SCIDIP-ES
SCIENCE DATA INFRASTRUCTURE FOR PRESERVATION - EARTH SCIENCE

D21.3 Process Virtualization Toolkit Installation and User Manual

Work package	WP21	Services/Toolkits Development and Adaptation
Task	T21.6 Process Virtualization and Emulation Toolkit Development	
Author (s)	Christian Ceprani	ACS
Author (s)	Fabrizio Tringali	ACS
Author (s)	Andrea Tringali	ACS
Authorized by		
Reviewer		
Doc Id		
Dissemination Level	~~CONFIDENTIAL/~~PUBLIC	
Issue	1.0	
Date	28/02/2014	

Abstract:

This document represents the Installation, Deployment and User Manual for the Process Virtualization Toolkit developed in the frame of SCIDIP-ES project. This document contains relevant information on how to install, configure and use the Toolkit.

Document Log

Date	Author	Changes	Version	Status
18/10/2013	Andrea Colapicchioni	First draft for internal review	0.1	Draft
21/02/2014	Fabrizio Tringali	Second draft for internal review	0.2	Draft
24/02/2014	Fabrizio Tringali	Version for Release	0.3	Draft
28/02/2014	Andrea Colapicchioni	Minor editorial changes	1.0	Published

TABLE OF CONTENTS

1 Introduction

1.1 Purpose and Scope

This document provides an overview of the Process Virtualization Toolkit focusing in particular to its installation and usage.

1.2 Who should read this document

Users who may want to understand, install and use the Process Virtualization Toolkit.

1.3 System Context

The Process Virtualization Toolkit is developed as part of the SCIDIP-ES preservation e-infrastructure. Its role is to preserve the capacity to visualise and process specific Earth Science Digital Objects in the long term. The Process Virtualization Toolkit achieves this by linking the preserved processing environment to the Representation Information (RepInfo[1]) Network by transforming the processing environment into a digital package suitable, and generating the relating Representation Information objects, for ingest into the SCIDIP-ES RepInfo Registry Service. The Toolkit uses the SCIDIP-ES Framework library to interact with the Registry Service and, optionally, the SCIDIP-ES Storage Service interface to store/retrieve the preserved environment digital package to/from a repository.

The Process Virtualization Toolkit adopts the Open Virtualization Format (OVF) to package the virtual processing environment. OVF is a common standard used by independent software vendors to package and deploy virtual components across multiple Virtualization platforms. The OVF standard (http://www.dmtf.org/standards/ovf) is not tied to any particular hypervisor or processor architecture. An OVF Package represents a single unit of packaging which may contain one or more virtual systems that can be deployed to a virtual machine.

Examples of Hypervisors supporting the OVF format are:

Product	OVF Support Since	Release Date
VirtualBox	2.2.0	April 2009
Red Hat Enterprise Virtualization	2.2	March 2010

[1] RepInfo is defined in the Open Archival Information System ISO standard for long term data preservation as the additional information that maps a data object into more meaningful concepts to facilitate its access and (re-)use.

VMware	ESX 3.5, Workstation 6.5, Player 3.1	Dec 2007
IBM POWER server AIX, Linux z/VM, IBM Systems Director (via VMControl Enterprise Edition plug-in, a cross-platform VM manager)		
IBM SmartCloud	IBM SmartCloud Enterprise 1.4	Oct 2011
OpenNode Cloud Platform	1.1	Nov 2010
Oracle VM	3.0	Aug 2011
rPath	4.0	c.2008
SUSE Studio		Oct 2010

The SCIDIP-ES implementation of the Process Virtualization Toolkit is based on a specific add-on to the Oracle's VirtualBox[2], which is a professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2.

The Process Virtualization Toolkit is being developed using the VirtualBox Software Developer Kit (SDK). The Toolkit provides a web application supported by the SCIDIP-ES Ontology for users to generate artifacts that capture and describe the use of the processing environment in a form suitable for ingest into the SCIDIP-ES Registry, and also to link it to the ES Digital Objects. The process involves:

- Exporting a standard OVF/OVA file representing the processing environment to be preserved
- Creating the required RepInfo Label and Manifest (see below) objects to describe and link the preserved processing environment to the relevant Earth Science Digital Data Objects. RepInfo Labels are used by the SCIDIP-ES Knowledge Backbone[3] to support discovery and access of the structured knowledge that facilitate interpretation and (re-)use of preserved Earth Science Data Objects.

