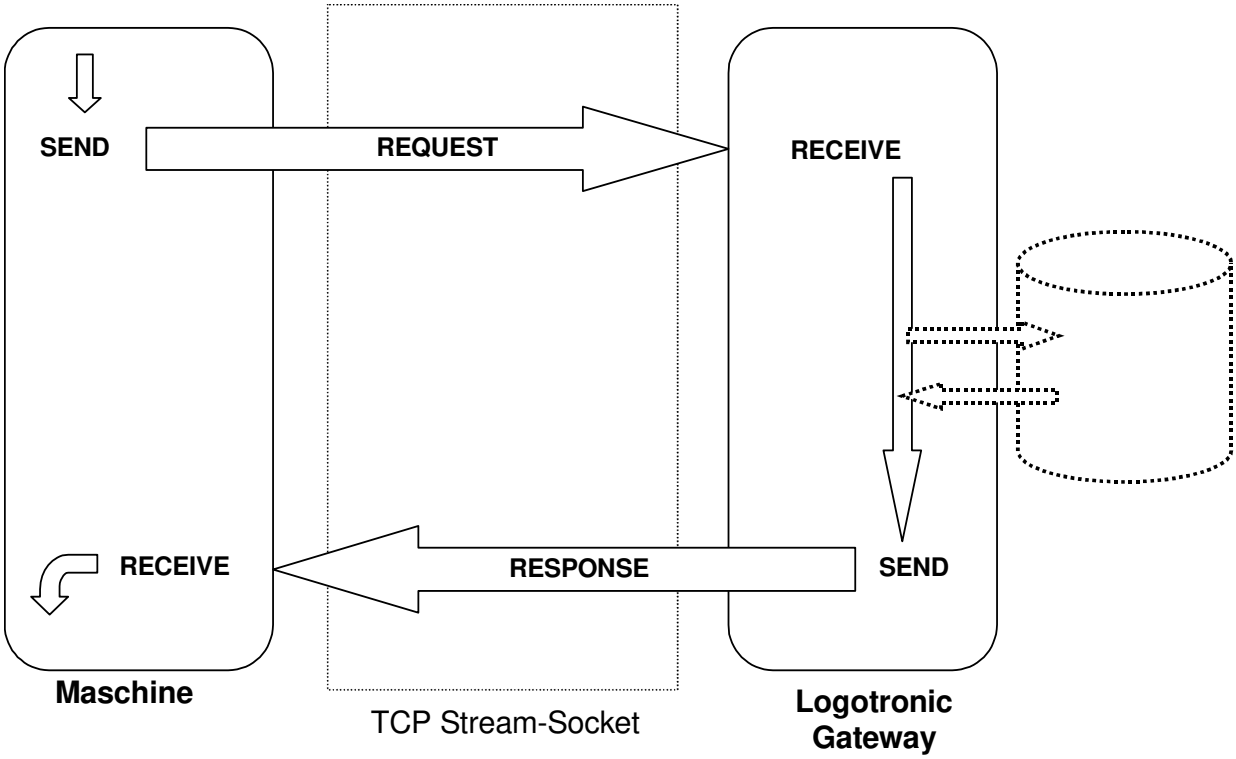
1 Introduction Interface Logotronic

Using this interface the Duran folding gluer should be able to

- receive job header data (e.g. customer name, job name, amount etc.),
- receive job preview pictures,
- receive job parameters like sheet width and length, paper thickness
- send cyclical or event triggered operational data,
    - current job and number,
    - sheet counter (gross and net),
    - operating hours,
    - machine state (idle, in production or malfunction),
    - current machine activities,
    - job state (job runs or no job is active),
    - time states (make-ready time, production, auxiliary or down time)
- save and restore repetition data for every job (in further development steps),
- provide the log in or log off of operators

2 Short overview of the Koenig&Bauer Logotronic communication protocol



The Logotronic interface bases on an Ethernet-TCP/IP-communication. The Logotronic works as server, the connecting machine is the client.

The protocol defines an Request-Response communication. The client/machine sends always requests, the Logotronic answers always with responses.

2.1 Structure of exchanged Data

The requests and responses are built of a binary frame which packs the content delivered in XML-format.
Exception of that are the initial exchanged telegrams used for instancing the communication. They contains their payload in binary format.
See *Logo_Com_Protocol_V1.3.docx.*

2.2 The fundamental binary frame

1. Binary data is transferred in network format as SHORT or LONG,
2. Strings are null-terminated. Accepted are all chars from ANSI-charset except an apostrophe,
3. All strings are sent in a buffer with a specified maximum length. Is the string shorter then this length the remaining buffer has to be filled up with NULLs

2.2.1 The request

Datafield	Type/Length	Description	Example
Version	unsigned long	Reserved (4 Bytes), always 0	0
TransactionID	unsigned long	Information for the assignment of request and response	218
WorkplaceID	char / 8		14
RequestType	unsigned long		WP_INFO
DataLength	unsigned long	Byte size of payload	1090
Request Data			
EDataLength	unsigned long	Repetition of the data fields of telegram header in reverse order	1090
ERequestType	unsigned long		WP_INFO
EWorkplaceID	char / 8		14
ETransactionID	unsigned long		218

Binary frame of an request

2.2.2 Description of data fields

Name of field	Value
Version	reserved, always 0
TransactionID/ ETransaction	a unique identifier for every telegram to assign a response to request. Its defined by the client, the logotronic rewrites this ID in its response of the regarding request
WorkplaceID/ EWorkplaceID	It's an unique six-digit ID for every workplace/machine in range of the logotronic. The byte array is 8 byte long, the last two chars are always 0.
RequestType/ ERequestType	the ID identifying a telegram

ResponseType/ EResponseType	
DataLength/ EDatalength	The length/size of payload contained by “Request Data”. The maximum allowed length is 65492 Bytes

The binary frame is closed by repetition of the opening data fields except of “Version” in reverse order. This should be used to ensure the consistency of the structure of the received telegram.

2.2.3 The response

The response sent by logotronic is built analogue to the request descriped above. The only difference is that the ResponseType/EResponseType is sent instead of RequestType/ERequestType.

Datafield	Type/Length	Description	Example
Version	unsigned long	Reserved (4 Bytes), always 0	0
TransactionID	unsigned long	Information for the assignment of request and response	218
WorkplaceID	char / 8		14
ResponseType	unsigned long		WP_INFO
DataLength	unsigned long	Byte size of payload	1090
Response Data			
EDataLength	unsigned long	Repetition of the data fields of telegram header in reverse order	1090
EResponseType	unsigned long		WP_INFO
EWorkplaceID	char / 8		14
ETransactionID	unsigned long		218

Binary frame of an response

The fields ResponseType/EResponseType contains the same content as the field RequestType/ERequestType of the corresponding Request.

The first field “Version” must be preceded to each individual telegram. Because of its static content it is no longer explicitly mentioned in the following chapters.

3 Binary Requests must be used by folding gluer

3.1 Retrieving “informational responses” of the Logotronic

There are a special kind of responses for some requests which can be sent instead of an actual expected response.

There are two such responses described below. In both cases their headers contains the transaction id of the request to which this partial response regards.

