|  |  |
| --- | --- |
| Approved by | Creator |
| <Company> | Suman S B |
| <Unit> | <Tel.> |
| <Name> |  |

Enterprise Information Archiving Service (EIAS) – 4.0

|  |  |
| --- | --- |
|  |  |
| Description |  |
|  |  |
| **Project owner** | **Project manager** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Checked by** |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Approved by** |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Purpose** |  |
| This guide provides information about creation of InfoArchive4.1 application. This guide is meant for the developers creating InfoArchive4.1 application |  |

Table of contents

1 General 3

2 Preparations 3

3 Files needs to be modified 3

3.1 Build.properties 3

3.2 Build.xml 4

3.3 Files needs to be deleted 5

3.4 070-space-root-folder.xml 5

3.5 150-retention-policy.xml 5

3.6 190-aic.xml 5

3.7 220-query.xml 6

3.8 Pdi-schema.xsd 7

3.9 Ingest.xml 7

3.10 Pdi.xml 8

3.11 Result-configuration-helper.xml 13

# General

TeliaSonera Enterprise Information Archive Service (EIAS) is a solution that uses standardized and unified policies, systems and procedures to archive content that meets the corporate standards. Projects and teams within TeliaSonera can use this service to decommission legacy applications and archive data which is no longer in active use. Once archived, the data can be accessed for reading purposes only.

EIAS is based on EMC's **Info Archive** product which complies with the Open Archival Information System standard (OAIS).

EIAS 3.2 platform has already been setup. Due to architectural changes in InfoArchive latest version, EIAS 4.1 will be setup and data should be migrated to new platform.

This document describes the creation steps of EMC info archive4.1 application.

# 2 Preparations

Make copy of PhoneCalls application folder in /opt/eias/infoarchive/tools/applications

Rename the folder to correspond your application name e.g. Ind

# Files needs to be modified

Below listed files needs to be updated/modified as per the requirement.

* Build.properties
* Build.xml
* 070-space-root-folder.xml
* 150-retention-policy.xml
* 190-aic.xml
* 220-query.xml
* Pdi-schema.xsd
* Ingest.xml
* Pdi.xml
* Result-configuration-helper.xml

## build.properties file

Below listed properties will be modified. The following properties are modified/updated based on “Ind” application creation.

Change “xdbDatabase” name

Change “application” name

Change “holding” name

Change “pdiSchema” name

Change/add “spaceRootFolder” attribute

Change/add “ageingPeriodDays”(retention) attribute

Change “federationPassword” and “xdbPassword” password

Change “description” name

Change “category” (get this value from business attribute present in eas.pdi.xml which is zipped in sip package) as shown below

Ind Example

|  |
| --- |
| xdbDatabase= Ind-xdb  application= Ind  holding= Ind  pdiSchema= urn:eias:gdw:Ind:1.0  category= Bill Invoice Management  description= The application has Ind data from GDW  federationPassword= xdbadmin123  xdbPassword=secret  spaceRootFolder=//opt/eias/infoarchive/data/space  ageingPeriodDays=2555 - (retention period) |

## Build.xml file

Change “project name"

Delete rows 010, 130 and 180 both from include list and target list

Ind Example

|  |
| --- |
| Changing project name  <project name="Ind application" default="all">  Remove below 3 rows from include list  <include as="010" file= ---------------------------------/>  <include as="130" file= ------------------------------------/>  <include as="180" file= ------------------------------------/>  -----------------------------------------------------------------------  Remove below 3 lines from target list  010.crypto-object,  130.pdi-crypto,  180.holding-crypto, |

## Files need to be deleted

From /opt/eias/infoarchive/tools/applications/Ind/resources delete below files

1. 010-crypro-object
2. 130-pdi-crypto
3. 180-holding-crypto

And also delete “**pdi-crypto.xml**” file located in

opt/eias/infoarchive/tools/applications/Ind/resources/content location

## . 070-space-root-folder

Change the path from default to spaceRootFolder(it is a variable defined in build.properties file).