[2] https://www.virtualbox.org/
[3] The SCIDIP-ES Knowledge Backbone is an instance of the SCIDIP-ES e-infrastructure

- Uploading the Manifest which maps the virtualised environment digital package to the RepInfo Label to the SCIDIP-ES Registry
- Uploading the virtualised environment digital package to the SCIDIP-ES repository.

VirtualBox runs on all the mainstream Operating Systems and supports a large number of guest operating systems.

2 Installation Guide

2.1 Overview

There are four main steps for the Process Virtualization Toolkit installation and setup:

1. JDK 7 setup (if not already installed)
2. Oracle VirtualBox 4.1.x setup
3. Servlet Engine setup (if not already installed)
4. Process Virtualization Toolkit setup

2.2 Prerequisites

2.2.1 Software prerequisites

- Java SE Development Kit 7
- Oracle VirtualBox 4.1.x
- A Generic Servlet Engine (e.g. Jetty 9+)

2.2.2 Hardware prerequisites

- N/A

2.3 Download Information

The software can be downloaded from the SCIDIP-ES interactive platform, at the address

http://int-platform.scidip-es.eu/

2.4 License Information And Terms of Use

The SCIDIP-ES Process Virtualization Toolkit is licensed under the Apache License, Version 2.0 (the "License");

You may not use this software except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

2.5 Process Virtualization Toolkit Overall Architecture

The SCIDIP-ES Process Virtualization Toolkit is a simple Java 7 web application using Spring MVC. The following list includes all the dependencies of the project.

- Spring MVC (3.2.6)
- Spring Security (3.2.0)
- JSTL (1.2)
- Tiles (2.2.2)
- Oracle VirtualBox (4.1.4)
- scidipes-framework (1.1.2)
- Apache HttpClient (4.3.2)
- slf4j (1.6.1)

2.6 OSS/COTS Installation

2.6.1 Java SE Development Kit 7

Download and install Java SE Development Kit 7 following the installation guide at http://docs.oracle.com/javase/7/docs/webnotes/install/index.html

2.6.2 Oracle VirtualBox 4.1.x

Download and install Oracle VirtualBox (from https://www.virtualbox.org/wiki/Downloads) following the installation guide at https://www.virtualbox.org/manual/UserManual.html

2.6.3 Servlet Engine

Download and install the distribution of a generic servlet engine.

Using Jetty 9 as servlet engine, the zip distribution could be downloaded from http://www.eclipse.org/jetty/ .

Unzip the package and launch Jetty in a shell (or at the Windows prompt) executing the commands

```
cd <JETTY_HOME>
<JAVA_HOME>/bin/java -jar start.jar
```

Replace JETTY_HOME with the unzipped directory containing start.jar .

Replace JAVA_HOME with the path of the Java SE Development Kit 7 directory.

2.7 Process Virtualization Toolkit Installation

Rename the pvtk-NNNNNNNNNN.war to pvtk.war and copy it into the webapps directory of the servlet engine. If necessary, restart the engine.

2.8 Uninstallation

Delete the pvtk.war from the webapps directory of the servlet engine. If necessary, restart the engine.

Optionally, uninstall the servlet engine (for Jetty delete the unzipped directory) and the JDK 7 environment.

3 Using SCIDIP-ES Process Virtualization Toolkit

The usage of the Process Virtualization Toolkit is explained in this section by means of two use cases:

- Preservation of a processing environment
- Restoration and usage of a preserved environment

The case study is based on preserving the processing environment for operating the BEAM (http://www.brockmann-consult.de/cms/web/beam/) software, an open-source toolbox and development platform for viewing, analysing and processing of remote sensing raster data. In this case study, BEAM is used to process the Envisat's MERIS product.

BEAM is originally developed to facilitate the utilisation of image data from Envisat's optical instruments. It currently supports a growing number of other raster data formats such as GeoTIFF and NetCDF as well as data formats of other EO sensors such as MODIS, AVHRR, AVNIR, PRISM and CHRIS/Proba. Various data formats and algorithms are supported by dedicated extension plug-ins and hence this use case is representative of the types of process that SCIDIP-ES should preserve.

