INFORMATION - nvarchar(max), NOT NULL DEFAULT "".

Existing packages will be modified to allow the creating, editing and selecting of the new field, including but not limited to:

- EPOD_LOAD_INSERT
- EPOD_LOAD_SEARCH
- EPOD_LOAD_SELECT
- EPOD_LOAD_SELECT_DATE
- EPOD_LOAD_SELECT_PDA
- EPOD_LOAD_UPDATE

The XML Export of EPOD_LOAD (through the DAL) will be modified to add these two new fields to the end of the EPOD_LOAD section. The XML export format in this case will be as follows:

```
<EPOD_LOAD>
  <EPL_SITE_ID></EPL_SITE_ID>
  <EPL_LOAD_ID></EPL_LOAD_ID>
  <EPL_LOAD_START_PLANNED_DATE></EPL_LOAD_START_PLANNED_DATE>
  <EPL_LOAD_START_PLANNED_TIME></EPL_LOAD_START_PLANNED_TIME>
  <EPL_LOAD_END_PLANNED_DATE></EPL_LOAD_END_PLANNED_DATE>
  <EPL_LOAD_END_PLANNED_TIME></EPL_LOAD_END_PLANNED_TIME>
  <EPL_LOAD_START_ACTUAL_DATE></EPL_LOAD_START_ACTUAL_DATE>
  <EPL_LOAD_START_ACTUAL_TIME></EPL_LOAD_START_ACTUAL_TIME>
  <EPL_LOAD_END_ACTUAL_DATE></EPL_LOAD_END_ACTUAL_DATE>
  <EPL_LOAD_END_ACTUAL_TIME></EPL_LOAD_END_ACTUAL_TIME>
  <EPL_LOAD_DISTANCE_PLANNED></EPL_LOAD_DISTANCE_PLANNED>
  <EPL_LOAD_DISTANCE_ACTUAL></EPL_LOAD_DISTANCE_ACTUAL>
  <EPL_VEHICLE_ID></EPL_VEHICLE_ID>
  <EPL_USER_ID></EPL_USER_ID>
  <EPL_STATUS></EPL_STATUS>
  <EPL_MILEAGE_START></EPL_MILEAGE_START>
  <EPL_MILEAGE_END></EPL_MILEAGE_END>
  <EPL_LAST_CHANGED_DATE></EPL_LAST_CHANGED_DATE>
  <EPL_LAST_CHANGED_TIME></EPL_LAST_CHANGED_TIME>
  <EPL_UDF_LOAD_START></EPL_UDF_LOAD_START>
  <EPL_UDF_LOAD_END></EPL_UDF_LOAD_END>
  <EPL_LOAD_INFORMATION></EPL_LOAD_INFORMATION>
  <EPOD_JOBS>...</EPOD_JOBS>
<EPOD_LOAD>
```

The EPOD_JOB DAL object will be modified to add any additional addresses to any exports to the PDA - any other existing EPOD_JOB XML formats will be unaffected (e.g. Extracts of data through ePOD_Data_Service.asmx).

The additional address details will be exported if:

- The new EPOD_SITE flag EPL_PDA_FULL_ADDRESSES is set to "Y"
- A record exists on EPOD_JOB_ADDRESSES where its EPL_JOB_TYPE value is not the EPL_JOB_TYPE of the job.

A single EPOD_JOB_ADDRESS tag and contents will be produced for each address found that matches.

The XML export format in this case will be as follows:

```
<EPOD_JOBS>
  <EPOD_JOB>
    <EPL_SITE_ID></EPL_SITE_ID>
    <EPL_JOB_ID></EPL_JOB_ID>
    <EPL_LOAD_ID></EPL_LOAD_ID>
    <EPL_JOB_TYPE></EPL_JOB_TYPE>
    <EPL_JOB_GROUP></EPL_JOB_GROUP>
    <EPL_JOB_INSTRUCTION></EPL_JOB_INSTRUCTION>
    <EPL_JOB_SIGNATURE></EPL_JOB_SIGNATURE>
    <EPL_REASON_CODE></EPL_REASON_CODE>
    <EPL_STATUS></EPL_STATUS>
    <EPL_CUSTOMER_CODE></EPL_CUSTOMER_CODE>
    <EPL_PHOTO_ID></EPL_PHOTO_ID>
    <EPL_ENG_SIGNATURE></EPL_ENG_SIGNATURE>
    <EPL_SEQUENCE></EPL_SEQUENCE>
    <EPL_START_PLANNED_DATE></EPL_START_PLANNED_DATE>
    <EPL_START_PLANNED_TIME></EPL_START_PLANNED_TIME>
    <EPL_END_PLANNED_DATE></EPL_END_PLANNED_DATE>
    <EPL_END_PLANNED_TIME></EPL_END_PLANNED_TIME>
```


```
      <EPL_START_ACTUAL_DATE></EPL_START_ACTUAL_DATE>
      <EPL_START_ACTUAL_TIME></EPL_START_ACTUAL_TIME>
      <EPL_END_ACTUAL_DATE></EPL_END_ACTUAL_DATE>
      <EPL_END_ACTUAL_TIME></EPL_END_ACTUAL_TIME>
      <EPL_DISTANCE_PLANNED></EPL_DISTANCE_PLANNED>
      <EPL_DISTANCE_ACTUAL></EPL_DISTANCE_ACTUAL>
      <EPL_DRIVING_TIME></EPL_DRIVING_TIME>
      <EPL_CUSTOMER_NAME></EPL_CUSTOMER_NAME>
      <EPL_JOB_ADDRESS></EPL_JOB_ADDRESS>
      <EPL_ADDRESS_1></EPL_ADDRESS_1>
      <EPL_ADDRESS_2></EPL_ADDRESS_2>
      <EPL_ADDRESS_3></EPL_ADDRESS_3>
      <EPL_ADDRESS_4></EPL_ADDRESS_4>
      <EPL_ADDRESS_5></EPL_ADDRESS_5>
      <EPL_POSTCODE></EPL_POSTCODE>
      <EPL_CONTACT></EPL_CONTACT>
      <EPL_TELEPHONE></EPL_TELEPHONE>
      <EPL_EMAIL></EPL_EMAIL>
      <EPL_INVOICED></EPL_INVOICED>
      <EPL_CUST_SIGNATORY></EPL_CUST_SIGNATORY>
      <EPL_JOB_CODE></EPL_JOB_CODE>
      <EPL_CUST_REF></EPL_CUST_REF>
      <EPL_OFFICE_INSTRUCTION></EPL_OFFICE_INSTRUCTION>
      <EPL_SIGNED_UNCHECKED></EPL_SIGNED_UNCHECKED>
      <EPL_SO_NUMBER></EPL_SO_NUMBER>
      <EPL_ORDER_DATE></EPL_ORDER_DATE>
      <EPL_SALES_CONTACT></EPL_SALES_CONTACT>
      <EPL_USER_NOTES></EPL_USER_NOTES>
      <EPL_OWNER_NAME></EPL_OWNER_NAME>
      <EPL_SERVICE_LEVEL></EPL_SERVICE_LEVEL>
      <EPL_LAST_CHANGED_DATE></EPL_LAST_CHANGED_DATE>
      <EPL_LAST_CHANGED_TIME></EPL_LAST_CHANGED_TIME>
      <EPL_ARRIVAL_DATE></EPL_ARRIVAL_DATE>
      <EPL_ARRIVAL_TIME></EPL_ARRIVAL_TIME>
      <EPL_XF_VALUES></EPL_XF_VALUES>
      <EPL_EXT_REF></EPL_EXT_REF>
      <EPL_TRAILER_ID></EPL_TRAILER_ID>
      <EPL_PF_DEPOT></EPL_PF_DEPOT>
      <EPL_PF_TRACKING_NO></EPL_PF_TRACKING_NO>
      <EPL_JOB_STATUS></EPL_JOB_STATUS>
      <EPL_UDF_JOBDETS></EPL_UDF_JOBDETS>
      <EPOD_JOB_ADDRESS>
        <EPL_ADDRESS_TYPE></EPL_ADDRESS_TYPE>
        <EPL_ADDRESS_1></EPL_ADDRESS_1>
        <EPL_ADDRESS_2></EPL_ADDRESS_2>
        <EPL_ADDRESS_3></EPL_ADDRESS_3>
        <EPL_ADDRESS_4></EPL_ADDRESS_4>
        <EPL_ADDRESS_5></EPL_ADDRESS_5>
        <EPL_POSTCODE></EPL_POSTCODE>
        <EPL_CONTACT></EPL_CONTACT>
        <EPL_TELEPHONE></EPL_TELEPHONE>
        <EPL_EMAIL></EPL_EMAIL>
      <EPOD_JOB_ADDRESS>
      <EPOD_CONTAINERS />
      <EPOD_SERVICES />
    </EPOD_JOB>
  </EPOD_JOBS>
```

4.2 Import/Export Messages

Both the dataservice and PDA web request servers (ePOD_DataService.asmx, ePOD_DataService2.asmx and Calidus_ePOD.asmx) and the AutoExport applications will need to import and export the new field EPOD_LOAD.EPL_LOAD_INFORMATION in their XML requests. This should be handled by the DAL changes above. The responses affected are:

- AutoExport:
  - EPOD_EXPORT_LOAD
- ePOD_DataService:
  - EPOD_EXPORT_LOAD_RESPONSE
- Calidus_ePOD:
  - LOAD_RESPONSE
  - AUTO_UPDATE_RESPONSE

Both the dataservice and PDA web request servers (ePOD_DataService.asmx, ePOD_DataService2.asmx and Calidus_ePOD.asmx) and the AutoImport application will need to import the new field EPOD_JOB_ADDRESS tag and contents in their XML requests. This should be handled by the DAL changes above. The responses affected are:

- AutoImport and ePOD_DataService:
  - EPOD_XML_IMPORT
- Calidus_ePOD:
  - JOB_LOCK_RESPONSE
  - LOAD_RESPONSE
  - AUTO_UPDATE_RESPONSE