|  |
| --- |
| <ia-select type="file-system-root" where="?[path=='${spaceRootFolder}']" property="self"  resultProperty="fileSystemRootSelf"/> |

## . 150-retention-policy

Change the retention period as shown below

|  |
| --- |
| <agingPeriod>  <units>DAYS</units>  <value>${ageingPeriodDays}</value>  </agingPeriod>  Where - ”**ageingPeriodDays**”is a variable declared in ”build.properties” file |

## . 190-aic.xml

Add attributes which you use in search form

Change name,label and type inside criteria tag

PhoneCalls example

|  |
| --- |
| <criterias>  <name>CallEndDate</name>  <label>Call End Date</label>  <type>DATETIME</type>  </criterias> |

Ind Example

|  |
| --- |
| <criterias>  <name>ObjectName</name>  <label>ObjectName</label>  <type>STRING</type>  </criterias> |

For data types (Type) Refer .xsd file which is used to create sip packages using documentum connector

## . 220-query.xml

Change path inside entityPath tag as shown below

PhoneCalls Example

|  |
| --- |
| <entityPath>/n:Calls/n:Call</entityPath> |

Ind Example

|  |
| --- |
| <entityPath>/n:objects/n:object</entityPath> |

Change path, type and name inside operand tag

PhoneCalls Example

|  |
| --- |
| <operands>  <name>CallStartDate</name>  <path>n:CallStartDate</path>  <type>DATETIME</type>  </operands> |

Ind Example

|  |
| --- |
| <operands>  <name>ObjectName</name>  <path>n:uiattrs/n:ObjectName</path>  <type>STRING</type>  <indexed>true</indexed>  </operands> |

|  |
| --- |
| Delete <template> and <prolog> element from the file |
|  |
|  |

## . pdi-schema.xsd

Change the content (copy the contents from .xsd file which is used while creating sip packages using Documentum connector)

## . ingest.xml

To compress files add below highlighted lines inside processor(<processor>) element named “CI compression”

Ind Example

|  |
| --- |
| <processor>  <name>CI compression</name>  <class>com.emc.ia.ingestion.processor.content.CiCompressProcessor</class>  <id>ci.compress</id>  <data>  <select.query>  **<![CDATA[**  **declare namespace n = "urn:eias:gdw:Ind:1.0";**  **for $ci in distinct-values(/n:objects/n:object/n:file/n:file)**  **return $ci**  **]]>**  </select.query>  </data>  </processor>  Add/change - path **(/n:objects/n:-----/n:---/n:-- )** to refer file/content which needs to be compressed as highlighted above. |

Comment or delete below code section.

|  |
| --- |
| <processor>  <name>PDI Transformer</name>  <class>com.emc.ia.ingestion.processor.transformer.PdiTransformerProcessor</class>  <id>pdi.transformer</id>  <data>  <resultSchema></resultSchema>  </data>  </processor> |

## pdi.xmlfile

Index all elements which will be part of search (including files) as shown below

PhoneCalls Example

|  |
| --- |
| <path.value.index>  <name>CallStartDate</name>  <path>  /{urn:eas-samples:en:xsd:phonecalls.1.0}Calls/{urn:eas-samples:en:xsd:phonecalls.1.0}Call[{urn:eassamples:en:xsd:phonecalls.1.0}CallStartDate&lt;DATE\_TIME>]  </path>  <compressed>false</compressed>  <unique.keys>false</unique.keys>  <concurrent>false</concurrent>  <build.without.logging>false</build.without.logging>  </path.value.index> |

