

CHALMERS



Mapping of CPM LCA database SPINE format to ILCD data format

JOHAN TIVANDER

Environmental Systems Analysis

CPM – the Swedish Life Cycle Center

CPM Report No. 2013:1

CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden, 2013

Acknowledgements

The project participants would like to express their gratitude to Vinnova for the financial support.

Introduction

In the project “Life cycle data conversion to international standard” financed by Vinnova in 2012 (see project description in Appendix B), the main goal was to produce a conversion function that enables LCI data to be downloaded in ILCD format from the CPM Database. This report defines the mapping from SPINE formatted data in CPM LCA Database to the ILCD format implemented in the conversion function.

Mapping definition from SPINE to ILCD

The mapping is based on the implementation of SPINE in CPM LCA Database (CPM 2013), the original definition of SPINE (Carlson et al 1995), the mapping between SPINE and ISO/TS 14048 (Tivander et al 2003), and the ILCD SDK development aid documents (EC-JRC 2013).

The mapping from each relevant SPINE database table and field is described in table 1. In addition to this preset text (or automatically generated GUID identifiers) are added to several ILCD fields described in table 2. A number of permanent xml reference files for definition of flow properties, contacts, etc. have also been created.

The implemented code based on the mapping is found in Appendix A.

Carlson R, Löfgren G, Steen B, et al, 1995, “SPINE – a relation database structure for life cycle assessments.”, Technical and Environmental Planning, Chalmers University of Technology, Gothenburg, Sweden

CPM 2013, “CPM LCA Database”, public LCA database, online <http://cpmdatabase.cpm.chalmers.se>

Tivander J, Carlson R, Erlandsson M, Flemström K, Pålsson A-C, Tidstrand U, 2003, “Data format mapping between SPINE and ISO/TS 14048”, CPM report 2003:8

EC-JRC, 2012, “The ILCD data set format SDK (ILCD Format SDK 1.1 Maintenance Release (MR) 2)”, programming development support files, online: <http://lca.jrc.ec.europa.eu/lcainfohub/developerPage.vm>

Table 1 Mapping from SPINE to ILCD

From SPINE		->	To ILCD field	Mapping Comment
CPM LCA Database	SPINE Database table and field			
Administrative data				
ObjectOfStudy Id	ObjectOfStudy.Id	Never	-	Not mapped. See Activity.Id
Activity Id	Activity.Id	Always	Administrative information - Publication and ownership - RegistrationNumber	Activity.Id is sufficient identifier of a process data set as it is a 1 to 1 mapping between ObjectOfStudy.Id and Activity.Id in CPM LCA Database
Finished	Activity.Finished	Never	-	Only processes with Activity.Finished value "yes" are published in CPM LCA Database
Date Completed	Inventory.DateCompleted	If data exists	Administrative information - Data entry by - Other content	Mapped together with fixed text: "Original CPM LCA Database dataset completed: " [Date Completed]
		If data exists	Administrative information - Publication and ownership - Date of last revision	Identical.
Copyright	Inventory.Copyright	Never	-	Not mapped. The ILCD Administrative information - Publication and Ownership - Copyright? is set to fixed value: "true". Furthermore the ILCD Administrative information - Publication and Ownership - Access and use restrictions is set to fixed value: "The CPM LCA Database is the property of the CPM consortium partners. Since year 2008 the CPM LCA Database is accessible free of charge through the website http://cpmdatabase.cpm.chalmers.se . Any commercial redistribution of data originating from CPM LCA Database; either in separate parts or complete database; either in the data formats provided at this website or translated into other formats; is not allowed without a prior written agreement with the CPM consortium." based on the Copyright statement text of CPM LCA Database.
Availability	Inventory.Availability	Never	-	Not mapped. CPM LCA Database data is always public. ILCD Administrative information - Publication and Ownership - Access is the best matching concept which is mapped with fixed text (see above).

From SPINE		->	To ILCD field	Mapping Comment
CPM LCA Database	SPINE Database table and field			
Technical system data				
Process Name	ObjectOfStudy.Name	Always	Process information - Key Data Set Information - Name - Base name	Identical.
Functional Unit	Inventory.FunctionalUnit	Always	Process information - Key Data Set Information - Name - Quantitative product or process properties	Identical.
		Always	Process information - Quantitative reference - Functional unit, Production period, or Other parameter	Identical.
Process Type	ObjectOfStudy.Category	Always	Modelling and validation - LCI method and allocation - Other content	Identical.
Site	JuridicalPerson via ObjectOfStudy.Site	If data exists	Process information - Key Data Set Information - Geographical representativeness - Sub-location(s) - Geographical representativeness description	Concatenated string of all fields in table JuridicalPerson. The location attribute of ILCD Sub-location(s) is set to "GLO".
Sector	ObjectOfStudy.Sector	If data exists	Process information - Key Data Set Information - Classification () - Class name	ILCD Class name - Hierarchy level attribute get a fixed value: "1". Note: it seems the ILCD schema implementation does not fully match the ILCD ProcessDataSet definition. The classification system and the hierarchy level is not shown in the html output rendered by the schema.
Owner	JuridicalPerson via ObjectOfStudy.Owner	If data exists	Process information - Key Data Set Information - Technological representativeness - Technology description including background system	Mapped as a concatenated string of all fields in table JuridicalPerson. Concatenated with ObjectOfStudy.Function and fixed string: "Owner: ".
Function	ObjectOfStudy.Function	Always	Process information - Key Data Set Information - Technological representativeness - Technology description including background system	Always mapped but also concatenated with ObjectOfStudy.Owner if Owner data exists.

From SPINE		->	To ILCD field	Mapping Comment
CPM LCA Database	SPINE Database table and field			
System boundary data				
Functional Unit Explanation	Inventory.FUExplanation	If data exists	Process information - Quantitative reference - Other content	Identical.
Nature Boundary	Inventory.NatureBoundary	If data exists	Process information - Key Data Set Information - Technological representativeness - Other content	Mapped if SPINE data exists. Concatenated with Other Boundaries.
Time Boundary	Inventory.TimeBoundary	If data exists	Process information - Key Data Set Information - Time representativeness - Reference year	Mapped here if Time Boundary can be converted to a Date format.
		If data exists	Process information - Key Data Set Information - Time representativeness - Time representativeness description	Mapped here if Time Boundary cannot be converted to a Date format.
Geographical Boundary	Inventory.GeographicalBoundary	If data exists	Process information - Key Data Set Information - Geographical representativeness - Location - Geographical representativeness description	The location attribute of ILCD Location is set to "GLO".
Other Boundaries	Inventory.OtherBoundaries	If data exists	Process information - Key Data Set Information - Technological representativeness - Other content	Concatenated with Nature Boundary.
Allocations	Inventory.Allocations	If data exists	Modelling and validation - LCI method and allocation -Deviations from LCI method approaches / explanations	Concatenated with System Expansions
System Expansions	Inventory.LateralExpansion	If data exists	Modelling and validation - LCI method and allocation -Deviations from LCI method approaches / explanations	Concatenated with Allocations

From SPINE		->	To ILCD field	Mapping Comment
CPM LCA Database	SPINE Database table and field			
General Flow Metadata				
General Activity QMetadata	QMetadata.Activity.Metald	Never		
Date Conceived	QMetadata.DateConceived	If data exists	Modelling and validation - Data sources, treatment, and representativeness - Data selection and combination principles	Concatenated with all QMD fields but Literature reference. Concatenated with fixed value: "See general comment on data set".
Data Type	QMetadata.DataType	If data exists	Modelling and validation - Data sources, treatment, and representativeness - Data selection and combination principles	Concatenated with all QMD fields but Literature reference. Concatenated with fixed value: "See general comment on data set".
Method	QMetadata.Method	If data exists	Modelling and validation - Data sources, treatment, and representativeness - Data selection and combination principles	Concatenated with all QMD fields but Literature reference. Concatenated with fixed value: "See general comment on data set".
Literature Reference	QMetadata.LitteratureRef	If data exists	Modelling and validation - Data sources, treatment, and representativeness - Data source(s) used for this data set	Mapped only to ILCD shortDescription element. No URI defined.
Represents	QMetadata.Represents	If data exists	Modelling and validation - Data sources, treatment, and representativeness - Data selection and combination principles	Concatenated with all QMD fields but Literature reference. Concatenated with fixed value: "See general comment on data set".
Notes	QMetadata.Notes	If data exists	Modelling and validation - Data sources, treatment, and representativeness - Data selection and combination principles	Concatenated with all QMD fields but Literature reference. Concatenated with fixed value: "See general comment on data set".

From SPINE		->	To ILCD field	Mapping Comment
CPM LCA Database	SPINE Database table and field			
Specific flow data				
Flow Number	Flow.FlowNumber	Always	Inputs and Outputs - Inputs and Outputs - Data set internal ID	Identical.
Direction	Flow.Subtype	Always	Inputs and Outputs - Inputs and Outputs - Exchange direction	Identical.
		Always	FLOW DATA SET: Flow information - Data set information - Elementary flow categorization	If elementary flow. Translated as "from" if input or as "to" if output. See Flow Type.
		Always	FLOW DATA SET: Flow information - Data set information - Classification	If non elementary flow. Translated as "from" if input or as "to" if output. See Flow Type.
Flow Type	Flow.Category	Always	FLOW DATA SET: Flow information - Data set information - Elementary flow categorization	If elementary flow. Concatenated with Direction and Environment: [Flow Type] [to/from] [Environment]
		Always	FLOW DATA SET: Flow information - Data set information - Classification	If non elementary flow. Concatenated with Direction and Environment: [Flow Type] [to/from] [Environment]
Substance	Substance.DefaultName via Flow.Substanceld	Always	Inputs and Outputs - Inputs and Outputs - Flow	To ILCD element common:shortDescription displays as text of the link to corresponding FLOW DATA SET xml file.
		Always	FLOW DATA SET: Flow information - Data set information - Name - Base name	Identical.
Quantity	Flow.Quantity	If data exists	Inputs and Outputs - Inputs and Outputs - Mean amount	Recalculated to match ILCD standard unit if SPINE data is given in other unit.
		If data exists	Inputs and Outputs - Inputs and Outputs - Resulting amount	Recalculated to match ILCD standard unit if SPINE data is given in other unit.
Min	Flow.QuantityMin	If data exists	Inputs and Outputs - Inputs and Outputs - Minimum amount	Recalculated to match ILCD standard unit if SPINE data is given in other unit.