The processing of import jobs (in AutoImport and ePOD_Dataservice) must be modified to update these new tags. The EPOD_JOB tag contents will now look like this:

```
<EPOD_JOB>
  <EPL_SITE_ID></EPL_SITE_ID>
  <EPL_LOAD_ID></EPL_LOAD_ID>
  <EPL_JOB_ID></EPL_JOB_ID>
  <EPL_JOB_CODE></EPL_JOB_CODE>
  <EPL_JOB_TYPE></EPL_JOB_TYPE>
  <EPL_JOB_GROUP></EPL_JOB_GROUP>
  <EPL_CUST_REF></EPL_CUST_REF>
  <EPL_JOB_INSTRUCTION></EPL_JOB_INSTRUCTION>
  <EPL_OFFICE_INSTRUCTION></EPL_OFFICE_INSTRUCTION>
  <EPL_START_PLANNED_DATE></EPL_START_PLANNED_DATE>
  <EPL_START_PLANNED_TIME></EPL_START_PLANNED_TIME>
  <EPL_END_PLANNED_DATE></EPL_END_PLANNED_DATE>
  <EPL_END_PLANNED_TIME></EPL_END_PLANNED_TIME>
  <EPL_DISTANCE_PLANNED></EPL_DISTANCE_PLANNED>
  <EPOD_CUSTOMER></EPOD_CUSTOMER>
  <EPL_CUSTOMER_CODE></EPL_CUSTOMER_CODE>
  <EPL_CUSTOMER_NAME></EPL_CUSTOMER_NAME>
  <EPL_ADDRESS_1></EPL_ADDRESS_1>
  <EPL_ADDRESS_2></EPL_ADDRESS_2>
  <EPL_ADDRESS_3></EPL_ADDRESS_3>
  <EPL_ADDRESS_4></EPL_ADDRESS_4>
  <EPL_ADDRESS_5></EPL_ADDRESS_5>
  <EPL_POSTCODE></EPL_POSTCODE>
  <EPL_CONTACT></EPL_CONTACT>
  <EPL_TELEPHONE></EPL_TELEPHONE>
  <EPL_EMAIL></EPL_EMAIL>
  <EPOD_JOB_ADDRESS>
    <EPL_ADDRESS_TYPE></EPL_ADDRESS_TYPE>
    <EPL_ADDRESS_1></EPL_ADDRESS_1>
    <EPL_ADDRESS_2></EPL_ADDRESS_2>
    <EPL_ADDRESS_3></EPL_ADDRESS_3>
    <EPL_ADDRESS_4></EPL_ADDRESS_4>
    <EPL_ADDRESS_5></EPL_ADDRESS_5>
    <EPL_POSTCODE></EPL_POSTCODE>
    <EPL_CONTACT></EPL_CONTACT>
    <EPL_TELEPHONE></EPL_TELEPHONE>
    <EPL_EMAIL></EPL_EMAIL>
  <EPOD_JOB_ADDRESS>
  <EPL_OWNER_NAME></EPL_OWNER_NAME>
  <EPL_SO_NUMBER></EPL_SO_NUMBER>
  <EPL_ORDER_DATE></EPL_ORDER_DATE>
  <EPL_ORDER_TIME></EPL_ORDER_TIME>
  <EPL_SALES_CONTACT></EPL_SALES_CONTACT>
  <EPL_SERVICE_LEVEL></EPL_SERVICE_LEVEL>
  <EPL_TRAILER_ID></EPL_TRAILER_ID>
  <EPL_JOB_STATUS></EPL_JOB_STATUS>
  <EPL_USER_ID></EPL_USER_ID>
  <EPL_EXT_REF></EPL_EXT_REF>
  <EPL_COL_DATE></EPL_COL_DATE>
  <EPL_SEQUENCE></EPL_SEQUENCE>
  <EPL_LINKED_ID></EPL_LINKED_ID>
  <EPOD_CONTAINERS></EPOD_CONTAINERS>
  <EPOD_PRODUCTS></EPOD_PRODUCTS>
  <EPOD_SERVICES></EPOD_SERVICES>
</EPOD_JOB>
```

This processing should be added after the creation of any existing addresses and customers.

If the new tag is present, the processes should check to see if an EPOD_JOB_ADDRESS record of that type already exists for this job and address type (mapped to EPL_JOB_TYPE). If it does not, then the processes should create an EPOD_JOB_ADDRESS record at this time.

The responses to PDA requests for job information (listed above) will produce XML in the format shown in the previous section.

Note: All XSDs should be modified to include the new fields, along with standard documentation. This includes but is not limited to:

- XMLUpload.xsd

The new EPOD_LOAD field EPL_LOAD_INFORMATION should be added as follows, as the last item before EPOD_JOBS are specified:

```
<xsd:element type="xsd:string" name="EPL_LOAD_INFORMATION" nillable="true" minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>
      To be populated with any information to be displayed to the user regarding the Load.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

The new Address elements should be added to the XSDs, in all places where EPOD_JOB is specified. All elements should be added as per their existing counterparts in the main message, with EPL_ADDRESS_TYPE being created with a similar definition as EPL_JOB_TYPE, as follows:

```
<xsd:element name="EPOD_JOB_ADDRESS" nillable="true" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="EPL_ADDRESS_TYPE" nillable="false" minOccurs="1">
        <xsd:annotation>
        </xsd:annotation>
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="D"/>
            <xsd:enumeration value="C"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="EPL_ADDRESS_1" nillable="false" minOccurs="1">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:maxLength value="40" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="EPL_ADDRESS_2" nillable="true" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:maxLength value="40" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="EPL_ADDRESS_3" nillable="true" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:maxLength value="40" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="EPL_ADDRESS_4" nillable="true" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:maxLength value="40" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="EPL_ADDRESS_5" nillable="true" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
```


```
            <xsd:maxLength value="40" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="EPL_POSTCODE" nillable="true" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:maxLength value="8" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="EPL_CONTACT" nillable="true" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:maxLength value="40" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="EPL_TELEPHONE" nillable="true" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:maxLength value="50" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="EPL_EMAIL" nillable="true" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:maxLength value="255" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

4.3 Admin Changes

The Site screen site_header.aspx will be modified to allow the system to be configured for:

- Send Full Addresses - add field EPL_PDA_FULL_ADDRESSES to the PDA tab as a check-box.

Note: Ensure as part of this change that the existing check-boxes line up.

Note: No changes will be made at this time to display or allow entry of the new address types against a job - this is not required and is a function of messages interface to *CALIDUS* ePOD from external sources e.g. *CALIDUS* TMS.

4.4 PDA Changes

4.4.1 PDA DAL/Database Changes

The existing EPOD_LOAD table will be modified to add the new field as follows:

- EPL_LOAD_INFORMATION, ntext, default "".

This field will not be added to the XML export back to the server.

The existing EPOD_JOB table will be modified to add the new fields as follows:

- New address fields, for those who have it.
- EPL_ALT_ADDRESS_LINE_1/2/3/4/5, nvarchar(40), default "".
- EPL_ALT_POST_CODE nvarchar(8), default "".
- EPL_ALT_PHONE_NO nvarchar(50), default "".

Note: It is not necessary at this time to add the new flag EPL_PDA_FULL_ADDRESSES to the EPOD_SITE table, or to any functions that create the record, as this flag is not required on the device.

4.4.2 Job List

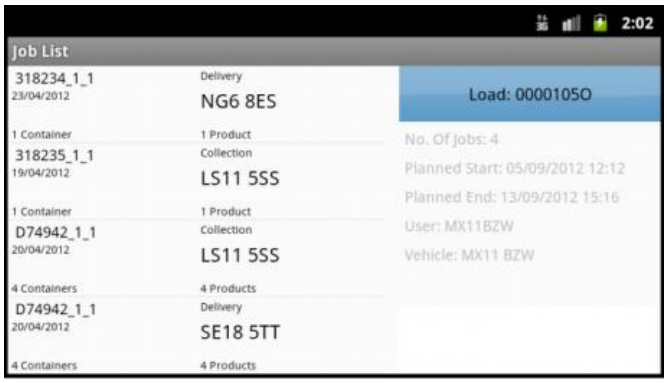
After Load Metrics have been entered (if required), the device will display the Job List screen.

The screen must display load-level information, but will operate differently when the device is in Portrait or Landscape mode.

In Landscape mode, the view to display this Load information already exists, and is populated by an existing function.

This view and function will be modified to:

- Remove the existing planned load date and time
- Add the Load information



Job List (Landscape mode)

In Portrait mode, the device will display to the driver the Packaging requirements for the whole load by adding a call-back to the existing Metrics function, which displays a pop-up Load Information window, if there is anything in the new EPL_LOAD_INFORMATION field. This window will be show the Landscape view with some extra functionality, as described below:

- A title (Load: EPL_LOAD_ID)
- A view containing information about the Load, including:
  - Total number of jobs on the Load.
  - User and Vehicle (the descriptions of the associated EPL_USER_ID and EPL_VEHICLE_ID records)
  - Any Load Information, with formatting preserved from the incoming message.
  - A **Close** button.

The window should be scrollable and fixed to the normal pop-up windows size.

If there is no data in the EPL_LOAD_INFORMATION field, The Load Information label should not be created.

The label should be size to the height of the contents within it.

The **Close** button, when pressed, will clear the pop-up and the jobs on this Load will be displayed in the Job List screen.

The title label of the Job List screen should be modified to indicate that there are instructions for this load, through an icon on the right of the title. This should display as a white letter 'i' in a circle. The picture should have a transparent background, to show the background of the title.

- 

Job List

Pressing this Information image should display the Load Information pop-up.

The menu on this screen should also be modified to show these instructions. The menu item should be labelled "Instructions" and have the information image above associated with it.

No changes are required to the Job List layouts - this has been completed under other specifications. The Job Identifier will display as that configured through EPL_JOB_DISP_ID, again specified elsewhere.

No changes are required to the way that load modifications from the server are highlighted to the user.

4.4.3 Job Details

The Job Details screen will be configured to display the customer's reference for the job in the title of the screen, through EPL_JOB_DISP_ID, specified elsewhere.

The *Instructions* tab will be modified to add a new resizeable UDF item under the existing instructions, for each job. Furthermore, the existing Instructions label will be added for each job, rather than a single consolidated label for all jobs.
Note: This directly contradicts the changes as specified in Job Consolidation (AUNZ-018, referenced in Appendix A) - this functionality takes precedence.

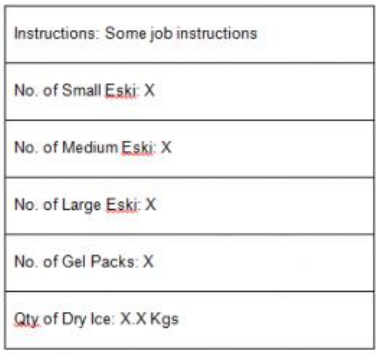
The whole window will be made into a scrolling view (if it is not already due to other changes).

