

Loftware Label Manager Version 12.1.1 Release Notes March 2021

What new features are available in this release?

Improved logging: Information about the actual point size used for a TextBox field with the Dynamic Point Size property enabled was added to logging.

Improved TTF support: The actual TTF used at print time in label fields is now identified and documented.

What fixes are available in this release?

- Resolved an issue that caused the On Demand Print Client to crash when printing labels with fixed image paths to TIFF files with more than 265 characters in the file name.
- Resolved a device driver memory leak that occurred when barcode images fail to render correctly.
- Resolved an issue that prevented LWReporter from correctly reporting strings containing only spaces.

What issues are known about this release?

Automatic updating of existing scripts: In LPS/LLM 12.1, the old Spider Monkey scripting engine was removed and replaced by the new Microsoft Chakracore scripting engine. If upgrading from a version prior to LPS/LLM 12.1, saving an existing label with a script automatically updates that script to be compatible with the Microsoft Chakracore scripting engine even if the user did not change the script. After being saved, the script is incompatible with LPS/LLM versions prior to LPS/LLM 12.0.2.

Important! Prior to upgrading, back up LPS/LLM.

Differences between the new Microsoft Chakracore scripting engine and the old Spider Monkey scripting engine: The Microsoft Chakracore scripting engine offers significant performance improvements when printing labels with scripts. However, there are some behavioral differences that may require changes in some situations.

- The old scripting engine used XPath XML queries, whereas the new scripting engine uses JSON queries and could present an Unexpected Character error.
- The old scripting engine touched every field, whereas the new scripting engine only updates fields that are required per the script.
- The old scripting engine had a property named **length** to describe the maximum length of data, whereas the new scripting engine reserves the property named **length** for an array of fields. Instead, the new scripting engine has a property named **maxChars** to describe the maximum length of data. Note that **maxChars** is in units of bytes rather than characters. If using multi-byte characters, divide this value by the number of bytes per character to determine the maximum number of characters allowed. For example, Simplified Chinese characters require 3 bytes per character, so if **maxChars** is set to 6, then 2 Simplified Chinese characters are allowed.
- The old scripting engine trimmed spaces from every field, whereas the new scripting engine does not do this.



ESCC substitution characters in Datamatrix barcode data: Changes were made to support the printing of Datamatrix barcodes on Zebra printers that relate to FNC1, the underscore (_) character, and the Escape Sequence Control Character (ESCC) command. Zebra's default value for the ESCC was the underscore (_) in old firmware but was changed to the tilde (~) in newer firmware. Until this change, Loftware's ZPL driver for LPS hardcoded the underscore as the ESCC. After this change, the tilde is used as the default ESCC, but if a tilde appears in the user data then another character is substituted.

- In the following situations, the data will not be encoded correctly by default.
 - A Zebra printer with newer firmware is used and the user data contains all of the ESCC substitution candidate characters. The ESCC defaults back to the tilde even though it is not available for use.
 - A Zebra printer with older firmware is used and the user data contains a tilde.
- If the data is not encoded correctly because of one of the preceding situations, the following workarounds are available.
 - If there are no underscores in the data, you can restore the old behavior of using the underscore for both the ESCC and the FieldHex character. This is a system-wide setting and is not configurable in Preferences. To make this change, configure the following setting in LLMWDN32.INI.

```
[Zebra]
DmtrxUScoreEscc=1
```

- You can configure Datamatrix barcodes to print as a bitmap. This is a printer model specific setting. To make this change, configure the following setting in LLMWDN32.INI.

```
[BarcodeBitmaps]
datamatrix=~printerId~
```

Setting changes that you can revert if necessary:

- Changes were made to correctly encode greater than (>), tilde (~), and caret (^) symbols when they are included in a Code 128 or GS1-128 Generic barcode when printing to a Zebra printer. If you want to restore the behavior that existed prior to these changes, you can do so by configuring the following settings.
 - Previously, the character mask for GS1-128 Generic did not include some symbolic characters that should be supported according to the specification. The greater than symbol was one of these characters.

The following are all of the characters that are now part of the default character mask for GS1-128 Generic.

```
!"#%&'*+,-./0123456789:;<=>?ABCDEFGHIJKLMNOPQRSTUVWXYZ_abcdefghijklmnopqrstuvwxyz()
```

If you want to restore the previous behavior, configure the following setting in LLMWDN32.INI . This setting affects all drivers. If this setting is omitted from the INI file, its default value is 1.

```
[Options]
GS1ISO646forGS1_128GEN=0
```



- Previously, when stepping through the data, the greater than symbol was left as is in the data, regardless of the field type or printer. Now when encountering a greater than symbol in a native Code 128 or GS1-128 Generic barcode on a Zebra (ZPL) printer, the greater than symbol is replaced with character sequence of ">0".

If you want to restore the previous behavior, configure the following setting in LLMWDN32.INI . Although this setting affects all drivers, Zebra (ZPL) is the only one that substitutes a different character sequence for the symbol. If this setting is omitted from the INI file, its default value is 1.

```
[Options]
SubstGTinCode128=0
```

- For a Zebra (ZPL) printer, the Code 128 and GS1-128 Generic data string includes embedded switch codes. When building the data string, the presence of a greater than sequence (>0) is processed as a trigger to switch to subset B if necessary. This same setting applies to the tilde (>=) and caret (><) sequences. None of these can be individually configured.

If you want to restore the previous behavior, configure the following setting in LLMWDN32.INI . This setting affects only Zebra. If this setting is omitted from the INI file, its default value is 1.

```
[Zebra]
Switch2B4SpecialChar=0
```

- Previously, paragraph fields with right justification that were partially off the label were printing left justified. This behavior has been changed so that if the right justification setting is enabled, then the text will always print right justified, regardless of whether the paragraph field extends off the label.

If you want to restore the previous behavior, configure the following setting in LLMWDN32.INI . If this setting is omitted from the INI file, its default value is 0.

```
[Options]
ForceOffLabelJustifyToLprnt=1
```

End of Support Notice for Windows Server 2008 R2: New LPS family products no longer support Windows Server 2008 R2 (beginning with LPS 12.1). Customers running Loftware solutions on this operating system are encouraged to upgrade to Windows Server 2019 or other supported operating systems.

End of Support Notice for Windows 7: New LPS family products no longer support Windows 7 (beginning with LPS 12.1). Customers running Loftware solutions on this operating system are encouraged to upgrade to Windows 10 or other supported operating systems.