3.1 Use Case #1: how to preserve the BEAM environment

Rationale: Preservation of the MERIS data visualization and basic processing

Objectives:

- Preservation of Operating System: UBUNTU Linux Distribution (http://www.ubuntu.com/desktop)
- Preservation of the Java Runtime Environment : the BEAM package already contains a Java Runtime Environment
- Preservation of the BEAM Software (http://www.brockmann-consult.de/cms/web/beam/), i.e.:
  - VISAT - An intuitive desktop application to be used for visualization, analyzing and processing of remote sensing raster data.
  - A set of scientific tools running either from the command line or invoked by VISAT, also entirely written in Java.
  - A rich Java API for the development of new remote sensing applications and BEAM extension plug-ins. This following table lists the data product formats which are supported by BEAM using the reader modules provided in the standard installation. Information about the access to these products is given on the data sources page.

Instrument	Platform	Formats
MERIS L1b/L2	Envisat	Envisat N1
MERIS L3	Envisat	NetCDF
AATSR L1b/L2	Envisat	Envisat N1
ASAR	Envisat	Envisat N1
ATSR L1b/L2	ERS	Envisat N1, ERS
SAR	ERS	Envisat N1
OLCI1	Sentinel-3	NetCDF/SAFE
SLSTR1	Sentinel-3	NetCDF/SAFE
MSI1	Sentinel-2	JPEG2000/SAFE
CHRIS L1	Proba	HDF4
AVNIR-2 L1/L2	ALOS	CEOS
PRISM L1/L2	ALOS	CEOS
MODIS L2	Aqua, Terra	HDF4
AVHRR/3 L1b	NOAA-KLM	NOAA, METOP
TM	Landsat 4	GeoTIFF
TM	Landsat 5	GeoTIFF, FAST
ETM+	Landsat 7	GeoTIFF
OLI, TIRS	Landsat 8	GeoTIFF
SPOT VEGETATION	SPOT	HDF

BEAM supports the following generic raster formats:

Format	Support	Description
BEAM-DIMAP	read + write	The standard BEAM I/O format. It comprises an XML header based on the SpotImage/CNES DIMAP schema and ENVI images for the raster data.
GeoTIFF	read + write	A widely used EO data format, e.g. for Quickbird, LANDSAT, SPOT.
NetCDF	read + write	A widely used EO data format. BEAM supports NetCDF files conforming to the NetCDF CF Metadata Convention.
HDF-EOS	read	BEAM supports the HDF-EOS profile (HDF4) used by NASA Ocean Color data products of SeaWiFS, MODIS, OCTS, CZCS, and the gridded MODIS L3 products.

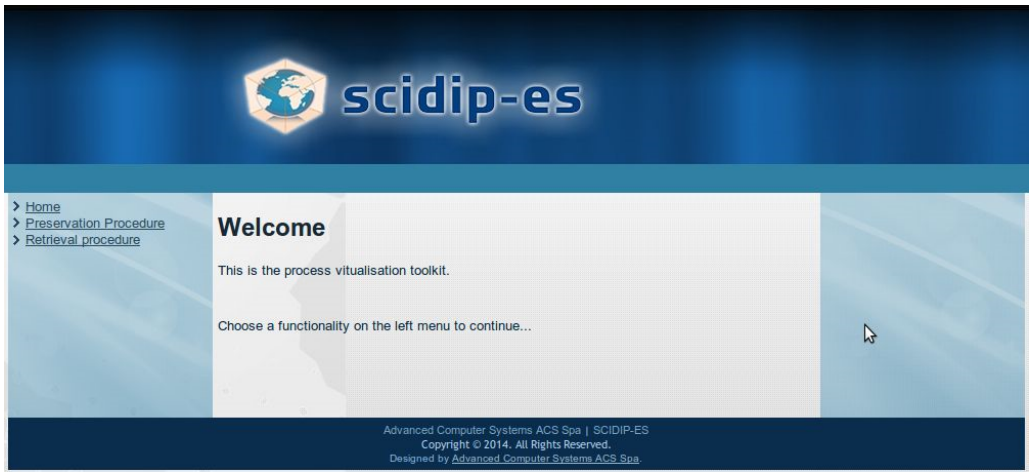
Prerequisites: Oracle VirtualBox and SCIDIP-ES Process Virtualization Toolkit installed and configured on a desktop machine (see section 2.6).