If no packaging information exists for the job, a UDF object will not be added for this job. If no Instructions information exists for the job, an Instructions label will not be added for this job.

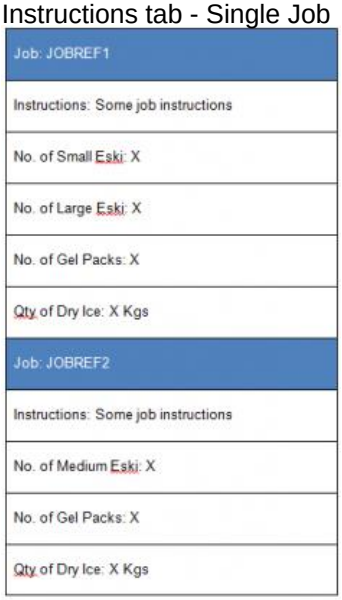
All objects added will be full width, but will resize in height to the contents.

If there are multiple jobs, the packaging information should be added for each job in sequence.

If there are multiple jobs, a title label will be added to each set of information displayed, showing "Job: EPL_DISP_JOB_ID".

- 

Instructions tab - Single Job

- 

Instructions tab - Consolidated jobs

No changes are required to the way that load modifications from the server are highlighted to the user.

4.4.4 Collection/Delivery

If the LFS style is selected on the device, the screen will move to the Containers (Packages) tab after the screen has been created.

The pop-up Container Information screen (accessible by long-pressing against a container in the table and selecting *Info* from the pop-up menu) will be modified to display the label of item EPL_CODE_1 of EPL_CONTAINER as "Dims", EPL_DESCRIPTION_LONG as "Commodity" and EPL_CONTAINER_PACKAGE_CODE as "Shipping Temp". This will be achieved through the custom LabelIDs object of the LFS style.

Each tab on the Collection/Delivery screen will be configured to display the customer's reference for the job in the title of the screen, through EPL_JOB_DISP_ID, specified elsewhere.

The *Job Details* tab of this screen displays the Contact and Instruction information - the contact area has been extended to show a Tabbed area in other specifications, which will be extended again as follows:

- Tab 1 - *Info*
  - ♦ Contact
  - ♦ Telephone
  - ♦ Planned Start Date Time

- Tab 2 - *Job Address* - the address to which you have just travelled.
- Tab 3 - *Collection/Delivery Address*

The new tab 3 will only be added if this information has been passed to *CALIDUS* ePOD.

The title of tab 3 will depend on the type of job being displayed.

- For a Delivery job, this will say "From"
- For a Collection job, this will say "To"

The information on this tab will be taken from the additional address information passed to the device, and stored in the following fields:

- EPL_ALT_ADDRESS_LINE_1/2/3/4/5
- EPL_ALT_POST_CODE
- EPL_ALT_PHONE_NO

In all other ways, the layout of this tab will be identical to the *Job Address* tab.

The Container text entry field will be modified to trigger an automatic click of the **Collect/Deliver** button, to ensure that those devices with a scanner (such as the Intermec CN51, expected to be in use for this operation) will immediately enter the item once scanned. This requires the scanned to be configured correctly (i.e. to have a Return tagged on the end of the barcode content). Note that this functionality will also work if the user presses the Enter button on the keyboard.

4.4.5 Signature Capture

If the device is configured in the style of LFS:

- The Signatory text-box will not be defaulted from the customer contact - this will be left blank, by setting the default to EPL_SIGNATORY rather than EPL_CONTACT. The user must be forced to enter this in the box before signature can be successfully confirmed.
- A label will be added behind the signature. "ORDER RECEIVED IN FULL AND GOOD CONDITION" will be displayed behind the signature. The signature box will be made partially transparent to show this text. The transparency should be applied to the view rather than the image, as a transparent image will cause issues with reports. See the opacity property in the documentation for more information.

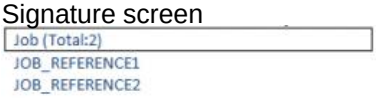
If the signature is being captured for a consolidated group of jobs, a tab will be added to the lower section of the screen, labelled as *Jobs*. This will display the total number of jobs being signed for, and a list of all the delivered job references. This will be added in a similar way to the existing tabs on this screen - particular attention must be made to any other parts of the screen that reference the tabs directly - this will now have to be through reference to an object list. For example, if the screen references tab 1 and expects it to be the Containers tab, this is now incorrect. Instead, create a tabref object, which puts the key value (1) in a property named as the name of the tab ("Containers"). Then, whenever referencing a tab on this this object, use the tabref object rather than the direct key (i.e. look up the value for "Containers" in the tabref object so: tab = tabref("Containers"). This tabref object can be added to the existing customer tab object used to create the tabs originally.

The *Pallets* (*Containers*) tab will be modified to show the total number of packages being signed for, by changing the ID label on the table from "ID" to "ID (Total: X)", where X is the total number of delivered packages only (i.e. not cancelled)

Note: As LFS have no loose products to deliver and no products within packages to scan, there will be no *Products* tab displayed on this screen.

Sample screenshots and layouts:

- 

Signature screen
- 

Jobs tab layout
- 

Containers tab layout

5 Appendix A: Quote & Document References

Cost Details				
Activity	**Estimate No. of Days**	**No. of Days**	**Rate per Day (?)**	**Cost (? Exc. VAT)**
Requirements	0.00	0.00	0	?0.00
Change Request Evaluation	0.00	0.00	0	?0.00
Functional Specification	0.00	3.50	0	?0.00
Technical Specification	0.00	0.00	0	?0.00
Development	0.00	10.50	0	?0.00
Testing and Release	0.00	2.00	0	?0.00
Implementation	0.00	0.25	0	?0.00
Project Management	First argument to "number_format" must be a number.	First argument to "number_format" must be a number.	0	?First argument to "number_format" must be a number.
TOTAL	First argument to "number_format" must be a number.	First argument to "number_format" must be a number.		?First argument to "number_format" must be a number.

Estimate excludes training, release to live and go live support.

A.1 References

Ref No	Document Title & ID	Version	Date
1			

A.2 Glossary

Term	Definition
EPOD	Electronic Proof of Delivery. The OBS EPOD system is *CALIDUS* ePOD.
CALIDUS eSERV	The OBS mobile system to complete Service functionality in the field. This is part of the *CALIDUS* ePOD system.
PDA	The mobile device on which the C-ePOD system will run in the field. This can be a Phone, EDA or industrial PDA, running Android.
DAL	Data Access Layer. A mechanism for accessing data by the system that is removed from the application, allowing for simplified access and providing protection to the data, as only approved DAL methods can be used to modify it.
GPS	Global Positioning System. A mechanism of retrieving accurate positioning information in the form of Latitude and Longitude (Lat-Long) co-ordinates from a device.
GPRS, 3G, HSDPA, Data Service	All terms referring to mobile device network connectivity, and the speed at which the device connects to the internet.

A.3 Authorised By

Matt Tipping Project Manager _____

_____(PRINT) Client Representative _____

Note: *CALIDUS* TMS functionality is mentioned only by reference in this solution - see elsewhere for specification of the changes to that system.

6 Set-up

6.1 Pre-requisites

- A working *CALIDUS* ePOD system.
- A working *CALIDUS* TMS system, accessible to the *CALIDUS* ePOD system.

6.2 Menu Structure

None

6.3 Data

All Job Groups on EPL_JOB_GROUP that require this functionality enabled must have the new flag set to "Enabled"

7 Functional Description

7.1 Database/DAL

Table EPOD_SITE requires the following modification:

- EPL_SCAN_AT_VEHICLE - nvarchar(1), NOT NULL DEFAULT 'N', values 'Y' or 'N'.

Existing packages will be modified to allow the creating, editing and selecting of the new flag, including but not limited to:

- EPOD_SITE_INSERT
- EPOD_SITE_SELECT
- EPOD_SITE_SELECT_UPDATED_DATA
- EPOD_SITE_UPDATE

The existing EPOD_SITE DAL object will be changed to:

- Export the new flag in XML requests
- Read the new flag

Note: It is not necessary to add this flag as a searchable item. However, if allowing this keeps the packages and DAL objects standard in design, then this can also be done, within the DAL and the packages.

Example: The XML Export of EPOD_SITE records will now look as follows:

```
<EPOD_SITE>
  <EPL_SITE_ID></EPL_SITE_ID>
  <EPL_DESCRIPTION></EPL_DESCRIPTION>
  <EPL_SERVICE_POD_FORMAT></EPL_SERVICE_POD_FORMAT>
  <EPL_DELIVERY_POD_FORMAT></EPL_DELIVERY_POD_FORMAT>
  <EPL_COLLECTION_POD_FORMAT></EPL_COLLECTION_POD_FORMAT>
  <EPL_SERVICE_ACTIVITIES></EPL_SERVICE_ACTIVITIES>
  <EPL_SERVICE_PREWORK></EPL_SERVICE_PREWORK>
  <EPL_SERVICE_INFO></EPL_SERVICE_INFO>
  <EPL_SERVICE_PRODUCTS></EPL_SERVICE_PRODUCTS>
  <EPL_SERVICE_MC_REF></EPL_SERVICE_MC_REF>
  <EPL_SERVICE_DIAGNOSIS></EPL_SERVICE_DIAGNOSIS>
  <EPL_SERVICE_POSTWORK></EPL_SERVICE_POSTWORK>
  <EPL_AUTO_COMPLETE_EMAIL></EPL_AUTO_COMPLETE_EMAIL>
  <EPL_AD_HOC_COLLECTION></EPL_AD_HOC_COLLECTION>
  <EPL_DEL_DRIVER_SIGN></EPL_DEL_DRIVER_SIGN>
  <EPL_COL_DRIVER_SIGN></EPL_COL_DRIVER_SIGN>
  <EPL_DELIVERY_PAYMENT></EPL_DELIVERY_PAYMENT>
  <EPL_DOCUMENT_PHOTO></EPL_DOCUMENT_PHOTO>
  <EPL_CONTAINER_ONLY></EPL_CONTAINER_ONLY>
  <EPL_PDA_DIS_JOB_CODE></EPL_PDA_DIS_JOB_CODE>
  <EPL_LINKED_C_D></EPL_LINKED_C_D>
  <EPL_METRICS_ENTRY></EPL_METRICS_ENTRY>
  <EPL_NOTES></EPL_NOTES>
  <EPL_FORCED_ENTRY></EPL_FORCED_ENTRY>
  <EPL_SYSTEM_TYPE></EPL_SYSTEM_TYPE>
  <EPL_UPDATE_FUNCTIONS></EPL_UPDATE_FUNCTIONS>
  <EPL_VEHICLE_CHECK_CONFIG></EPL_VEHICLE_CHECK_CONFIG>
  <EPL_JOB_LIST_CFG></EPL_JOB_LIST_CFG>
  <EPL_ARRIVAL_FLAG></EPL_ARRIVAL_FLAG>
  <EPL_SCAN_ERROR_FLAG></EPL_SCAN_ERROR_FLAG>
  <EPL_LAST_CHANGED_DATE></EPL_LAST_CHANGED_DATE>
  <EPL_LAST_CHANGED_TIME></EPL_LAST_CHANGED_TIME>
  <EPL_RESEQUENCE></EPL_RESEQUENCE>
  <EPL_CLAUSE_DELIVERY></EPL_CLAUSE_DELIVERY>
  <EPL_JOB_STATUS></EPL_JOB_STATUS>
  <EPL_CONSOLIDATION></EPL_CONSOLIDATION>
  <EPL_VEHICLE_STOCK_FLAG></EPL_VEHICLE_STOCK_FLAG>
  <EPL_SCAN_AT_VEHICLE></EPL_SCAN_AT_VEHICLE>
</EPOD_SITE>
```