From SPINE		->	To ILCD field	Mapping Comment
CPM LCA Database	SPINE Database table and field			
Max	Flow.QuantityMax	If data exists	Inputs and Outputs - Inputs and Outputs - Maximum amount	Recalculated to match ILCD standard unit if SPINE data is given in other unit.
Sdev	Flow.StandardDeviation	Never		
Unit	Flow.Unit	Always	FLOW DATA SET: Quantitative reference - Reference flow property	Translated to corresponding ILCD standard unit if SPINE data is another unit. Mapped as uri reference to permanent ILCD formatted flow property xml files.
Environment	Flow.ImpactMedia	Always	FLOW DATA SET: Flow information - Data set information - Elementary flow categorization	If elementary flow. Concatenated with Direction and Environment: [Flow Type] [to/from] [Environment]
		Always	FLOW DATA SET: Flow information - Data set information - Classification	If non elementary flow. Concatenated with Direction and Environment: [Flow Type] [to/from] [Environment]
Geography	Geography.AreaName via Flow.ImpactRegion	If data exists	Inputs and Outputs - Inputs and Outputs - Location	Identical.
Specific Flow QMetaData	QMetaData.Id via Flow.Metald	Never		
Date Conceived	QMetaData.DateConceived	If data exists	Inputs and Outputs - Inputs and Outputs - Comment	Concatenated with all QMD fields but Literature reference.
Data Type	QMetaData.DataType	If data exists	Inputs and Outputs - Inputs and Outputs - Comment	Concatenated with all QMD fields but Literature reference.
Method	QMetaData.Method	If data exists	Inputs and Outputs - Inputs and Outputs - Comment	Concatenated with all QMD fields but Literature reference.
Literature Reference	QMetaData.LitteratureRef	If data exists	Inputs and Outputs - Inputs and Outputs - Data source(s) - Data source(s)	Mapped only to ILCD shortDescription element. No URI defined.
Represents	QMetaData.Represents	If data exists	Inputs and Outputs - Inputs and Outputs - Comment	Concatenated with all QMD fields but Literature reference.
Notes	QMetaData.Notes	If data exists	Inputs and Outputs - Inputs and Outputs - Comment	Concatenated with all QMD fields but Literature reference.

From SPINE		->	To ILCD field	Mapping Comment
CPM LCA Database	SPINE Database table and field			
About Inventory				
Publication	Inventory.Publication	If data exists	Process information - Key Data Set Information - Data set LCA report, background info	Mapped only to ILCD shortDescription element. No URI defined.
Intended User	Inventory.IntendedUser	If data exists	Administrative information - Commissioner and goal - Intended applications	Concatenated with General purpose, Detailed purpose, and Applicability
General Purpose	Inventory.GeneralPurpose	If data exists	Administrative information - Commissioner and goal - Intended applications	See Intended user
Detailed Purpose	Inventory.DetailedPurpose	If data exists	Administrative information - Commissioner and goal - Intended applications	See Intended user
Commissioner	JuridicalPersion via Inventory.Commissioner	If data exists	Administrative information - Commissioner and goal - Commissioner of data set	Mapped only to ILCD shortDescription element. No URI defined.
Practitioner	JuridicalPersion via Inventory.Practitioner	If data exists	Administrative information - Data set generator / modeller - Data set generator / modeller (contact data set)	Mapped only to ILCD shortDescription element. No URI defined.
Reviewer	JuridicalPersion via Inventory.Reviewer	If data exists	Modelling and validation - Validation - Review - Reviewer name and institution	Concatenated with all QMD fields but Literature reference. Mapped only to ILCD shortDescription element. No URI defined.
Applicability	Inventory.Applicability	If data exists	Administrative information - Commissioner and goal - Intended applications	See Intended user
About Data	Inventory.Data	If data exists	Process information - Key Data Set Information - General comment on data set	Identical.
Notes	Inventory.Notes	If data exists	Modelling and validation - Data sources, treatment, and representativeness - Other content	Identical.

From SPINE		->	To ILCD field	Mapping Comment
CPM LCA Database	SPINE Database table and field			
Other				
CPM quality indicator	Prettyness.Ddocumentation	Always	Modelling and validation - Validation - Review - Review details	Concatenated with fixed value: "All LCI data sets in the CPM LCA Database have unergone review according to the CPM review process in order to ensure fulfilment of CPM's documentation criteria. See documentation at http://cpmdatabase.cpm.chalmers.se/AboutDatabase . This data set has been categorised as: " [CPM Quality indicator] ILCD Review Scope added with fixed value: "Documentation" ILCD Method(s) of review added with fixed value: "Expert judgement" ILCD Reviewer name and institution added fixed value "CPM" with link to permanent CPM contact data set.

Table 2 additional fields added with fixed values

ILCD field	Preset text	Comment
PROCESS DATA SET		
version	"1.1"	Header of process xml data file
location	"../ILCDLocations.xml"	Header of process xml data file. Relative link to location table.
Meta data only	"false"	Header of process xml data file. Only data sets with quantitative flow data are available in CPM LCA Database.
Process information		
Key Data Set Information		
UUID of Process data set	Automatically generated UUID string	the unique identifier of the ILCD process data set. The xml file is also named: "CPM_process_" + [UUID] + ".xml"
Data set LCA report, background info	"This data set has been exported from the CPM LCA Database"	Referencing the ILCD source data set CPM_LCA_Database_ddb15c9a-ca85-480b-adb0-d5b7a2d30395.xml
Modelling and validation		
LCI method and allocation		
Type of data set	"LCI result"	Chosen as best match from ILCD preset nomenclature
LCI method principle	"Other"	Chosen as best match from ILCD preset nomenclature
Deviation from LCI method principle / explanations	"none"	No deviation from other (arbitrary) method.
LCI method approaches	Not applicable	CPM LCA Database datasets has no indicator of normative LCI method(s).
Data sources, treatment, and representativeness		
Use advice for data set	"See intended applications"	Intended User, General Purpose, Detailed Purpose, Applicability are all mapped to ILCD Intended applications.
Validation		
Review		
Type of review	"Independent external review"	Chosen as best match from ILCD preset nomenclature
Scope of review - Scope name	"Documentation"	Chosen as best match from ILCD preset nomenclature
Method(s) of review - Method name	"Documentation" and "Expert judgement"	Chosen as best match from ILCD preset nomenclature
Review details	"All LCI data sets in the CPM LCA Database have unergone review according to the CPM review process in order to ensure fulfilment of CPM's documentation criteria. See documentation at http://cpmdatabase.cpm.chalmers.se/AboutDatabase ."	If CPM Quality indicator data exists it is appended to this text (see table 1).

ILCD field	Preset text	Comment
Reviewer name and institution	"CPM"	Referencing the ILCD contact data set CPM_0b8d9a23-1f44-4f17-999f-f1d1120701ee.xml
Compliance declarations		
Compliance - Compliance system name	"ILCD Compliance"	Referencing the ILCD source data set ILCD_Compliance_88d4f8d9-60f9-43d1-9ea3-329c10d7d727.xml
Compliance - Approval of overall compliance	"Not defined"	Chosen as best match from ILCD preset nomenclature. No review of the compliance of this mapping has been conducted.
Compliance - Nomenclature compliance	"Not defined"	Chosen as best match from ILCD preset nomenclature. No review of the compliance of this mapping has been conducted.
Compliance - Methodological compliance	"Not defined"	Chosen as best match from ILCD preset nomenclature. No review of the compliance of this mapping has been conducted.
Compliance - Review compliance	"Not defined"	Chosen as best match from ILCD preset nomenclature. No review of the compliance of this mapping has been conducted.
Compliance - Documentation compliance	"Not defined"	Chosen as best match from ILCD preset nomenclature. No review of the compliance of this mapping has been conducted.
Compliance - Quality compliance	"Not defined"	Chosen as best match from ILCD preset nomenclature. No review of the compliance of this mapping has been conducted.
Administrative information		
Data entry by		
Time stamp (last saved)	Automatically generated time data	Date + Time + Timezone
Data set format(s)	"ILCD format"	Referencing the ILCD source data set ILCD_Format_a97a0155-0234-4b87-b4ce-a45da52f2a40.xml
Converted original data set from:	"CPM LCA Database SPINE format"	Referencing the ILCD source data set CPM_LCA_Database_SPINE_Format_d4c9462b-f7aa-467a-85ef-b369960fa732.xml
Data entry by:	"See: General comment on data set (No URI available)"	No uri reference.
Official approval of data set by producer/operator:	"CPM"	Referencing the ILCD contact data set CPM_0b8d9a23-1f44-4f17-999f-f1d1120701ee.xml
Publication and ownership		
Data set version	"01.00.00"	
Permanent data set URI	"cpmdatabase.cpm.chalmers.se/ILCD/data/processes/CPM_process_" + [UUID of process data set] + ".xml"	
Workflow and publication status	Data set finalised; entirely published	Chosen as best match from ILCD preset nomenclature.

