iai industrial systems
part of HID Global

Host Interface Description Personalisation Systems

Software

Confidential information

Exported on
23/April/2025

Table of Contents

Introduction

This document describes the interface between IAI Personalisation Systems and the host environment providing the personal data. The structure of the document is as follows:

- Explanation of systems, which protocols, which applications each system supports and a guide to which chapters to read.
- Overview of the conceptual interfaces between system and host.
- Detailed explanation of the protocols.
- Description of data format's
- Error codes

This document is maintained and may change without prior notice. Please check with IAI for the latest version applicable.

Reading guide

For an overview of the interfaces and a choosing guide, read chapters:

- Conceptual overview of interfaces
- Systems
- Protocols

For the interface definition based upon a file share, read chapters:

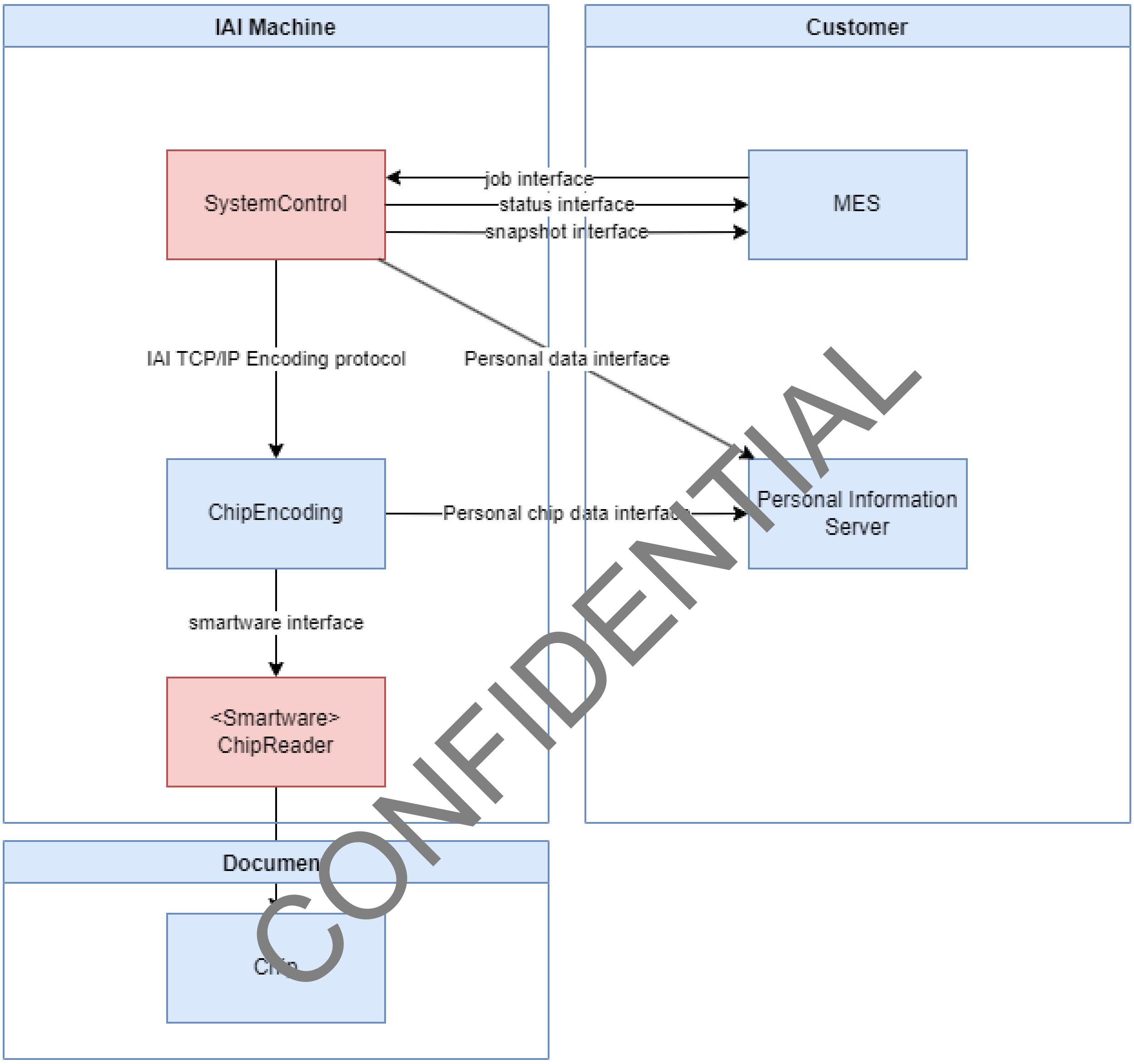
- Conceptual overview of interfaces
- Systems
- File share
- Jobfiles
- Job processing
- Data File Structures
- Status files
- Error codes

For the interface definition based upon HTTP REST, read chapters:

- Conceptual overview of interfaces
- Systems
- HTTP REST
- Jobfiles
- Data File Structures
- Error codes

Conceptual overview of interfaces

The diagram below describes the relevant interfaces and components for this document.



Any red component is provided by IAI, blue blocks are not provided by IAI and the responsibility of the integrator or customer. This document describes the following main components and interfaces:

Components (blocks in the above diagram):

1. **SystemControl** main software component of the IAI machines controlling all other processes in the machine. Communicates with the MES and Personal Information System.
2. **MES** Manufacturing Executing System, a customer system which controls the planning and status of production.
3. **Personal Information Server** a customer system which contains the personal information to put on the product.

Note that this is a conceptual diagram: a customer might not have an actual MES or the MES and Personal Information Server are integrated into one service.

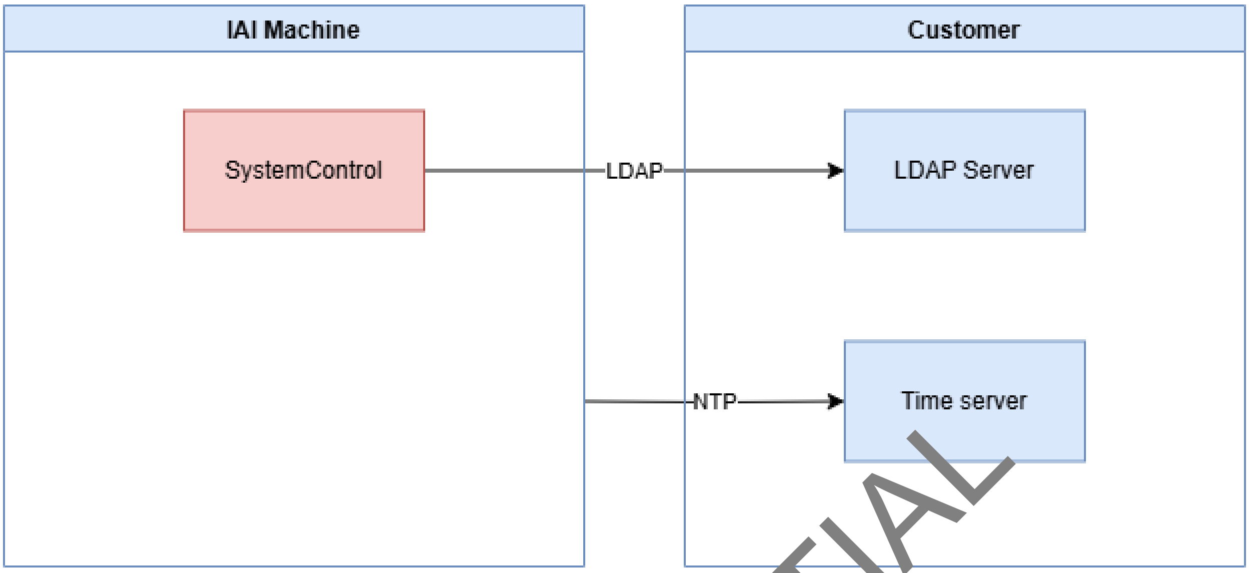
Interfaces (arrows in the above diagram):

1. **Job interface** - add/de;ete job - which instructs the system which products to create with which recipe.
2. **Status interface** - write status - which allows the customer systems to observe the status of the machine, the jobs and of the products.
3. **Personal interface** - get personal data - which allows the system to access the personal information from the customer systems.
4. **Snapshot interface** - write snapshot - which allows the customer system to retrieve verification results. As these may contain personal information it is separated from the status interface.

Out of scope of this document are:

1. **IAI TCP/IP Encoding protocol** this interface is described in "Chip Encoding Interface Personalisation Systems".
2. **ChipEncoding / Personal chip data interface / Chip** the IAI system does not handle the content of the chip encoding data as it's customer / integrator specific.
3. **ChipReader** the IAI machines contain smartware readers for the chip encoding process, but the details of this interface are not described here.

Supporting interfaces



Aside from the main interfaces the IAI machines also support:

1. **LDAP** standard protocol to connect the IAI machine to an LDAP Server (like OpenLDAP or Active Directory) at the customer, allowing for managing credentials in the LDAP Server and not locally on the machine.
2. **NTP** synchronize the time of the machine with the customer network time.

Systems

There are six types of system for which this document is applicable. In the table below the systems are depicted, with their supported application, protocol and document type.

