

Emulation and Rendering Tools (CPP-015)

CPP-Identifier	CPP-015
CPP-Label	Emulation and Rendering Tools
Author	Kris Dekeyser
Contributors	Bertrand Caron, Johan Kylander
Evaluators	Felix Burger, Maria Benauer
Date of edition completed	29.08.2025
Change history	Comments
Version 1.0 - 29.08.2025	Milestone version

1. Description of the CPP

The TDA enables the rendering of *Objects* via the application of emulation and/or other specialist tools.

Inputs and outputs

Input(s)	
Data	<i>Representation</i>
Documentation / guidance	Emulation and rendering policy
Output(s)	
Data	<i>Representation</i> rendered in a suitable environment

Definition and scope

Digital preservation is all about ensuring long-term access to digital content despite the challenges of obsolescence, degradation, and technological change. Emulation and Rendering tools are important strategies in achieving that goal, and together with File migration ensure that digital content remains usable to the designated community.

Emulation attempts to recreate the original habitat - both hardware and software - of the digital *Object* in order to interact with it in the way it was intended. This is achieved by preserving both the content and the experience of the original environment. The purpose of emulation is to simulate the original operating system and the tools that are needed to interact with the *Object*.

Rendering tools interpret and display the digital *Objects* in the current environment using viewers based on modern technology. It is not uncommon for rendering tools to convert the digital *Object* into a more accessible form before displaying it. In such cases, viewers and converters collaborate to achieve the required result.

In addition to maintaining the *Bitstream* integrity of its holdings, semantic digital preservation entails ensuring their continued readability and usability. This necessitates that the TDA identifies the required environments and tools to support the actions its *Designated Community* is authorised to perform, regardless of whether direct access to the holdings is currently being provided.

Emulation and Rendering are two very different techniques to achieve a similar goal. They differ in some essential aspects:

- *Authenticity*: Emulation aims for the best authenticity by reproducing the entire look and feel of the original environment. The output of the rendering process may do a pretty decent job in reproducing the original *Representation* of the *Object*, but the interaction will be simulated at best. The rendering's conversion operation can be lossy and generate a less accurate result. Rendering is also more vulnerable to slight format changes and thus requires Format Validation.
- *User Experience*: Again, Emulation obtains the highest score as it will reinstate the original hard- and software as good as possible and allows the user to interact with

the *Object* to its full potential. Rendering will typically reduce the experience to a read-only view of the data with a limited set of operations available.

- *Preservation Scope*: In order to render the *Object*, only the content needs to be preserved. For emulation, however, both content and its environment need to be preserved.
- *Technical Overhead*: The requirements for Emulation are very high. A lot of resources (e.g. CPU and memory) are required to fire up the emulation. Moreover, the configuration and maintenance of an emulation environment is complex and time consuming. Performance may suffer and emulation does not scale easily. Rendering on the other hand, is relatively simple to set up, especially when no conversion is required or the conversion is done beforehand. Performance is mostly great, unless the *Object* is stored in slow off-line storage. The technique causes little problems for scaling.
- *Licensing*: Licensing of the operating system and software interacting with the *Object* may be a concern for emulation. For rendering of most common file formats, open source libraries and tools are available and licensing is less of a concern.

The choice for emulation versus rendering is captured in the Emulation and Rendering Policy. There are many possible decision factors and strategies for choosing emulation over rendering. For example:

- Manual configuration by archive maintainer;
- Requested by consumer;
- Decision based on format;
- Decision based on *Technical metadata*;
- Decision based on collection preferences, policies, agreements with producer, etc.

Process description

Trigger event(s)

Trigger event	CPP-identifier
Access requested to a given <i>Object</i>	CPP-025 (Enabling Access)

Step-by-step description

No	Supplier	Input	Steps	Output	Customer
	Initial strategy selection				
1	CPP-025 (Enabling Access)	<i>Representation</i> identifier	Collect the <i>Object's Representation</i>	<i>Representation Technical metadata</i>	
2		<i>Representation</i>	Collect the Environment <i>Objects</i> associated with the <i>Representation</i> (if present)	Environment <i>Objects</i>	
		<i>Technical metadata</i>			
3		<i>Representation</i>	Decide on strategy	Emulation (proceed to "Emulation")	
		<i>Technical metadata</i>			
		Environment <i>Object</i>			
	CPP-012 (Risk Mitigation)	Emulation and rendering policy		Rendering (Proceed to "Rendering")	