Sequence of Operations:

- Step 1 : create the virtual reference platform:
    - Download and install Ubuntu: http://www.ubuntu.com/download/desktop/thank-you?release=latest&bits=64&distro=desktop&status=zeroc
    - Download and install the BEAM Software:

- http://www.brockmann-consult.de/cms/web/beam/dlsurvey?p_p_id=downloadportlet_WAR_beamdownloadportlet10&what=software/beam/4.11/beam_4.11_linux64_installer.sh

- Step 2 : Configure the virtual machine to preserve and turn it off.
- Step 3 : Use the Process Virtualization Toolkit to preserve the environment:



  - If prompted, login using configured credentials (default is test/test)

  - Set the Virtual machine manager address to see your inventory

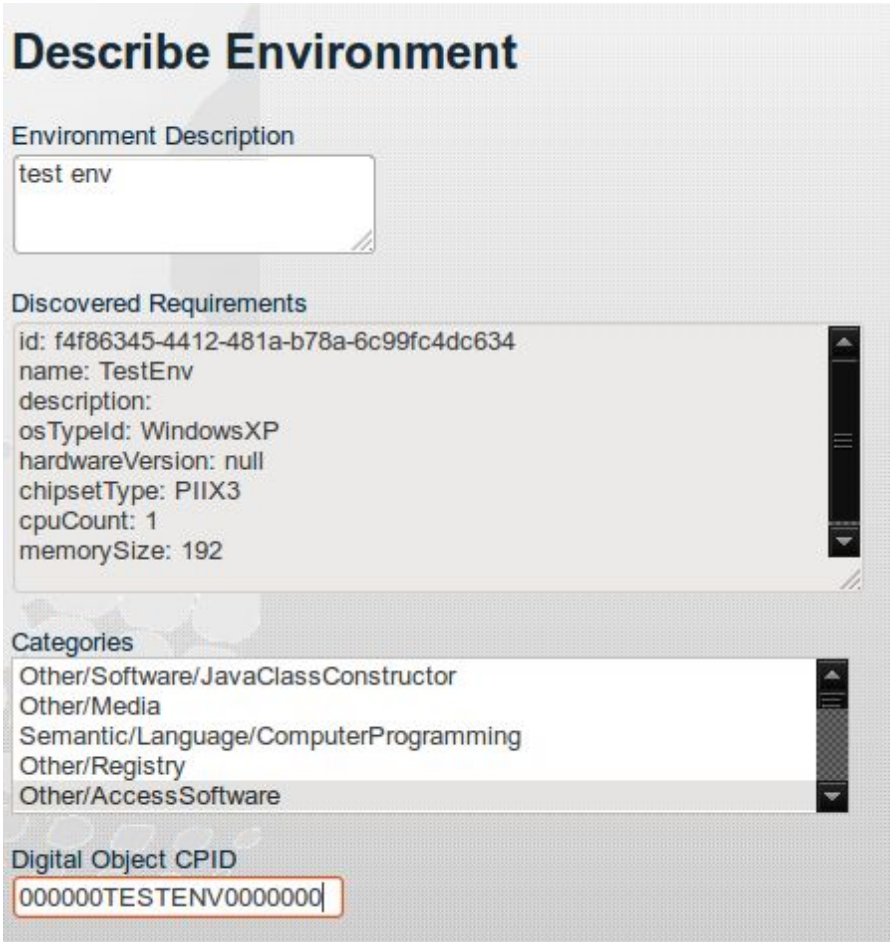
- Choose the Virtual machine among your inventory

- Search the related RepInfo Label (RIL)

- Select the RIL

- Select the Registry to save the manifest to

- Fill in the description and set a digital object Curation Persistent IDentifier (CPID). Hardware requirements are automatically gathered from the Virtual Machine



- Submit the Virtual Machine package/Manifest to the Repository/Registry

3.2 Use Case #2: Retrieval of a Preserved Process

Rationale: retrieval of a preserved virtualized process package

Objectives: To show the capability of retrieving the tool (BEAM) and use it to open/edit a MERIS image