System Name	Document type	Supported Protocol for host	Supporting interfaces
BookMaster One	Passport	▪ HTTP(s) REST ▪ File share ▪ Job files ▪ STS files ▪ CSV audit file ▪ XML audit file	LDAP NTP
CardMaster One	ID-1	• File share • Job files • STS files	LDAP NTP
BookMaster Pro	Passport	• File share • Job files • STS files • CSV audit file	NTP
CardMaster Desk	ID-1	• File share • Job files • STS files	NTP
BookMaster Desk	Passport	• File share • Job files • STS files	NTP
Element	ID-1	• File share • Job files • STS files	LDAP NTP

Table 1: Overview

Hardened / unhardened

There are two versions of the systems available, which differ greatly in their security setup:

1. Unhardened systems
2. Hardened systems

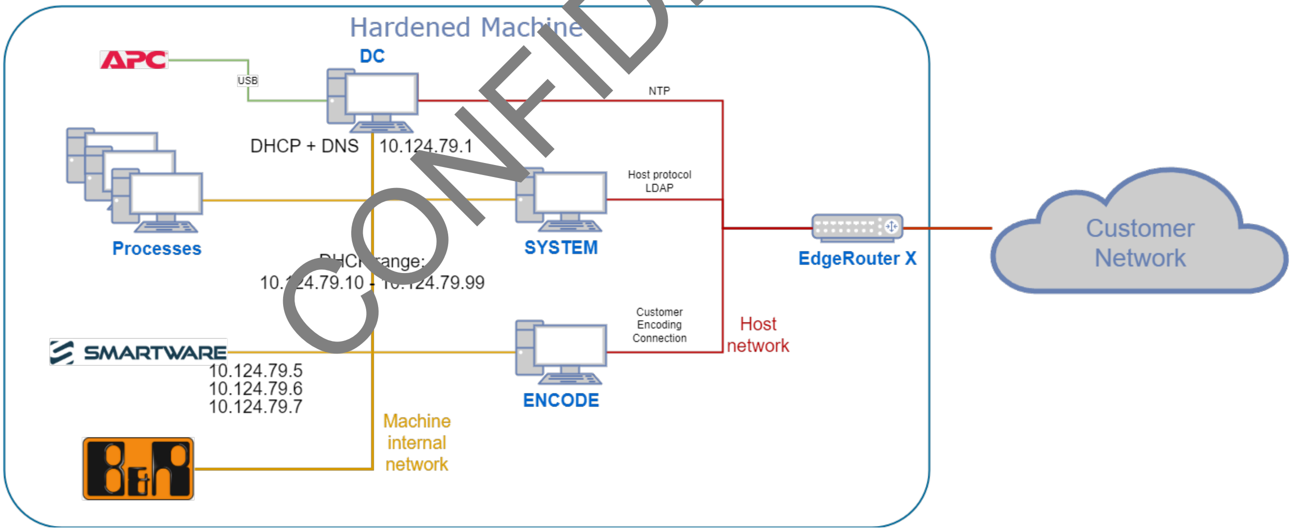
Newly delivered systems will always be hardened systems.

Network

Requirements

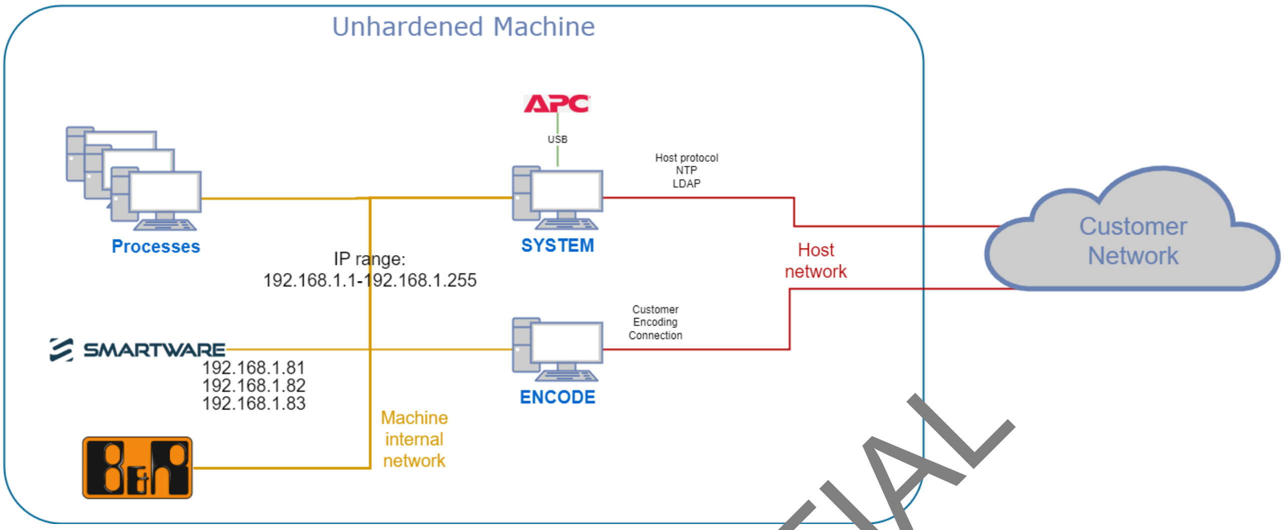
Parameter	**Value**
Network	100/1000 Mbit/sec.
Physical connection	UTP
Host computer response time incl. all data transfer time over the network for one document	< 0.5 sec.
IP range	Configurable Range 192.168.1.x (unhardened machines) or 10.124.79.* (hardened machines) is not possible as it is reserved for the internals of the machine.

Network layout hardened BMO/CMO



The EdgeRouter can be configured as switch/router or whatever seems fit for integration with the customer network.

Network layout unhardened BMO/CMO



Supporting interfaces

LDAP

Users can be administered locally on the machine or remotely in the customers Active directory. In case Active Directory is preferred the system should be configured using the service screens. With these screens a group in the IAI software can be coupled to an active directory group in the customer Active Directory. Corresponding rights need to be configured locally on the machine.

Security log

All PCs in the machine maintain a security log: major system interactions, like user log in and out for Windows users, are logged to a special security logfile on the BookMaster and CardMaster systems. The security interactions in the SystemControl are also logged to the logfile.

NTP

In case of unhardened systems NTP can be configured to synchronize time at startup.

In case of hardened systems the internal domaincontroller needs to be NTP connected to the customer network, to be able to synchronize time.

Credentials are not required but the source needs to be acknowledged. Exact specifications to be discussed.

Host protocols

There are two different protocols supported by the systems.

- File share
    - Windows share
    - Communication through files
- HTTP(s) – REST api
    - Support for encryption (requires handling of certificates by customer)
    - Authentication through client certificate (requires handling of certificates by customer)
    - Contains the most detailed information
    - Direct communication to the system

Versioning

With software updates or new machines a newer version of interface can become available. The following strategy is followed:

- Any communication that is received from the host system will remain compatible with the same functionality. Any modifications will be additions which do not modify the existing behavior.
- Any communication that is send by the machine might contain extra information, but existing fields will not be removed or changed behavior.

If for whatever reason the interface cannot be kept compatible, the customer will be informed.

File share

The personalised data for the IAI personalisation system is available through a network share. The related data for each application or feature in the document is retrieved by the IAI software from this host network share (see below diagram).

Parameter	**Value**
Host computer Operating System	MS Networking e.g. WIN2000 Server or Unix with Samba
Data location	On the host computer a separate network share is available for each IAI system on which the system has enough privileges to create, read, rename and delete files. NOTE: It is configurable whether the data is stored in a folder per job, or all data is stored in the root of the specified network share (which can lead to huge amount of files in one folder)

Table 2: Specification network

The systemcontrol will allow to configure the credentials to login to the shared folder on the customer network. On the BMO, CMO and Element it is also possible to use the logged in user to access the shared folder.

The content of the files are described in chapters Jobfiles, Data File structures and Status files.

Note that for hardened systems by default SMB1 is not supported as it has known vulnerabilities.

Folder structure

The folder structure is configurable on the machines, but there are network / OS driven limitations which need to be taken into consideration. A common setup would be:

- Folder Jobs
  - Job1.xml
  - Job2.xml
  - ...
  - Folder Job1 (containing perso files)
  - Folder Job2 (containing perso files)
  - ...
- Folder Audit
  - Folder Job1 (containing output sts files)
  - Folder Job2 (containing output sts files)
  - ...

This setup isolates the folders as much as possible, allowing the machine to efficiently query the files over the network. In contrast: putting perso files in the same folder as the job files requires the machine to querying through a large amount of files for querying which job files are there, which will slow down the machine significantly.

Job files

Parameter	Value
Format	XML encoding UTF-8
Filename	<name>.job e.g. 123E456.job
Postfix	Not Applicable
Extension	Job
Name	Letters (A..Z) & digits (0..9)Max 25 characters

The operator selects the appropriate job file from a list of job files on the display of the System. This list is a result of searching on the designated share on the host computer for all job files (*.job). The operator can select more than one job via the job selection screen. A job can have the following status reflected in the extension:

State	Extension
New	.job
Active	.active
Done	Job file is removed
Aborted	.aborted

When a job is queued, the system verifies if all the related operation files are available on the host computer. Only when all files are available, the system will accept the job. A started job can either be completed, or aborted at any time after the job was started.

Completing a job:

When a job is completed the job file is deleted on the host. A job will be completed when all products for that job are taken in from the input hopper, and are either transported to the output bins (output tray/rework/reject) or taken out of the machine manually due to an error.

Aborting a job:

When a job is aborted, the job file will be renamed to .aborted on the host. Additionally the documents that are not already in production will be marked as remake (audit file on host will be renamed to *.RMK0_0). Products which were already in the machine at the time of the job abort, will be finished as normal.

Status files

The extension of a status file is used to give the status of a product in the system. The status file will either be provided by the host environment, and when it is not provided it will be created by the IAI software on the host share.