Name of field	Value
TransactionID	TransactionId if the regarding request
WorkplaceID	0 or WorkPlaceId of the machine
ResponseType	RSP_INFO (254) or RSP_ERROR (255)
DataLength	4 + Length of response data

3.1.1 The informational response (RSP_INFO = 254)

Data field	Type/Length	Description	Example
InfoCode	long int	Type of information	INFO_CREATING_WP=17
ServerInfo	char / 255 + 1	Additional information as zero terminated text	Workplace is being created. This will take about 10 seconds.0x00

Possible values for InfoCode are:

Name of field	Value	Description
INFO_UNSUBSCRIBED	12	The client is not registered via WP_INFO right now.
INFO_SERVER_CLOSED	4	The logotronic is going to close the socket.
INFO_SEND_VERSION	16	The Logotronic requests information about the client version. The client must send a valid REQ_VERSIONINFO (see chapter 3.5).
INFO_CREATING_WP	17	A new client is registering, a new Workplace and WorkplaceID must be created.
INFO_REQ_NOT_SUPPORTED	24	The regarding client request (see TransactionID) is not supported/does not exists

There is no specification as how the individual client must react to the particular information type.

3.1.2 The error response (RSP_ERROR = 255)

Data field	Type/Length	Description
ServerInfo	char / 255 + 1	information about the error as zero terminated text

There is no specification as how the individual client must react to the error.

3.2 stablishing a logotronic connection

To connect and logon to the logotronic a special sequence of binary requests/responses must be sent and processed. For a detailed reference of these necessary telegrams named below see **“Logo_Com_Protocol_V1.5-v07.docx”.**

After opening a socket to the Logotronic it (the Logotronic) sends a binary connect response called “**Accept-Response**”.

Please not that these response is sent without a request before. So the responses header contains only

Name of field	Value
TransactionID	0
WorkplaceID	0
ResponseType	ACCEPT
DataLength	260

The payload looks like this:

Data field	Type/Length	Description	Example
CurrentIndex	unsigned short	The number of already connected Clients including the new connection. If the maximum is already exhausted, it contains NO_INDEX. If NO_INDEX is sent, the socket is closed immidately after sending response	8
MaxConnections	unsigned short	The maximum permitted number of connections.	30
ServerInfo	char / 255 + 1	Version information of the responding Logotronic	„1.0.2.9“

3.3 Registering a new Client / Machine (WP_SETUP)

The Logotronic identifies every single client by its WorkPlaceID. The logotronic uses this WorkPlaceID as part of the telegram header of every request and response to internally assign them to every single client.

But if the client connects for the very first time it cannot know its own WorkPlaceID. So it must retrieve its Workplace-ID by requesting it using WP_SETUP.

Otherwise the client send immediately the WP_INFO request as descriped in the next chapter.

The header of the request leaves the WorkPlaceID empty.

Datenfeld	Erklärung
TransactionID	Generated by Client
WorkplaceID	Request: empty Response: new WorkplaceID
RequestType	WP_SETUP = 1
DataLength	Request: 46 + Length oft he workplace specific data Response: 4

WP_SETUP request data:

Data field	Type/Length	Description	Example
WorkplaceName	char / 30+1	Machine name as defined inside the Logotronic (usually specified by the customer)	RA162-4
WorkplaceType	char / 10+1	Sheet offset machine (DM) Digital Sheet offset machine (DS) Roll offset machine (WEB) Folder gluer (FG) Platen die cutter (FDC) Rotation die cutter (RDC)	DM
WorkplaceDataLength	unsigned long	Length of the workplace specific data in Bytes	0
WorkplaceData	unsigned char	Workplace specific Data	

Using the WorkplaceName and WorkplaceType the Logotronic tries to determine the corresponding WorkplaceID. Because the client was never logged on before it creates a new one and sends it with the regarding response.

To find the fitting machine data the logotronic needs some seconds. To avoid that the client runs in any timeout the Logotronic sends an additional info response (InfoCode= INFO_CREATING_WP (17), see Chapter 3.1.1) before it sends the proper WP_SETUP response.

The WorkplaceID is sent in the field WorkplaceID of the response header. Apart from that the response contains only the return code:

Data field	Type/Length	Description	Example
ReturnCode	long int	Successfully created: RC_CREATED (=1), Workplace id was updated: RC_UPDATED (=2) otherwise: ERR_WRONG_NAME	RC_CREATED

If the logotronic could not find a machine of the specified WorkplaceType and with this WorkPlaceName sent with the WP_SETUP request, the return code is ERR_WRONG_NAME.

> *Be aware that the client has to save the received Workplace-ID itself. The client needs it to know next time that he has already logged in once and to mark its requests and identify the regarding responses.*

3.4 Getting workplace specific data (WP_INFO)

After the client has retrieved its WorkplaceID or, in case it already knows its WorkplaceID, after receiving the ACCEPT response (Chapter 3.2) it must request its appropriate machine information.

Data field	Description
TransactionID	Generated by Client
WorkplaceID	WorkplaceID of the requesting machine
RequestType	WP_INFO = 2
DataLength	Request: 0 Response: 46 + Length of the workplace specific data

3.4.1 The request

The WP_INFO is sent as header-only request and contains no data.

3.4.2 The response

If the Logotronic could not find the specified WorkplaceID, it sends an error-response ERR_WP_DATA_NOT_FOUND. In case of a not properly certified machine, an invalid WorkPlaceType or the internal Logotronic database connection is interrupted a regarding error response is sent.

In case of success the response contains the instant of time when the client should backup of its job repetition data.

Data field	Type/Length	Description	Example
WorkplaceName	char / 30+1	Maschinenname	RA162-4
WorkplaceType	char / 10+1	Sheet offset machine (DM) Digital Sheet offset machine (DS) Roll offset machine (WEB) Folder gluer (FG) Platen die cutter (FDC) Rotation die cutter (RDC)	DM
WorkplaceDataLength	unsigned long	Length of the worplace specific data in Bytes	2
WorkplaceData	unsigned char	**1st Byte: point of time to backup repetition data:** char ‚0': the print run is completed char ‚1': Backup on first good sheet char ‚2': No automatic backup **2nd Byte: Language Number:** 0: German 1: Englisch etc. (IDs are specified by Koenig&Bauer Radebeul)	‚1' 0

3.5 Exchanging version information (REQ_VERSIONINFO)

The control console should tell to the logotronic which protocol version it supports. So immediately after retrieving the workplace information the control console sends the request REQ_VERSIONINFO (=253) to the Logotronic.

Data field	Description
TransactionID	Generated by Client
WorkplaceID	WorkplaceID oft the requesting machine
RequestType	REQ_VERSIONINFO = 253 (in case of error try 50)
DataLength	Request: 51 Response: 55

Neither the Logotronic nor the client process this version information. That's why the particular members of of the request as well as the response have only informational character and are used only for logging for later debugging in case of communication errors.

3.5.1 The request

Data field	Type/Length	Description	Example
ProtocolVersion	char / 16+1	Version number of the supported protocol	„1.0“
ClientVersion	char / 16+1	Version number of client/machine software	„3.0.0“
ClientRevision	char / 16+1	Internal version number oft the client communication program.	„1.2.3“

3.5.2 The response

Data field	Type/Length	Description	Example
CommFrame	unsigned long	Until now always 0	0
ProtocolVersion	char / 16+1	Version number of the supported protocol	„1.0“
Logotronic	char / 16+1	Version number Logotronic	„1.0.3.1“
ServerRevision	char / 16+1	Internal revision number of Logotronic TCP/IP Gateway	„1.0.2.11“

3.6 Synchronizing the client time with the time of Logotronic (REQ_TIME)

Especially for the production data acquisition it is important that client time stamp of its sent data equals to the system time of the Logotronic. For that the Logotronic serves as time server and the client/machine requests after every login the current time from the Logotronic to synchronize itself to the LogoTronic.

