# Just In Time Printing Guide

**Rev 1.00**

**Date: 24 May 2024**

## Revision History

| Doc. Version | JSL SW Release | Date | Details |
|---|---|---|---|
| 1.00 | R5.3.0 and later | 24 May 2024 | • Initial Release |

# Contents



## Figures

# 1 Introduction

This document is part of the OEM-facing technical documentation suite for Memjet DuraFlex® module-based printing systems. It references, and therefore requires access to, additional documentation available for download from your Memjet Partner Site.

## 1.1 Aim and Audience

This document outlines and explains the configuration of the Just in Time printing mode on a DuraFlex system.

This document should be read by anyone planning to implement or evaluate Just in Time printing on a DuraFlex system.

## 1.2 Prerequisites and Scope

The Just in Time (JIT) printing mode is a new printing mode supported with DuraFlex release R5.3.0 and later. It also requires use of Memjet JSL version 12.5.2 or later for generating print data.

This document does not cover any other aspects of the DuraFlex printing system or software.

## 1.3 Typographic Conventions

Throughout this document, the following typographic conventions are used:

| | |
|---|---|
| `Code Character` | `Courier` font is used to identify HTTP GET and POST commands with associated arguments, as well as references to source code, job states, registry settings, directory/file names, XCI commands, and XML settings. |
| **Bold** | Text that appears on-screen in the user interface is shown in **bold**. This includes UI buttons, engine states, warning codes, and fault codes. |
| Yellow Highlighting | Yellow highlighting indicates sections that are new or updates in this version of the document, compared to the previous version. |

## 1.4 Related Documentation

Other documents, besides this guide, provide further details for specific readers:

- *System Overview* – For OEM managers and non-technical personnel charged with evaluating the DuraFlex components for use within their products. This document describes the DuraFlex concept and Memjet-supplied DuraFlex components and gives an overview of the operational considerations. It introduces the components an OEM is required to design and manufacture to ensure the DuraFlex Modules function as designed in a DuraFlex-based print engine.
- *Mechanical and Fluidic Databook and Design Guide* – For mechanical design engineers and developers, providing details of the Memjet hardware modules and components (including printhead and maintenance system) and specifications of the ink delivery system fluidics.
- *Electrical Databook and Design Guide* – For electrical design engineers and developers, providing details of the Memjet power requirements, electronic assemblies, and connections.
- *Software Databook and Design Guide* – For software and firmware engineers who need to understand the software interfaces, commands, scripts, and reference software applications.
- *Demo GUI User Guide* – For OEM personnel using the DuraFlex Demo GUI reference application.
- Installation and Commissioning Guide – For OEM personnel who are installing and commissioning a new printing system.
- *Operations Guide* – For OEM engineers and operators to perform operational tasks.




- Troubleshooting Guide – For OEM engineers and technicians to identify symptoms and resolve issues.
- *Service and Repair Guide* – For OEM engineers and technicians to perform DuraFlex inspection and maintenance tasks and component and consumable replacement.
- *Job Submission Library Guide* – For OEM software engineers to incorporate the Job Submission Library (JSL) into their chosen Raster Image Processor (RIP).
- *Technical Bulletins* – For various audiences to announce product or process update or to provide specifics on single-subject technical topics.
- *CAD and Schematics* – For various audiences to provide detailed dimensions related to specific areas.

| | |
|---|---|
| Note: | All technical documentation is available on your Memjet Partner Site. |

## 1.5 Glossary

For terms, acronyms, and abbreviations used in this guide and some product-specific terms, see the *DuraFlex Glossary*.

| | |
|---|---|
| Note: | This document is hyperlinked to the glossary. For offline reading, download the DuraFlex Glossary file from your Memjet Partner Site. |

## 1.6 Additional Documentation or Access

For additional product-related technical documents, go to your Memjet Partner Site.

If you need Partner Site access, enter a case in Service Desk (*https://OEMsupport.memjet.com*), send an email to Memjet Customer Support (*customer.support@memjet.com*), or contact your Technical Account Manager.