	CPP-018 (Community Watch)				
	Emulation				
E1		Environment <i>Object</i>	Retrieve the emulation properties from the Environment: -Hardware to be emulated -Operating System type and version -Tools to be installed in the OS	Emulation information	
E2		Emulation information	Prepare Emulation environment: -initialise the virtual machine based on operating system requirement -install the software requirements	Initialised emulated environment	
E3		Emulated environment	Inject or copy the <i>Representation</i> content in the emulated environment	Complete emulated environment	
		<i>Representation</i> Information			
E4		Complete emulated environment	Start the emulated environment and redirect input and output to the consumer		
	Rendering				
R1	CPP-025 (Enabling Access)	<i>Representation</i> identifier	Get the <i>Object's representation</i>	<i>Representation Technical metadata</i>	
R2		<i>Representation Technical metadata</i>	Retrieve the <i>Representation</i> content	<i>Representation</i> content	

R3		<i>Representation</i> Information	Make a choice of rendering tool	Rendering Tool	
	CPP-012 (Risk Mitigation) CPP-018 (Community Watch)	Emulation and rendering policy			
R4		<i>Representation</i> content	Execute the Rendering tool and hand over the <i>Representation</i> content		
		Rendering tool			

Rationale(s)¹ and worst case(s)

Rationale	Impact of inaction or failure of the process
It is good practice to provide the tools to let the consumer interact with the data (Except in the case of dark archives)	

2. Dependencies and relationships with other CPPs

Dependencies

CPP-ID	CPP-Title	Relationship description
CPP-025	Enabling Access	The access request is the trigger to invoke the rendering process or start up the emulated environment.
CPP-010	File Format Validation	In order to have a decent level of confidence in the rendering process, the <i>File</i> 's format needs to be validated.
CPP-012	Risk Mitigation	Risk Mitigation is in charge of defining the emulation and rendering policy that is meant to be applied by this CPP.
CPP-022	Significant Properties Definition	Rendering should be evaluated based on significant properties. The faulty rendering of an <i>Object</i> because of unsuitable hardware or software may go unnoticed and produce wrong interpretations from the TDA's community.

Other relations

Relation	CPP-ID	CPP-Title	Relationship description
Required by	CPP-013	Object Management Reporting	The process of selecting tools for emulation and rendering provides data to the TDA for reporting to the designated communities.
Facilitated by	CPP-014	File Migration	May be needed to support rendering tools in the long term

¹ Term derived from PREMIS.

Facilitated by	CPP-026	Normalisation	Normalisation of the file formats can reduce the effort and complexity for rendering and emulation significantly.
Facilitated by	CPP-028	Creation of Derivatives	Derivatives in formats and/or structures that are easier to render can be created prior to the rendering process to avoid time and resource consuming conversion processes.

3. Links to frameworks

Certification

Certification framework	Term used in framework to refer to the CPP	Section
CTS Link	Preservation Plan	R09 - Preservation Plan refers to emulation as a possible preservation strategy
Nestor Seal Link	Emulation strategy	C24 Interpretability of the archival information
ISO 16363 Link	/	In scope of 3.3.1 “The repository shall have defined its Designated Community and associated knowledge base(s) and shall have these definitions appropriately accessible.”

Other frameworks and reference documents

Reference Document	Term used in framework to refer to the process	Section
OAIS Link	Representation Information Rendering Software and Access Software	4.3.2.3.3 Representation Information Networks
PREMIS Link	Emulation Render	Emulation is mentioned several times w.r.t. PreservationLevel and Environment Objects. Rendering is mentioned in sections discussing Representations, Relations and Environment objects. Both processes are also described in the glossary.

4. Reference implementations

Example use case(s)

Rendering images with ICC profile embedded

Institutional Background	
Institution	Bibliothèque nationale de France, France
Hyperlink	/
Description	
Trigger event	In addition to its digital library, Gallica, BnF might provide access to its digital images with embedded ICC profiles on its public computer stations.
Problem statement	The diversity of born-digital images collected by BnF requires rendering software aware of its complexity. The rendering tool should be able to handle several file formats, different colour models, and many other more or less common characteristics of image <i>Files</i> : multi-image TIFFs, orientation indicated by an EXIF tag, etc. In particular, it should use the embedded ICC profile to enable proper colour management.
Proposed solution	BnF selected the XnView freeware as its solution for displaying images to its Designated Community. Though the software was natively supporting embedded ICC profiles, the setting was not activated by default, which illustrates the importance of adjusting the tool settings to ensure faithful rendering of digital <i>Objects</i> .

Publicly available documentation

Institution	Organisation type	Language	Hyperlink
TIB – Leibniz Information Centre for Science and Technology and University Library, Germany	National library	English	https://wiki.tib.eu/confluence/spaces/lza/pages/93608641/Preservation+Management#PreservationManagement-Emulation
	Non-commercial digital preservation service		
	Research infrastructure		
	Research performing organisation		
CSC – IT Center for Science Ltd., Finland	Non-commercial digital preservation service	English	https://urn.fi/urn:nbn:fi-fe2025040925236 (section 8)