The appropriate request REQ_TIME(=252) is sent after getting the REQ_VERSIONINFO response by the client.

Data field	Description
TransactionID	Generated by Client
WorkplaceID	WorkplaceID oft the requesting machine
RequestType	REQ_TIME = 252
DataLength	Reqest: 0 Response: 6

3.6.1 The request

The REQ_TIME is a header-only request.

3.6.2 The response

Data field	Type/Length	Description	Example
TimeStamp	unsigned long	Current time (UTC-based in UNIX-Format: seconds from 1st January 1970 00:00)	
SummerTime	unsigned short	0 = “common” time 1 = Summer time	

4 Requests for the production data acquisition

4.1 Introduction

For a proper production acquisition the client machine sends its production data cyclically (the Rapidas do that every 60 seconds) as well as on particular events e.g. a job was “loaded”, started or has finished.

So the client machine needs to specify which order/partial order (production)/job is currently processed. For that in the HMI the machine operator must be able to select a single job from the list deployed by the Logotronic. As soon as the operator performs activities which belong to a particular job, the client machine send the regarding job id together with its production data telegrams.

In case of the Rapidas there is no situation where no order is loaded in the machine except directly after commissioning. That’s why the print run id is always sent with every production data telegram. According to the job state (loaded, running, finished, interrupted), time data specification and the current activity the Logotronic is able to assign the sent production data to special time categories e.g. production time, downtime and so on.

For the implementation of the production data acquisition several telegram types are necessary: telegrams to retrieve jobs, which has to be done (JOBLIST, KBA_JOBLIST, ORDER_NOTE, ORDER_JOBINFO, KBA_JOBINFO) and the OPERATIONAL_DATA-telegram. One of the transmitted jobs must be defined as “Currently active job” which is currently processed in the machine. So a job id of a currently processed job can be transmitted with each OPERATIONAL_DATA.

4.2 The structure type INFO_PACK

With some of the following telegrams for retrieving job data a special structure INFO_PACK is used. This INFO_PACKs allow the transmitting of lists of data of the same content for several jobs.

The structure of an InfoPack of type INFO_PACK looks as follows:

Data field	Type/Length	Description	Example
InfoPackType	INFOPACK_TYPE	Bit mask for the desired and returned data	IPK_KUNDENNAME
InfoPackLength	long int	Byte size of data contained by this “InfoPack”	20
InfoPackData	unsigned char	Data (null-terminated and null-separated)	Bavaria

Inside the Logotronic there are no limitations for the size of the InfoPackData member. Its length depends on which InfoPackType is specified. For instance until now the IPK_AUFTRNR got 80 chars in maximum so the necessary size of InfoPackData would be 81 chars including the terminating NULL. The client can limit its receiving buffer as it needs.

4.3 Requesting the joblist (JOBLIST)

For the production data acquisition the Logotronic needs to know which job is currently loaded or printed by the machine. The purpose of that knowledge is the ability to assign the production data to the proper job and to categorize the data properly. For the later selection of a particular job the JobList-request (RequestType=3) must be sent.

Logotronic provides the current planning status of the order list. A flag is required for a more precise specification of the desired result. This is formed as follows (if necessary as OR-link of several specifications)

The planned operations can be requested for a certain machine (flag JLF_WORKPLACE) or for all machines of the same type (JLF_WORKPLACEGROUP). If a request is made for all machines of the same type, the *WhichWorkplace* entry serves as a reference machine type, that is, its type affiliation is evaluated.

Optional: If no data fields are specified in the flag, the Logotronic provides always InfoPacks of the type 0x010F 3B77.

Data field	Description
TransactionID	Generated by Client
WorkplaceID	WorkplaceID oft the requesting machine
RequestType	JOBLIST = 3
DataLength	Request: 15 Response: 11 + regarding to number of InfoPacks and the size of each of them

4.3.1 The request

Data field	Type/Length	Description	Example
Flags	long int	Specifies the range of the requested job list and the containing data: Jobs for requesting machine (WorkPlaceId)/all jobs for the same type of the requesting machine + Flags specifing the desired data	JLF_WORKPLACE
NumberOfJobs	long int	Maximum number of jobs to transmit	20
WhichWorkplace	char / 6+1	WorkPlaceIdID of machine, for which or for their type the planned joblist is requested	14

In “Flags” you can specify several data in an InfoPack by using an OR operator (next chapter).

4.3.2 Available types of InfoPacks

InfoPack-Typ	Define and Meaning
IPK_AUFTRNR	0x0000 0001: Order number
IPK_PRODNR	0x0000 0002: Production number (Partial order number)
IPK_AGNR	0x0000 0004: Job number (Printing pass number)
IPK_AUFTRBEZ	0x0000 0010: Order name
IPK_PRODBEZ	0x0000 0020: Production name
IPK_AGBEZ	0x0000 0040: Job name
IPK_KUNDENNAME	0x0000 0100: Customer name
IPK_MENGE	0x0000 0200: Amount/Copies (job related)
IPK_MATERIAL	0x0000 0400: Paper
IPK_STATUS01	0x0000 0800: Job status Available states of a job: „0“: started „1“: not started „2“: interrupted
IPK_BREITE	0x0000 1000: Sheet width[mm]
IPK_LAENGE	0x0000 2000: Sheet width [mm]

InfoPack-Typ	Define and Meaning
IPK_BEARBZP	0x0001 0000: planned production time [industrial-h]
IPK_RUESTZP	0x0002 0000: planned make-ready time [industrial-h]
IPK_FBFDATUM	0x0004 0000: earliest planned start date, format: YYYYMMDD
IPK_FESDATUM	0x0008 0000: latest planned finishing date, format: YYYYMMDD
IPK_FBFZEIT	0x0010 0000: earliest planned start time, format: HHMM
IPK_APNR	0x0100 0000: WorkPlaceId of planned machine of the certain job
IPK_KUNDENNR	0x0200 0000: Customer number
IPK_MATERIALNR	0x0400 0000: Material number of printing material

4.3.3 The response

Data field	Type/Length	Description	Example
ReturnCode	long int	Number of transmitted InfoPacks in the result set of response	12
WhichWorkplace	char / 6+1	WorkplaceId of machine the result setis related to	14
InfoPack[0]	INFO_PACK	The requested data of planned jobs according to the specified flags as InfoPack	...
InfoPack[1]	"		
…			

The data is transferred in an InfoPack in which the single data is separated from each other by a binary zero (sequence as in the list above, ascending, according to the value of the InfoPack)

4.4 Auftragsinfo anfordern (ORDER_JOBINFO)

This telegram allows the request of detailed data for a single job.

Datenfeld	Erklärung
TransactionID	Generated by Client
WorkplaceID	WorkplaceID oft the requesting machine
RequestType	ORDER_JOBINFO = 5
DataLength	Req.: 37 Rsp.: 4+ the size of returned data

4.4.1 The request

Datenfeld	*Typ/Länge*	Erklärung	Beispiel
OrderNo	char / 10+1	Order number	8921021
ProdNo	char / 10+1	Production run number	1
JobNo	char / 10+1	Job number	1
InfoPackType	INFOPACK_TYPE	Bit mask of the desired data	IPK_KUNDENNAME

4.4.2 The response

Datenfeld	*Typ/Länge*	Erklärung	Beispiel
ReturnCode	long int	Anzahl der übertragenen “InfoPacks”	3
InfoPack[0]	INFO_PACK	The requested data of of the specified job according to the InfoPackType in the request	siehe unten
InfoPack[1]	INFO_PACK		
…			

4.5 Requesting a filtered Joblist (KBA_JOBLIST)

The JOBLIST request allows to request all planned (not already finished) jobs for the requesting machine itself, for another machine or for all machines of the same type. Additionally it is possible to limit the number of the transmitted jobs. The last point is of particular importance for the folder gluer PLC.