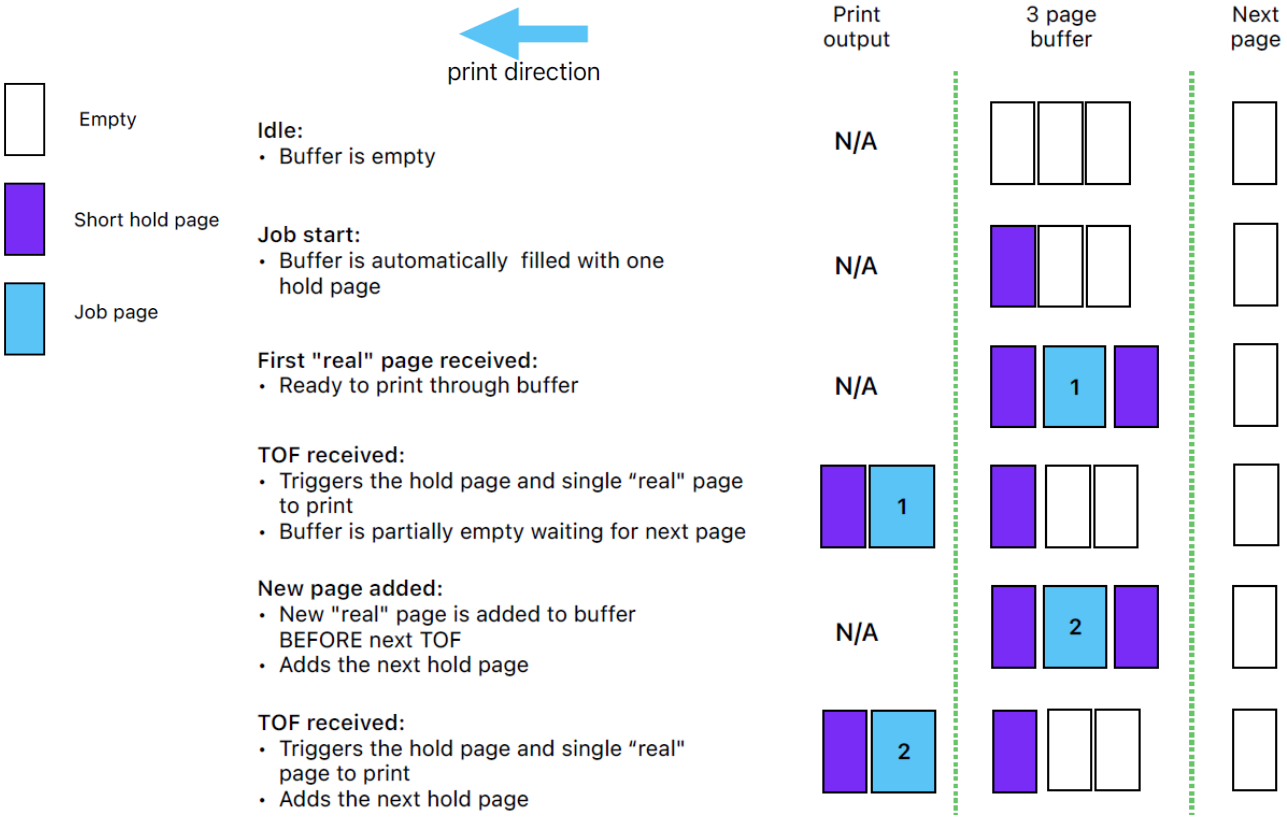
# 2 Description and Configuration

## 2.1 The JIT Print Mode

DuraFlex R5.3.0 software (and later) supports Just in Time (JIT) printing mode. This mode allows the print data for a page to be supplied just before the corresponding TOF signal however it can only be used with TOF-triggered printing modes such as `ALL` and `TRANSACTIONAL`.

JIT printing works by internally generating a blank hold page which fulfills the internal queue requirement to have a page ready to print. The hold page is managed entirely within the DuraFlex system and is based on several configuration settings. When the TOF signal is received, the system prints the hold page, then any in-job maintenance ink and then prints the job page. The print data for the print job page must arrive in the print engine pipeline (PEP) queue in time otherwise a buffer underrun will occur. The buffer sequence is shown in detail in *Figure 1*.

**Figure 1 – Buffer Sequence During JIT Printing**




The hold page is automatically configured to be half the size of the `interPageGap` setting. The hold page itself must be large enough to ensure there is enough time to process the start of page interrupts. The minimum hold page size must be larger than the intra-printhead gap in the DuraFlex printhead plus a bit more. So, using a hold page size of at least 13mm should always work. This means the `interPageGap` must be set to 26,000 µm (slightly larger than 1 inch). In addition, the `mediaReadyOffset` values, the distance between the TOF sensor and the print zone, must be greater than the hold page length plus any in-job maintenance that is configured.

## 2.2 Configuration and Use

### 2.2.1 Activation

To activate JIT print mode, you will need to change the following configuration settings on the print module and the RIP.