Table EPOD_JOB_GROUP requires the following modification:

- EPL_SCAN_AT_VEHICLE - nvarchar(1), NOT NULL DEFAULT 'N', values 'Y' or 'N'.

Existing packages will be modified to allow the creating, editing and selecting of the new flag, including but not limited to:

- EPOD_JOB_GROUP_INSERT
- EPOD_JOB_GROUP_SEARCH
- EPOD_JOB_GROUP_SELECT
- EPOD_JOB_GROUP_SELECT_UPDATED_DATA
- EPOD_JOB_GROUP_UPDATE

The existing EPOD_JOB_GROUP DAL object will be changed to:

- Export the new flag in XML requests
- Read the new flag

Note: It is not necessary to add this flag as a searchable item. However, if allowing this keeps the packages and DAL objects standard in design, then this can also be done, within the DAL and the packages.

Example: The XML Export of EPOD_JOB_GROUP records will now look as follows:

```
<EPOD_JOB_GROUP>
  <EPL_SITE_ID></EPL_SITE_ID>
  <EPL_JOB_GROUP></EPL_JOB_GROUP>
  <EPL_DESCRIPTION></EPL_DESCRIPTION>
  <EPL_SERVICE_POD_FORMAT></EPL_SERVICE_POD_FORMAT>
  <EPL_DELIVERY_POD_FORMAT></EPL_DELIVERY_POD_FORMAT>
  <EPL_COLLECTION_POD_FORMAT></EPL_COLLECTION_POD_FORMAT>
  <EPL_SERVICE_ACTIVITIES></EPL_SERVICE_ACTIVITIES>
  <EPL_SERVICE_PREWORK></EPL_SERVICE_PREWORK>
  <EPL_SERVICE_INFO></EPL_SERVICE_INFO>
  <EPL_SERVICE_PRODUCTS></EPL_SERVICE_PRODUCTS>
  <EPL_SERVICE_MC_REF></EPL_SERVICE_MC_REF>
  <EPL_SERVICE_DIAGNOSIS></EPL_SERVICE_DIAGNOSIS>
  <EPL_SERVICE_POSTWORK></EPL_SERVICE_POSTWORK>
  <EPL_AUTO_COMPLETE_EMAIL></EPL_AUTO_COMPLETE_EMAIL>
  <EPL_AD_HOC_COLLECTION></EPL_AD_HOC_COLLECTION>
  <EPL_DEL_DRIVER_SIGN></EPL_DEL_DRIVER_SIGN>
  <EPL_COL_DRIVER_SIGN></EPL_COL_DRIVER_SIGN>
  <EPL_DEL_CUST_SIGN></EPL_DEL_CUST_SIGN>
  <EPL_COL_CUST_SIGN></EPL_COL_CUST_SIGN>
  <EPL_SIGN_CHECK_LBL></EPL_SIGN_CHECK_LBL>
  <EPL_DELIVERY_PAYMENT></EPL_DELIVERY_PAYMENT>
  <EPL_DOCUMENT_PHOTO></EPL_DOCUMENT_PHOTO>
  <EPL_CONTAINER_ONLY></EPL_CONTAINER_ONLY>
  <EPL_METRICS_ENTRY></EPL_METRICS_ENTRY>
  <EPL_NOTES></EPL_NOTES>
  <EPL_TNCS></EPL_TNCS>
  <EPL_LAST_CHANGED_DATE></EPL_LAST_CHANGED_DATE>
  <EPL_LAST_CHANGED_TIME></EPL_LAST_CHANGED_TIME>
  <EPL_JOB_STATUS></EPL_JOB_STATUS>
  <EPL_CONSOLIDATION></EPL_CONSOLIDATION>
  <EPL_SCAN_AT_VEHICLE></EPL_SCAN_AT_VEHICLE>
</EPOD_JOB_GROUP>
```

The existing database package EPOD_SETUP will be modified to ensure that the new flags are defaulted appropriately.

7.2 Import/Export Message

The PDA web request service (Calidus_ePOD.asmx) will export the new field EPL_SCAN_AT_VEHICLE on EPOD_SITE and EPOD_JOB_GROUP in the XML responses. This should be handled by the DAL changes above. The responses affected are:

- Calidus_ePOD:
  - LOGIN_RESPONSE

7.3 Admin Changes

The Job Group and Site screens (job_group.aspx and site_header.aspx respectively) will be modified to allow the system to be configured for:

- Scan at Vehicle - add field EPL_SCAN_AT_VEHICLE to the PDA tab as a check-box.

Note: Ensure as part of this change that the existing check-boxes line up.

7.4 PDA Changes

7.4.1 PDA DAL/Database Changes

A new column will be added to the EPOD_SITE table:

- EPL_SCAN_AT_VEHICLE - This will be a one character field with of Values 'Y' and 'N'. The default value is 'N'.

This new field should be added to every procedure as part of the PDA_SITE DAL object.

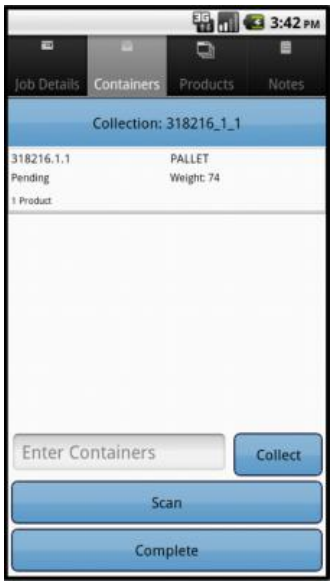
A new column will be added to the EPOD_JOB_GROUPS table:

- EPL_SCAN_AT_VEHICLE - This will be a one character field with of Values 'Y' and 'N'. The default value is 'N'.

This new field should be added to every procedure as part of the PDA_JOB_GROUP DAL object.

7.4.2 Collection / Delivery

Note: The processes for Collection and Delivery are identical within *CALIDUS* ePOD. This new functionality must affect Delivery-type jobs *ONLY*.

- 

Collection/Delivery - Containers Tab

Note: For the purposes of deciding upon which settings are used to control how the consolidated jobs are processed, the Job Group of the first job will be used to obtain the values of flags.

For Collection-type jobs, or when the new EPOD_JOB_GROUP flag EPL_SCAN_AT_VEHICLE is disabled ('N'), this

screen (created in ColDel.js) will act as it does now.

The existing function to display all the packages on the screen's Container table (BindContainerData) will be modified to:

- First fetch the required display package data itself from SelectAllContainersForJob(true) i.e. retrieving only unconfirmed data.
- If this returns no data, and this is a Delivery-type job, and the new EPOD_JOB_GROUP flag EPL_SCAN_AT_VEHICLE is enabled ('Y'), then call SelectAllContainersForJob(false) i.e. retrieving all data, including confirmed data.
- Populate the table from the returned data.
- Add status classes if the status of the container being added is unconfirmed (i.e. not 'P' or 'I').

All calls to BindContainerData will be modified to not require the passing of the data to be bound, as the function now does this itself.

Selecting containers from the list will be modified, if the status of the container is already confirmed. In this case, all actions, regardless of how they occurred (long-press, press, scan, text entry) will result in displaying the Container Action dialogue (through function ContainerActionWithDialogue). The actions allowed against the container depend on the status:

- *Deliver* - This action sets the status to Confirmed. If the status is already Cancelled ('X'), a confirmation dialogue should be displayed, asking "Changing from Cancelled to Delivered - Please confirm" with options "Yes" and "No". If the use chooses "No", no action will be taken. If the user chooses "Yes", the existing value of the reason code (EPL_REASON_CODE) will be blanked, and the status changed to Confirmed ('C'). If the Container was already at confirmed status, choosing this option will not do anything.
- *Cancel* - This action sets the status to Cancelled. If the status of the container was Confirmed (Delivered), the normal Cancellation screen (CancelContainer) will be called. If the user backs out of this cancellation in the Cancellation screen, no action will be taken. If cancellation is confirmed, the container should be set to Cancelled status in the normal manner (with reason code), and the Container list refreshed. If the Container was already at Cancelled status, choosing this option will not do anything.
- *Info* - Display some information about the package.
- *Close* - Close the dialogue.

Note: These former two actions are slightly different to the current functionality where the containers have contained products, or this is the Loose Products container. In these cases, the screen would normally display the Products tab automatically, which in this instance will now be suppressed.

Note: The actions may need to be driven by a 'screen state', maintained to show that the screen is in '1st Stage' scanning (i.e. at the vehicle) or 2nd Stage scanning (in front of the customer).

Note: Functionally, 1-stage scanning (i.e. the operation of the system before this change) should be identical to 1st-stage scanning.