In contrast, KBA_JOBLIST allows the result set to be filtered by job, part job or print run numbers. The result also includes jobs that have already been completed.

In this request it does not matter if flags contains JLF_WORKPLACE or JLF_WORKPLACEGROUP. Instead in any case an order number must be specified to retrieve all production run numbers related to this order.

If an order number is specified as well as an production run number the according jobs will be returned.

If an production run contains jobs first an InfoPack with the data of the related production run is transmitted. Then it follows the InfoPacks for all jobs belonging to thiat production run.

Wildcards as * or ? will be processed.

Data field	Description
TransactionID	Generated by Client
WorkplaceID	WorkplaceID oft the requesting machine
RequestType	KBA_JOBLIST = 60
DataLength	Req.: 26 Rsp.: 4 + according the number of InfoPacks and the size of each of them

The KBA_JOBLIST header

4.5.1 The request

Data field	Type/Length	Description	Example
Flags	long int	Specifies the desired data.	…

OrderNo	char / 10+1	Order number (or *)	*0815*
ProdNo	*char / 10+1*	*If applicable the production run number*	*1*

4.5.2 The response

Datenfeld	*Typ/Länge*	Erklärung	Beispiel
ReturnCode	long int	Number of transmitted InfoPacks in the result set of response	12
InfoPack[0]	INFO_PACK	The requested data of planned jobs according to the specified flags as InfoPack	...
InfoPack[1]	"		
…			

4.6 Requesting extended order info (KBA_JOBINFO)

This telegram allows the request of more detailed data for a single job. The only difference zwischen this and the ORDER_JOBINFO described in chapter 4.4 is that the two additional informations "separation numbers in production run" and "separation numbers in job" can be requested.

Datenfeld	Erklärung
TransactionID	Generated by Client
WorkplaceID	WorkplaceID oft the requesting machine
RequestType	KBA_JOBINFO = 61
DataLength	Req.: 37 Rsp.: 4+ the size of returned data

4.6.1 The request

Datenfeld	*Typ/Länge*	Erklärung	Beispiel
OrderNo	char / 10+1	Order number	8921021
ProdNo	char / 10+1	Production run number	1
JobNo	char / 10+1	Job number	1
InfoPackType	INFOPACK_TYPE	Bit mask of the desired data	IPK_KUNDENNAME

4.6.2 The response

Datenfeld	*Typ/Länge*	Erklärung	Beispiel
ReturnCode	long int	Anzahl der übertragenen "InfoPacks"	3
InfoPack[0]	INFO_PACK	The requested data of of the specified job according to the InfoPackType in the request	siehe unten
InfoPack[1]	INFO_PACK		

…			

4.6.3 Available types of InfoPacks for KBA_JOBINFO

InfoPack-Typ	**Define and Meaning**
IPK_AUFTRNR	0x0000 0001: Order number
IPK_PRODNR	0x0000 0002: Production number (Partial order number)
IPK_AGNR	0x0000 0004: Job number (Printing pass number)
IPK_AUFTRBEZ	0x0000 0010: Order name
IPK_PRODBEZ	0x0000 0020: Production name
IPK_AGBEZ	0x0000 0040: Job name
IPK_KUNDENNAME	0x0000 0100: Customer name
IPK_MENGE	0x0000 0200: Amount/Copies (job related)
IPK_MATERIAL	0x0000 0400: Paper
IPK_STATUS01	0x0000 0800: Job status Available states of a job: „0“: started „1“: not started „2“: interrupted
IPK_BREITE	0x0000 1000: Sheet width[mm]
IPK_LAENGE	0x0000 2000: Sheet width [mm]
IPK_BEARBZP	0x0001 0000: planned production time [industrial-h]
IPK_RUESTZP	0x0002 0000: planned make-ready time [industrial-h]
IPK_FBFDATUM	0x0004 0000: earliest planned start date, format: YYYYMMDD
IPK_FESDATUM	0x0008 0000: latest planned finishing date, format: YYYYMMDD
IPK_FBFZEIT	0x0010 0000: earliest planned start time, format: HHMM
IPK_APNR	0x0100 0000: WorkPlaceId of planned machine of the certain job
IPK_KUNDENNR	0x0200 0000: Customer number
IPK_MATERIALNR	0x0400 0000: Material number of printing material
IPK_PRODFZ	0x2000 0000: separation number ‚Front/Back’ of production run
IPK_JOBFZ	0x4000 0000: ggf. Farbzahl ‚Oben/Unten’ des Arbeitsgangs

4.7 Auftragsnotiz anfordern (ORDER_NOTE)

Logotronic supplies a note for a job which can be displayed at the machine when the job is loaded (character stream in the globally agreed code page).

Data field	Description
TransactionID	Generated by Client
WorkplaceID	WorkplaceID oft the requesting machine
RequestType	ORDER_NOTE = 6
DataLength	Req.: 11 Rsp.: 4 + length of transmitted text

4.7.1 The request

Datenfeld	*Typ/Länge*	Erklärung	Beispiel
OrderNo	char / 10+1	Order number	8890123

4.7.2 The response

Datenfeld	*Typ/Länge*	Erklärung	Beispiel
ReturnCode	long int	>0: Number of bytes in the returned text 0: No notes available <0: Failure	892
OrderNote	unsigned char	Codepage coded char stream containing the order note	...

4.8 Machine specific, freely defined messages (SD_ERRORTEXTS)

One type of messages which can be told to the Logotronic using OPERATIONAL_DATA are freely defined messages which are workplace specific.

If the Logotronic detects after exchanging the version numbers (VERSIONINFO-request) that the version of the machine software does not correspond to the version stored in the Logotronic for this workplace it sends an INFO-response which requests the workplace to inform the Logotronic about all message texts in the installed languages.

4.8.1 The informational response to the workplace client

Data field	Type/Length	Description	Example
InfoCode	unsigned long	Synchronizing message texts: 32	32
ServerInfo	char / 255+1	Requested language ids separated by ‚+'	„0+2“

The following language ids are supported:

Language	Koenig&Bauer language id
German	0
Englich (GB)	1
Englich (US)	8
French	2
Italian	3
Hungary	4
Spain	5
Swedish	6
Danish	7
Dutch	9
Portuguese	10
Polish	11
Russian	12
Greek	13
Chinese	14
Czech	15
Korean	16
Turkish	17
Croatian	18
Finnish	19
	20
Japanese	21
Slovecia	22
Romanian	23
Vietnamese	24
Arabic	25
Thai	26
Slowenian	27
Chinese (Traditional)	28
Hebrew	29
Lithuanian	30
Portuguese (Brasil)	31
Bulgarian	32
Estonian	33
Latvian	34
Norwegian	35
Farsi	36

After retrieving of that INFO-response with Info-Code 32 the machine sends a SD_ERRORTEXTS-request containing the message texts of any desired language to the Logotronic. If an language is not available it should be sent the English texts or, if even English is not available, in German.