ILCD field	Preset text	Comment
Registration authority	"CPM"	Referencing the ILCD contact data set CPM_0b8d9a23-1f44-4f17-999f-f1d1120701ee.xml
Owner of data set	"CPM"	Referencing the ILCD contact data set CPM_0b8d9a23-1f44-4f17-999f-f1d1120701ee.xml
Copyright?	"true"	
License type	Free of charge for some user types or use types	Chosen as best match from ILCD preset nomenclature.
Access and use restrictions	"The CPM LCA Database is the property of the CPM consortium partners. Since year 2008 the CPM LCA Database is accessible free of charge through the website http://cpmdatabase.cpm.chalmers.se . Any commercial redistribution of data originating from CPM LCA Database; either in separate parts or complete database; either in the data formats provided at this website or translated into other formats; is not allowed without a prior written agreement with the CPM consortium."	Based on CPM Copyright disclaimer.
FLOW DATA SET		
Flow information		
Data set information		
UUID of flow data set	Automatically generated UUID string	the unique identifier of the ILCD flow data set. The xml file is also named: "CPM_flow_" + [UUID] + ".xml"
Modelling and validation		
Compliance declarations		
Compliance - Compliance system name	"ILCD Data Network compliance"	
Compliance - Approval of overall compliance	"Not defined"	Chosen as best match from ILCD preset nomenclature.
Administrative information		
Data entry by		
Time stamp (last saved)	Automatically generated time data	Date + Time + Timezone
Data set format(s)	"ILCD format"	Referencing the ILCD source data set ILCD_Format_a97a0155-0234-4b87-b4ce-a45da52f2a40.xml
Data entry by:	"CPM Swedish Life Cycle Center - LCA Database"	Referencing the ILCD contact data set CPM_0b8d9a23-1f44-4f17-999f-f1d1120701ee.xml
Publication and ownership		
Data set version	"01.00.00"	
Permanent data set URI	"cpmdatabase.cpm.chalmers.se/ILCD/data/flows/CPM_flow_" + [UUID of process data set] + ".xml"	

ILCD field	Preset text	Comment
Owner of data set	"CPM"	Referencing the ILCD contact data set CPM_0b8d9a23-1f44-4f17-999f-f1d1120701ee.xml
Flow properties		
Flow property		
Mean value (of flow property)	"1.0"	
Minimum value	"1.0"	
Maximum value	"1.0"	
Uncertainty distribution type	"undefined"	Chosen as best match from ILCD preset nomenclature. (This is really a definition of the base property and hence no of the words from the ILCD preset nomenclature is a good match.)
Relative StdDev in %	"0"	
Data derivation type / status	"Measured"	Chosen as best match from ILCD preset nomenclature. (This is really a definition of the base property and hence no of the words from the ILCD preset nomenclature is a good match.)

Appendix A CPM LCA Database ILCD.asp webpage code

```
Function MakeILCDProcessDataSetXMLString(strAId, strGuid)
```

```
    strSQL = "SELECT ObjectOfStudy.Id AS OId, Activity.Id AS AId, ObjectOfStudy.Name,
ObjectOfStudy.ActivityType, ObjectOfStudy.Sector, ObjectOfStudy.Site, ObjectOfStudy.Owner,
ObjectOfStudy.Category, ObjectOfStudy.Function, Inventory.Practitioner, Inventory.Reviewer,
Inventory.Commissioner, Inventory.IntendedUser, Inventory.GeneralPurpose, Inventory.DetailedPurpose,
Inventory.FunctionalUnit, Inventory.FUEExplanation, Inventory.Copyright, Inventory.Availability,
Inventory.Publication, Inventory.DateCompleted, Inventory.Applicability, Inventory.Data,
Inventory.LateralExpansion, Inventory.Allocations, Inventory.NatureBoundary, Inventory.TimeBoundary,
Inventory.GeographicalBoundary, Inventory.OtherBoundaries, Inventory.Notes, Activity.MetaId,
QMetaData.DataType AS QMDDataType, QMetaData.Method AS QMDMethod, QMetaData.DateConceived AS
QMDDateConceived, QMetaData.LitteratureRef AS QMDLiteratureRef, QMetaData.Notes AS QMDNotes,
QMetaData.Represents AS QMDRepresents "
    strSQL = strSQL & "FROM ObjectOfStudy INNER JOIN ((Activity INNER JOIN Inventory ON Activity.Id =
Inventory.ActivityId) INNER JOIN QMetaData ON Activity.MetaId = QMetaData.Id) ON ObjectOfStudy.Id =
Activity.ObjectId "
    strSQL = strSQL & "WHERE Activity.Id = '" & strAId & "'"

    Set rsProc = db.Execute(strSQL)

    strXML = "<?xml version=""1.0"" encoding=""UTF-8""?>"
    strXML = strXML & vbCrLf & "<?xml-stylesheet type='text/xsl'
href='../././stylesheets/process2html.xsl' ?>"
    strXML = strXML & vbCrLf & "<processDataSet xmlns=""http://lca.jrc.it/ILCD/Process""
xmlns:common=""http://lca.jrc.it/ILCD/Common""
xmlns:process=""http://lca.jrc.it/ILCD/Process""
xmlns:xsi=""http://www.w3.org/2001/XMLSchema-
instance""
locations=""../ILCDLocations.xml""
metaDataOnly=""false""
Version=""1.1""
xsi:schemaLocation=""http://lca.jrc.it/ILCD/Process
../././schemas/ILCD_ProcessDataSet.xsd"">"
    strXML = strXML & vbCrLf & "<processInformation>"
    strXML = strXML & vbCrLf & "<dataSetInformation>"
    strXML = strXML & vbCrLf & "<common:UUID>" & strGuid & "</common:UUID>"
    strXML = strXML & vbCrLf & "<name>"
' OOS.Name
    strXML = strXML & vbCrLf & "<baseName xml:lang=""en"">" & FixXML(rsProc.fields("Name")) &
"</baseName>"
    strXML = strXML & vbCrLf & "<treatmentStandardsRoutes
xml:lang=""en"">N/A</treatmentStandardsRoutes>"
    strXML = strXML & vbCrLf & "<mixAndLocationTypes xml:lang=""en"">N/A</mixAndLocationTypes>"
' Inventory.FunctionalUnit
    strXML = strXML & vbCrLf & "<functionalUnitFlowProperties xml:lang=""en"">" &
FixXML(rsProc.fields("FunctionalUnit")) & "</functionalUnitFlowProperties>"
    strXML = strXML & vbCrLf & "</name>"
    strXML = strXML & vbCrLf & "<classificationInformation>"
    strXML = strXML & vbCrLf & "<common:classification name=""CPM"">"
' OOS.ActivityType
    strXML = strXML & vbCrLf & "<common:class level=""0"">" & rsProc.fields("ActivityType") &
"</common:class>"
' OOS.Sector
    If Not Trim(rsProc.fields("Sector")) = "" Then
        strXML = strXML & vbCrLf & "<common:class level=""1"">" & rsProc.fields("Sector") &
"</common:class>"
    End If
    strXML = strXML & vbCrLf & "</common:classification>"
    strXML = strXML & vbCrLf & "</classificationInformation>"
' Inventory.Data
    strXML = strXML & vbCrLf & "<common:generalComment xml:lang=""en"">" &
FixXML(rsProc.fields("Data")) & "</common:generalComment>"
' PRESET MAPPING TEXT
```



```

strXML = strXML & vbCrLf & "<referenceToExternalDocumentation type=""source data set""
refObjectId=""ddb15c9a-ca85-480b-adb0-d5b7a2d30395"" Version = ""01.00""

uri=""../sources/CPM_LCA_Database_ddb15c9a-ca85-480b-adb0-d5b7a2d30395.xml"">
strXML = strXML & vbCrLf & "<common:shortDescription xml:lang=""en"">This data set has been
exported from the CPM LCA Database</common:shortDescription>"
strXML = strXML & vbCrLf & "</referenceToExternalDocumentation>"

'Inventory.Publication
If Not Trim(rsProc.fields("Publication")) = "" Then
strXML = strXML & vbCrLf & "<referenceToExternalDocumentation type=""source data set""
refObjectId=""00000000-0000-0000-0000-000000000000"" Version = ""00.00"" uri="""">"
strXML = strXML & vbCrLf & "<common:shortDescription xml:lang=""en"">" &
FixXML(rsProc.fields("Publication")) & "</common:shortDescription>"
strXML = strXML & vbCrLf & "</referenceToExternalDocumentation>"
End If
strXML = strXML & vbCrLf & "</dataSetInformation>"

'Inventory.FunctionalUnit
'Inventory.FunctionalUnitExplanation
strXML = strXML & vbCrLf & "<quantitativeReference type=""Funtional unit"">"
strXML = strXML & vbCrLf & "<functionalUnitOrOther xml:lang=""en"">" &
FixXML(rsProc.fields("FunctionalUnit")) & "</functionalUnitOrOther>"
strXML = strXML & vbCrLf & "<other>Functional unit explanation: " &
FixXML(rsProc.fields("FUEExplanation")) & "</other>"
strXML = strXML & vbCrLf & "</quantitativeReference>"
strXML = strXML & vbCrLf & "<time>"

'Inventory.TimeBoundary
If IsDate(Trim(rsProc.fields("TimeBoundary"))) Then
strXML = strXML & vbCrLf & "<common:referenceYear>" & Trim(rsProc.fields("TimeBoundary")) &
"</common:referenceYear>"
Else
strXML = strXML & vbCrLf & "<common:timeRepresentativenessDescription xml:lang=""en"">" &
FixXML(rsProc.fields("TimeBoundary")) & "</common:timeRepresentativenessDescription>"
End If
strXML = strXML & vbCrLf & "</time>"

'Inventory.GeographicalBoundary
strXML = strXML & vbCrLf & "<geography>"
strXML = strXML & vbCrLf & "<locationOfOperationSupplyOrProduction latitudeAndLongitude=""0;0""
location=""GLO"">"
strXML = strXML & vbCrLf & "<descriptionOfRestrictions xml:lang=""en"">" &
FixXML(rsProc.fields("GeographicalBoundary")) & "</descriptionOfRestrictions>"
strXML = strXML & vbCrLf & "</locationOfOperationSupplyOrProduction>"

'OOS.Site
If Not Trim(rsProc.fields("Site")) = "" Then
strXML = strXML & vbCrLf & "<subLocationOfOperationSupplyOrProduction
latitudeAndLongitude=""0;0"" subLocation=""GLO"">"
strXML = strXML & vbCrLf & "<descriptionOfRestrictions xml:lang=""en"">" &
FixXML(GetJuridicalPersonString(Trim(rsProc.fields("Site")))) & "</descriptionOfRestrictions>"
strXML = strXML & vbCrLf & "</subLocationOfOperationSupplyOrProduction>"
End If

strXML = strXML & vbCrLf & "</geography>"

'OOS.Function
'OOS.Owner
strTechDescription = rsProc.fields("Function")
If Not Trim(rsProc.fields("Owner")) = "" Then
strTechDescription = strTechDescription & vbCrLf & " Owner: " &
GetJuridicalPersonString(Trim(rsProc.fields("Owner")))
End If
strXML = strXML & vbCrLf & "<technology>"
strXML = strXML & vbCrLf & "<technologyDescriptionAndIncludedProcesses xml:lang=""en"">" &
FixXML(strTechDescription) & "</technologyDescriptionAndIncludedProcesses>"

'Inventory.NatureBoundary
'Inventory.OtherBoundaries
strOtherTechContent = ""
If Not Trim(rsProc.fields("NatureBoundary")) = "" Then
strOtherTechContent = "Nature boundary: " & rsProc.fields("NatureBoundary") & " " & vbCrLf
End If
If Not Trim(rsProc.fields("OtherBoundaries")) = "" Then
strOtherTechContent = "Other boundaries: " & rsProc.fields("OtherBoundaries")
End If