1. On the RIP, update the `JslConfigs.xml` file by uncommenting the highlighted code. Remove both of the `<!--` and `-->` tags before and after the element, and change the element value from `0` to 100.

```
<!--
        The JSL API normally requires adding a page with PageDescription::isLastPage set in
        order to call jslibCloseJob(). In the event of an error, a job is canceled by first
        calling jslibAbortJob() followed by jslibCloseJob(). In JIT applications there are
        scenarios
        where the user doesn't know whether there is another page to follow when sending the
        current page to set the PageDescription::isLastPage appropriately. By setting the
        following to a value greater than 0, a user can change the behaviour of the API to
        allow calling jslibCloseJob() without sending a page with PageDescription::isLastPage
        set to indicate completion of a job. The value indicates the time to wait in
        milliseconds after sending the BOR_END_JOB block, before closing the connection to
        ensure that the other end receive the job end block. A value of 0 disables this
        feature. The value must be an unsigned integer.
        -->
        <!--
        <closeJobWithoutLastPageDelayMs>0</closeJobWithoutLastPageDelayMs>
        -->
```

   The reason for modifying this setting is to remove the need to set the `isLastPage` flag to true when finishing a job. On non-JIT jobs, the `isLastPage` is set to true to indicate to the system that the job is ending. However, this is not practical in JIT printing as pages are not in a well-defined order or quantity.

2. Next, the print engine settings must be changed. The setting of `defaultPepQueueMaxPages` must be adjusted in the `hwparamstore.json` file. Under normal, non-JIT printing, this value is usually set to 3 or more. To activate the JIT print mode, it must be set to 2. Additionally, the TOF trigger needs to be set to `ALL_PAGES` or `TRANSACTIONAL` mode. While in this mode, there are several constraints:

   - The TOF sync mode can only be set to `ALL_PAGES` or `TRANSACTIONAL`
   - The `interPageGap` must be larger than 13mm
   - The `mediaReadyOffset` setting must be larger than the hold page distance plus any pre-page spit distance, declog and pre-page spit gap (if they are being used).
   - If pre-page spit/declog is used, a spit target area must still be allocated in the same way as described above. However, the use of a hold page adds an additional constraint in that the secondary spit target area may not be smaller than the first page target area (even when it is not normally required, such as in `FIRST_PAGE` declog mode).
   - Because this is a TOF-triggered mode, the `interPageGap` setting does not cause any additional gap to be added to the inter-maintenance page (other than the hold page itself).
   - The relationship between the hold page and the other PES settings is only checked when preparing to print, and not when loading/updating settings.
   - Pausing: In JIT printing mode, pausing and mid-job servicing is not expected to work as the hold page would be stranded in the PEP queue. However, the print session may be completed at any time (using `finishPrinting`) without queuing or printing any more pages, so the printer controller can complete the current job, perform any necessary maintenance, and then start a new job.




3. The configuration settings can be set via the PES interface. Refer to the *DuraFlex Operations Guide* for details on how to log into the PES command line:

```
>>> ss = pes.getSettings()
>>> ss.engineStage[1].mediaReadyOffset.isFactoryDefault = False
>>> ss.engineStage[1].mediaReadyOffset.value = 40000
>>> ss.interPageGap.isFactoryDefault = False
>>> ss.interPageGap.value = 26000
>>> ss.mediaTiming[1].tofSyncMode = 3   # must be ALL_PAGES (3) or TRANSACTIONAL (4)
>>> ss.mediaTiming[1].printOnMedia = False
>>> pes.storeSettings(ss)
```

4. There is also a setting that must be configured after the print engine is powered on. This setting will also need to be performed again if the printer power is cycled. However, this will not be required in the final DuraFlex R5.3.2 release. Login to the DuraFlex unit and get to a command prompt.

   Enter the following:
   `> nc localhost 9000`

   This will then show the cursor on a blank line. Then enter:

   `tcl pep end_of_page_poll_period 200`

   Then press CTRL+D to close the `nc` connection.

### 2.2.2 Printing

After the configuration is complete, you can print jobs where the main constraint is that the page to print must be fully in the buffer before the TOF for that page is triggered. Provided the page is in the buffer prior to the TOF trigger, the print job should continue without needing to keep the single page buffer full. The high-level flow of a print job is given below:

1. Open a new job via the JSL API. This generates the job header information and allows a job to be started via the PES API even if no pages have been added to the buffer.
2. The job state will change from `ASSEMBLING` to `IDLE`.
3. Send a `prepareToPrint` command via the PES API.
4. Send a `startPrinting` command via the PES API.
5. Send pages(s) to the print engine via the JSL API and make sure a full page is in the buffer before the TOF that triggers the printing of that page. When no more pages are to be printed, close the job via the JSL API.
6. Wait for the job completion via the PES API, then call `finishPrinting` via the PES API.

## 2.3 Troubleshooting

### 2.3.1 TOF Triggering Issues

It is imperative that the page buffer contains full page in it when a TOF signal arrives for that page to print. In practice, this means that the media feed must be fully controlled to avoid sending extra blank pages after there are no more pages to put in the buffer. If a TOF signal arrives when the buffer is empty, a data underrun fault will occur.