Data field	Description
TransactionID	Generated by Client
WorkplaceID	WorkplaceID oft the requesting machine
RequestType	SD_ERRORTEXTS = 38
DataLength	Req.: 8 + size of payload (in maximum 64Kb) Rsp.: 4

4.8.2 The request

Data field	Type/Length	Description	Example
TextCount	long int	Number of sent messages	1021
LanguageNr	long int	Language id regarding to the table above	0 (deutsch)
Text[0]	ERROR_INFO	“Message text”-data field as described below	...
Text[1]	„		
...			

4.8.3 The response

Data field	Type/Length	Description	Example
ReturnCode	long int	Number of successfully imported messages	1020

4.8.4 The “Message text”-data field

Data field	Type/Length	Description	Example
ErrorNumber	char / 6+1	Message text number is it is send in OPERATIONAL_DATA	$1000
ErrorText	char / 100+1	Codepage coded message text	DW #1: Pneumatiksch...

4.9 Sending operational data (OPERATIONAL_DATA)

The OperationalData (RequestType=11) allows the sending of general and job regarded production data. This telegram should be sent cyclicly as well as event driven for instance in case of a malfunction or production start.

Data field	Description
TransactionID	Generated by Client
WorkplaceID	WorkplaceID oft the requesting machine
RequestType	OPERATIONAL_DATA = 11
DataLength	Req.: 252 Rsp.: 4

4.9.1 The request

The following table contains the definitions of the single record fields. Optional record fields are displayed italic. Should an optional record field not be used all bytes of it must be filled with binary NULL.

Data field	Type/Length	Description	Example
TimeStamp	long int	Machine speed at the time the telegram is sent	
MachineState	long int	Machine state	MSt_Production
Amount	char / 10+1	Job related amount of good (net) products (in pieces)	25043
TotalAmount	char / 10+1	Job related amount of gross products (in pieces)	26211
TotalCount	char / 10+1	*Overall total counter of the machine*	*1632329*
Speed	char / 10+1	Machine speed at the time the telegram is sent	14000
Machine Times	long int	Time classification	MTi_MakeReady
ActivityNo	char / 20+1	Fixed machine independent messages (activities)	@95
ActivityName	*char / 20+1*	*Activity name*	
ActivityValue	*char / 10+1*	*(optional)Activity related value, e.g. Unit number*	*1*
Units	*char / 2+1*	*Signs whether messsage starts or is gone* *„1“ = incoming, „0“ =gone*	*1*
CostcenterNo	*char / 10+1*	*Free cost center number, if an free message is active otherwise Null*	*808*
PersNo	*char / 10+1*	*Always Null (determined by Logotronic)*	
MachineJob	long int	Order assignment	MJo_JobRunning
OrderNo	char / 10+1	Order number	1922432
ProdNo	char / 10+1	Production number (partial job number)	1
JobNo	char / 10+1	Job number (printing run number	2
Note	char / 80+1	*Notes*	*...*

Table of record fields OPERATIONAL_DATA

TimeStamp (mandatory): Unix-Time format: Number of seconds, which have passed since 01.01.1970 00:00 GMT

The state of the machine is identified by three independent attributes: the operating state **MachineState**, the time classification **MachineTimes** and the job assignment **MachineJob**. A machine event may require a state transition in one or more attributes. The new machine state is communicated to the Logotronic system with each request of the type OPERATIONAL_DATA.

Event	Order state		Following state	
	MachineJob	Order state	MachineJob	Order state
Start of job	NoJobActive	Planned	JobRunning	Started
Continue job		Interrupted		
Job interrupted	JobRunning	Started	NoJobActive	Interrupted
Job finished				Finished

MachineState (mandatory):

- 0 = MSt_Idle (is the basic state of a machine. In this state the machine is basically ready for operation, but does not produce)
- 1 = MSt_Production (a material flow takes place through the machine. At least the value of the gross counter is constantly increased.)
- 2 = MSt_Malfunction (a technical fault of the machine)

MachineTimes (mandatory)

Possible values for the time classification

Value	Meaning
285212672 (0x11000000)	MTi_MakeReady
?????????	MTi_Processing
301989888 (0x12000000)	MTi_Production
318767104 (0x13000000)	MTi_OtherProcessing
536870912 (0x20000000)	MTi_Auxiliary
1073741824 (0x40000000)	MTi_Down

MachineJob (mandatory):

- MJo_JobRunning = 0x0000 1000 „Order is assigned“
- MJo_NoJobActive = 0x0000 2000 „No order is assigned“

ActivityNo:

It is separated between 3 kinds of messages:

- Machine independent messages, starting with “@”,
- Machine specific messages, starting with “$”,
  freely defined messages as they was sent in SD_ERRORTEXTS (Chapter 4.8)

For freely defined messages this field contains the source where the message is caused. The source/place is identified by its id/number as it was sent in SD_ERRORTEXTS. For instance this could be 112 for the Rapida delivery.

The following activity numbers indicates **Machine independent messages**:

ActivityNo	Message texts
@17	**Machine counters**
@18	**Production run interval started**
@41,42	Incoming/Outgoing Plate change Printing unit #1
@45,46	Incoming/Outgoing Impression cylinder washing
@47,48	Incoming/Outgoing Blanket washing Printing unit #1
@49,50	Incoming/Outgoing Inking roller washing Printing unit #1
@51,52	*Incoming/Outgoing Ink duct roller washing*
@53,54	Incoming/Outgoing Printing plate Printing unit #1
@55,56	*Incoming/Outgoing Dampening color washing*
@57,58	Incoming/Outgoing Late sheet
@59,60	Incoming/Outgoing Early sheet
@61,62	Incoming/Outgoing Misaligned sheet
@63,64	Incoming/Outgoing Double sheet
@65,66	*Incoming/Outgoing Sheet arrival controller at the limit of control range*
@67,68	*Incoming/Outgoing Sidelay error*
@69,70	*Incoming/Outgoing Sheet incoming error*
@95	**Job started**
@96	**Job interrupted**
@97	**Job finished**
@102,103	Incoming/Outgoing Identifying printing plate
@106	**Job continued**
@117	**Production run interval finished**
@118	**Production started**
@119	**Production finished**
@120,121	**Incoming/Outgoing Malfunction**
@122	**Makeready time started**
@123	**Production time started**
@124	**Other Processing time started**
@125	**Down time started**
@126	**Auxiliary time started**
@127	**Reset production counters**
@132	Discard plate scan
@133	Plate scan finished
@200	#1 logged in as craftsman
@201	#1 logged in as shift foreman

@202	#1 logged in as shift forman and press operator
@203	#1 takes a break
@204	#1 returns from break
@500,501	Sheet travel
@502,503	Ink up
@504,505	Engage/disengage printing unit
@506,507	Perfecting/inline mode change
@508,509	Blanket change
@510,511	Manual sheet travel
@512,513	Semi automatic plate change
@514,515	Guide segment adjustment
@516,517	Guide roller adjustment
@518,519	Plate correction
@520,521	Machine positioning for ICS
@522,523	Positioning coating form cylinder
@2000	Count progress before manipulation
@2100	Order changed
@2101	End Printing/Gluing
@2102	Totalisator manipulation
@2000	Cylinder change
@2100	Start of shift
@2101	End of shift
@2102	Shift changed
@2200	Day changed
@2300	Press connected
@2301	Press disconnected
@2302	Press switched off
@2400	Terminal connected
@2401	Terminal disconnected
@2310	Press connected by DB
@2311	Press disconnected by DB
@3000	CIP3 interpretation failed
@4000	Pile changeover
@4001	Pile - Job Change
@4002	Pile - Lot Change

The die cutting machine should limit their activities to these whose seem to fit to their needs, for instance the bold activities in the table above.