```

```

If Not strOtherTechContent = "" Then
    strXML = strXML & vbCrLf & "<other>" & FixXML(strOtherTechContent) & "</other>"
End If
strXML = strXML & vbCrLf & "</technology>"
strXML = strXML & vbCrLf & "</processInformation>"

strXML = strXML & vbCrLf & "<modellingAndValidation>"
strXML = strXML & vbCrLf & "<LCIMethodAndAllocation>"

'PRESET MAPPING TEXT
strXML = strXML & vbCrLf & "<typeOfDataSet>LCI result</typeOfDataSet>"
strXML = strXML & vbCrLf & "<LCIMethodPrinciple>Other</LCIMethodPrinciple>"
strXML = strXML & vbCrLf & "<deviationsFromLCIMethodPrinciple
xml:lang=""en"">none</deviationsFromLCIMethodPrinciple>"
strXML = strXML & vbCrLf & "<LCIMethodApproaches>Not applicable</LCIMethodApproaches>"

'Inventory.Allocation
'Inventory.LateralExpansion
strLCIMethod = "No data"
If Not Trim(rsProc.fields("Allocations")) = "" Then
    strLCIMethod = "Allocations: " & rsProc.fields("Allocations") & " " & vbCrLf
End If
If Not Trim(rsProc.fields("LateralExpansion")) = "" Then
    strLCIMethod = strLCIMethod & "Lateral expansion: " & rsProc.fields("LateralExpansion") & " " &
vbCrLf
End If
strXML = strXML & vbCrLf & "<deviationsFromLCIMethodApproaches xml:lang=""en"">" &
FixXML(strLCIMethod) & "</deviationsFromLCIMethodApproaches>"

'ObjectOfStudy.Category (technical scope)
strXML = strXML & vbCrLf & "<other>" & rsProc.fields("Category") & "</other>"
strXML = strXML & vbCrLf & "</LCIMethodAndAllocation>"

'General QMD.DataType
'General QMD.Method
'General QMD.DateConceived
'General QMD.Represents
'General QMD.Notes
strXML = strXML & vbCrLf & "<dataSourcesTreatmentAndRepresentativeness>"
strDataTreatment = ""
If Not (Trim(rsProc.fields("QMDDataType"))) = "" Or Trim(rsProc.fields("QMDDataType"))) =
"Unspecified") Then
    strDataTreatment = "General data treatment type: " & Trim(rsProc.fields("QMDDataType")) & " " &
vbCrLf
End If
If Not (Trim(rsProc.fields("QMDMethod"))) = "" Or Trim(rsProc.fields("QMDDataType"))) = "Unknown" Or
Trim(rsProc.fields("QMDDataType"))) = "Not given" Or Trim(rsProc.fields("QMDDataType"))) =
"Inventory") Then
    strDataTreatment = strDataTreatment & "General data treatment method: " &
Trim(rsProc.fields("QMDDataType")) & " " & vbCrLf
End If
If Not Trim(rsProc.fields("QMDDateConceived")) = "" Then
    strDataTreatment = strDataTreatment & "Quantitative data conceived date: " &
Trim(rsProc.fields("QMDDateConceived")) & " " & vbCrLf
End If
If Not Trim(rsProc.fields("QMDRepresents")) = "" Then
    strDataTreatment = strDataTreatment & "Represents: " & Trim(rsProc.fields("QMDRepresents")) & "
" & vbCrLf
End If
If Not Trim(rsProc.fields("Notes")) = "" Then
    strDataTreatment = strDataTreatment & "Notes: " & Trim(rsProc.fields("Notes")) & " " & vbCrLf
End If
strDataTreatment = strDataTreatment & "See general comment on data set."
strXML = strXML & vbCrLf & "<dataSelectionAndCombinationPrinciples xml:lang=""en"">" &
FixXML(strDataTreatment) & "</dataSelectionAndCombinationPrinciples>"

'General QMD.LiteratureRef
If Not Trim(rsProc.fields("QMDLiteratureRef")) = "" Then
    strXML = strXML & vbCrLf & "<referenceToDataSource type=""source data set""
refObjectId=""00000000-0000-0000-0000-000000000000"" Version=""00.00"" uri="""">"
    strXML = strXML & vbCrLf & "<common:shortDescription xml:lang=""en"">" &
FixXML(rsProc.fields("QMDLiteratureRef")) & "</common:shortDescription>"
    strXML = strXML & vbCrLf & "</referenceToDataSource>"
End If

'PRESET MAPPING TEXT

```

```

    strXML = strXML & vbCrLf & "<useAdviceForDataSet xml:lang=""en"">See intended
applications</useAdviceForDataSet>"

'Inventory.Notes
If Not Trim(rsProc.fields("Notes")) = "" Then
    strXML = strXML & vbCrLf & "<other>Additional note on dataset: " &
FixXML(rsProc.fields("Notes")) & "</other>"
End If
strXML = strXML & vbCrLf & "</dataSourcesTreatmentAndRepresentativeness>"

'PRESET MAPPING TEXT
strXML = strXML & vbCrLf & "<validation>"
strXML = strXML & vbCrLf & "<review type=""Independent external review"">"
strXML = strXML & vbCrLf & "<common:scope name=""Documentation"">"
strXML = strXML & vbCrLf & "<common:method name=""Documentation""/>"
strXML = strXML & vbCrLf & "<common:method name=""Expert judgement""/>"
strXML = strXML & vbCrLf & "</common:scope>"
strReview = "All LCI data sets in the CPM LCA Database have unergone review according to the CPM
review process in order to ensure fulfilment of CPM's documentation criteria. See
documentation at http://cpmdatabase.cpm.chalmers.se/AboutDatabase."

'Prettyness.Ddocumentation
Set rsPretty = db.Execute("SELECT Ddocumentation from Prettyness WHERE activityid = '" &
rsProc.fields("Aid") & "'")
If Not Trim(rsPretty.fields("Ddocumentation")) = "" Then
    strReview = strReview & " This dataset has been categorised as: " &
rsPretty.fields("Ddocumentation") & " (out of three possible values - Unsatisfying, Acceptable, and
Sufficient)."
```

```

End If
strXML = strXML & vbCrLf & "<common:reviewDetails xml:lang=""en"">" & FixXML(strReview) &
"</common:reviewDetails>"
strXML = strXML & vbCrLf & "<common:referenceToNameOfReviewerAndInstitution type=""contact data
set"" uri=""../contacts/CPM_0b8d9a23-1f44-4f17-999f-f1d1120701ee.xml"">"
refObjectId=""0b8d9a23-1f44-4f17-999f-f1d1120701ee"">"
strXML = strXML & vbCrLf & "<common:shortDescription xml:lang=""en"">CPM</common:shortDescription>"
strXML = strXML & vbCrLf & "</common:referenceToNameOfReviewerAndInstitution>"
Set rsPretty = Nothing

'Inventory.Reviewer
If Not Trim(rsProc.fields("Reviewer")) = "" Then
    GetJuridicalPersonString (Trim(rsProc.fields("Reviewer")))
    strXML = strXML & vbCrLf & "<common:referenceToNameOfReviewerAndInstitution type=""contact data
set"" uri="" "" refObjectId=""00000000-0000-0000-0000-000000000000"">"
    strXML = strXML & vbCrLf & "<common:shortDescription xml:lang=""en"">" &
FixXML(GetJuridicalPersonString(Trim(rsProc.fields("Reviewer")))) & "</common:shortDescription>"
    strXML = strXML & vbCrLf & "</common:referenceToNameOfReviewerAndInstitution>"
End If
strXML = strXML & vbCrLf & "</review>"
strXML = strXML & vbCrLf & "</validation>"

'PRESET MAPPING TEXT
strXML = strXML & vbCrLf & "<complianceDeclarations>"
strXML = strXML & vbCrLf & "<compliance>"
strXML = strXML & vbCrLf & "<common:referenceToComplianceSystem type=""source data set""
refObjectId=""88d4f8d9-60f9-43d1-9ea3-329c10d7d727"">"
uri=""../sources/ILCD_Compliance_88d4f8d9-60f9-43d1-9ea3-329c10d7d727.xml"">"
strXML = strXML & vbCrLf & "<common:shortDescription xml:lang=""en"">ILCD
Compliance</common:shortDescription>"
strXML = strXML & vbCrLf & "</common:referenceToComplianceSystem>"
strXML = strXML & vbCrLf & "<common:approvalOfOverallCompliance>Not
defined</common:approvalOfOverallCompliance>"
strXML = strXML & vbCrLf & "<common:nomenclatureCompliance>Not
defined</common:nomenclatureCompliance>"
strXML = strXML & vbCrLf & "<common:methodologicalCompliance>Not
defined</common:methodologicalCompliance>"
strXML = strXML & vbCrLf & "<common:reviewCompliance>Not defined</common:reviewCompliance>"
strXML = strXML & vbCrLf & "<common:documentationCompliance>Not
defined</common:documentationCompliance>"
strXML = strXML & vbCrLf & "<common:qualityCompliance>Not defined</common:qualityCompliance>"
strXML = strXML & vbCrLf & "</compliance>"
strXML = strXML & vbCrLf & "</complianceDeclarations>"
strXML = strXML & vbCrLf & "</modellingAndValidation>"

strXML = strXML & vbCrLf & "<administrativeInformation>"