Note: For clarification, all events against containers will show a pop-up action dialogue except in the following conditions:

- At 1-stage or 1st-stage:
  - Entered by any means where there are products for the container selected (and Container Only is disabled)
  - Scan or Delivery/Collection button or Auto-entry (through keyboard or wedge scanner)
- At 2nd-stage:
  - Always show dialogue (i.e. ignore products)

Currently, the screen automatically completes jobs (i.e. moves on to signature capture) that have no more pending containers to scan (and have passed validation). This is done in the ContainerAction... functions (by calling ValidateJob, then CompleteJob). If this is a Delivery-type job, and the new EPOD_JOB_GROUP flag EPL_SCAN_AT_VEHICLE is enabled ('Y'), this CompleteJob function will not be called here - it will only be called when clicking the **Complete** button.

Note: The same modification (regarding preventing automatic job completion) should be made to the ProductAction... functions.

Once the driver has cancelled all packages that the signatory does not wish to sign for, the driver can press the **Complete** button to go back to the Signature screen - any changes to delivered packages will be shown on the summary Containers (packages) tab as now.

All calls to the existing SignatureCapture function (from existing function nextJobEndTask) should now include a 'Back' callback function, which is executed by the Signature screen is the user chooses to back out. In this case, this function should be set to the existing BindContainerData function, to force a refresh of the Container list.

7.4.3 Job Completion

The existing Signature screen (created in Signature.js) requires modifications.

- 

Signature screen

Note: For the purposes of deciding upon which settings are used to control how the consolidated jobs are processed, the Job Group of the first job will be used to obtain the values of flags.

Note: The general functionality of this screen when signing for a consolidated group of jobs is as follows:

- One signature is required for all jobs
- One set of T&Cs will be displayed (from the first job) and used against all jobs.
- The *Containers* tab will show a consolidated view of all containers for all jobs.
- The *Products* tab will show a consolidated view of all Products for all jobs.

The main function SignatureCapture will be modified to accept a backout callback function as a parameter.

For Collection-type jobs, or when the new EPOD_JOB_GROUP flag EPL_SCAN_AT_VEHICLE is disabled ('N'), this screen will act as it does now.

For Delivery-type jobs, when the new EPOD_JOB_GROUP flag EPL_SCAN_AT_VEHICLE is enabled ('Y') the screen will be modified to allow the driver to back out of the screen. If this happens, the screen must execute the callback function passed to it (to refresh the Delivery processing screen's list of packages). **Note:** As this function is used throughout the system in many different places, the screen must check whether this is passed parameter is null or undefined. If so, this should not be executed.

8 Appendix A: Quote & Document References

Cost Details				
Activity	**Estimate No. of Days**	**No. of Days**	**Rate per Day (?)**	**Cost (? Exc. VAT)**
Requirements	0.00	0.00	0	?0.00
Change Request Evaluation	0.00	0.00	0	?0.00
Functional Specification	0.00	0.75	0	?0.00
Technical Specification	0.00	0.00	0	?0.00
Development	0.00	3.50	0	?0.00
Testing and Release	0.00	0.50	0	?0.00
Implementation	0.00	0.25	0	?0.00
Project Management	First argument to "number_format" must be a number.	First argument to "number_format" must be a number.	0	?First argument to "number_format" must be a number.
TOTAL	First argument to "number_format" must be a number.	First argument to "number_format" must be a number.		?First argument to "number_format" must be a number.

Estimate excludes training, release to live and go live support.

A.1 References

Ref No	Document Title & ID	Version	Date
1			

A.2 Glossary

Term	Definition
EPOD	Electronic Proof of Delivery. The OBS EPOD system is *CALIDUS* ePOD.
CALIDUS eSERV	The OBS mobile system to complete Service functionality in the field. This is part of the *CALIDUS* ePOD system.
PDA	The mobile device on which the C-ePOD system will run in the field. This can be a Phone, EDA or industrial PDA, running Android.
DAL	Data Access Layer. A mechanism for accessing data by the system that is removed from the application, allowing for simplified access and providing protection to the data, as only approved DAL methods can be used to modify it.
GPS	Global Positioning System. A mechanism of retrieving accurate positioning information in the form of Latitude and Longitude (Lat-Long) co-ordinates from a device.
GPRS, 3G, HSDPA, Data Service	All terms referring to mobile device network connectivity, and the speed at which the device connects to the internet.

A.3 Authorised By

Matt Tipping Project Manager _____

_____(PRINT) Client Representative _____

The included functionality to be specified is shown below:

From 313754/AUNZ-018

LFS can arrive at a collection point at an RDC and pick up multiple collection pallets these pallets will contain multiple jobs. When the driver arrives back at the depot the pallets will be broken down and then distributed to

individual loads. A change will be made to allow the jobs to be consolidated on the PDA based on a new customer setting and a new location setting. If both of these settings are set to allow bulk collections then the data will be transferred through from the CTMS system with a flag to say these jobs should be consolidated. The driver will then see a consolidated order on the Job List screen. They will click this job which will take them to the Job details screen where they can start the job. Each pallet will then be scanned without the need to complete multiple jobs.
Bulk pick-up work for several orders collected/loaded at the same location - these should show as one job on ePOD. The same applies for pick/pack orders outbound from LFS depot - show one location with many packages to scan on - not segregated into lots of jobs for each order at same loading location.
Similar ability to consolidate to delivery addresses where the same address (and orders not already part of a shipment).
These should be configurable at customer level to control how the data is presented to ePOD.

From 313630/AUNZ-030

A new Link ID will be added to the Customer addresses table. If multiple jobs are sent to the PDA device from the CTMS which have the same link ID these Jobs will be consolidated on the PDA as one job.
A new link ID will be added to the location table of the CTMS. This link will determine which locations are for the same delivery point. A change will be made to the location entry screen and the location upload process.
The inbound and outbound flow to and from EPOD will be modified to accept a Link ID.
There is also a need for when a new location is entered the CTMS will check all Latitude and Longitude records to see if there are any locations sharing the same latitude and longitude. If this is the case the locations will be automatically linked by the CTMS"
A requirement was identified whereby when a manual consolidation has occurred on the device a record will be stored of this in a link ID field this will be sent back to the CTMS and the addresses will now be linked so they will not need to be manually consolidated by the driver again.
This must be configurable so the option to consolidate can be deactivated.

8.1 Solution Overview

CALIDUS TMS will consolidate jobs through to *CALIDUS* ePOD under the following conditions:

- Collections from depot (loading tasks for inter-warehouse transfers and cross-dock trips)
- Deliveries to depot (unloading tasks for inter-warehouse transfers and cross-dock trips)
- Linked locations (remembered from drivers' manual consolidations in the past)

Jobs that have been pre-consolidated in this way will be displayed as a single line on the Job List screen on the Android device. The layout of these rows will be modified slightly to make it clear that this row represents a consolidated group of jobs, and the number of jobs in that group.

The user will also be able to create a consolidated group of jobs manually, by long holding on a row, starting a 'Consolidation Mode', and selecting the jobs (or existing groups of jobs) to merge together. There will be some criteria that will be checked when selecting jobs, to ensure that only jobs that can be consolidated together are selected. When complete, these selected rows will be merged into one on the device, displaying just like a pre-consolidated group of jobs. It will also be possible to break these groups back down, in case of any mistakes.

This manual consolidation of jobs can be completed before leaving the depot, on all jobs that need it, or as the deliveries are being made, on demand. In either case, the device will remember any consolidations made, as long as the load is not taken away from the user, or reloaded onto the device from scratch.

Completing consolidated groups of jobs will be very similar to completing a single job. For example, the user will still be able to:

- Select the group to see the details
- Start and Arrive to the group of jobs
- Scan, enter or select the containers (parcels) to be delivered or collected on that group of jobs
- Obtain a customer signature for the group of jobs, showing the containers and any terms and conditions.

The differences in processing are to allow the user to see details of each of the jobs in the group. For example:

- When the group is selected, the Job Detail screen will allow the user to scroll through all the jobs in the group, so that they can see the details of all the individual jobs in the group, including:
    - ♦ Contact details (Name, telephone)

    - ♦ Job Address details
    - ♦ Any special instructions for the group of jobs, all collected onto one scrollable page, for ease of use.
- When arrived at jobs and starting to complete the group of collections or deliveries, the user will also be able to scroll through these job details again, to help find the right contacts for the jobs.
- The list of parcels to be delivered will show all the packages for all the grouped jobs together. The list will be grouped by the sequence *CALIDUS* TMS requires them to be loaded (for consolidated collections) or unloaded (for consolidated deliveries). The list of packages will also show a little more information, allowing the user to see the Job that the item came from, to help in visually identifying items together.
- The user can break all the items from a job out of a consolidation from the list of all the packages in one long-press from the list of packages.

When all packages have been collected or delivered, the device will move on to job confirmation. This will work as it does for single jobs, in that it will request only one customer signature. This screen will show all the collected or delivered items here, as well as the chosen terms and conditions from the first job. Collection from and deliveries to a depot (i.e. loading and unloading tasks) may be configured to not require a signature at all.

When complete, all the jobs will updated to *CALIDUS* TMS with the same signature information, if entered on the jobs.

CALIDUS TMS will remember the locations linked together (consolidated) by the device users, and will ensure that those locations are automatically consolidated together again when the next loads are produced. **Note:** *CALIDUS* TMS will not attempt to reflect when a location has been split from a group by the device users (the drivers) - this can be reflected manually within *CALIDUS* TMS screens if required.

8.2 Scope

Note: This functionality will be developed in the Android application only and will not be available in the Windows Mobile application.

Note: The scanning functionality for Containers/Packages depends on the IDs for these items being unique for all the packages for jobs within the network.

Note: No functionality need be added to the *CALIDUS* ePOD Admin screens to allow manual consolidation - this is a function of messages being imported from a TMS only.

Note: *CALIDUS* TMS functionality is mentioned only by reference in this solution - see elsewhere for specification of the changes to that system.

Note: *CALIDUS* TMS will sequence, route and combine orders into stops based on configuration information imported from existing customer solutions.

Note: Several changes have been completed within *CALIDUS* ePOD regarding consolidation. This functionality builds on that. This is referenced as FS-308745 - ID5 - Job Consolidation in Appendix A.

9 Set-up

9.1 Pre-requisites

- A working *CALIDUS* ePOD system.
- A working *CALIDUS* TMS system, accessible to the *CALIDUS* ePOD system.
- Although specified elsewhere, it should be noted that the following data is required to be sent from *CALIDUS* TMS, in additions to the normal data, for the described functionality to work:
    - ♦ EPOD_JOB:
        - ◊ EPL_LINKED_ID - if this job is required to be consolidated
        - ◊ EPL_SEQUENCE - from the stop-level information.