- When a product is inserted, a status with extension <product_number>_sts.txt will be created.
- When the first operation is applied, the file is renamed to <product_number>_sts.**IBW**.
- When the document is successfully processed by the personalisation system, the file is renamed to <product_number>_sts.**OK**.
- When an error occurred during the processing of the document, the file is renamed to <product_number>_sts.**REJ[errorcode].**
- When the product is not physically touched by any process, the file is renamed into <product_number>_sts.**RMK[errorcode].**
- When the product is marked as suspect (e.g. by verification), the file is renamed into <product_number>_sts.**SUS[errorcode]**.There is also an option in the customer configuration, SuspectStateOnHost, to mark suspect products either as reject or as OK.
- When the operator aborts a job, all host files of the unused numbers are deleted. The extension is renamed into <product_number>_sts.**RMK.**

The error code is represented by a global error and a specific error (GLOBALERROR_SPECIFICERROR, e.g. 3_32). Thus the filename of a rejected product will be e.g. <productnumber>_sts.**REJ3_32**. In the next section the meaning of the known error codes.

The machines can be configured to use lower case, upper case and strict casing (following the casing in this document) for integration with case sensitive systems.

Additional Status files

CSV Audit File

ℹ Currently this format is not supported by CMO / Element

In addition to the current status files there is a possibility to extend the information.

Every time a booklet is leaving the machine a CSV file is created providing the following information:

- JobName
- ProductNumber
- Barcode
- ChipResultData
  Every time a passport is processed by the encode unit and is done successfully the integrator application gives a Chip Result Data back. The integrator can define a custom key to store the data.
- Date
- Time
- UserName
- MachineNumber

- GlobalDocStatus
  Decimal number, existing error code major number
  In case product has not been touched, this value is -1 to indicate a rework
  In the special case that an unknown error has occurred, the GlobalDocStatus will be -2 indicating it has been rejected and the DetailedDocStatus will be 0.
- DetailedDocStatus
  Decimal number, existing error code minor number
- > The following properties are BMO only
- RecipeName
- EncodeLocation
  - According to the format *location-sublocation*, where the location is the unit number within the machine and the sublocation is the location within the encode unit itself (1-4)
  - For example: 6-1.
  - Will be empty for aborted products.
- EngraveLocation
  - According to the format *location.side*, where the location is the unit number within the machine and the side is the side of the exchanger on which the product was (A or B)
  - For example: 17-A.
  - Will be empty for aborted products.

XML Audit File

> Currently this format is only supported by BMO

XML Audit Files give an insight where all the different products has been produced. Once the job has finished a file is genereated showing which operations has been executed on the different products in the job. See Host Interface Description Personalisation Systems#Job status format.

HTTP REST

Currently this type of interface is only supported by BMO

The data is communicated with the machine over HTTP REST. All interfaces can be modified to use HTTPS instead of HTTP, to ensure encryption and authentication.

Job interface

Post job

Description	Make a job available to the machine. An operator has to schedule the job before it is executed.
Direction	Machine – server Host – client
URL	http://{MachineName}/Host/Jobs/ProJob0001.xml
Method	POST
Request content	See Job files.
Reply content	• Always empty • In case of OK: http status 200 • In case of invalid job file: http status 400

Delete job

Description	Remove a job from the machine. All personal data and status information associated with the job is deleted as well.
Direction	Machine – server Host – client
URL	http://{MachineName}/Host/Jobs/ProJob0001.xml
Method	DELETE
Request content	Empty

Reply content	• Always empty • In case of OK: http status 200 • In case of job not allowed to be removed: http status 400 • In case of job not found: http status 400

Delete all jobs

Description	Removes all job from the machine that are not active. All personal data and status information associated with the job is deleted as well.
Direction	Machine – server Host – client
URL	http://{MachineName}/Host/Jobs/
Method	DELETE
Request content	Empty
Reply content	• Always empty • http status 200

Abort job

Description	Abort a job from the machine. This is only allowed when the job is active. No new products will be picked for the job, but the products in the machine will be finished.
Direction	Machine – server Host – client
URL	http://{MachineName}/Host/Jobs/Abort?{JobName}
Method	POST
Request content	Empty
Reply content	• Always empty • In case of OK: http status 200 • In case of job not allowed to be aborted: http status 400 • In case of job not found: http status 400

Status interface

Job list

Description	Request all the jobs that are posted to the machine.
Direction	Machine – server Host – client
URL	http://{MachineName}/Host/Jobs/
Method	GET
Request content	Empty
Reply content	• See below • Http status 200

Reply content

Parameter	**Value**
Job	List of jobs in the machine
Job.Name	The unique name of the job
Job.Status	The status of the job (see next chapter for detailed information)

```xml
<?xml version="1.0" encoding="utf-8"?>
<HostJobOverview>
<Job>
<Name>Test</Name>
<Status>Queued</Status>
</Job>
</HostJobOverview>
```

Job and product Status

Description	Request the status of a job on the machine. Note that multiple products are in the machine at the same time, so multiple products can have overlapping states (in progress, but not encoded for instance).
Direction	Machine – server Host – client
URL	http://{MachineName}/Host/Jobs/{JobID}.xml
Method	GET
Request content	Empty
Reply content	• See below • In case of OK: http status 200 • In case of job not found: http status 400

Reply content

See Host Interface Description Personalisation Systems#Job status format.

Job activation

Description	Send the job status to the host when the job is started. When an error occurs the job is not started and production is paused. Note: the machine will wait for this call to complete, slow handling of this call can affect the performance of the machine.
Direction	Machine – client Host – server
URL	Configurable
Method	POST
Request content	See "Reply Content" of "Job and product status"
Reply content	• Always empty • In case of OK: http status 200 • In case of not OK: job iwll not start.

Job end

Description	Send the job status to the host when the job is finished. Errors are ignored.
Direction	Machine – server Host – client
URL	Configurable
Method	POST
Request content	See "Reply Content" of "Job and product status"
Reply content	• Always empty • Errors are ignored

Snapshots

Description	Request all verification snapshots for a job. Verification snapshots become available as soon as a product has been finished. Optionally the reduceResolutionPercentage can be set to indicate a resolution reduction. For example: when this value is set to 50, a snapshot of 1000x1000 will be send as 500x500. Another option is the filtering based upon jobDefinedNumber and/or documentNumber. This allows to retrieve only the snapshots for one product. Filtering on both numbers is possible, the product will have to match both numbers in that case. Arguments have to be separated by the '&' character.
Direction	Machine – server Host – client

URL	http://{MachineName}/Host/Snapshots/{JobID}.xml
	http://{MachineName}/Host/Snapshots/{JobID}.xml?reduceResolutionPercentage=50
	http://{MachineName}/Host/Snapshots/{JobID}.xml?jobDefinedNumber={value}
	http://{MachineName}/Host/Snapshots/{JobID}.xml?documentNumber={value}

Method	GET

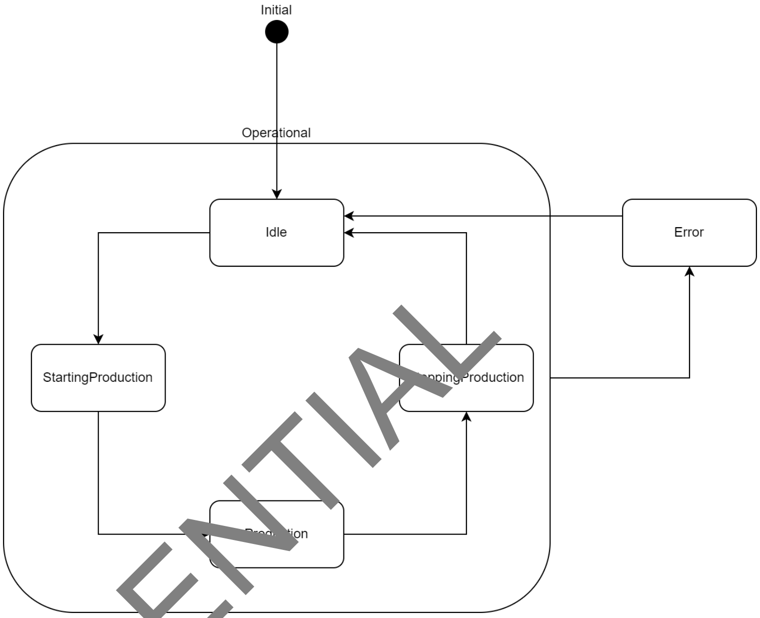
Request content	Empty
Reply content	• See below • In case of OK: http status 200 • In case of job or document not found: http status 400

Reply Content

Parameter	**Value**
Job.Name	The unique name of the job
Job.Product	A list of all finished products in the job.
Job.Product.JobDefinedNumber	The number as defined in the job.
Job.Product.DocumentNumber	Optional. The number read from the physical document (either by barcode/chip/ocr).
Job.Product.Snapshot	A list of all available snapshots for the product.
Job.Product.Snapshot.Id	The unique name of the snapshot.
Job.Product.Snapshot.Content	Base64 encoded content of the snapshot.

```xml
<?xml version="1.0" encoding="utf-8"?>
<SnapshotOverview xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<JobName>Test</JobName>
<Product>
<JobDefinedNumber>0</JobDefinedNumber>
<Snapshot Id="snap.jpg">/9j/4AAQS</Snapshot>
</Product>
<Product>
<JobDefinedNumber>1</JobDefinedNumber>
<Snapshot Id="snap.jpg">/9j/4AAQSkZJR</Snapshot>
</Product>
</SnapshotOverview>
```

Machine status

Description	Requests the status of the machine. The status follows this state diagram:  Initial Operational Idle Error StartingProduction StoppingProduction Production
Direction	Machine – server Host – client
URL	http://{MachineName}/Host/machinestatus/
Method	GET
Request content	Empty
Reply content	• See below • http status 200