```

```

strXML = strXML & vbCrLf & "<common:commissionerAndGoal>"

'Inventory.Commissioner
If Not Trim(rsProc.fields("Commissioner")) = "" Then
    strCommissioner = FixXML(GetJuridicalPersonString(Trim(rsProc.fields("Commissioner"))))
Else
    strCommissioner = "No data"
End If
strXML = strXML & vbCrLf & "<common:referenceToCommissioner type=""contact data set"" uri="" ""
refObjectId = "" "">""
strXML = strXML & vbCrLf & "<common:shortDescription xml:lang=""en"">" & strCommissioner &
"</common:shortDescription>"
strXML = strXML & vbCrLf & "</common:referenceToCommissioner>"

'Inventory.Applicability
'Inventory.IntendedUser
'Inventory.GeneralPurpose
'Inventory.DetailedPurpose
strIntApp = "No data"
If Not Trim(rsProc.fields("Applicability")) = "" Then
    strIntApp = "Applicability: " & rsProc.fields("Applicability") & " " & vbCrLf
End If
If Not Trim(rsProc.fields("IntendedUser")) = "" Then
    strIntApp = strIntApp & "Intended user: " & rsProc.fields("IntendedUser") & " " & vbCrLf
End If
If Not Trim(rsProc.fields("GeneralPurpose")) = "" Then
    strIntApp = strIntApp & "General purpose: " & rsProc.fields("GeneralPurpose") & " " & vbCrLf
End If
If Not Trim(rsProc.fields("DetailedPurpose")) = "" Then
    strIntApp = strIntApp & "Detailed purpose: " & rsProc.fields("DetailedPurpose")
End If
strXML = strXML & vbCrLf & "<intendedApplications xml:lang=""en"">" & FixXML(strIntApp) &
"</intendedApplications>"
strXML = strXML & vbCrLf & "</common:commissionerAndGoal>"

'Inventory.Practitioner
If Not Trim(rsProc.fields("Practitioner")) = "" Then
    strGenerator = FixXML(GetJuridicalPersonString(Trim(rsProc.fields("Practitioner"))))
Else
    strGenerator = "No data"
End If
strXML = strXML & vbCrLf & "<dataGenerator>"
strXML = strXML & vbCrLf & "<common:referenceToPersonOrEntityGeneratingTheDataSet type=""contact
data set"" uri="" "" refObjectId = "" "">""
strXML = strXML & vbCrLf & "<common:shortDescription xml:lang=""en"">" & strGenerator &
"</common:shortDescription>"
strXML = strXML & vbCrLf & "</common:referenceToPersonOrEntityGeneratingTheDataSet>"
strXML = strXML & vbCrLf & "</dataGenerator>"

'PRESET MAPPING TEXT
strXML = strXML & vbCrLf & "<dataEntryBy>"
strXML = strXML & vbCrLf & "<common:timeStamp>" & Date & "T" & FormatDateTime(Now, 3) &
"+01:00</common:timeStamp>"
strXML = strXML & vbCrLf & "<common:referenceToDataSetFormat type=""source data set""
uri=""../sources/ILCD_Format_a97a0155-0234-4b87-b4ce-a45da52f2a40.xml ""
refObjectId=""a97a0155-0234-4b87-b4ce-a45da52f2a40 "">""
strXML = strXML & vbCrLf & "<common:shortDescription xml:lang=""en"">ILCD
format</common:shortDescription>"
strXML = strXML & vbCrLf & "</common:referenceToDataSetFormat>"
strXML = strXML & vbCrLf & "<common:referenceToConvertedOriginalDataSetFrom type=""source data
set"" uri =
""../sources/CPM_LCA_Database_SPINE_Format_d4c9462b-f7aa-467a-85ef-b369960fa732.xml ""
refObjectId=""d4c9462b-f7aa-467a-85ef-b369960fa732 "">""
strXML = strXML & vbCrLf & "<common:shortDescription>CPM LCA Database SPINE
format</common:shortDescription>"
strXML = strXML & vbCrLf & "</common:referenceToConvertedOriginalDataSetFrom>"
strXML = strXML & vbCrLf & "<common:referenceToPersonOrEntityEnteringTheData type=""contact data
set"" uri="" "">""
strXML = strXML & vbCrLf & "<common:shortDescription xml:lang=""en"">See: General comment on data
set</common:shortDescription>"" ""
strXML = strXML & vbCrLf & "</common:referenceToPersonOrEntityEnteringTheData>"
strXML = strXML & vbCrLf & "<common:referenceToDataSetUseApproval type=""contact data set""
uri=""../contacts/CPM_0b8d9a23-1f44-4f17-999f-f1d1120701ee.xml ""
refObjectId=""0b8d9a23-1f44-4f17-999f-f1d1120701ee "">""

```

```

strXML = strXML & vbCrLf & "<common:shortDescription xml:lang=""en"">CPM</common:shortDescription>"
strXML = strXML & vbCrLf & "</common:referenceToDataSetUseApproval>"

'Inventory.DateCompleted
If Not Trim(rsProc.fields("DateCompleted")) = "" Then
    strXML = strXML & vbCrLf & "<other>Original CPM LCA Database dataset completed: " &
FixXML(rsProc.fields("DateCompleted")) & "</other>"
End If
strXML = strXML & vbCrLf & "</dataEntryBy>"
strXML = strXML & vbCrLf & "<publicationAndOwnership>"
strXML = strXML & vbCrLf & "<common:dateOfLastRevision>" & FixXML(rsProc.fields("DateCompleted")) &
" </common:dateOfLastRevision>"

'PRESET MAPPING TEXT
strXML = strXML & vbCrLf & "<common:dataSetVersion>01.00.000</common:dataSetVersion>"
strXML = strXML & vbCrLf & "<common:permanentDataSetURI>" & gstrPermanentURIPath &
"processes/CPM_Process" & strGuid & ".xml</common:permanentDataSetURI>"
strXML = strXML & vbCrLf & "<common:workflowAndPublicationStatus>Data set finalised; entirely
published</common:workflowAndPublicationStatus>"
strXML = strXML & vbCrLf & "<common:referenceToRegistrationAuthority type=""contact data set""
uri=""../contacts/CPM_0b8d9a23-1f44-4f17-999f-f1d1120701ee.xml""

refObjectId=""0b8d9a23-1f44-4f17-999f-f1d1120701ee"">"
strXML = strXML & vbCrLf & "<common:shortDescription xml:lang=""en"">CPM</common:shortDescription>"
strXML = strXML & vbCrLf & "</common:referenceToRegistrationAuthority>"

'Activity.Id
strXML = strXML & vbCrLf & "<common:registrationNumber>" & FixXML(rsProc.fields("Aid")) &
"</common:registrationNumber>"
strXML = strXML & vbCrLf & "<common:referenceToOwnershipOfDataSet type=""contact data set""
uri=""../contacts/CPM_0b8d9a23-1f44-4f17-999f-f1d1120701ee.xml""

refObjectId=""0b8d9a23-1f44-4f17-999f-f1d1120701ee"">"
strXML = strXML & vbCrLf & "<common:shortDescription xml:lang=""en"">CPM</common:shortDescription>"
strXML = strXML & vbCrLf & "</common:referenceToOwnershipOfDataSet>"
strXML = strXML & vbCrLf & "<common:copyright>true</common:copyright>"
strXML = strXML & vbCrLf & "<common:licenseType>Free of charge for some users and
uses</common:licenseType>"
strXML = strXML & vbCrLf & "<common:accessRestrictions xml:lang=""en"">The CPM LCA Database is the
property of the CPM consortium partners. Since year 2008 the CPM LCA Database is accessible free of
charge through the website http://cpmdatabase.cpm.chalmers.se. Any commercial redistribution of data
originating from CPM LCA Database; either in separate parts or complete database; either in the data
formats provided at this website or translated into other formats; is not allowed without a prior
written agreement with the CPM consortium.</common:accessRestrictions>"
strXML = strXML & vbCrLf & "</publicationAndOwnership>"
strXML = strXML & vbCrLf & "</administrativeInformation>"

strXML = strXML & vbCrLf & "<exchanges>"
strSQL = "SELECT Flow.*, Substance.DefaultName FROM Substance INNER JOIN Flow ON Substance.Id =
Flow.SubstanceId WHERE Flow.ActivityId = '" & rsProc.fields("Aid") & "' ORDER BY FlowNumber"
Set rsFlow = db.Execute(strSQL)
Do While Not rsFlow.EOF

'Flow.Substance.DefaultName
'Flow.FlowNumber
'Flow.Unit

'Make flow xml file
strFlowGuid = GetGuid
strFlow = MakeILCDFlowDataSetXMLString(rsProc.fields("Aid"), rsFlow.fields("FlowNumber"),
strFlowGuid)
    MakeXMLfile "flows\CPM_flow_" & strFlowGuid & ".xml", strFlow

    strXML = strXML & vbCrLf & "<exchange dataSetInternalID="" & rsFlow.fields("FlowNumber") &
"">"
    strXML = strXML & vbCrLf & "<referenceToFlowDataSet type=""flow data set""
uri=""../flows/CPM_flow_" & strFlowGuid & ".xml"" refObjectId="" & strFlowGuid & "">"
    strXML = strXML & vbCrLf & "<common:shortDescription xml:lang=""en"">" &
FixXML(rsFlow.fields("DefaultName")) & "</common:shortDescription>"
    strXML = strXML & vbCrLf & "</referenceToFlowDataSet>"

'Flow.ImpactRegion
If Not Trim(rsFlow.fields("ImpactRegion")) = "" Then
    Set rsGeo = db.Execute("SELECT AreaName From Geography WHERE Id = '" &
rsFlow.fields("ImpactRegion") & "'")
    strXML = strXML & vbCrLf & "<location>" & FixXML(rsGeo.fields("AreaName")) & "</location>"
    Set rsGeo = Nothing