9.2 Menu Structure

None

9.3 Data

No additional configuration due to these changes is expected.

10 Functional Description

10.1 Database/DAL

Table EPOD_JOB requires the following modification:

- EPL_LINKED_ID - nvarchar(40)

Existing packages will be modified to allow the creating, editing and selecting of the new field, including but not limited to:

- EPOD_JOB_INSERT
- EPOD_JOB_SEARCH
- EPOD_JOB_SELECT
- EPOD_JOB_SELECT_DATE_RANGE
- EPOD_JOB_UPDATE

The existing EPOD_JOB DAL object will be changed to:

- Export the new field in XML requests
- Read the new field

Note: It is not necessary to add this field as a searchable item. However, if allowing this keeps the packages and DAL objects standard in design, then this can also be done, within the DAL and the packages.

Example: The XML Export of EPOD_JOB records will now look as follows:

```
<EPOD_JOB>
  <EPL_SITE_ID></EPL_SITE_ID>
  <EPL_JOB_ID></EPL_JOB_ID>
  <EPL_LOAD_ID></EPL_LOAD_ID>
  <EPL_JOB_TYPE></EPL_JOB_TYPE>
  <EPL_JOB_GROUP></EPL_JOB_GROUP>
  <EPL_JOB_INSTRUCTION></EPL_JOB_INSTRUCTION>
  <EPL_JOB_SIGNATURE></EPL_JOB_SIGNATURE>
  <EPL_REASON_CODE></EPL_REASON_CODE>
  <EPL_LINKED_REASON></EPL_LINKED_REASON>
  <EPL_STATUS></EPL_STATUS>
  <EPL_CUSTOMER_CODE></EPL_CUSTOMER_CODE>
  <EPL_PHOTO_ID></EPL_PHOTO_ID>
  <EPL_PHOTO></EPL_PHOTO>
  <EPL_ENG_SIGNATURE></EPL_ENG_SIGNATURE>
  <EPL_SEQUENCE></EPL_SEQUENCE>
  <EPL_START_PLANNED_DATE></EPL_START_PLANNED_DATE>
  <EPL_START_PLANNED_TIME></EPL_START_PLANNED_TIME>
  <EPL_END_PLANNED_DATE></EPL_END_PLANNED_DATE>
  <EPL_END_PLANNED_TIME></EPL_END_PLANNED_TIME>
  <EPL_START_ACTUAL_DATE></EPL_START_ACTUAL_DATE>
  <EPL_START_ACTUAL_TIME></EPL_START_ACTUAL_TIME>
  <EPL_END_ACTUAL_DATE></EPL_END_ACTUAL_DATE>
  <EPL_END_ACTUAL_TIME></EPL_END_ACTUAL_TIME>
  <EPL_DISTANCE_PLANNED></EPL_DISTANCE_PLANNED>
  <EPL_DISTANCE_ACTUAL></EPL_DISTANCE_ACTUAL>
  <EPL_DRIVING_TIME></EPL_DRIVING_TIME>
  <EPL_CUSTOMER_NAME></EPL_CUSTOMER_NAME>
  <EPL_INVOICED></EPL_INVOICED>
  <EPL_CUST_SIGNATORY></EPL_CUST_SIGNATORY>
  <EPL_JOB_CODE></EPL_JOB_CODE>
  <EPL_CUST_REF></EPL_CUST_REF>
  <EPL_OFFICE_INSTRUCTION></EPL_OFFICE_INSTRUCTION>
  <EPL_SIGNED_UNCHECKED></EPL_SIGNED_UNCHECKED>
  <EPL_SO_NUMBER></EPL_SO_NUMBER>
  <EPL_TNCS></EPL_TNCS>
  <EPL_ORDER_DATE></EPL_ORDER_DATE>
  <EPL_ORDER_TIME></EPL_ORDER_TIME>
  <EPL_SALES_CONTACT></EPL_SALES_CONTACT>
  <EPL_USER_NOTES></EPL_USER_NOTES>
  <EPL_OWNER_NAME></EPL_OWNER_NAME>
  <EPL_SERVICE_LEVEL></EPL_SERVICE_LEVEL>
  <EPL_LAST_CHANGED_DATE></EPL_LAST_CHANGED_DATE>
  <EPL_LAST_CHANGED_TIME></EPL_LAST_CHANGED_TIME>
  <EPL_ARRIVAL_DATE></EPL_ARRIVAL_DATE>
  <EPL_ARRIVAL_TIME></EPL_ARRIVAL_TIME>
```


```
    <EPL_XF_VALUES></EPL_XF_VALUES>
    <EPL_EXT_REF></EPL_EXT_REF>
    <EPL_TRAILER_ID></EPL_TRAILER_ID>
    <EPL_PF_DEPOT></EPL_PF_DEPOT>
    <EPL_PF_TRACKING_NO></EPL_PF_TRACKING_NO>
    <EPL_JOB_STATUS></EPL_JOB_STATUS>
    <EPL_UDF_JOBDETS></EPL_UDF_JOBDETS>
    <EPL_LINKED_ID></EPL_LINKED_ID>
    <EPOD_CONTAINERS></EPOD_CONTAINERS>
    <EPOD_SERVICES></EPOD_SERVICES>
</EPOD_JOB>
```

10.2 Import/Export Message

Both the dataservice and PDA web request servers (ePOD_DataService.asmx, ePOD_DataService2.asmx and Calidus_ePOD.asmx) and the AutoExport and AutoImport applications will need to import and export the new field EPOD_JOB.EPL_LINKED_ID in their XML requests. This should be handled by the DAL changes above. The responses affected are:

- AutoExport:
  - EPOD_EXPORT_LOAD
  - EPOD_EXPORT_JOB
- ePOD_DataService:
  - EPOD_EXPORT_LOAD_RESPONSE
  - EPOD_EXPORT_JOB_RESPONSE
- AutoImport and ePOD_DataService:
  - EPOD_XML_IMPORT
- Calidus_ePOD:
  - JOB_LOCK_RESPONSE
  - LOAD_RESPONSE
  - AUTO_UPDATE_RESPONSE

The processing of Job Update messages from the PDA (in Calidus_ePOD.asmx, JOB_UPDATE_REQUEST) must be modified to update this new field. The EPOD_JOB tag contents will now look like this:

```
<EPL_SITE_ID></EPL_SITE_ID>
<EPL_LOAD_ID></EPL_LOAD_ID>
<EPL_JOB_ID></EPL_JOB_ID>
<EPL_USER_ID></EPL_USER_ID>
<EPL_VEHICLE_ID></EPL_VEHICLE_ID>
<EPL_START_DATE_TIME></EPL_START_DATE_TIME>
<EPL_ARRIVAL_DATE_TIME></EPL_ARRIVAL_DATE_TIME>
<EPL_END_DATE_TIME></EPL_END_DATE_TIME>
<EPL_STATUS></EPL_STATUS>
<EPL_CONTACT></EPL_CONTACT>
<EPL_INVOICED></EPL_INVOICED>
<EPL_SIGNED_UNCHECKED></EPL_SIGNED_UNCHECKED>
<EPL_USER_NOTES></EPL_USER_NOTES>
<EPL_TNCS></EPL_TNCS>
<EPL_AMENDED_FLAG></EPL_AMENDED_FLAG>
<EPL_JOB_STATUS></EPL_JOB_STATUS>
<EPL_UDF_JOBDETS></EPL_UDF_JOBDETS>
<EPL_LINKED_ID></EPL_LINKED_ID>
<CONFIRMATION>
    <EPL_ENG_SIGNATURE></EPL_ENG_SIGNATURE>
    <EPL_JOB_SIGNATURE></EPL_JOB_SIGNATURE>
</CONFIRMATION>
<EXCEPTION type="job">
    <EPL_REASON_CODE></EPL_REASON_CODE>
</EXCEPTION>
<SERVICES>
...
</SERVICES>
<CONTAINERS>
...
</CONTAINERS>
```

The processing of import jobs (in AutoImport and ePOD_Dataservice) must be modified to update this new field. The EPOD_JOB tag contents will now look like this:

```
<EPOD_JOB>
  <EPL_SITE_ID></EPL_SITE_ID>
  <EPL_LOAD_ID></EPL_LOAD_ID>
  <EPL_JOB_ID></EPL_JOB_ID>
  <EPL_JOB_CODE></EPL_JOB_CODE>
  <EPL_JOB_TYPE></EPL_JOB_TYPE>
  <EPL_JOB_GROUP></EPL_JOB_GROUP>
  <EPL_CUST_REF></EPL_CUST_REF>
  <EPL_JOB_INSTRUCTION></EPL_JOB_INSTRUCTION>
  <EPL_OFFICE_INSTRUCTION></EPL_OFFICE_INSTRUCTION>
  <EPL_START_PLANNED_DATE></EPL_START_PLANNED_DATE>
  <EPL_START_PLANNED_TIME></EPL_START_PLANNED_TIME>
  <EPL_END_PLANNED_DATE></EPL_END_PLANNED_DATE>
  <EPL_END_PLANNED_TIME></EPL_END_PLANNED_TIME>
  <EPL_DISTANCE_PLANNED></EPL_DISTANCE_PLANNED>
  <EPOD_CUSTOMER></EPOD_CUSTOMER>
  <EPL_CUSTOMER_CODE></EPL_CUSTOMER_CODE>
  <EPL_CUSTOMER_NAME></EPL_CUSTOMER_NAME>
  <EPL_ADDRESS_1></EPL_ADDRESS_1>
  <EPL_ADDRESS_2></EPL_ADDRESS_2>
  <EPL_ADDRESS_3></EPL_ADDRESS_3>
  <EPL_ADDRESS_4></EPL_ADDRESS_4>
  <EPL_ADDRESS_5></EPL_ADDRESS_5>
  <EPL_POSTCODE></EPL_POSTCODE>
  <EPL_CONTACT></EPL_CONTACT>
  <EPL_TELEPHONE></EPL_TELEPHONE>
  <EPL_EMAIL></EPL_EMAIL>
  <EPL_OWNER_NAME></EPL_OWNER_NAME>
  <EPL_SO_NUMBER></EPL_SO_NUMBER>
  <EPL_ORDER_DATE></EPL_ORDER_DATE>
  <EPL_ORDER_TIME></EPL_ORDER_TIME>
  <EPL_SALES_CONTACT></EPL_SALES_CONTACT>
  <EPL_SERVICE_LEVEL></EPL_SERVICE_LEVEL>
  <EPL_TRAILER_ID></EPL_TRAILER_ID>
  <EPL_JOB_STATUS></EPL_JOB_STATUS>
  <EPL_USER_ID></EPL_USER_ID>
  <EPL_EXT_REF></EPL_EXT_REF>
  <EPL_COL_DATE></EPL_COL_DATE>
  <EPL_SEQUENCE></EPL_SEQUENCE>
  <EPL_LINKED_ID></EPL_LINKED_ID>
  <EPOD_CONTAINERS></EPOD_CONTAINERS>
  <EPOD_PRODUCTS></EPOD_PRODUCTS>
  <EPOD_SERVICES></EPOD_SERVICES>
</EPOD_JOB>
```