Ind example

|  |
| --- |
| <path.value.index>  <name>ObjectName</name>  <path>  /{urn:eias:gdw:Ind:1.0}objects/{urn:eias:gdw:Ind:1.0}object/{urn:eias:gdw:Ind:1.0}uiattrs[{urn:eias:gdw:Ind:1.0}ObjectName&lt;STRING>]  </path>  <compressed>false</compressed>  <unique.keys>false</unique.keys>  <concurrent>false</concurrent>  <build.without.logging>false</build.without.logging>  </path.value.index> |

Below code snippet shows how to index a file

Ind example

|  |
| --- |
| <path.value.index>  <name>file</name>  <path>  /{urn:eias:gdw:Ind:1.0}objects/{urn:eas-samples:en:xsd:phonecalls.1.0}object/{urn:eas-samples:en:xsd:phonecalls.1.0}file/{urn:eas-samples:en:xsd:phonecalls.1.0}file&lt;STRING>  </path>  <compressed>false</compressed>  <unique.keys>false</unique.keys>  <concurrent>false</concurrent>  <build.without.logging>false</build.without.logging>  </path.value.index> |

**<id>pdi.aiu.id</id>** : Change schema name. Change for loop to point to AIU. Change return to return any element from AIU as shown below (see PhoneCalls example)

PhoneCalls Example

|  |
| --- |
| <id>pdi.aiu.id</id>  <select.query>  <![CDATA[  declare namespace n = "urn:eas-samples:en:xsd:phonecalls.1.0";  for $call in /n:objects/n:object  return ($call, $call/n:uiattrs/n:ObjectName/text())  ]]>  </select.query> |

Ind Example

|  |
| --- |
| <data>  <id>pdi.aiu.id</id>  <select.query>  <![CDATA[  declare namespace n = "urn:eias:gdw:Ind:1.0";  for $call in /n:objects/n:object  return ($call, $call/n:uiattrs/n:ObjectName/text())  ]]>  </select.query>  </data> |

**<id>pdi.aiu.cnt</id>** Change schema name. Change count to count AIUs. As shown below

PhoneCalls Example

|  |
| --- |
| <data>  <id>pdi.aiu.cnt</id>  <select.query>  <![CDATA[  declare namespace n = "urn:eas-samples:en:xsd:phonecalls.1.0";  count(/n:objects/n:object)  ]]>  </select.query>  </data> |

Ind Example

|  |
| --- |
| <data>  <id>pdi.aiu.cnt</id>  <select.query>  <![CDATA[  declare namespace n = "urn:eias:gdw:Ind:1.0";  count(/n:objects/n:object)  ]]>  </select.query>  </data> |

**<id>pdi.pkeys</id>** for the dateTime01 and dateTime02: Change schema name. Change min and max functions to point to datetime field. Delete values01 pkey element. (Refer 3.2 files while selecting keys)

PhoneCalls Example

|  |
| --- |
| <data>  <id>pdi.pkeys</id>  <pkey attr="dateTime01">  declare namespace n = "urn:eas-samples:en:xsd:phonecalls.1.0";  min(/n:Calls/n:Call/xs:dateTime(n:CallStartDate))  </pkey>  <pkey attr="dateTime02">  declare namespace n = "urn:eas-samples:en:xsd:phonecalls.1.0";  max(/n:Calls/n:Call/xs:dateTime(n:CallStartDate))  </pkey>  </data> |

Ind Example

|  |
| --- |
| <data>  <id>pdi.pkeys</id>  <pkey attr="dateTime01">  declare namespace n = "urn:eias:gdw:Ind:1.0";  min(/n:objects/n:object/n:uiattrs/xs:date(n:MessageDate))  </pkey>  <pkey attr="dateTime02">  declare namespace n = "urn:eias:gdw:Ind:1.0";  max(/n:objects/n:object/n:uiattrs/xs:date(n:MessageDate))  </pkey>  </data> |

**<id>ri.init</id>:** If there is attachments change this otherwise delete it. Change schema name. Change the distinct values to point to the element having the file name. Change the content type.