```

```

Else
    strXML = strXML & vbCrLf & "<location/>"
End If

'Flow.SubType
    strXML = strXML & vbCrLf & "<exchangeDirection>" & rsFlow.fields("SubType") &
"</exchangeDirection>"

'Flow.Quantity
'Flow.QuantityMin
'Flow.QuantityMax
'Flow.Unit
    strQuantity = ""
    strQuantityMin = ""
    strQuantityMax = ""
    Select Case rsFlow.fields("unit")
    Case "kg", "MJ", "m2", "m3", "Nm3", "m3sub", "m3 fub", "kBq", "m2a", "m2 year", "m3a", "tonne
km", "m", "pce", "pkm"
        'no need to convert unit
        If Not Trim(rsFlow.fields("Quantity")) = "" Then
            strQuantity = Replace(rsFlow.fields("Quantity"), ".", ",")
        End If
        If Not Trim(rsFlow.fields("QuantityMin")) = "" Then
            strQuantityMin = FormatSci(Replace(rsFlow.fields("QuantityMin"), ".", ","))
        End If
        If Not Trim(rsFlow.fields("QuantityMax")) = "" Then
            strQuantityMax = FormatSci(Replace(rsFlow.fields("QuantityMax"), ".", ","))
        End If
    Case "tonne", "g", "mg", "ug", "ng"
        If Not Trim(rsFlow.fields("Quantity")) = "" Then
            strQuantity = FormatSci(UnitConversion(CDbl(Replace(rsFlow.fields("Quantity"), ".",
",")), rsFlow.fields("unit"), "kg"))
        End If
        If Not Trim(rsFlow.fields("QuantityMin")) = "" Then
            strQuantityMin = FormatSci(UnitConversion(CDbl(Replace(rsFlow.fields("QuantityMin"),
".", ",")), rsFlow.fields("unit"), "kg"))
        End If
        If Not Trim(rsFlow.fields("QuantityMax")) = "" Then
            strQuantityMax = FormatSci(UnitConversion(CDbl(Replace(rsFlow.fields("QuantityMax"),
".", ",")), rsFlow.fields("unit"), "kg"))
        End If
    Case "TJ", "GJ", "GWh", "MWh", "kJ", "kWh", "kcal", "kJ", "J", "Wh"
        If Not Trim(rsFlow.fields("Quantity")) = "" Then
            strQuantity = FormatSci(UnitConversion(CDbl(Replace(rsFlow.fields("Quantity"), ".",
",")), rsFlow.fields("unit"), "MJ"))
        End If
        If Not Trim(rsFlow.fields("QuantityMin")) = "" Then
            strQuantityMin = FormatSci(UnitConversion(CDbl(Replace(rsFlow.fields("QuantityMin"),
".", ",")), rsFlow.fields("unit"), "MJ"))
        End If
        If Not Trim(rsFlow.fields("QuantityMax")) = "" Then
            strQuantityMax = FormatSci(UnitConversion(CDbl(Replace(rsFlow.fields("QuantityMax"),
".", ",")), rsFlow.fields("unit"), "MJ"))
        End If
    Case "km2", "ha", "mm2"
        If Not Trim(rsFlow.fields("Quantity")) = "" Then
            strQuantity = FormatSci(UnitConversion(CDbl(Replace(rsFlow.fields("Quantity"), ".",
",")), rsFlow.fields("unit"), "m2"))
        End If
        If Not Trim(rsFlow.fields("QuantityMin")) = "" Then
            strQuantityMin = FormatSci(UnitConversion(CDbl(Replace(rsFlow.fields("QuantityMin"),
".", ",")), rsFlow.fields("unit"), "m2"))
        End If
        If Not Trim(rsFlow.fields("QuantityMax")) = "" Then
            strQuantityMax = FormatSci(UnitConversion(CDbl(Replace(rsFlow.fields("QuantityMax"),
".", ",")), rsFlow.fields("unit"), "m2"))
        End If
    Case "m3", "l", "cm3", "ml"
        If Not Trim(rsFlow.fields("Quantity")) = "" Then
            strQuantity = FormatSci(UnitConversion(CDbl(Replace(rsFlow.fields("Quantity"), ".",
",")), rsFlow.fields("unit"), "m3"))
        End If
        If Not Trim(rsFlow.fields("QuantityMin")) = "" Then
            strQuantityMin = FormatSci(UnitConversion(CDbl(Replace(rsFlow.fields("QuantityMin"),
".", ",")), rsFlow.fields("unit"), "m3"))
        End If
        If Not Trim(rsFlow.fields("QuantityMax")) = "" Then

```

```

        strQuantityMax = FormatSci(UnitConversion(CDb1(Replace(rsFlow.fields("QuantityMax"),
".", ",")), rsFlow.fields("unit"), "m3"))
    End If
    Case "Bq"
        If Not Trim(rsFlow.fields("Quantity")) = "" Then
            strQuantity = FormatSci(UnitConversion(CDb1(Replace(rsFlow.fields("Quantity"), ".",
","), rsFlow.fields("unit"), "kBq")))
        End If
        If Not Trim(rsFlow.fields("QuantityMin")) = "" Then
            strQuantityMin = FormatSci(UnitConversion(CDb1(Replace(rsFlow.fields("QuantityMin"),
".", ",")), rsFlow.fields("unit"), "kBq")))
        End If
        If Not Trim(rsFlow.fields("QuantityMax")) = "" Then
            strQuantityMax = FormatSci(UnitConversion(CDb1(Replace(rsFlow.fields("QuantityMax"),
".", ",")), rsFlow.fields("unit"), "kBq")))
        End If
    Case "kgkm"
        If Not Trim(rsFlow.fields("Quantity")) = "" Then
            strQuantity = FormatSci(UnitConversion(CDb1(Replace(rsFlow.fields("Quantity"), ".",
","), rsFlow.fields("unit"), "tonne km")))
        End If
        If Not Trim(rsFlow.fields("QuantityMin")) = "" Then
            strQuantityMin = FormatSci(UnitConversion(CDb1(Replace(rsFlow.fields("QuantityMin"),
".", ",")), rsFlow.fields("unit"), "tonne km")))
        End If
        If Not Trim(rsFlow.fields("QuantityMax")) = "" Then
            strQuantityMax = FormatSci(UnitConversion(CDb1(Replace(rsFlow.fields("QuantityMax"),
".", ",")), rsFlow.fields("unit"), "tonne km")))
        End If
    End Select

    strXML = strXML & vbCrLf & "<meanAmount>" & Replace(strQuantity, ",", ".") & "</meanAmount>"
    strXML = strXML & vbCrLf & "<resultingAmount>" & Replace(strQuantity, ",", ".") &
"</resultingAmount>"
    strXML = strXML & vbCrLf & "<minimumAmount>" & Replace(strQuantityMin, ",", ".") &
"</minimumAmount>"
    strXML = strXML & vbCrLf & "<maximumAmount>" & Replace(strQuantityMax, ",", ".") &
"</maximumAmount>"

    If Not Trim(rsFlow.fields("MetaId")) = "" Then
        Set rsFlowQMD = db.Execute("SELECT * From QMetaData WHERE Id = '" & rsFlow.fields("MetaId")
& "'")

'Flow QMD.LitteratureRef
    If Not Trim(rsFlowQMD.fields("LitteratureRef")) = "" Then
        strXML = strXML & vbCrLf & "<referencesToDataSource>"
        strXML = strXML & vbCrLf & "<referenceToDataSource type=""flow data set"" uri=""""
refObjectId="""">"
        strXML = strXML & vbCrLf & "<common:shortDescription xml:lang=""en"">" &
FixXML(rsFlowQMD.fields("LitteratureRef")) & "</common:shortDescription>"
        strXML = strXML & vbCrLf & "</referenceToDataSource>"
        strXML = strXML & vbCrLf & "</referencesToDataSource>"
    End If

'Flow QMD.DataType
'Flow QMD.Method
'Flow QMD.DateConcieved
'Flow QMD.Represents
'Flow QMD.Notes
    strFlowComment = ""
    If Not Trim(rsFlowQMD.fields("DataType")) = "" Then
        strFlowComment = "Derivation: " & rsFlowQMD.fields("DataType") & vbCrLf
    End If
    If Not Trim(rsFlowQMD.fields("Method")) = "" Then
        strFlowComment = strFlowComment & " Method: " & rsFlowQMD.fields("Method") & vbCrLf
    End If
    If Not Trim(rsFlowQMD.fields("DateConcieved")) = "" Then
        strFlowComment = strFlowComment & " DateConcieved: " & rsFlowQMD.fields("Method") &
vbCrLf
    End If
    If Not Trim(rsFlowQMD.fields("Represents")) = "" Then
        strFlowComment = strFlowComment & " Represents: " & rsFlowQMD.fields("Represents") &
vbCrLf
    End If
    If Not Trim(rsFlowQMD.fields("Notes")) = "" Then
        strFlowComment = strFlowComment & " Notes: " & rsFlowQMD.fields("Notes") & vbCrLf
    End If

```

```

        If Not strFlowComment = "" Then
            strXML = strXML & vbCrLf & "<generalComment>" & FixXML(strFlowComment) &
"</generalComment>"
        End If
        Set rsFlowQMD = Nothing
    Else
        strXML = strXML & vbCrLf & "<referencesToDataSource/>"
    End If

    strXML = strXML & vbCrLf & "</exchange>"
    rsFlow.MoveNext
Loop
Set rsFlow = Nothing

strXML = strXML & vbCrLf & "</exchanges>"

strXML = strXML & vbCrLf & "</processDataSet>"

MakeILCDProcessDataSetXMLString = strXML

Set rsProc = Nothing

End Function

Function MakeILCDFlowDataSetXMLString(ActivityId, FlowNumber, strGuid)
'response.write "fl <br>" & ActivityId & " nr: " & FlowNumber

    Set rsFlow = db.Execute("SELECT Flow.ActivityId, Flow.FlowNumber, Flow.SubType, Flow.Category,
Flow.Quantity, Flow.QuantityMin, Flow.QuantityMax, Flow.StandardDev, Flow.Unit, Flow.ImpactMedia,
Flow.ImpactRegion, Flow.MetaId, Flow.SubstanceID, (SELECT Substance.DefaultName as DName FROM Substance
WHERE Substance.Id = Flow.SubstanceId) as DefaultName, (SELECT Geography.AreaName as GName FROM
Geography WHERE Geography.ID=Flow.ImpactRegion) as GeoName FROM Flow WHERE Flow.ActivityID = '" &
ActivityId & "' AND Flow.FlowNumber = " & FlowNumber)

    strXML = "<?xml version=""1.0"" encoding=""UTF-8""?>"
    strXML = strXML & vbCrLf & "<?xml-stylesheet type='text/xsl' href='../..//stylesheets/flow2html.xsl'
?>"
    strXML = strXML & vbCrLf & "<flowDataSet xmlns=""http://lca.jrc.it/ILCD/Flow""
xmlns:common=""http://lca.jrc.it/ILCD/Common"" xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance""
version=""1.1"" xsi:schemaLocation=""http://lca.jrc.it/ILCD/Flow ../..//schemas/ILCD_FlowDataSet.xsd"">"

    strXML = strXML & vbCrLf & "<flowInformation>"
    strXML = strXML & vbCrLf & "<dataSetInformation>"
    strXML = strXML & vbCrLf & "<common:UUID>" & strGuid & "</common:UUID>"

'Flow.Substance.DefaultName
    strXML = strXML & vbCrLf & "<name>"
        strXML = strXML & vbCrLf & "<baseName xml:lang=""en"">" & FixXML(rsFlow.fields("DefaultName"))
    & "</baseName>"
        strXML = strXML & vbCrLf & "</name>"

'Flow.SubType
'Flow.Category
'Flow.ImpactMedia
    Select Case rsFlow.fields("Category")
    Case "Product"
        strFlowType = "Product"
    Case "Input Product"
        strFlowType = "Product"
    Case "By-product"
        strFlowType = "Product"
    Case "Co-product"
        strFlowType = "Product"
    Case "Refined resource"
        strFlowType = "Product"
    Case "Cargo"
        strFlowType = "Product"
    Case "Emission"
        strFlowType = "Elementary"
    Case "Natural resource"
        strFlowType = "Elementary"
    Case "Resource"