Reply Content

Parameter	**Value**
Status	• Idle • StartingProduction • Production • StoppingProduction • Error

```
<?xml version="1.0" encoding="utf-8"?>
<MachineStatusData>
<Status>Idle</Status>
</MachineStatusData>
```

Machine Status push

Description	Send the machine status to the host upon change. Errors are ignored
Direction	Machine – client Host – server
URL	Configurable
Method	POST
Request content	See "Reply Content" of "Machine status"
Reply content	• Always empty • Errors are ignored

Alarms

Description	Requests all the occurred alarms on the machine. Alarms are reset when the machine is restarted and a maximum of 100 alarms is cached. As part of the URL the date/time from which the alarms should be shown can be given, providing a way to filter out alarms that have already been detected.
Direction	Machine – server Host – client
URL	http://{MachineName}/Host/alarms/ http://{MachineName}/Host/alarms/?2020-04-23T12:35:56.6020589+02:00
Method	GET
Request content	Empty
Reply content	• See below • In case of OK: http status 200 • In case of job not found: http status 400

Reply Content

Parameter	**Value**
Alarm	A list of alarms
Alarm.Time	The date/time this alarm has occurred.
Alarm.Type	Error – error will cause the machine to stop and it will not be available for production until resolved Warning – an event occurred which requires attention, but it does not block production.
Alarm.Message	The English message as shown to the user.
Alarm.ErrorCode	A code which represents the error. Starts with E for error, W for warning, followed by a unique identifier.

```
<?xml version="1.0" encoding="utf-8"?>
<Alarms>
<Alarm>
<Time>2020-04-23T12:39:36.5850044+02:00</Time>
<Type>Error</Type>
<Message>Could not activate job Test</Message>
<ErrorCode>E74.1685</ErrorCode>
</Alarm>
</Alarms>
```

Alarms push

Description	Send the machine alarm to the host upon occurence. Errors are ignored
Direction	Machine – client Host – server
URL	Configurable
Method	POST
Request content	See "Reply Content" of "Alarms"

Reply content	- Always empty - Errors are ignored

Message to operator

Description	Pushes a message from the host to the machine to notify the operator.
Direction	Machine – server Host – client
URL	http://{MachineName}/Host/Messages/
Method	POST
Request content	Raw string to be displayed to operator.
Reply content	- Empty - Http status 200

Personal data

Description	Requests the personal information from the system of the customer. The url is configurable in the IAI system. Three parameters can be added that depend on product parameters: - JobName - JobDefinedNumber (the number of the product in the job) - DocumentNumber (the number read from the product) The IAI system will request the information once all data is available to do the request and before it is required by one of the processes in the machine.
Direction	Machine – client Host – server
URL	Configurable. Example: http://host/{JobName}/{JobDefinedNumber}{DocumentNumber}.xml
Method	GET
Request content	Empty

Reply content	• See below • Errors will be logged and the associated product will be marked as rework by the IAI machine.

Reply content

The system uses separate data files for each operation (e.g. ImagePerf, NumberPerf, Laser-Engraving, etc). The exact naming and content of these files are described in next chapter. Thus, depending on the configuration of the machine the required data files can be different. The data structures and formats, which the system(s) uses for their operations, are given separately in an overview below. The content of file is expected to be base64 encoded data. When the same file should be used for multiple inputs a reference can be made. In this case the content needs to be empty and the reference needs to be equal to the ID of the referenced file. Aspect-ratio, file type and required minimal resolution should be compatible between referenced files.

```
<?xml version="1.0" encoding="utf-8"?>
<Files xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/
2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="../../Schemas/Files.xsd">
  <File Id="_CINK.txt">AQJlZg==</File>
  <File Id="..." Reference="_CINK.txt />
  <File Id="...">AQJlZg==</File>
</Files>
```

Encryption and authentication

The HTTP REST interface support encryption using TLS. These mechanisms rely on the certificates for both encryption and authentication. The customer is responsible for providing the certificate and keys and thus for the certificate and key infrastructure. The machine can run in the following modes:

Description	**HTTP REST**	**Certificates on machine**
No encryption	Supported	None
Encryption Machine authentication – host validates machine	Supported	Root certificate Machine certificate Machine private key
Encryption Machine authentication Client authentication by root – machine validates host by checking if certificate is signed by correct root	Supported	Root certificate Machine certificate Machine private key

Encryption Machine authentication Client authentication by host – machine validates host by checking if certificate is the expected host certificate	Supported	Root certificate Machine certificate Machine private key Host certificate

The following key formats are supported:

Description	**Format**
Machine certificate + private key	pkcs
Root certificate	cer
Host certificate	cer

Job data format

Jobs are used to tell the system which processes, layouts and data to use.

Product numbers are expected to be unique over jobs, at least while the jobs are active in the machine.

Job data format specification

The XML file contains the following elements. See Host Interface Description Personalisation Systems#Appendix A for the XSD specification.

Name	Description
Description	A small descriptive text which described the intent of the job
RecipeName	Name of machine recipe file which defines which operations are applied, e.g. RegularPassport.rcp. The customer can define the desired names for the different types of products, and IAI will supply these recipe files preinstalled on the system.
UseNumberReading	Boolean (true/false) that defines if the machine should read a number from the document. The NumberReadingType defines what kind of number reading is applicable (e.g. barcode).
NumberReadingType	Can be Barcode or Chip, defaults to Barcode. Barcode means in this case depending on the recipe/machineconfiguration to read a visible identification of the document, not only a barcode itself.
NumberOfProducts	Specifies the number of products, should be used when the product numbers field is left empty.
JobScenario	Defines the data files reference number scenario. Data files will be collected from the host according to the numbers created by the scenario. The following values are available: *JobDefinedNumber* When this scenario is chosen the product numbers that must be processed for the given job, are prescribed in the job file itself (in the element <ProductNumbers>). The value of each element must be unique and will be used as data file reference. Hence it should only contain valid filename characters. Space character is not allowed. *ReadNumber* The number will be read from the product, e.g. by a barcode reader installed in the machine.
RunType	Internal use only.

Name	Description
ProductNumbers	If JobData.JobScenario is set to JobDefinedNumber, the products that must be processed by this job are specified in this section. Product numbers cannot contain the following characters < > : " / \ | ? *
ReadNumbersInFixedOrder	When this value is set to true, the product number is read from the product, and the order of the products that are entered into the machine is prescribed by the list of product numbers in the (in the element <ProductNumbers>).
DeleteJobFile	Do not use: this property is meant for testing purposes only, not for production scenarios. In case the property is set to false, the job file will not be deleted and can be reused after it is finished.
DeleteHostDataOnOutput	If set to true, the data files of a product are removed when a product is taken out of the system (either to the output tray, the reject/rework bins or when a product is taken out of the system manually.
DataLocation	Optional path that defines the data file location per job. If this element is not filled in, then the path from the CustomerConfiguration.xml is used to determine the location of the job. Note: in the CustomerConfiguration it is also possible to define the data to be stored per job, this is done using the $JobName tag in the path name (e.g. <DataLocationOnHost>S:\LocalHost\PersonalData\ $JobName</DataLocationOnHost>)
AutoQueue	Optional, if true the job will be added to the queue automatically.
Priority	Optional for use when AutoQueue is enabled. A higher priority will be executed before jobs in the queue will a lower priority.
Inputs/Outputs	Optional for use when AutoQueue is enabled. Selects the index for the inputs/outputs to use.

Job example

Example

```
<?xml version="1.0" encoding="utf-8" ?>
<JobData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://
www.w3.org/2001/XMLSchema">
```

```
<Description>Example</Description>
<RecipeName>ExamplePassport.rcp</RecipeName>
<UseNumberReading>false</UseNumberReading>
<JobScenario>JobDefinedNumber</JobScenario>
<ReadNumbersInFixedOrder>false</ReadNumbersInFixedOrder>
<ProductNumbers>001</ProductNumbers>
<ProductNumbers>002</ProductNumbers>
</JobData>
```

Production scenarios

Below the supported scenario's, with their corresponding properties in the job file.

	JobData.UseNumberReading	**JobData.JobScenario**	**JobData.ReadNumbersInFixedOrder**	**JobData.NumberOfProducts**	**File checking**
Product number in job, no number reading	False	JobDefined Number	N/A	N/A	All files for all defined products should be present when loading job
Product number in job, with number reading, fixed order	True	ReadNumber	true	N/A	All files for all defined products should be present when loading job
Product number in job, with number reading, no fixed order	True	ReadNumber	false	N/A	All files for all defined products should be present when loading job
No product number in job, with number reading	True	ReadNumber	N/A	Number of products in job	For each product the files are copied (thus must be present) after a specified operation

Job data format CMO/Element specification

The job format for the CMO and Element differs from the standard format. See Appendix B for the XSD specification.