If additional ativities become necessary they can be attached. But it must kept in mind that these new activities will also be available for all other machine classes.

There are no machine class specific activities.

> ***Please care for to define new @ActivityNumbers, which does not interfere with the already existing ones. These messages must also be told to the Logotronic development for integration into the logotronic.***

ActivityName (optional):

A short name describing the Activity, defined by the client. For instance for the activity "@41" and "@42" the Rapida assign the values "BEGIN_PLC" and "END_PLC" to the name-attribute.

ActivityValue (optional):

With this field you can pass additionally information like the number of the affected print unit. For instance for these both activities mentioned above the Rapida sets the attribute "value" to the number of the unit where the plate change takes place.

> **If a free defined message (see SD_ERRORTEXTS)** was sent the ActivityValue contains the message number according to the free message.

Units (optional): Free messages of the type „Incoming/Gone" must contain this field to mark the begin and the end of a message.

- „1" = Message comes in
- „0" = Message is gone

CostcenterNo (optional):

If no free message is active, the cost center will be determined automatically by Logotronic, e.g. by the current time classification. Otherwise a cost centre number can be entered to assign the time period covered by the "Come/Goes" of a free message to assign the message to a certain cost centre.

PersNo (optional):

Always NULL: Only the Logotronic knows about the currently active machine operator (if a personnel administration must be implemented at all)

4.9.2 The response

For the die cutter only the return code should be processed.

5 Saving and Loading job repetition data

5.1 Saving job related machine data (SAVE_REPETITIONDATA)

The Logotronic stores a repletion data record for a specific job (Printing pass). If the same repetition data identifier is used, previously saved data will be overwritten if necessary. If no data is transferred in this request, possibly existing data will be deleted.

Data field	Description
TransactionID	Generated by Client
WorkplaceID	WorkplaceID oft the requesting machine
RequestType	SAVE_REPETITIONDATA = 50
DataLength	Req.: 33 + Byte size of data to transmit (in maximum 64 KByte) Rsp.: 4

5.1.1 The request

Data field	Type/Length	Description	Example
OrderNo	char / 10+1	Order number	2829232
ProdNo	char / 10+1	Prduction run number	2
JobNo	char / 10+1	Job number	1
RepeatDataId	char / 10+1	Data identifier: defined are: **KBA_SET01 data in standard format**, KBA_DAT01 data in XML-format	KBA_DAT01
RepeatData	unsigned char	Job Repetition dta	Text-/Binary data

5.1.2 The response

Responsedaten

Data field	Type/Length	Description	Example
ReturnCode	long int	Number of successfully stored bytes <0: Fehler	

5.2 Reading repetition data (READ_REPETITIONDATA)

The Logotronic supplies to the desired job number the calculated color preset data for offset printing machines or the repetition data saved before depending to the request.

Additionally it can be checked whether repetition data (or preset data) is available for the specified job number.

Data field	Description
TransactionID	Generated by Client
WorkplaceID	WorkplaceID oft the requesting machine
RequestType	READ_REPETITIONDATA = 49
DataLength	Req.: 33 Rsp.: 4 + Byte size oft he transmitted data (in maximum 64 KByte)

5.2.1 The request

Data field	Type/Length	Description	Example
OrderNo	char / 10+1	Order number	2829232
ProdNo	char / 10+1	Production run	2
JobNo	char / 10+1	Job number	1
RepeatDataId	char / 10+1	Data-Identifier: - KBA_INFO: requests a job note which could be stored before by meaning of SAVE_REPETITIONDATA - KBA_STATE: is repetition data available? - KBA_PRE01: requests preset data in standard format - KBA_PRE02: requests preset data in XML format - else: Identifier of the format, the repetition data was saved	**KBA_SET01**

5.2.2 The response

Data field	Type/Length	Description	Example
ReturnCode	long int	>0: Byte size of the returned data 0: No data <0: Failure Exceptions are: KBA_STATE: -1, if the job could not be found 0, Neither preset data nor repetition data is available	
RepeatData	unsigned char	Data	Text-/Binary data

6 Creating and Deleting a Job

Are there no jobs underneath a production runs the operator has to create a new job. This can be done by meaning of the KBA_CREATEJOB request as described as follows.

As well the oprator must have the option to delete a job which was created by a wrong entry or which was was parametrized not properly. For that the operator can us the KBA_DELETEJOB request.

6.1 Creating a job from client side (KBA_CREATEJOB)

Data field	Description
TransactionID	Generated by Client
WorkplaceID	WorkplaceID oft the requesting machine
RequestType	KBA_CREATEJOB = 63
DataLength	Req.: 165 Rsp.: 4

6.1.1 The request

Data field	Type/Length	Description	Example
OrderNo	char / 10+1	*Order number*	*8921021*
ProdNo	char / 10+1	*Production run number*	*1*
JobNo	char / 10+1	*Job number*	*2*
JobName	char / 30+1	Job name	Front
Amount	long int	*Copies*	*10000*
Addition	long int	Sunsidy	150
MakeReadyTime	char / 5+1	Planned make-ready time	00:30
PrintTime	char / 5+1	Planned production time	01:30
Comment	char / 80+1	Comments	Umgeplant

6.1.2 The response

Data field	Type/Length	Description	Example
ReturnCode	long int	1 if job was succesfully created, otherwise: 106=Oorder does not exist 206=Job: Belonging production run does not exist 301=Invalid chars in job number 303=Job could not be created 306=Job number already exists	1

6.2 Deleting a job from client side (KBA_DELETEJOB)

Because a single job is always identified by its number as well as by the workplace id where this job should be processed it is necessary that the deletion request always contain the order number, the production run number and job number as well as the workplace id.

The Logotronic can only process deletion requests for a job if the requesting workplace id is the same as the workplace id the job was planned for.

Data field	Description
TransactionID	Generated by Client
WorkplaceID	WorkplaceID oft the requesting machine
RequestType	KBA_DELETEJOB = 65
DataLength	Req.: 64 Rsp.: 4

6.2.1 The request

Data field	Type/Length	Description	Example
OrderNo	char / 10+1	*Order number*	*8921021*
ProdNo	char / 10+1	*Production run number*	*1*
JobNo	char / 10+1	*Job number*	*2*
WhichWorkplace	char / 30+1	*Workplace id*	*25*

6.2.2 The response

Data field	Type/Length	Description	Example
ReturnCode	long int	1 = The job could be deleted <> 1 Failure, Job could not be deleted	0

7 Proposals for working with orders

7.1 Management of folding gluer jobs

Right now in agreement to Bavaria Digital a folding gluer job belongs always to that production run which the appropriate printing job belongs to. So the folding gluer job is placed under this production run.

Beside their plan time jobs for the folding gluer machines needs no parameter else. So it is not necessary to implement an special configuration panel for a folding gluer job as it is needed by an printing job.