```



```

        strFlowType = "Elementary"
    Case "Waste"
        strFlowType = "Waste"
    Case "Residue"
        strFlowType = "Waste"
    End Select

    strXML = strXML & vbCrLf & "<classificationInformation>"
    if strFlowType = "Elementary" then
        strXML = strXML & vbCrLf & "<common:elementaryFlowCategorization>"
        strXML = strXML & vbCrLf & "<common:category level=" & "0" & ">" & rsFlow.fields("Category") &
"</common:category>"
        If Not Trim(rsFlow.fields("ImpactMedia")) = "" Then
            If rsFlow.fields("SubType") = "Input" Then
                strXML = strXML & vbCrLf & "<common:category level=" & "1" & ">" & rsFlow.fields("Category")
& " from " & FixXML(rsFlow.fields("ImpactMedia")) & "</common:category>"
            Else
                strXML = strXML & vbCrLf & "<common:category level=" & "1" & ">" & rsFlow.fields("Category")
& " to " & FixXML(rsFlow.fields("ImpactMedia")) & "</common:category>"
            End If
        End If
        strXML = strXML & vbCrLf & "</common:elementaryFlowCategorization>"
    Else
        strXML = strXML & vbCrLf & "<common:classification>"
        strXML = strXML & vbCrLf & "<common:class level=" & "0" & ">" & rsFlow.fields("Category") &
"</common:class>"
        If Not Trim(rsFlow.fields("ImpactMedia")) = "" Then
            If rsFlow.fields("SubType") = "Input" Then
                strXML = strXML & vbCrLf & "<common:class level=" & "1" & ">" & rsFlow.fields("Category") &
from " & FixXML(rsFlow.fields("ImpactMedia")) & "</common:class>"
            Else
                strXML = strXML & vbCrLf & "<common:class level=" & "1" & ">" & rsFlow.fields("Category") &
to " & FixXML(rsFlow.fields("ImpactMedia")) & "</common:class>"
            End If
        End If
        strXML = strXML & vbCrLf & "</common:classification>"
    End If
    strXML = strXML & vbCrLf & "</classificationInformation>"
    strXML = strXML & vbCrLf & "</dataSetInformation>"

    strXML = strXML & vbCrLf & "<quantitativeReference>"
    strXML = strXML & vbCrLf & "<referenceToReferenceFlowProperty>0</referenceToReferenceFlowProperty>"
    strXML = strXML & vbCrLf & "</quantitativeReference>"
    strXML = strXML & vbCrLf & "</flowInformation>"

    strXML = strXML & vbCrLf & "<modellingAndValidation>"
    strXML = strXML & vbCrLf & "<LCIMethod>"

    strXML = strXML & vbCrLf & "<typeOfDataSet>" & strFlowType & " flow</typeOfDataSet>"
    strXML = strXML & vbCrLf & "</LCIMethod>"

'PRESET MAPPING TEXT
    strXML = strXML & vbCrLf & "<complianceDeclarations>"
    strXML = strXML & vbCrLf & "<compliance>"
        strXML = strXML & vbCrLf & "<common:referenceToComplianceSystem
uri=" & "../sources/ILCD_Compliance_88d4f8d9-60f9-43d1-9ea3-329c10d7d727.xml" & " type=" & "source data set" & ">"
        strXML = strXML & vbCrLf & "<common:shortDescription>ILCD Data Network
compliance</common:shortDescription>"
        strXML = strXML & vbCrLf & "</common:referenceToComplianceSystem>"
        strXML = strXML & vbCrLf & "<common:approvalOfOverallCompliance>Not
defined</common:approvalOfOverallCompliance>"
        strXML = strXML & vbCrLf & "</compliance>"
    strXML = strXML & vbCrLf & "</complianceDeclarations>"

    strXML = strXML & vbCrLf & "</modellingAndValidation>"

'PRESET MAPPING TEXT
    strXML = strXML & vbCrLf & "<administrativeInformation>"
    strXML = strXML & vbCrLf & "<dataEntryBy>"
    strXML = strXML & vbCrLf & "<common:timeStamp>" & Date & "T" & FormatDateTime(Now, 3) &
"+01:00</common:timeStamp>"
    strXML = strXML & vbCrLf & "<common:referenceToDataSetFormat type=" & "source data set" & " uri =
" & "../sources/CPM_LCA_Database_SPINE_Format_d4c9462b-f7aa-467a-85ef-b369960fa732.xml" & "
refObjectId=" & "d4c9462b-f7aa-467a-85ef-b369960fa732" & ">"

```

```

    strXML = strXML & vbCrLf & "<common:shortDescription>CPM LCA Database SPINE
format</common:shortDescription>"
    strXML = strXML & vbCrLf & "</common:referenceToDataSetFormat>"
    strXML = strXML & vbCrLf & "<common:referenceToPersonOrEntityEnteringTheData
uri="\"../contacts/CPM_0b8d9a23-1f44-4f17-999f-f1d1120701ee.xml\"" type="contact data set"

refObjectId="\"0b8d9a23-1f44-4f17-999f-f1d1120701ee">"
    strXML = strXML & vbCrLf & "<common:shortDescription xml:lang=\"en\">CPM Swedish Life Cycle Center
- LCA Database</common:shortDescription>"
    strXML = strXML & vbCrLf & "</common:referenceToPersonOrEntityEnteringTheData>"
    strXML = strXML & vbCrLf & "</dataEntryBy>"
    strXML = strXML & vbCrLf & "<publicationAndOwnership>"
    strXML = strXML & vbCrLf & "<common:datasetVersion>01.00.000</common:datasetVersion>"
    strXML = strXML & vbCrLf & "<common:permanentDataSetURI>" & gstrPermanentURIPath &
"flows/CPM_flow_" & strGuid & ".xml</common:permanentDataSetURI>"
    strXML = strXML & vbCrLf & "<common:referenceToOwnershipOfDataSet uri="\"../contacts/CPM_0b8d9a23-
1f44-4f17-999f-f1d1120701ee.xml\"" type="contact data set"

refObjectId="\"0b8d9a23-1f44-4f17-999f-f1d1120701ee">"
    strXML = strXML & vbCrLf & "<common:shortDescription xml:lang=\"en\">CPM Swedish Life Cycle Center
- LCA Database</common:shortDescription>"
    strXML = strXML & vbCrLf & "</common:referenceToOwnershipOfDataSet>"
    strXML = strXML & vbCrLf & "</publicationAndOwnership>"
    strXML = strXML & vbCrLf & "</administrativeInformation>"

    strXML = strXML & vbCrLf & "<flowProperties>"
    strXML = strXML & vbCrLf & "<flowProperty dataSetInternalID=\"0\">"

'Flow.Unit
    strUri = ""
    strRefObjId = ""
    strEntity = "undefined unit"

Select Case rsFlow.fields("unit")
Case "kg", "tonne", "g", "mg", "ug", "ng"
    strUri = "../flowproperties/Mass_08f485ac-ed8d-4764-b8f6-ed3c220abdb8.xml"
    strRefObjId = "08f485ac-ed8d-4764-b8f6-ed3c220abdb8"
    strEntity = "Mass"
Case "MJ", "TJ", "GJ", "GWh", "MWh", "kJ", "kWh", "kcal", "kJ", "J", "Wh"
    strUri = "../flowproperties/Net_calorific_value_cb0f1996-b781-48f0-83d0-d4c0eb002fd6.xml"
    strRefObjId = "cb0f1996-b781-48f0-83d0-d4c0eb002fd6"
    strEntity = "Net_calorific_value"
Case "m2", "km2", "ha", "mm2"
    strUri = "../flowproperties/Area_4dacb230-6e13-4250-8371-dc59641d89c8.xml"
    strRefObjId = "4dacb230-6e13-4250-8371-dc59641d89c8"
    strEntity = "Area"
Case "m3", "Nm3", "m3sub", "m3 fub", "l", "cm3", "ml"
    strUri = "../flowproperties/Volume_dce009b7-a56a-4274-be6b-d17ded68a5bf.XML"
    strRefObjId = "dce009b7-a56a-4274-be6b-d17ded68a5bf"
    strEntity = "Volume"
Case "kBq", "Bq"
    strUri = "../flowproperties/Radioactivity_3135446d-27f9-4d92-8d53-e2bd65650a26.xml"
    strRefObjId = "3135446d-27f9-4d92-8d53-e2bd65650a26"
    strEntity = "Radioactivity"
Case "m2a", "m2 year"
    strUri = "../flowproperties/Area_time_df96e341-9c3a-4f28-aa21-9f05666667be.xml"
    strRefObjId = "df96e341-9c3a-4f28-aa21-9f05666667be"
    strEntity = "Area_time"
Case "m3a"
    strUri = "../flowproperties/Volume_time_a24cb362-0c2f-4a49-9139-9046eede88a7.xml"
    strRefObjId = "a24cb362-0c2f-4a49-9139-9046eede88a7"
    strEntity = "Volume_time"
Case "tonne km", "kgkm"
    strUri = "../flowproperties/Mass_length_838aaa21-0117-11db-92e3-0800200c9a66_02.01.000.xml"
    strRefObjId = "838aaa21-0117-11db-92e3-0800200c9a66"
    strEntity = "Mass_length"
Case "m"
    strUri = "../flowproperties/Length_5d298ca0-0523-4407-bbb8-99476e49d91e.xml"
    strRefObjId = "5d298ca0-0523-4407-bbb8-99476e49d91e.xml"
    strEntity = "Length"
Case "pce"
    strUri = "../flowproperties/Number_91797c96-7672-4121-ab91-553315d3b2c7.xml"
    strRefObjId = "91797c96-7672-4121-ab91-553315d3b2c7.xml"
    strEntity = "Number"
Case "pkm"
    strUri = "../flowproperties/Person_distance_b0dc5ff7-41a6-4519-ae20-7a833c9b39cc.xml"
    strRefObjId = "b0dc5ff7-41a6-4519-ae20-7a833c9b39cc.xml"