Recipe	Name of machine recipe file which defines which operations are applied, e.g. RegularIDCard.rcp. The customer can define the desired names for the different types of products, and IAI will supply these recipe files preinstalled on the system.
Inputs	List of cassettes indicating which positions the machine will be pick cards from.
Outputs	List of cassettes indicating which positions the machine will be push successfully produced cards into.
Delete	Do not use: this property is meant for testing purposes only, not for production scenarios. In case the property is set to false, the job file will not be deleted and can be reused after it is finished. Also the perso data will not be deleted.
JobDataFolder	Optional path that defines the data file location per job. If this element is not filled in, then the path from the CustomerConfiguration.xml is used to determine the location of the job. Note: in the CustomerConfiguration it is also possible to define the data to be stored per job, this is done using the $JobName tag in the path name (e.g. <DataLocationOnHost>S:\LocalHost\PersonalData\ $JobName</DataLocationOnHost>)
SuspectDestination	The output destination used for suspect products. When not specified the user has the option to select a suspect destination.
ReworkDestination	The output destination used for rework products. When not specified the user has the option to select a rework destination.
RejectDestination	The output destination used for reject products. When not specified the user has the option to select a reject destination.
ReadNumber	Boolean (true/false) that defines if the machine should read a number from the document. The NumberReadingType defines what kind of number reading is applicable (e.g. barcode).
NumberReadingType	Can be Vision, ConcactChip, ContactlessChip or ChipUhf.
ReadNumberAsProductNumber	Use the number read from the document as identification for the operator and on the host.
Products	A list of all products to produce.

Products.Product.Inputs	Optional: overrides the list of the job, for this product only this list of inputs will be used.
Products.Product.Outputs	Optional: overrides the list of the job, for this product only this list of outputs will be used.
Products.Product.Number	The product number for this product. An alpha-numerical reference to the product data required for the product Product numbers cannot contain the following characters < > : " / \ | ? *
Products.Product.Recipe	Optional: overrides the recipe of the job, for this job the recipe of this product will be used.

CMO/Element Job example

The example below defines two products. All products are defined **without** product numbers, as they are read from the card.

```
<?xml version="1.0"?>
<Job xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://
www.w3.org/2001/XMLSchema">
  <Recipe>UTO\Inkjet\UTO-Inkjet.rcp</Recipe>
  <Inputs>
    <Cassette Index="1"/>
    <Cassette Index="2"/>
    <Cassette Index="3"/>
    <Cassette Index="4"/>
    <Cassette Index="5"/>
    <Cassette Index="6"/>
  </Inputs>
  <Outputs>
    <Cassette Index="1"/>
    <Cassette Index="2"/>
    <Cassette Index="3"/>
    <Cassette Index="4"/>
    <Cassette Index="5"/>
    <Cassette Index="6"/>
  </Outputs>
  <SuspectDestination>7</SuspectDestination>
  <ReworkDestination>8</ReworkDestination>
  <RejectDestination>0</RejectDestination>
  <ReadNumber>true</ReadNumber>
  <NumberReadingType>ContactChip</NumberReadingType>
  <ReadNumberAsProductNumber>true</ReadNumberAsProductNumber>
  <Products>
    <Product Number="XA0001" />
    <Product Number="XA0002" />
  </Products>
</Job>
```

CMO/Element Job extended example

The example below is the same as the previous example but allows different recipe and input/output selection per product within a job.

```
<?xml version="1.0"?>
<Job xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://
www.w3.org/2001/XMLSchema">
<SuspectDestination>7</SuspectDestination>
<ReworkDestination>8</ReworkDestination>
<RejectDestination>0</RejectDestination>
<ReadNumber>true</ReadNumber>
<NumberReadingType>ContactChip</NumberReadingType>
<ReadNumberAsProductNumber>true</ReadNumberAsProductNumber>
<Products>
<Product Number="XA0001" Recipe="IAI_Card\UTO_ID_Card.rcp">
<Inputs>
<Cassette Index="1" />
<Cassette Index="2" />
<Cassette Index="3" />
<Cassette Index="4" />
</Inputs>
<Outputs>
<Cassette Index="1" />
<Cassette Index="2" />
<Cassette Index="3" />
</Outputs>
</Product>
<Product Number="XA0002" Recipe="IAI_Card\UTO_ID_Card.rcp">
<Inputs>
<Cassette Index="1" />
<Cassette Index="2" />
<Cassette Index="3" />
<Cassette Index="4" />
</Inputs>
<Outputs>
<Cassette Index="1" />
<Cassette Index="2" />
<Cassette Index="3" />
</Outputs>
</Product>
</Products>
</Job>
```

Job status format

This format is used to indicate the detailed status of the job with its product. See Host Interface Description Personalisation Systems#Appendix C for the XSD specification.

Parameter	**Value**
Job.Name	The unique name of the job
Job.RecipeName	The recipe used for this job
Job.Status	The current state of the job, can be: • Available • Queued • Active • Finished • Aborted • AbortedFinished
Job.StartTime	Start time of the job according to the date/time format (see below). As long as the job is not started, the element will be unavailable.
Job.EndTime	End time of the job according to the date/time format (see below). As long as the job is not finished, the element will be unavailable.
Job.StartedBy	The name of the operator who started the job
Job.Product	A list of all products in the job.
Job.Product.JobDefinedNumber	The number as defined in the job.
Job.Product.DocumentNumber	Optional. The number read from the physical document (either by barcode/chip/ocr).
Job.Product.ErrorCode	Optional. The error code for this product as described in the error code chapter.
Job.Product.Status	The current status of the product: • Idle • InProgress • OK • Suspect • Rework • Reject

Job.Product.StartTime	Time the product entered the machine according to the date/time format (see below). As long as the product is not started, the element will be unavailable.
Job.Product.EndTime	Time the product left the machine according to the date/time format (see below). As long as the product is still in the machine, the element will be unavailable.
Job.Product.Operation	A list of all operations for the product. Will be empty when the product has not entered the machine yet.
Job.Product.Operation.Name	The name of the operation, possible values are: - ContactEncode - ContactLessEncode - Inkjet - Engrave - ImagePerf - NumberPerf - LabelPrint - Vision - EngraveBack
Job.Product.Operation.Status	The status of the operation, possible values: - Pending (operation is not executed yet) - Busy (operation is being executed) - Done (operation has been finished succesfully) - Failed (operation has been finished with a failure) - Suspect (operation has been finished with a suspected failure)
Job.Product.Operation.StartTime	Time the operation on this product started according to the date/time format (see below). As long as the operation is not started, the element will be unavailable.
Job.Product.Operation.EndTime	Time the operation on this product finished according to the date/time format (see below). As long as the operation is not finished, the element will be unavailable.
Job.Product.Operation.ErrorCode	Optional. The error code for this product as described in the error code chapter. Only set if this operation has failed or is suspect.
Job.Product.Operation.Location	The location of the unit the operation is executed, only available after the operation has started. This is a number counting from 1 up till the number of units in the machine. In case of chip encoding, the location will also contain the readerboard on which the chip was programmed. The format will be *unitpos.readerboard.*

Job.Product.Operation.SubLocation	Optional. The sub location of the unit indicates in which (buffered) location within the unit the product was processed. For example Parallel Engrave Side A or Side B. Or for an Encode unit, location 1..4
Job.Product.Operation.FullLocation	The combination of location and sub location concatenated as " `Location. SubPosition` " Example #1: a product which was encoded on encode2 (e.g. unit position 6) in slot 3, this field would contain the text "6.3" Example #2: a product which engraved on parallel engrave 2 (e.g. unit position 18) on the B side of the rotating holder, this field would contain the text "18.B"
Job.Product.Operation.ChipResultData	Optional. The customer dependent chip result information communicated via the chip result file will be communicated back through this xml element. Only present on operation ContactEncode or ContactLessEncode.

```
<?xml version="1.0" encoding="utf-8"?>
<Job xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://
www.w3.org/2001/XMLSchema">
  <Name>Test</Name>
  <RecipeName>recipe</RecipeName>
  <Status>Finished</Status>
  <StartTime xsi:nil="true" />
  <EndTime xsi:nil="true" />
  <Product>
    <JobDefinedNumber>0</JobDefinedNumber>
    <ErrorCode />
    <Status>OK</Status>
    <StartTime>2020-04-17T17:04:02.0229955+02:00</StartTime>
    <EndTime>2020-04-17T17:04:50.8781769+02:00</EndTime>
    <Operation>
      <Name>Engrave</Name>
      <Status>Done</Status>
      <StartTime>2020-04-17T17:04:22.1471817+02:00</StartTime>
      <EndTime>2020-04-17T17:04:22.1522129+02:00</EndTime>
      <Location>10</Location>
      <SubLocation>A</SubLocation>
      <ChipResultData>customer dependent...</ChipResultData>
     </Operation>
  </Product>
</Job>
```

Data File Structures

The system uses separate data files for each operation (e.g. ImagePerf, NumberPerf, Laser-Engraving, etc). The exact naming and content of these files are described in next chapter. Thus, depending on the configuration of the machine the required data files can be different. The data structures and formats, which the system(s) uses for their operations, are given separately in an overview below.