Note: All XSDs should be modified to include the new field, along with standard documentation. This includes but is not limited to:

- EPOD_EXPORT_LOAD_RESPONSE.xsd
- XMLUpload.xsd
- EPOD_EXPORT_JOB.xsd
- EPOD_EXPORT_JOB_RESPONSE.xsd
- ePOD_ImportResponce.xsd

The element should be added to the XSDs, in all places where EPOD_JOB is specified, as follows:

```
<xsd:element name="EPL_LINKED_ID" nillable="true" minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>
      An identifier that informs the system that this job should be linked (consolidated) with other job
      The format is irrelevant - if multiple jobs on the same load have the same link ID, they will be c
    </xsd:documentation>
  </xsd:annotation>
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="40" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```


10.3 PDA Changes

10.3.1 General PDA Changes

A custom LFS style will be created, changing the displays as follows:

- The LFS logo will be displayed on the login screen.
- All occurrences of the word "Container" or "Containers" will be exchanged for the words "Package" or "Packages".

10.3.2 PDA DAL/Database Changes

The existing EPOD_JOB table will be modified to add the new fields as follows:

- EPL_LINKED_ID - nvarchar(40) default
- EPL_LINK_ID - nvarchar(40) default
- EPL_LINKED_JOB_ID - nvarchar(10) default null
- EPL_LINKED_SEQUENCE - nvarchar(4) default

These new fields should be added to every procedure as part of the PDA_JOB DAL object.

The method ToUpdateXML of the EPOD_PRODUCT DAL should be modified to add the new EPL_LINKED_ID field to the message, as follows:

```
<EPL_SITE_ID></EPL_SITE_ID>
<EPL_LOAD_ID></EPL_LOAD_ID>
<EPL_JOB_ID></EPL_JOB_ID>
<EPL_USER_ID></EPL_USER_ID>
<EPL_VEHICLE_ID></EPL_VEHICLE_ID>
<EPL_START_DATE_TIME></EPL_START_DATE_TIME>
<EPL_ARRIVAL_DATE_TIME></EPL_ARRIVAL_DATE_TIME>
<EPL_END_DATE_TIME></EPL_END_DATE_TIME>
<EPL_STATUS></EPL_STATUS>
<EPL_CONTACT></EPL_CONTACT>
<EPL_INVOICED></EPL_INVOICED>
<EPL_SIGNED_UNCHECKED></EPL_SIGNED_UNCHECKED>
<EPL_USER_NOTES></EPL_USER_NOTES>
<EPL_TNCS></EPL_TNCS>
<EPL_AMENDED_FLAG></EPL_AMENDED_FLAG>
<EPL_JOB_STATUS></EPL_JOB_STATUS>
<EPL_UDF_JOBDETS></EPL_UDF_JOBDETS>
<EPL_LINKED_ID></EPL_LINKED_ID>
<CONFIRMATION>
    <EPL_ENG_SIGNATURE></EPL_ENG_SIGNATURE>
    <EPL_JOB_SIGNATURE></EPL_JOB_SIGNATURE>
</CONFIRMATION>
<EXCEPTION type="job">
    <EPL_REASON_CODE></EPL_REASON_CODE>
</EXCEPTION>
<SERVICES>
...
</SERVICES>
<CONTAINERS>
...
</CONTAINERS>
```

Note: None of the other new fields above are to be added.

A new column will be added to the EPOD_SITE table:

- EPL_CONSOLIDATION - This will be a one character field with of Values 'Y' and 'N'. The default value is 'N'.

This new field should be added to every procedure as part of the PDA_SITE DAL object.

A new column will be added to the EPOD_JOB_GROUPS table:

- EPL_CONSOLIDATION - This will be a one character field with of Values 'Y' and 'N'. The default value is 'N'.

This new field should be added to every procedure as part of the PDA_JOB_GROUP DAL object.

The PDA_CONTAINER DAL object will be modified to allow retrieval of some job-level information with the container. This will be achieved with a new boolean flag against the creation of the PDA_CONTAINER object. If set, this will get the container information linking to its job record to also retrieve:

- EPL_CUSTOMER_NAME
- EPL_JOB_DISP_ID (a combination of many fields - see PDA_JOB for details).
- EPL_LINKED_SEQUENCE

If the boolean flag is not set, these new fields will be left blank.

The PDA_JOB DAL object will be modified to find and store any jobs linked to in in a new field EPL_LINKED_JOB_ID. This will make retrieval of any linked job easier and faster. This will be done when creating a job on the database, by getting any job on the device with the same EPL_JOB_CODE, EPL_SITE_ID and EPL_LOAD_ID but different EPL_JOB_ID. If one is not found, this new field should be left as null. Modify the existing getLinkedJob methods to use this field instead.

Additionally, this object will also be modified to retrieve the value of EPL_SEQUENCE of any linked job into a new field. Again, at the time of creating a job on the database, store the EPL_SEQUENCE value from the linked job into EPL_LINKED_SEQUENCE. If this is not found, the sequence from the existing job should be stored here.

10.3.3 Job List

- 

Job List screen

The Job List screen will allow users to enter 'Consolidation Mode' through a long-press on a job's row, if configured to do so, through the EPOD_SITE EPL_CONSOLIDATION flag. This will display a pop-up action, allowing the option 'Group Jobs Together'. **Note:** This mode may be used to consolidate jobs together whilst 'on-the-road' or before leaving the depot, whilst loading the vehicle at the depot.

When in Consolidation Mode, selecting Jobs from the Job list will highlight them as selected. Selecting an already-highlighted job will deselect it.

Note: There is no criteria used to identify whether jobs in the collection may be consolidated with other jobs (e.g. Post Code, etc), other than that different Job Types may not be consolidated together (i.e. a delivery with a collection).

A **Consolidate** button will be shown on the bottom of the screen. When clicked, all jobs selected will be consolidated together (by setting EPL_LINK_ID against each job selected to be a unique counting integer for that group on that load) and the Job Details screen will be shown. The jobs that have been consolidated together will be passed to this screen as an object array. Additionally, these jobs grouped together in this way will be consolidated onto a single row in the Job List, displaying identically to pre-consolidated jobs - see below for details.

The **Back** button on the device will exit Consolidation Mode and will not save any changes.

Any jobs that are marked as pre-consolidated (i.e. have a unique value of EPL_LINKED_ID, or a blank value) and jobs that have been manually consolidated together (i.e. have a value of EPL_LINK_ID not blank) will be consolidated together, as long as other jobs exist for those linking id's. This will require the screen to take all jobs in the list with an identical value in this field and group them together into an EPOD_JOBS collection, and create a single entry on the Job List for this consolidation. If no other jobs exist for that linking ID, the linking ID should be blanked.

The fields displayed in Job List rows are:

- Job Identifier - One of: Job ID, Job Code. This is be configured at Site level in *CALIDUS* ePOD and is subject to a change being specified elsewhere to allow other values (e.g. Cust Ref, SO Ref, External Ref). For rows showing a group of consolidated jobs, this will display "X Consignments" instead, where X is the number of jobs grouped together.
- Job Type - "Collection" or "Delivery"
- Customer Name - This is the location name of the address being delivered to or collected from. For an LFS depot collections / driver loading tasks this will show as the depot name.
- Start Planned Date/Time - The planned date and time for the delivery.
- Post Code - Not displayed, at customer request.
- Container Count - The number of containers (packages) on the Job. For rows showing a group of consolidated jobs, this will show the total number of containers in all the grouped jobs.
- Product Count - Not displayed, as not applicable.

Note: Customer Name will take the majority of space on this row, across the center. Other elements will be distributed around these elements in a smaller font.

Sample New Job List Rows

Clicking on one of these consolidated jobs on the screen will display the Job Details screen. The jobs that have been consolidated together will be passed to this screen as an object array.

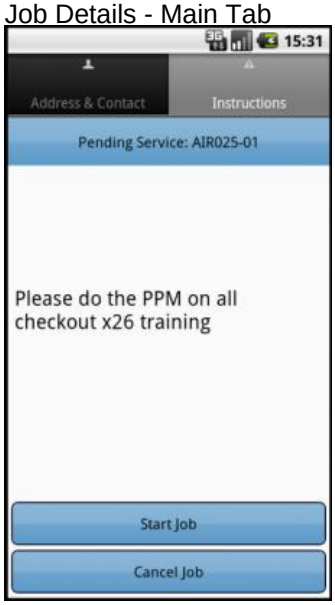
It will be possible to consolidate a pre-consolidated group of jobs with another group or other single jobs, if all the jobs being linked pass the criteria checks (at this time, Job Type is the same). In this case, *all* of the jobs, including the pre-consolidated ones, will be linked together by a common value in EPL_LINKED_ID, in preference to anything passed in to *CALIDUS* ePOD.

It will be possible to break a consolidated group of jobs into single jobs again, by long-holding on a consolidated group's row and choose *Break Group*. This will refresh the Job List and put all the group's constituent jobs back into the list on their own lines. **Note:** Any jobs that were originally pre-consolidated by TMS will be displayed back into their original consolidation group.

The option *Cancel Job* on the pop-up menu will display as *Cancel Group* when showing this menu when long-pressing on a consolidated group row. This will call the standard Cancellation screen once only, allowing the user to identify the reason for the cancellation and potentially also an image. Once confirmed, a cancellation message should be sent for all jobs with the same reason. **Note:** If an image is taken, that image will be attached to the first job ONLY. Regardless of how this cancellation was run, the user will be returned to the Job List screen, redisplaying the job list again.

10.3.4 Job Details

- 

Job Details - Main Tab
- 

Job Details - Instructions Tab

The Job Details screen will be modified to recognise whether it has been called with a consolidated jobs object array, or a single job, as now.

If the screen is called with a single job, this screen will act as it does now.

If the screen is called with a object array of jobs, several changes will be made:

- **<** and **>** (**Previous** and **Next**) buttons will be added to the title of the *Address & Contact* tab.
- The Title text on this tab will be modified to display the number the job is in the consolidated jobs object array (e.g. "1 of 5").
- The **Start Job** button will have its label changed to indicate this affects all jobs, for all labels of this button, i.e. **Start Jobs**, **Arrive Jobs** and **Continue Jobs** on both existing tabs.

When the **Prev** and **Next** buttons are pressed, this will navigate to the next or previous job in the consolidated group. If the **Prev** button is hit on the first job, this will move to the last job. If the **Next** button is hit on the last job, this will move to the first. When a new job is moved to, the *Address & Contact* tab will refresh, changing the data displayed to show the information for the job now selected. This will allow the user to see all detailed information for all jobs before choosing to start the jobs.

The *Job Instructions* tab will be modified to show *all* Job Instructions from all jobs on one scrollable screen. For each job that has instructions, a line will be shown, saying "Instructions for job

EPL_JOB_DISP_ID:\nEPL_JOB_INSTRUCTIONS\n".

When ready to drive to a group of jobs, the user will press the **Start Jobs** button. This will mark all of the jobs as In Progress, with the Actual Start Date and Time. When pressed, the label will change to **Arrive Jobs**.

Note: The functionality in the Android application for Job Lock updates will still be in force, whereby all job lock messages could potentially update the jobs at this point. The functionality in the existing Windows Mobile application should be copied where appropriate to ensure that this updating works as expected when processing a consolidated group.

When the user arrives at the destination, they will press the **Arrive Jobs** button. This will mark all of the jobs as Arrived, with the Arrival Date and Time. The label will change to **Continue Jobs** and the Collection or Delivery screen will be displayed. These will be called passing a collection of Jobs rather than a single job.

When processing a consolidated group of jobs, the **Cancel Job** button's label will be modified to **Cancel Jobs**. When pressed, the screen should offer a choice to the user as follows:

- Cancel Job EPL_JOB_DISP_ID only?
- Cancel all jobs in this group?
- Do nothing.

Regardless of which of the first two choices are made, the existing cancellation screen should be called once only, allowing the user to identify the reason for the cancellation and potentially also an image. Once confirmed, if this was a single job being cancelled, the screen should run as now, cancelling only that job. If the user chose to cancel all jobs, a cancellation message should be sent for all jobs with the same reason. **Note:** If an image is taken, that image will be attached to the first job ONLY. Regardless of how this cancellation was run, the user will be returned to the Job List screen, redisplaying the job list again.

10.3.5 Collection / Delivery

Note: The processes are identical within *CALIDUS* ePOD and are shown below. For convenience, these will be referred to as Deliveries below. For Collections, all labels will display as "Collect", "Collection", etc.

- 

Collection/Delivery - Job Details Tab

- 

Collection/Delivery - Containers Tab

Note: For the purposes of deciding upon which settings are used to control how the consolidated jobs are processed, the Job Group of the first job will be used.

The Delivery screen will be modified to recognise whether it has been called with an consolidated jobs object array, or a single job, as now.

If the screen is called with a single job, this screen will act as it does now.

If the screen is called with a jobs object array, several changes will be made:

- **<** and **>** (**Previous** and **Next**) buttons will be added to the title of the *Job Details* tab on this screen.
- The Title text on this tab will be modified to display the number the job is in the consolidated jobs object array (e.g. "1 of 5").
- The *Container* and *Products* tabs' titles will be changed to show the text "X Consignments", where X is the number of jobs in the consolidated group.
- **Note:** The style should have changed the title and tab content of the *Containers* tab to "Packages".

When the **Prev** and **Next** buttons are pressed, this will navigate to the next or previous job in the consolidated group. If the **Prev** button is hit on the first job, this will move to the last job. If the **Next** button is hit on the last job, this will move to the first. When a new job is moved to, the *Job Details* tab will refresh, changing the data displayed to show the information for the job now selected. This will allow the user to see all detailed information for all jobs individually.

The existing area used to display the contact, telephone and planned times will be modified into a custom tab object, an example of which can be found on the Signature screen. This will be populated as follows:

- Tab 1 - Info
  - Contact
  - Telephone
  - Planned Start Date Time
- Tab 2 - Job Address

Other tabs can then be used for further extended information, for example to display original collection/ultimate delivery address, a potential future change.

The *Job Details* tab on this screen already displays (and will continue to display):

- Any special instructions **Note:** This is just for the job being displayed at this time, unlike the *Job Details* screen above.
- Any additional check flags
- Any additional entry fields

This section of the screen is scrollable by the user.

The *Container* and *Products* tabs will also be modified if the screen has been passed a jobs object array.

The Container list will be modified from its current contents:

For standard jobs:

- Container ID
- Description - The Distribution Unit passed in (e.g. PACKAGE, PALLET).
- Weight - As received
- Product Count - This will not be displayed as there are no products.
- Status - this will be left at this time, as a further future change may require this to be visible.

For consolidations:

- Job Identifier - EPL_JOB_DISP_ID
- Customer Name - from the Job.

Additionally, the list will be shown in reverse Job sequence, as provided by the *CALIDUS* TMS system against the Delivery jobs.

Note: To achieve this, the EPOD_CONTAINER DAL object will need to be called requesting this extra Job-level information - this is specified above.

Long-pressing on a container and choosing *Info* will show more information about this container in a pop-up screen. This will be modified to add the following fields, if this is part of a consolidated group of jobs:

- Job Identifier - EPL_JOB_DISP_ID
- Customer Name - from the Job.

If a job needs to be deconsolidated from the list the driver can long hold the container for the job, this will open up the further options menu where a 'Deconsolidate Job' menu item will be added. If the driver selects the 'Deconsolidate Job' menu item the job and all containers associated to this job will be removed from the list (blanking its EPL_LINK_ID and EPL_LINKED_ID values) and the container list redisplayed. If all jobs have been completed at this time, this should automatically move on to job completion.

Note: The deconsolidated job will be placed back onto the Job List screen to be processed as a normal stand-alone job, the next time the user returns to this screen.

Once all containers (for all jobs if consolidated) have been marked as either delivered, deconsolidated or cancelled, the user will be taken to Job Completion stage automatically. If this is a consolidated group, this should pass the job collection to the screen.

10.3.6 Job Completion

If the location is configured to not require signature (as would be the case for Depot locations, where Loading and Unloading is handled through CALIDUS EPOD (through the configuration shown in section 1.2.5 Job Group Signature control), the driver will not be taken to the signature screen. If signature capture is required, the screen will be shown.

The existing Signature screen needs some modifications if the jobs being completed are consolidated.

- 

Signature screen

Note: For the purposes of deciding upon which settings are used to control how the consolidated jobs are processed, the Job Group of the first job will be used.

The general functionality of this screen when signing for a consolidated group of jobs is as follows:

- One signature is required for all jobs
- One set of T&Cs will be displayed (from the first job) and used against all jobs.
- The *Containers* tab will show a consolidated view of all containers for all jobs.
- The *Products* tab will show a consolidated view of all Products for all jobs.

Clicking on a container in the *Containers* tab will show more information about this container in a pop-up screen, allowing the customer to enter notes against a package. This pop-up screen will be modified to add the following fields, if this is part of a consolidated group of jobs:

- Job Identifier - EPL_JOB_DISP_ID
- Customer Name - from the Job.

When the signature has been recorded, the device will move onto the next stage. Generically, this could be a driver signature or job photograph. However, it is expected in this configuration to update the job or jobs.

Note: If Job Photo *is* used, this photo will be stored against the first job in the consolidation *only*.

For a consolidated group of jobs, all jobs will be marked with the T&Cs and signature from the signature screen and all jobs will be updated back to the server through the existing web services. This job update message will be modified to include the new EPL_LINKED_ID field (as specced above). This will be set to (in order of precedence):

- EPL_CUSTOMER_CODE (the location) of the first job in any group linked by EPL_LINK_ID)
- Blank (if not manually consolidated)

Note: This means that, even if the device has been passed a value in EPL_LINK_ID from *CALIDUS* TMS, this will *not* be passed back here. Only manual consolidations will be returned in this message, so that TMS knows which unlinked addresses to link together.

This will be saved in the server through the Job Update message on the Calidus_EPOD.asmx webservice.

Completed and Cancelled jobs will be automatically exported to *CALIDUS* TMS through the standard interface. This will include the new field values.

11 Appendix A: Quote & Document References

Cost Details				
Activity	**Estimate No. of Days**	**No. of Days**	**Rate per Day (?)**	**Cost (? Exc. VAT)**
Requirements	0.00	0.00	0	?0.00
Change Request Evaluation	0.00	0.00	0	?0.00
Functional Specification	0.00	2.00	0	?0.00
Technical Specification	0.00	0.00	0	?0.00
Development	0.00	20.00	0	?0.00
Testing and Release	0.00	3.00	0	?0.00
Implementation	0.00	0.25	0	?0.00
Project Management	First argument to "number_format" must be a number.	First argument to "number_format" must be a number.	0	?First argument to "number_format" must be a number.
TOTAL	First argument to "number_format" must be a number.	First argument to "number_format" must be a number.		?First argument to "number_format" must be a number.

Estimate excludes training, release to live and go live support.

A.1 References

Ref No	Document Title & ID	Version	Date
1	FS-308745 - ID5 - Job Consolidation	1.4	03/07/2013

A.2 Glossary

Term	Definition
EPOD	Electronic Proof of Delivery. The OBS EPOD system is *CALIDUS* ePOD.
CALIDUS eSERV	The OBS mobile system to complete Service functionality in the field. This is part of the *CALIDUS* ePOD system.
PDA	The mobile device on which the C-ePOD system will run in the field. This can be a Phone, EDA or industrial PDA, running Android.
DAL	Data Access Layer. A mechanism for accessing data by the system that is removed from the application, allowing for simplified access and providing protection to the data, as only approved DAL methods can be used to modify it.
GPS	Global Positioning System. A mechanism of retrieving accurate positioning information in the form of Latitude and Longitude (Lat-Long) co-ordinates from a device.
GPRS, 3G, HSDPA, Data Service	All terms referring to mobile device network connectivity, and the speed at which the device connects to the internet.

A.3 Authorised By

Matt Tipping Project Manager _____

_____(PRINT) Client Representative _____