PhoneCalls Example

|  |
| --- |
| <data>  <id>ri.init</id>  <select.query>  <![CDATA[  declare namespace n = "urn:eas-samples:en:xsd:phonecalls.1.0";  for $ci in distinct-values(/n:Calls/n:Call/n:Attachments/n:Attachment/n:FileName)  order by $ci  return <content type="audio/x-mpeg">{ $ci }</content>  ]]>  </select.query>  </data> |

Ind Example

|  |
| --- |
| <data>  <id>ri.init</id>  <select.query>  <![CDATA[  declare namespace n = "urn:eias:gdw:Ind:1.0";  for $ci in distinct-values(/n:objects/n:object/n:file/n:file)  order by $ci  return <content type="application/octet-stream">{ $ci }</content>  ]]>  </select.query>  </data> |

**<id>xdb.pdi.ci.id</id>** Change the schema name (n=..). Change the last for loop to point to the element having the file name. Change highlighted part

PhoneCalls example

|  |  |
| --- | --- |
| <data>  <id>xdb.pdi.ci.id</id>  <select.query>  <![CDATA[  declare variable $id as xs:string external;  declare variable $seqno\_start as xs:long external;  declare namespace n = "urn:eas-samples:en:xsd:phonecalls.1.0";  declare namespace ri = "urn:x-emc:ia:schema:ri";  let $pdi\_uri := root(.)  let $ri\_uri := replace(document-uri($pdi\_uri), '\.pdi$', '.ri')  for $ri in doc($ri\_uri)/ri:ris/ri:ri[@key]  for $n in /n:Calls/n:Call/n:Attachments/n:Attachment/n:FileName[. = $ri/@key]  return ($n,concat($id,":ci:", sum(($ri/@seqno,$seqno\_start))))  ]]>  </select.query>  </data> |  |

Ind example

|  |
| --- |
| <data>  <id>xdb.pdi.ci.id</id>  <select.query>  <![CDATA[  declare variable $id as xs:string external;  declare variable $seqno\_start as xs:long external;  declare namespace n = "urn:eias:gdw:Ind:1.0";  declare namespace ri = "urn:x-emc:ia:schema:ri";  let $pdi\_uri := root(.)  let $ri\_uri := replace(document-uri($pdi\_uri), '\.pdi$', '.ri')  for $ri in doc($ri\_uri)/ri:ris/ri:ri[@key]  for $n in /n:objects/n:object/n:file/n:file[. = $ri/@key]  return ($n,concat($id,":ci:", sum(($ri/@seqno,$seqno\_start))))  ]]>  </select.query>  </data> |

## . Result-configuration-helper.xml

**🡪** Change schema name

🡪Add repeating values in group

🡪 Add all attributes which will appear in search results and label them inside element tag as shown below

Adding attributes inside element tag

PhoneCalls Example

|  |
| --- |
| Change schema name - xmlns:n="urn:eas-samples:en:xsd:phonecalls.1.0  <element>  <name>CallStartDate</name>  <label>Call start</label>  <type>DATETIME</type>  <path>n:CallStartDate</path>  </element> |

Ind Example

|  |
| --- |
| <element>  <name>ObjectName</name>  <label>Tiedoston nimi</label>  <type>STRING</type>  <path>n:uiattrs/n:ObjectName</path>  </element> |

Adding repeating values in group (refer Ind application)

|  |
| --- |
| <group>  <name>Laskun tiedot</name>  <label>Laskun tiedot:</label>  <path>n:uiattrs/n:SellerInvoiceDetails/n:SellerInvoiceTypeDetails</path>  <element>  <name>SellerInvoiceTypeText</name>  <label>Laskun aihe:</label>  <type>STRING</type>  <path>n:SellerInvoiceTypeText</path>  </element>  <element>  <name>SellerInvoiceIdentifierText</name>  <label>Pyydettävä tunnistetieto:</label>  <type>STRING</type>  <path>n:SellerInvoiceIdentifierText</path>  </element>  </group> |