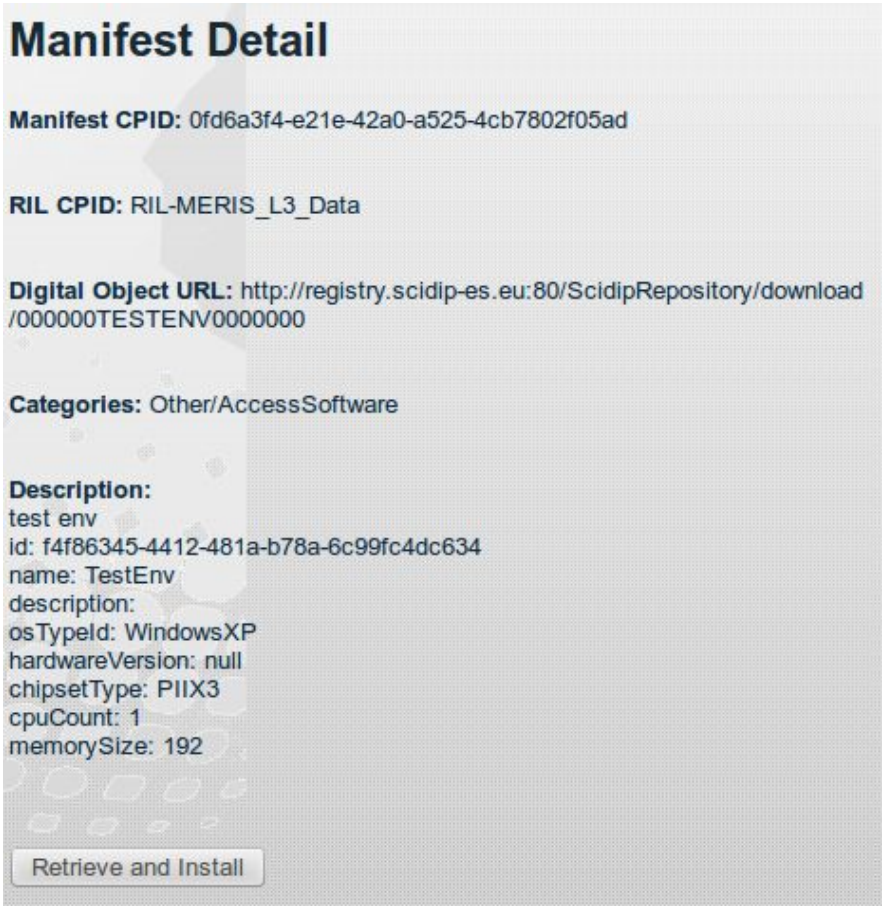
Prerequisites: Oracle VirtualBox installed and configured on a desktop machine.

Sequence of Operations:

- Step 1: Set the virtual machine manager address to import the environment from

- Step 2: search for the required manifest in the Registry

- Step 3: inspect the environment description



- Step 4 : retrieve (and import) the preserved process (BEAM) using the SCIDIP-ES Storage Service

4 Reference Manual

4.1 Keyboard shortcuts

None

4.2 Command-line commands

Access to the Oracle VirtualBox Manager is via the VirtualBox web service interface. The service is disabled by default, launch the *vboxwebsrv* executable to enable it.

The authentication manager must be properly configured (see Oracle VirtualBox documentation), it could be disabled by executing the command

VBoxManage setproperty websrvauthlibrary null

before launching the web service interface (*vboxwebsrv*).

4.3 Public APIs

None

5 Troubleshooting Common Issues

None

Annex A. Terminology

ACRONYM	DESCRIPTION
AIP	Archival Information Package
ARK	Archival Resource Key
CDMI	Cloud Management Interface
DOI	Digital Object Identifier
ES	Earth Science
GIS	Gap Identification Service
KB	Knowledge Base
OS	Orchestration Service
OWL	Web Ontology Language
PI	Persistent Identifier
PNM	Preservation Network Model
PURL	Persistent Uniform Resource Locator
RDF	Resource Description Framework
RepInfo	Representation Information
SNIA	Storage Networking Industry Association
SWKM	Semantic Web Knowledge Middleware
VM	Virtual Machine
WP	Work Package
XAM	eXtensible Access Method
XML	eXtensible Mark-up Language