Laser engraving, document data personalisation

The personalised data for laser engraving will meet the following specifications:

Parameter	**Value**
Format	**UNICODE** format (UCS-2 Little Endian) 2 bytes per character, files starts with Unicode file specifier 0xFFFE
Filename	<data file reference identification>_<postfix>.<extension> e.g. A1234567_LET.txt
Postfix filename	LET (Laser Engraved Text)
Extension	Txt

Contents	Personalised data like surname, given name, place of birth, etc by means of <data element>=<content> defined in section LET_Data. Quotes are not engraved and no length checking will be performed! E.g.
	[LET_Data]
	state = "UTO"
	nationality = "Utopian"
	surname = "IAI"
	given_names = "IAI"
	date_of_birth = "680513"
	place_of_birth = "ANYWHERE"
	address = "De Run"
	residence = "VILLAGE"
	authority = "You"
	date_of_issue = "040909"
	date_of_expiry = "090909"
	sex = "M"
	length = "1,78 m"
	personal_no = "12345682"
	MRZ1 = "P<UTOVON<UTOPIA<<UTONIS<<<<<<<<<<<<<<<<<<<<"
	MRZ2 = "AB123456<4UTO5711185M1210222123456789<<<<<70"
	MRZ3 = "<<<<" (only for id cards)

Table 3: Specification laser-engraved personalised data

Laser Engraving, document image personalisation

There can be multiple images depending on the layout. The document image data for laser-engraving will meet the following specifications:

Parameter	**Value**
Format	**JPEG** format: • Compression quality 80
Postfix filename	LEI1-LEI9 (Laser Engraving Image 1-9)
Filename	< data file reference identification >_<postfix>.<extension>e.g. A1234567_LEI1.jpg
Extension	Jpg
Resolution	To be defined by IAI
Ratio	To be defined by IAI
Size	To be defined by IAI
Background colour of the photograph	To be defined by IAI
Contents	Document photograph

Table 4: Specification document photograph data for laser-engraving

ImagePerf, document photograph personalisation

The document photograph data for ImagePerf will meet the following specifications:

Param	**Value**
Format	**JPEG** format: • Compression quality 80
Postfix filename	IP (ImagePerf)
Filename	< data file reference identification >_<postfix>.<extension> e.g. A1234567_IP.jpg
Extension	Jpg
Resolution	300 x 400 pixels
Ratio	3:4
Size	Approximately 16 Kbytes
Background colour of photograph	Uni-coloured, max. 18% grey
Contents	Document photograph

Table 9: Specification document photograph data for ImagePerf

Tilted Laser Number

The document data for Tilted Laser Number will meet the following specifications:

Parameter	**Value**
Format	**ASCII** format
Postfix filename	TLN
Filename	<data file reference identification>_<postfix>.<extension> e.g. A1234567_TLN.txt
Postfix filename	TLN
Extension	Txt
Contents	Date of birth in format: mmyyyy. E.g. 061962

Table 10: Specification TLN Data

NumberPerf, document number personalisation

The document number data for NumberPerf will meet the following specifications:

Parameter	**Value**
Format	**ASCII** format
Postfix	NP (NumberPerf)
Filename	< data file reference identification >_<postfix>.<extension> e.g. A1234567_NP.txt
Extension	Txt
Size	Document number length.
Contents	Document number.

Table 11: Specification NumberPerf data

Inkjet-printing, document special notes personalisation

The Special Notes data for inkjet-printing will meet the following specifications:

Parameter	**Value**
Format	**UNICODE** format (UCS-2 Little Endian) 2 bytes per character, files starts with Unicode file specifier 0xFFFE
Postfix	INK (Inkjet)
Filename	< data file reference identification >_<postfix>.<extension>e.g. A1234567_INK.txt
Extension	Txt
Minimum length	0 bytes
Contents	<Text><LF> All lines must end with a Linefeed character. The maximum number of lines is to be defined by IAI.
<text>	Special Notes. E.g. The bearer was registered at birth as: John Smith

Table 12: Specification inkjet-printed Special Notes data

Label printing, document label text personalisation

The Label text data for label-printing will meet the following specifications:

Parameter	**Value**
Format	**UNICODE** format (UCS-2 Little Endian) 2 bytes per character, files starts with Unicode file specifier 0xFFFE
Filename	< data file reference identification >_<postfix>.<extension> e.g. A1234567_LAB.txt
Postfix filename	LAB (Label Text)
Extension	Txt

Contents	Personalised data by means of <data element>=<content> defined in section LAB_Data. Quotes are not printed! E.g.
	[LAB_Data]
	Personal_no = "1234567"
	surname = "IAI"
	given_names = "IAI"
	authority = "You"
	personal_no = "12345682"

Supported characters	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890 /+-=*_<>"'` ?|()&^%$#@!~,.:; ÀÁÂÃÄÅĂĀĄ ÆÇĆĈČĊĐĎÈÉ ÊËĚĖĒĘĔĜĞĠ ĢĦĤĨİĪĮĬÌÍ ÎÏĴĶŁĹĽĻĿÑ ŃŇŅŊÒÓÔÕÖØ ŐŌŎŔŘŖŞŠŜŚ ŦŤŢŨŬŰŮŪŲÙ ÚÛÜŴŸŶŹŽŻŒ ĲÞ àáâãäåăāą æçćĉčċđďèé êëěėēęĕĝğġ ģħĥĩıīįĭìí îïĵķłĺľļŀñ ńňņŋòóôõöø őōŏŕřŗşŝšś ŧťţũŭűůūųù úûüŵÿŷźžżœ ĳß

Machine generated values **BookMasterOne only**	Some values cannot be generated in advance by the host software, because they will only be known by the machine at production time. The following: *fieldName* = *%BOOKNUMBER%* with booknumber *123* will produce: fieldName = 123 The %BOOKNUMBER% will be replaced by the value known at by the machine at production time. Multiple replacements, references to other fields and fixed texts can be combined: *randomTextProducedByHostSystem* = RandomText *generatedFields* = %BOOKNUMBER% fixedtext *%randomTextProducedByHostSystem%* with booknumber *123* will produce: *randomTextProducedByHostSystem* = RandomText *generatedFields* = 123fixedtextRandomText **Available replacements by machine** • BookNumber - the number read from the physical document • Status - the status of the document (OK, Reject, Rework, Suspect) • ErrorCode - the error code as defined in the chapter Error Codes in the form MajorErrorCode.MinorErrorCode • ErrorText - a detailed error description • ShortErrorText1 - first line of a abbreviated error description format • ShortErrorText2 - first line of a abbreviated error description format • JobName - name of the job this document belongs to • CreationDate - date the product is entered in the machine • Unit_Number - machine identifier • Day_Of_Month - day of the month (fixed to 2 digits) • Personalized_Product_Number - day counter of products excluding rework products

Table 13: Specification label printing personalised data

Colour Inkjet printing, document data personalisation

The personalised data for colour inkjet printing will meet the following specifications:

Parameter	**Value**
Format	**UNICODE** format (UCS-2 Little Endian) 2 bytes per character, files starts with Unicode file specifier 0xFFFE
Filename	< data file reference identification >_<postfix>.<extension> e.g. A1234567_CINK.txt
Postfix filename	CINK (Colour INKjet)
Extension	txt

Contents	Personalised data like surname, given name, place of birth, etc by means of <data element>=<content> defined in section CINK_Data. Quotes are not engraved and no length checking will be performed!
	Images are added by filename. A '*' can be used in the filename to refer to the product number, the software will replace the * with the product number of the book.
	
	E.g. (passport systems that deploy Colour Inkjet on the data page).
	[CINK_Data]
	state = "UTO"
	nationality = "Utopian"
	photo = "*_CINK1.JPG"
	signature = "*_CINK2.JPG"
	surname = "IAI"
	given_names = "IAI"
	date_of_birth = "680513"
	place_of_birth = "ANYWHERE"
	address = "De Run"
	residence = "VILLAGE"
	authority = "You"
	date_of_issue = "040909"
	date_of_expiry = "090909"
	sex = "M"
	length = "1,78 m"
	personal_no = "12345682"
	MRZ1 = "P<UTOVON<UTOPIA<<UTONIS<<<<<<<<<<<<<<<<<<<<<<<"
	MRZ2 = "AB123456<4UTO5711185M1210222123456789<<<<<70"
	
	E.g. (passport systems that deploy Colour Inkjet on page 3)
	[CINK_Data]
	height = 1.72 m
	eye_color = blue
	residence = Utopia City

Table 14: Specification printing personalised data

Document Colour Inkjet printing, document photograph personalisation

In case of passport systems that deploy colour inkjet on the page 3 this data is optional. The document photograph data for colour inkjet printing will meet the following specifications:

Parameter	**Value**
Format	**JPEG** format: • Compression quality 80 • Colour image
Postfix filename	CINK1 (Colour INKjet)
Filename	< data file reference identification >_<postfix>.<extension>e.g. A1234567_CINK1.jpg
Extension	Jpg
Resolution	720 DPI
Ratio	To be defined
Contents	Document photograph

Table 15: Specification document photograph data for printing

Colour Inkjet printing, document signature personalisation

In case of passport systems that deploy colour inkjet on the page 3 this data is optional. The document data for inkjet printing will meet the following specifications:

Parameter	**Value**
Format	**JPEG** format: • Compression quality 80 • Black and white image
Postfix filename	CINK2 (Color INKjet)
Filename	< data file reference identification >_<postfix>.<extension>e.g. A1234567_CINK2.jpg
Extension	jpg
Resolution	720DPI
Ratio	To be defined
Contents	Document signature

Table 16: Specification document signature data for printing

Visual Inspection, document MRZ information

MRZ information is used for visual inspection of laser engraved or colour inkjet printed MRZ data and meets with the following specification:

Parameter	**Value**
Format	**ASCII** format
Postfix filename	MRZ (Machine Readable Zone)
Filename	< data file reference identification >_<postfix>.<extension> e.g. A1234567_MRZ.txt
Extension	Txt
Size	Passport: 88 bytes, bytes 1 to 44 containing the upper MRZ line, bytes 45 to 88 containing the lower MRZ line. The file contains no LF. ID: 90 bytes, bytes 1 to 30 containing upper MRZ line, bytes 31 to 60, containing middle MRZ line, bytes 61 to 90, containing lower MRZ line.
Contents	MRZ-data E.g. (passports) P<UTOVON<UTOPIA<<UTONIS<<<<<<<<<<<<<<<<<<<<< AB123456<4UTO5711185M1210222123456789<<<<<70 E.g. (ID) I<UTOS1234567<2A0000000<<<<<<<7 805134M090814OUTO<<<<<<<<<<<4 VAN<DEN<AKKEREN<<JOHANNES<<<<<