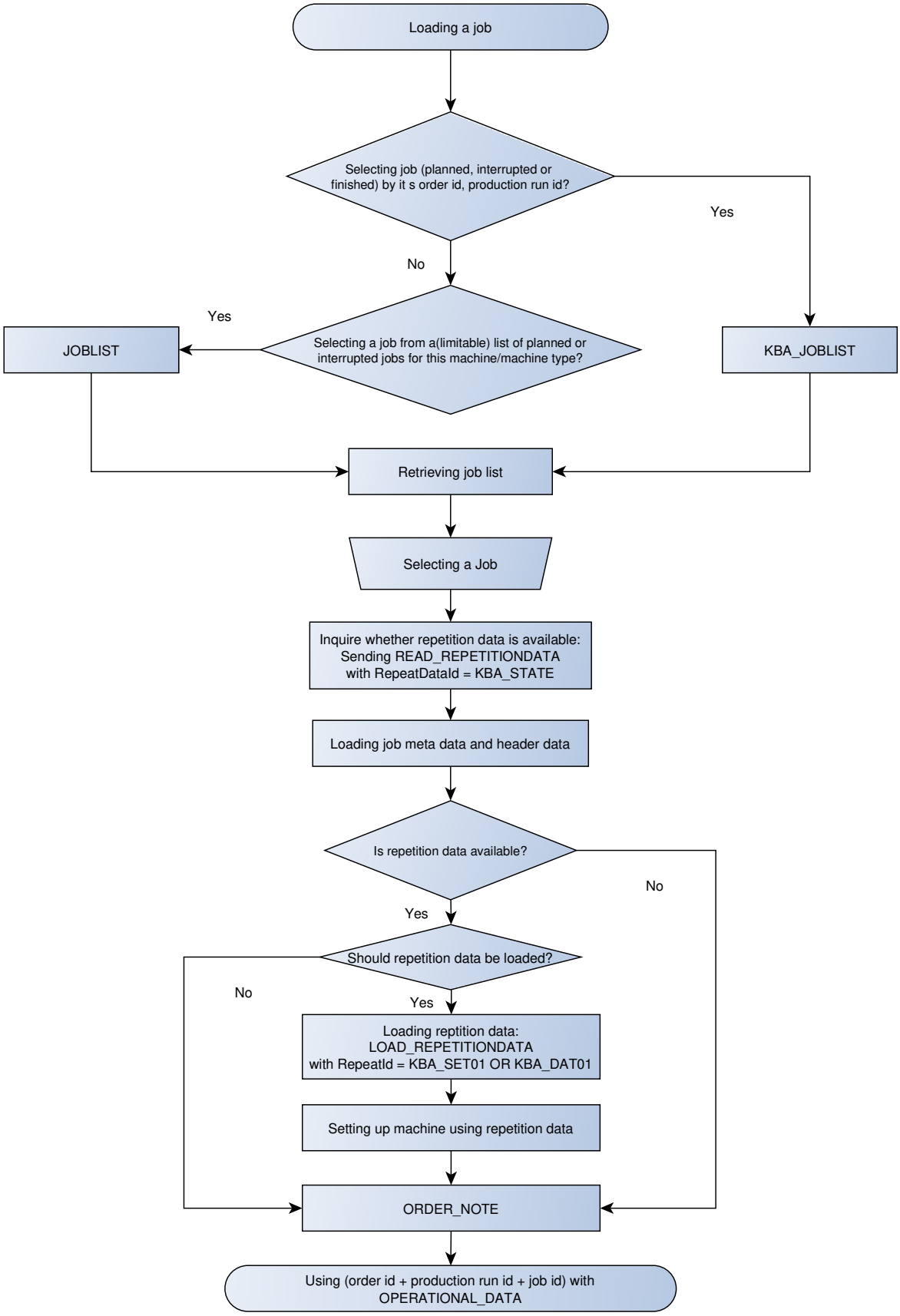
Folding gluer jobs can only and must be processed immediately after the printing product leave the delivery of the offset machine.

Regarding to that and as long as folding gluer jobs must not be planned in the Logotronic time table it makes no sense to create a job using the Logotronic navigator.

So the operator should create his job at the HMI of the folding gluer machine using the request described in chapter 6 so that the created job is attached to the appropriate production run.

The machine sends then all of its production data as well as its repetition data using the job id created by himself and the numbers of the regarding production run and order to the Logotronic.

7.2 Order of telegram communication for loading a job



8 Further telegrams

For later improvements the telegrams listed below can be used.

To retrieve data about one or more jobs and their corresponding meta data and preview pictures planned at the Logotronic the following telegrams can be used:

- XML-GetOrderNote (10006) (Chapter 2.2.4)
- XML-SetOrderNote (10007) (Chapter 2.2.5)
- XML-Preview (10093) (Chapter 2.2.9)

To save and restore machine adustments belonging to a job as repetition data the following telegrams must be used:

- XML-ReadRepetitionData (10049) (Chapter 2.5.1)
- XML-SaveRepetitionData (10050) (Chapter 2.5.2)

To journalize the staff assignment these telegrams are used

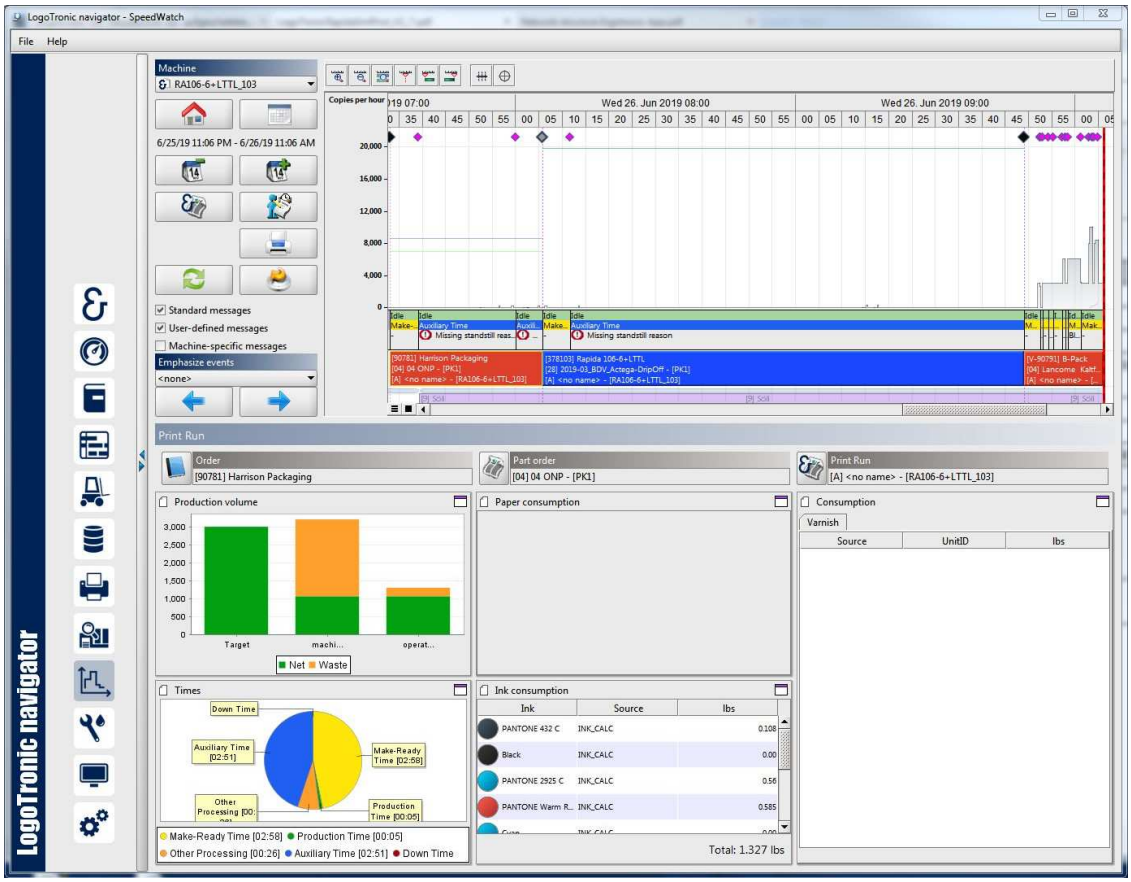
- XML-Personnel (10036) (Chapter 2.4.4)
- XML-BdePersonnel (10008) (Chapter 2.3.6)

A detailed reference of these telegrams is available in the LogoTronicRapidaXmlProt_V1_8.pdf.