```

```

        strEntity = "Person_distance"
    End Select
    strXML = strXML & vbCrLf & "<referenceToFlowPropertyDataSet uri="" & strUri & "" type=""flow
property data set"" refObjectId="" & strRefObjId & "">"
    strXML = strXML & vbCrLf & "<common:shortDescription xml:lang=""en"">" & strEntity &
"</common:shortDescription>"
    strXML = strXML & vbCrLf & "</referenceToFlowPropertyDataSet>"

'PRESET MAPPING TEXT
    strXML = strXML & vbCrLf & "<meanValue>1.0</meanValue>"
    strXML = strXML & vbCrLf & "<minimumValue>1.0</minimumValue>"
    strXML = strXML & vbCrLf & "<maximumValue>1.0</maximumValue>"
    strXML = strXML & vbCrLf & "<uncertaintyDistributionType>undefined</uncertaintyDistributionType>"
    strXML = strXML & vbCrLf & "<relativeStandardDeviation95In>0</relativeStandardDeviation95In>"
    strXML = strXML & vbCrLf & "<dataDerivationTypeStatus>Measured</dataDerivationTypeStatus>"
    strXML = strXML & vbCrLf & "</flowProperty>"
    strXML = strXML & vbCrLf & "</flowProperties>"

    strXML = strXML & vbCrLf & "</flowDataSet>"

rsFlow.Close
Set rsFlow = Nothing

MakeILCFlowDataSetXMLString = strXML

End Function

Function GetJuridicalPersonString(JId)
    Set rsJP = db.Execute("SELECT * FROM JuridicalPerson WHERE Id = ' " & JId & "'")
    strJP = Trim(rsJP.fields("Name"))
    If Not Trim(rsJP.fields("MailAddress")) = "" Then
        strJP = strJP & " " & rsJP.fields("MailAddress")
    End If
    If Not Trim(rsJP.fields("Telephone")) = "" Then
        strJP = strJP & ", tel: " & rsJP.fields("Telephone")
    End If
    If Not Trim(rsJP.fields("Fax")) = "" Then
        strJP = strJP & ", fax: " & rsJP.fields("Fax")
    End If
    GetJuridicalPersonString = strJP
    Set rsFlow = Nothing
End Function

Function GetGuid()
    Set TypeLib = CreateObject("Scriptlet.TypeLib")
    strGuid = Mid(CStr(TypeLib.Guid), 2, 36)
    strGuid = LCase(strGuid)
    GetGuid = strGuid
    Set TypeLib = Nothing
End Function

Function UnitConversion(value, unit, baseUnit)
    Set rsUC = db.Execute("SELECT * FROM UnitConversion WHERE UnitName ='" & unit & "' AND
BaseUnitName='" & baseUnit & "'")
    If Not rsUC.EOF Then
        UnitConversion = value / CDb(Replace(rsUC.fields("Factor"), ".", ",")) +
CDb(Replace(rsUC.fields("Offset"), ".", ","))
    Else
        UnitConversion = "no conversion found"
    End If
    Set rsUC = Nothing
End Function

Function FormatSci(floVal)
    floAbsVal = Abs(floVal)
    If floAbsVal <> 0 And (floAbsVal > 1000 Or floAbsVal < 0.1) Then
        intSgnVal = Sgn(floVal)
        intScale = Int(Log(floAbsVal) / Log(10))
        floScaled = floAbsVal / (10 ^ intScale)
        FormatSci = CStr(intSgnVal * floScaled) & "E" & CStr(intScale)
    Else
        FormatSci = CStr(floVal)
    End If
End Function

Function FixXML(s)
    s = Replace(s, "&", "&amp;")

```

```
s = Replace(s, "<", "&lt;")
s = Replace(s, ">", "&gt;")
FixXML = s
End Function

Function MakeXMLfile(FileName, strText)
Set fs = CreateObject("Scripting.FileSystemObject")
'syntax: object.CreateTextFile filename [, overwrite[, unicode]]
Set f = fs.CreateTextFile(gstrPath & FileName, True, True)
f.write (strText)
f.Close
Set f = Nothing
Set fs = Nothing
End Function
```

Appendix B Excerpt from Project description

Konvertering av livscykeldata till internationell standard (Life cycle data conversion to international standard)

Sammanfattning

Tillgång på produktrelaterad miljödata är en nödvändighet för att beräkna produkter och tjänsters miljöprestanda. Sverige var tidigt ute med att redan under 90-talet bygga upp en databas för produktrelaterad miljödata (CPM LCA Database). Denna databas innehåller idag ca 700 dataset vilka är fritt tillgängliga på nätet. Under senare år har ett nytt internationellt format för produktrelaterad miljödata tagits fram (ILCD-format). Projektets mål är att utveckla en konverteringsfunktion som översätter datamängderna inom CPM LCA Database till ILCDformat. Syftet är att på så sätt öka tillgängligheten av produktrelaterad miljödata som kan användas för att bedöma och förbättra produkters miljöprestanda. Detta är till stor nytta för framförallt små och medelstora företag som inte har möjlighet att investera i kommersiella databaser. Databasen i sig utgör en viktig infrastruktur för att tillgängliggöra och kommunicera data från svensk forskning och industri internationellt.

Syfte och mål

Projektet syftar till att öka tillgängligheten av produktrelaterad miljödata som kan användas för att bedöma och förbättra produkters miljöprestanda. Detta görs genom att utveckla en konverteringsfunktion som översätter datamängderna inom CPM LCA Database till det internationella standardformatet ILCD. Produktrelaterad miljödata är en nödvändighet för att beräkna produkter och tjänsters miljöprestanda i exempelvis carbon footprints eller fulla livscykelanalyser (LCA). Tillgången på data är ofta ett av de största hindren för att göra sådana analyser och datainsamlingen är den mest tidskrävande delen av en studie. Det har genom åren utvecklats ett antal olika databaser i olika länder med olika inriktning, och med datamängderna lagrade på olika format. För att harmonisera utvecklingen har European Commission Joint Research Center initierat ILCD; en internationell plattform för livscykeldata (The International Life Cycle Data System). ILCD har arbetat fram ett dataformat som etablerat sig som de-facto världsstandard för miljödata för produkter och processer. Sverige var tidigt ute med att bygga upp en databas för produktrelaterad miljödata (livscykelinventeringsdata) inom kompetenscentret CPM (centrum för produktrelaterad miljöanalys, www.lifecyclecenter.se), kallad "CPM LCA Database". Sedan 2008 finns denna databas fritt tillgänglig på nätet och innehåller idag ca 700 dataset. Den uppdateras kontinuerligt när nya forskningsdata finns tillgänglig (<http://cpmdatabase.cpm.chalmers.se/>). I takt med att allt fler organisationer intresserar sig för att göra olika former av miljöpåverkansberäkningar på produkter ökar också efterfrågan på data. Idag dominerar utbudet av data av ett par kommersiella aktörer. Ett flertal mindre öppna databaser finns (däribland CPM LCA Database), och flera initiativ tas även för att bygga upp databaser på nationell- eller sektorsövergripande nivå. De fria datamängderna används framförallt av SME, studenter och andra som inte har möjlighet att investera i kommersiella databaser. Databaser i sig utgör en viktig infrastruktur för att arkivera och tillgängliggöra data från svensk forskning och industri och kan även kommunicera data internationellt.

Resultat

Den föreslagna anpassningen till ILCD väntas leda till:

- Ökad tillgänglighet, spridning och användning av existerande och kommande datamängder i CPM LCA Database då format kan fås som är kompatibelt med ledande mjukvarutillverkare.

- Ökat intresse att lämna data till CPM LCA Database genom datamängdernas ökade spridning
- Ökad kompetens inom Sverige kring olika dataformats utformning och dokumentation (särskilt bland involverade utvecklare och testare)
- Sverige som ett gott exempel och föredöme internationellt på hur existerande öppna databaser kan anpassas till ILCD Projektet kommer att leverera en översättningsfunktion från SPINE-format till ILCD-format samt en mappningsrapport som beskriver hur formaten relaterar till varandra:

Översättningsfunktion från SPINE format till ILCD format: Funktion för automatisk mappning av livscykelinventeringsdata (LCI-data) från formatet SPINE som används i CPM LCA Database till ILCD formatet. Funktionen ska kunna användas av besökare på CPM LCA Database hemsida och genererar nedladdningsbara XML-filer formaterade enligt ILCD:s standardiserade format. Flera applikationer inklusive de ledande LCA mjukvarorna GaBi och SimaPro samt OpenSource mjukvaran OpenLCA är kompatibla med ILCD formatet och kan importera dessa filer.

Mappningsrapport: Rapport som beskriver hur begrepp, begreppshierarkier, relationer, nomenklaturer samt datatyper i formaten SPINE och ILCD mappar till varandra.

Nytta och skalbarhet

Funktionen att kunna använda data från CPM LCA Database direkt i andra applikationer är efterfrågad. CPM har de senaste åren regelbundet fått frågor från såväl livscykelanalytiker som mjukvaruutvecklare (i Sverige, men framförallt finns en internationell efterfrågan på information) om de data som finns kan importeras till olika mjukvaror, till exempel genom att fås på ILCD-format. En översättning till ILCD-format skulle vara av nytta för bland annat:

- Näringsliv/myndigheter/forskare som utför livscykelanalyser: då data tillgängliggörs i ett format som kan användas direkt i ledande mjukvaror.
- Studenter i högre utbildning samt små och medelstora företag: Dessa är särskilt beroende av att det finns publikt tillgängliga data utan kostnad för användaren.
- Utvecklare av programvaror för implementering av miljöaspekter i operativt arbete: Ett exempel är Chalmers produkt- och produktionsutveckling som med en konvertering till ILCD-format skulle kunna använda CPM data i sin simulator för produktionsutveckling. Den föreslagna översättningsfunktionen är skalbar i betydelsen att den är generell för alla typer av produkter och produktionsprocesser. Funktionen kan användas på samtliga existerande och kommande datamängder på SPINE-format.