Table 17: Specification MRZ data

Electronic chip inspection, document MRZ information

ICAO Data Group 1 information is used for electronic chip inspection of personalised documents and meets with the following specification:

Parameter	**Value**
Data Group number	DG 1
Format	**BINARY**-format
Postfix	DG1
Filename	< data file reference identification >_<postfix>.<extension> e.g. A1234567_DG1.bin
Extension	Bin
Contents	MRZ data

Table 18: Specification microchip programmable Data Group 1

Magstripe

The personalised data for magstripe will meet the following specifications:

Parameter	Value
Format	**UNICODE** format (UCS-2 Little Endian) 2 bytes per character, files starts with Unicode file specifier 0xFFFE
Filename	<data file reference identification>_<postfix>.<extension> e.g. A1234567_magstripe.txt
Postfix filename	magstripe
Extension	Txt

Contents	According to ISO/IEC 7811-6. Content does not include sentinel characters, and is enclosed in double quotes (“) or single quotes (‘). No text after the closing quote.
	[Data]
	Track1 = "ABCDEF"
	Track2 = "0123456789"
	Track3 = "0123456789"
	All tracks are optional, in case a track is omitted from the personalised data, this track will be untouched on the card.
	Tracks can be cleared by including the track in the personalised data, but defining it empty, the next example will clear all data from the card.
	[Data]
	Track1 = ""
	Track2 = ""
	Track3 = ""

Table 19: Specification magstripe personalized data

Error codes

The personalisation system knows the following global reject codes:

Global reject code	Name	Meaning
0	UNKNOWN_ERROR	An error has been generated where no other error code is defined for, the detailled error code will always be 0.
1	HOST_ERROR	Errors that occur when reading or writing files on the host
2	INSPECTION_ERROR	Process verification errors (vision and encoding)
3	SYSTEM_ERROR1	System failure errors during processing a document
4	SYSTEM_ERROR2	System failure errors during processing a document
5	MEASUREMENT_ERROR	Measurement error (fiducial, barcode, etc)
6	DOC_MISSING_ERROR	Document is manually removed from the system

Table 20: Global reject codes

Detailed reject	Name	Meaning
1	HOST_FILE_MISSING	Host file was removed after job loading and validating or bad or no network connection
2	HOST_FILE_ERROR	Host file reading error due to incorrect file content or bad network connection
4	HOST_FILE_CREATE	Host file creation error due to bad or no connection. This code can only be regained in the system loggings.
8	NP_MRZ_MISMATCH	Host file NP and MRZ have different document number
16	NUMBER_JOB_LOOKUP	Number is not available in the job file
32	Reserved for future use	

Table 21: Host error codes

Detailed reject code	Name	Meaning
1	NUMBER_ERROR	Pre-applied number could not be read by the number reader
2	FIDUCIAL_ERROR	Fiducial marker on document could not be measured by the vision system

Table 22: Measurement error codes

Detailed reject code	Name	Meaning
1	INSPECTION_FAIL_IP	Vision inspection failure on ImagePerf (holder page)
2	INSPECTION_FAIL_NP	Vision inspection failure on NumberPerf
4	INSPECTION_FAIL_PHOTOENGRAVE	Vision inspection failure on engraved Photo (holder page)
8	INSPECTION_FAIL_MRZENGRAVE	Vision inspection failure on engraved MRZ (holder page)
16	INSPECTION_FAIL_PRINTL	Vision inspection failure on Notes Printing
32	INSPECTION_FAIL_TEXT	Vision inspection failure on Text
64	INSPECTION_FAIL_TLN	Vision inspection failure on TLN
128	INSPECTION_FAIL_CHIP	Chip inspection (DG1) failure
256	INSPECTION_FAIL_MRZCINKJET	Vision inspection failure on printed MRZ
512	INSPECTION_FAIL_PHOTOCINKJET	Vision inspection failure on printed photo
1024	INSPECTION_FAIL_MRZ	
2048	INSPECTION_FAIL_FEATURE	Vision inspection failure on Feature Image
4096	INSPECTION_FAIL_SIGNATURE	Vision inspection failure on Signature Image

8192	INSPECTION_FAIL_BACKTEXT	Vision inspection failure on backside engraving text
16384	INSPECTION_FAIL_BACKCHIP	Chip inspection (DG1) failure
32768	INSPECTION_FAIL_DATA	Failure of inspection of crosscheck between printed/engraved data and the MRZ
65536	Reserved for future use	
131072	INSPECTION_FAIL_UV	Vision inspection failure of UV print
262144	INSPECTION_FAIL_PHOTO	
524288	INSPECTION_FAIL_SHAPE	
1048576	INSPECTION_FAIL_FEATUREPRESENCE	
2097152	INSPECTION_FAIL_BACKFEATUREPRESENCE	
4194304	INSPECTION_FAIL_FEATUREMATCH	
8388608	INSPECTION_FAIL_BACKFEATUREMATCH	
16777216	INSPECTION_FAIL_EXTERNAL	
33554432	INSPECTION_FAIL_MATCH	
67108864	INSPECTION_FAIL_FLOW	

Table 23: Inspection Errors

Detailed reject code	**Name**	**Meaning**
1	IP_CREATION	System failure preparing data for ImagePerf

2	NP_CREATION	System failure preparing data for NumberPerf
4	PHOTO_CREATION	System failure preparing data for Photo Engraving
8	DATA_CREATION	System failure preparing data for Data/ MRZ Engraving
16	PRINT_CREATION	System failure preparing data for Notes Printing
32	VISION_CREATION	System failure preparing data for Vision Inspection
64	LABEL_CREATION	System failure preparing data for Label Printing
128	FATAL_ERROR	System failure status of document is indefinite
256	ENCODING_CREATION	System failure, preparing data for encoding chip
512	ENCODING_TIMEOUT	System failure, timeout encoding chip
1024	ML_CREATION	System failure preparing data for MLI Engraving
2048	Reserved for future use	
4096	CINK_CREATION	System failure preparing data for Colour Inkjet

Table 24: System Errors 1

Detailed reject code	**Detailed reject code hexadecimal notation**	**Name**	**Meaning**
1	0x000000000001	CHIP_ENCODING_PROCESS_ERROR	Chip encoding station, processing error

2	0x000000000002	MACHINE_PROCESS_ERROR	A reason why the transport system rejects a product
4	0x000000000004	LASER_ENGRAVING_1_PROCESS_ERROR	Laser engraving station 1, processing error
8	0x000000000008	LASER_ENGRAVING_2_PROCESS_ERROR	Laser engraving station 2, processing error
16	0x000000000010	LASER_ENGRAVING_3_PROCESS_ERROR	Laser engraving station 3, processing error
32	0x000000000020	LASER_ENGRAVING_4_PROCESS_ERROR	Laser engraving station 4, processing error
64	0x000000000040	IMAGEPERF_1_PROCESS_ERROR	ImagePerf station 1, processing error
128	0x000000000080	NUMBERPERF_PROCESS_ERROR	NumberPerf station, processing error
256	0x000000000100	IMAGEPERF_2_PROCESS_ERROR	ImagePerf station 2, processing error
512	0x000000000200	CINKJET_PRINTING_PROCESS_ERROR	Colour Inkjet printer station, processing error
1024	0x000000000400	INKJET_PRINTING_PROCESS_ERROR	B&W Inkjet printer station, processing error In case of multiple Inkjet units on the machine, this represents first inkjet unit
2048	0x000000000800	LABEL_PRINTING_PROCESS_ERROR	Label printer station, processing error
4096	0x000000001000	LASER_ENGRAVING_5_PROCESS_ERROR	Laser engraving station 5, processing error
8192	0x000000002000	LASER_ENGRAVING_6_PROCESS_ERROR	Laser engraving station 6, processing error
16384	0x000000004000	ENCODING_READER_1_PROCESS _ERROR	Reader 1

32768	0x000000008000	ENCODING_READER_2_PROCESS _ERROR	Reader 2
65536	0x000000010000	ENCODING_READER_3_PROCESS _ERROR	Reader 3
131072	0x000000020000	ENCODING_READER_4_PROCESS _ERROR	Reader 4
262144	0x000000040000	ENCODING_READER_5_PROCESS _ERROR	Reader 5
524288	0x000000080000	ENCODING_READER_6_PROCESS _ERROR	Reader 6
1048576	0x000000100000	ENCODING_READER_7_PROCESS _ERROR	Reader 7
2097152	0x000000200000	ENCODING_READER_8_PROCESS _ERROR	Reader 8
4194304	0x000000400000	ENCODING_READER_9_PROCESS _ERROR	Reader 9
8388608	0x000000800000	ENCODING_READER_10_PROCESS _ERROR	Reader 10
16777216	0x000001000000	ENCODING_READER_11_PROCESS _ERROR	Reader 11
33554432	0x000002000000	ENCODING_READER_12_PROCESS _ERROR	Reader 12
67108864	0x000004000000	ENCODING_READER_13_PROCESS _ERROR	Reader 13
134217728	0x000008000000	ENCODING_READER_14_PROCESS _ERROR	Reader 14
268435456	0x000010000000	ENCODING_READER_15_PROCESS _ERROR	Reader 15
536870912	0x000020000000	ENCODING_READER_16_PROCESS _ERROR	Reader 16