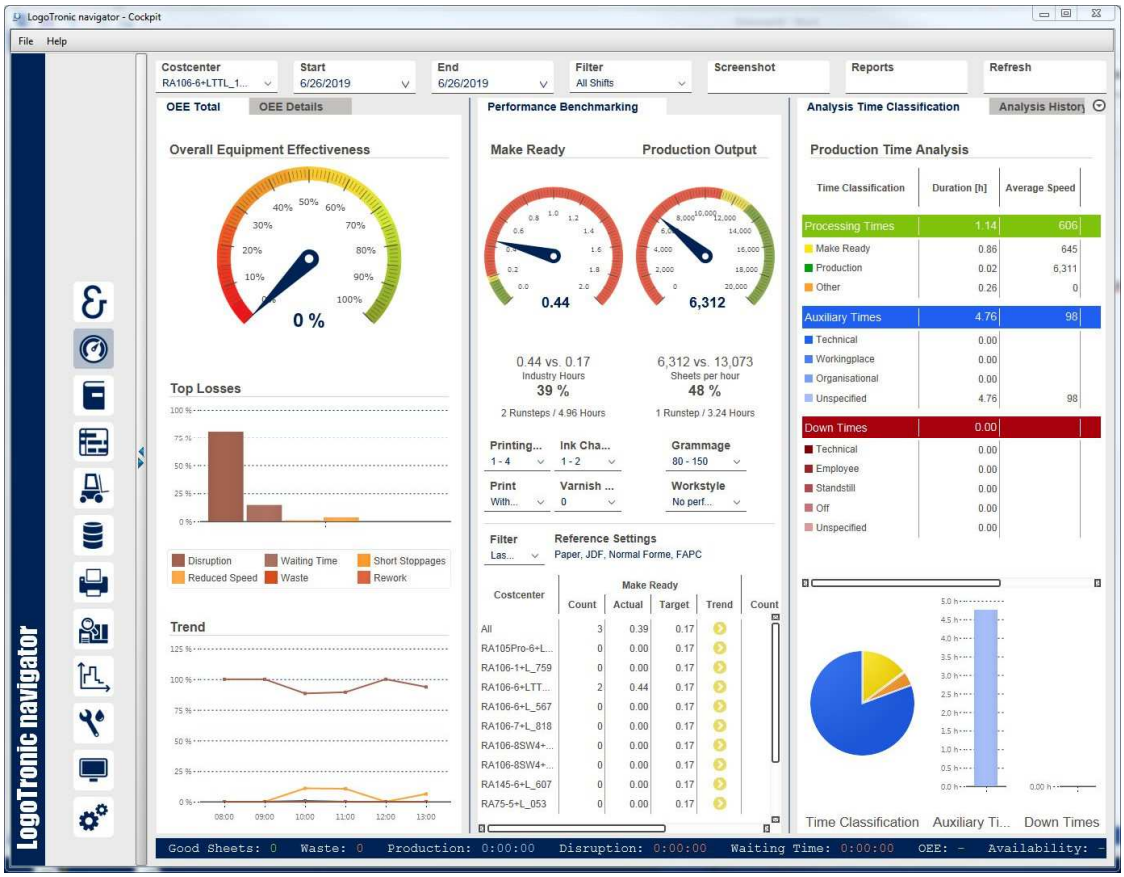
9 Example of displaying operational data in the Logotronic

The Logotronic features a panel displaying incoming operational data from every connected machine.

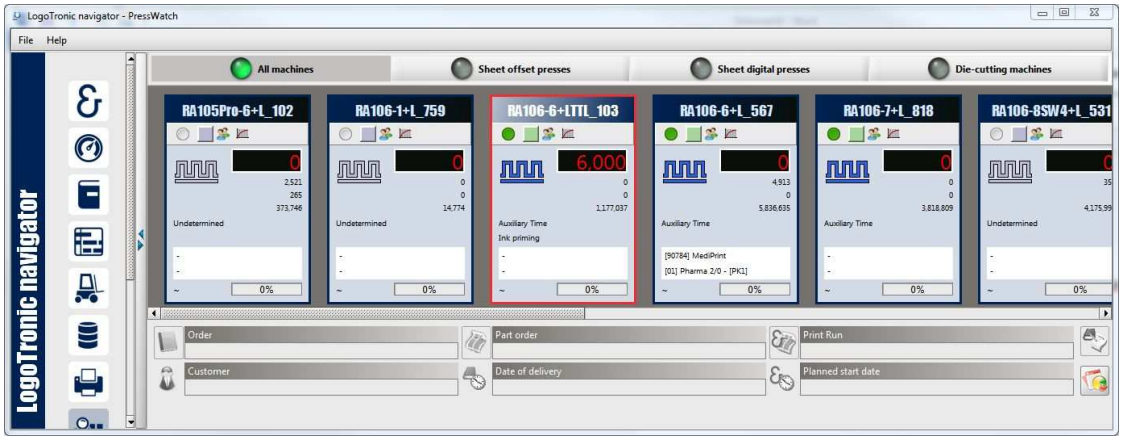
The client is able to send any states, message or machine situation to the logotronic. These state changes the logotronic renders in its “SpeedWatch”, “PressWatch” and “Logotronic Cockpit”:



Logotronic PressWatch



Logotronic Cockpit



Logotronic Speedwatch Panel

10 Appendix A: Machine Time Classification

Machine::timeState

Machine times								
Production times			Auxiliary times			Downtimes		
Makeready	Production time	Other processing times	Technical auxiliary time	Workplace-related auxiliary time	Organisational auxiliary time	Staff-related downtime	Technical downtime	Down time
From job start to the first good product e.g. net sheet	Machine speed > 0 and Net sheet counter on	Machine speed > 0 and Net sheet counter off or machine speed = 0 over a period of 6 min or less (e.g. Roller change, washing times)	Faults > 6 min to 8 hours (e.g. part damages, leakage)	Maintenance of production readiness (e.g. rubber blanket change, Cleaning)	Disturbance in the production process or organizational defects (e.g. Waiting for missing plates or auxiliary materials)	Paid absence of the operators (e.g. tariff holiday, works meeting, shift change)	Faults over 8 hours and major repair	Lack of orders

Machine::jobState:

JobRunning	A Job is loaded and the production is started or restarted in case of an interruption
NoJobActive	No job loaded

Machine::state

IDLE	No job is loaded and/or machine stands still; time period between job end and job start
PRODUCTION	Machine produces; Speed > 0
MALFUNCTION	There's no specification for sheetfed machines