1073741824	0x000040000000	MAGSTRIPE_PROCESS_ERROR	Magstripe processing error
2147483648	0x000080000000	LEFI1_PROCESS_ERROR	LEFI engraving station 1, processing error
4294967296	0x000100000000	LEFI2_PROCESS_ERROR	LEFI engraving station 2, processing error
8589934592	0x000200000000	BACK_ENGRAVING_1_PROCESS_ERROR	Laser engraving station 1, back side, processing error Note: available from BMO version 2.21.x and higher
17179869184	0x000400000000	BACK_ENGRAVING_2_PROCESS_ERROR	Laser engraving station 2, back side, processing error Note: available from BMO version 2.21.x and higher
34359738368	0x000800000000	INKJET2_PRINTING_PROCESS_ERROR	B&W Inkjet printer 2 station (will only occur in case of multiple inkjet units), processing error
68719476736	0x001000000000	HIGH_ACCURACY_LASER_1_PROCESS_ERROR	High Accuracy Laser station 1 processing error
137438953472	0x002000000000	HID_ELEMENT	HID Element error
274877906944	0x004000000000	HIGH_ACCURACY_LASER_2_PROCESS_ERROR	High Accuracy Laser station 2 processing error
549755813888	0x008000000000	HIGH_ACCURACY_LASER_3_PROCESS_ERROR	High Accuracy Laser station 3 processing error
1099511627776	0x010000000000	CHIP_UHF_ENCODING_PROCESS_ERROR	Chip encoding station UHF, processing error
2199023255552	0x020000000000	SURFACE_ENGRAVE_1_PROCESS_ERROR	Surface Engrave station 1 processing error

Table 25: System Errors 2

Detailed reject code	Name	Meaning
1	MISSING_AT_STARTUP	Document present, according to system administration is not detected by the document sensor at system start-up.

2	MISSING_BY_OPERATOR	While system is in standby, the operator removes a document and confirms removal to the system administration.

Table 25: Document missing errors

As an example when a document has the following extension A1234567_STS.REJ2_3, this means that the Inspection unit has detected an error on the ImagePerf and NumberPerf.

Appendix A: XSD Job data format

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="JobData" nillable="true" type="JobData" />
  <xs:complexType name="JobData">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:choice>
        <xs:element minOccurs="0" maxOccurs="1" name="Description" type="xs:string" /
>
        <xs:element minOccurs="1" maxOccurs="1" name="RecipeName" nillable="true"
type="xs:string" />
        <xs:element minOccurs="1" maxOccurs="1" name="UseNumberReading" type="xs:bool
ean" />
        <xs:element minOccurs="0" maxOccurs="1" name="NumberReadingType" type="Number
ReadingType" />
        <xs:element minOccurs="0" maxOccurs="1" name="NumberOfProducts" type="xs:int"
/>
        <xs:element minOccurs="1" maxOccurs="1" name="JobScenario" type="PersonalFile
sReferenceType" />
        <xs:element minOccurs="0" maxOccurs="1" name="RunType" type="RunType" />
        <xs:element minOccurs="0" maxOccurs="unbounded" name="ProductNumbers" type="P
roductNumberInfo" />
        <xs:element minOccurs="0" maxOccurs="1" name="ReadNumbersInFixedOrder" type="
xs:boolean" />
        <xs:element minOccurs="0" maxOccurs="1" name="DeleteJobFile" type="xs:boolean
" />
        <xs:element minOccurs="0" maxOccurs="1" name="DeleteHostDataOnOutput"
nillable="true" type="xs:boolean" />
        <xs:element minOccurs="0" maxOccurs="1" name="DataLocation" type="xs:string"
/>
        <xs:element minOccurs="0" maxOccurs="1" name="AutoQueue" type="xs:boolean" />
        <xs:element minOccurs="0" maxOccurs="1" name="Priority" type="xs:int" />
        <xs:element minOccurs="0" maxOccurs="1" name="Inputs" type="ArrayOfJobDataTra
ySelection" />
        <xs:element minOccurs="0" maxOccurs="1" name="Outputs" type="ArrayOfJobDataTr
aySelection1" />
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="NumberReadingType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Barcode" />
      <xs:enumeration value="Chip" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="PersonalFilesReferenceType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="SequenceNumber" />
      <xs:enumeration value="JobDefinedNumber" />
      <xs:enumeration value="ReadNumber" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="RunType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Normal" />
      <xs:enumeration value="Test" />
      <xs:enumeration value="Calibrate" />
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="ProductNumberInfo">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="Status" type="xs:int" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="ArrayOfJobDataTraySelection">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Input" nillable="true"
type="JobDataTraySelection" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="JobDataTraySelection">
    <xs:attribute name="Index" type="xs:int" use="required" />
  </xs:complexType>
  <xs:complexType name="ArrayOfJobDataTraySelection1">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Output" nillable="true"
type="JobDataTraySelection" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Appendix B: XSD Job data CMO / Element format

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Job" nillable="true" type="JobInfo" />
  <xs:complexType name="JobInfo">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Recipe" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="Inputs" type="ArrayOfCassette" />
      <xs:element minOccurs="0" maxOccurs="1" name="Outputs" type="ArrayOfCassette" /
>
      <xs:element minOccurs="0" maxOccurs="1" name="Delete" type="xs:boolean" />
      <xs:element minOccurs="0" maxOccurs="1" name="JobDataFolder" type="xs:string" /
>
      <xs:element minOccurs="0" maxOccurs="1" name="SuspectDestination" type="xs:unsi
gnedByte" />
      <xs:element minOccurs="0" maxOccurs="1" name="ReworkDestination" type="xs:unsig
nedByte" />
      <xs:element minOccurs="0" maxOccurs="1" name="RejectDestination" type="xs:unsig
nedByte" />
      <xs:element minOccurs="0" maxOccurs="1" name="ReadNumber" type="xs:boolean" />
      <xs:element minOccurs="0" maxOccurs="1" name="NumberReadingType" type="NumberRe
adingType" />
      <xs:element minOccurs="0" maxOccurs="1" name="ReadNumberAsProductNumber" type="
xs:boolean" />
      <xs:element minOccurs="0" maxOccurs="1" name="Products" type="ArrayOfProductInf
o" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ArrayOfCassette">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Cassette" nillable="true"
 type="Cassette" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Cassette">
    <xs:attribute name="Index" type="xs:unsignedByte" use="required" />
    <xs:attribute name="Id" type="xs:string" />
  </xs:complexType>
  <xs:simpleType name="NumberReadingType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="None" />
      <xs:enumeration value="Vision" />
      <xs:enumeration value="ContactChip" />
      <xs:enumeration value="ContactlessChip" />
      <xs:enumeration value="ChipUhf" />
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="ArrayOfProductInfo">
    <xs:sequence>
```

```
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Product" nillable="true" 
type="ProductInfo" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ProductInfo">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Inputs" type="ArrayOfCassette" />
      <xs:element minOccurs="0" maxOccurs="1" name="Outputs" type="ArrayOfCassette" /
>
    </xs:sequence>
    <xs:attribute name="Number" type="xs:string" />
    <xs:attribute name="Recipe" type="xs:string" />
  </xs:complexType>
</xs:schema>
```

Appendix C: XSD Job status format

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Job" nillable="true" type="Job" />
  <xs:complexType name="Job">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="RecipeName" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="MachineName" type="xs:string" />
      <xs:element minOccurs="1" maxOccurs="1" name="Status" type="JobStatus" />
      <xs:element minOccurs="1" maxOccurs="1" name="StartTime" nillable="true" type="
xs:dateTime" />
      <xs:element minOccurs="1" maxOccurs="1" name="EndTime" nillable="true" type="xs
:dateTime" />
      <xs:element minOccurs="0" maxOccurs="1" name="StartedBy" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Product" type="Product" /
>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="JobStatus">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Available" />
      <xs:enumeration value="Queued" />
      <xs:enumeration value="Active" />
      <xs:enumeration value="Finished" />
      <xs:enumeration value="Aborted" />
      <xs:enumeration value="AbortedFinished" />
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="Product">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="JobDefinedNumber" type="xs:string
" />
      <xs:element minOccurs="0" maxOccurs="1" name="DocumentNumber" type="xs:string"
/>
      <xs:element minOccurs="0" maxOccurs="1" name="ErrorCode" type="xs:string" />
      <xs:element minOccurs="1" maxOccurs="1" name="Status" type="ProductStatus" />
      <xs:element minOccurs="1" maxOccurs="1" name="StartTime" nillable="true" type="
xs:dateTime" />
      <xs:element minOccurs="1" maxOccurs="1" name="EndTime" nillable="true" type="xs
:dateTime" />
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Operation" type="Operatio
n" />
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="ProductStatus">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Idle" />
      <xs:enumeration value="InProgress" />
```

```
        <xs:enumeration value="OK" />
        <xs:enumeration value="Suspect" />
        <xs:enumeration value="Rework" />
        <xs:enumeration value="Reject" />
      </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="Operation">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string" />
        <xs:element minOccurs="1" maxOccurs="1" name="Status" type="OperationStatus" />
        <xs:element minOccurs="1" maxOccurs="1" name="StartTime" nillable="true" type="
  xs:dateTime" />
        <xs:element minOccurs="1" maxOccurs="1" name="EndTime" nillable="true" type="xs
  :dateTime" />
        <xs:element minOccurs="0" maxOccurs="1" name="ErrorCode" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="FullLocation" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="Location" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="SubLocation" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="ChipResultData" type="xs:string"
  />
      </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="OperationStatus">
      <xs:restriction base="xs:string">
        <xs:enumeration value="NoProcessing" />
        <xs:enumeration value="Pending" />
        <xs:enumeration value="Busy" />
        <xs:enumeration value="Done" />
        <xs:enumeration value="Skipped" />
        <xs:enumeration value="Failed" />
        <xs:enumeration value="Suspect" />
      </xs:restriction>
    </xs:simpleType>
  </xs:schema>
```