Elliott® V8.6 Release Notes

Introduction

Elliott Version 8.6 is built on top of 8.5 and is compatible with prior V8.x versions. The time to launch a task from Elliott Control Center had been improved. The existing tables and fields sizes are the same. Generally speaking, you can run Elliott 8.6 with any prior 8.x version side by side. The interoperability allows you to evaluate the new features in Elliott 8.6 before you switch everyone to this new version. Therefore, upgrading to Elliott V8.6 is relatively easy with minimum risk. For more information on the compatibility issue, please refer to the following KB article: https://support.elliott.com/knowledgebase/articles/1996810-elliott-8-6-requirements-and-features-compatibilit

One of the main features in Elliott 8.6 is Report Desk (RD) which is like a report writer but is a lot easier to use and safer. RD is not turned on in 8.6 by default. Before you start to use it, you will need to set up the necessary Elliott 8.6 PSQL relational databases. The Elliott 8.6 installation utility will help you to create the databases. By default, it will create Eli86DATA?? databases which are designed for alphabetic document number format. But if you did not convert to using the alphabetic document number format, then want to use the 8.2 format like the EliData?? database instead. For more information on creating the databases, please refer to the following KB article: https://support.elliott.com/knowledgebase/articles/1949389-how-to-use-el860db-exe-utility-to-create-psql-data

Similar to Report Desk, PowerSearch and Avalara Sales Tax Interface are also dependent on the relational engine. If you wish to use these features, please refer to the same KB article noted above. Avalara Sales Tax Interface only works in Elliott 8.6. Once you start using it, then certain functions in AR or COP must be run in Elliott 8.6.

During Elliott 8.6 installation, the system will bring up the Elliott 8.6 configuration utility automatically, which updates the <ElliottRoot>\Bin86\EL860.CFG file. This file contains critical configuration information for Elliott 8.6, including License Key Codes, Default Email Servers, Report Desk, Databases...etc. After installation, you can also update EL860.CFG by running the utility <ElliottRoot>\Bin86\EL860CF.EXE or use notepad to edit this file manually. For more information, please reference the following KB article: https://support.elliott.com/knowledgebase/articles/1937566-elliott-v8-6-configuration-utility-el860cf-exe

The following are major new features in Elliott 8.6:
- Report Desk
- PowerSearch
- Avalara Sales Tax Interface (add-on Option)
- Document Notes
- Optionally Eliminate ASCII Data Files
- Support Multiple Email Servers and Email Throttling
- Defer Process Supports PDF PostOffice

- Century Year Cutoff Is Changed from 30 to 50

For a general description of other changes made since Elliott V8.5, please refer to the section "What's New since Elliott V8.5." You may also find more information on our support portal https://support.elliott.com in the "Elliott Version 8.6" section, or use the following URL directly: https://support.elliott.com/knowledgebase/topics/149254-elliott-version-8-6

Report Desk

Report Desk provides a powerful developing environment for Netcellent to deliver modern reports with rich elements of proportional spacing fonts, graphics, and line draws. It allows us to output reports to PDF, CSV, and XLSX formats. In addition, Report Desk uses the PSQL relational engine to access Elliott's database. As a result, Report Desk can be significantly faster than traditional Elliott reports.

The vast majority of RD reports are user-definable. This means, as an admin user, you can modify Elliott's pre-defined RD report layouts by adding or removing columns, changing report sorting sequence, and choosing different filtering criteria to print the report. The process is a lot easier and safer than the traditional report writer. To get started with Report Desk, please refer to the following KB article: https://support.elliott.com/knowledgebase/articles/1968144-report-desk-running-designing-reports-basics

PowerSearch

PowerSearch allows users to search Elliott master tables through the PSQL relational engine and hence is more versatile than the traditional Elliott searches based on the transactional engine. Privilege users can modify the basic PowerSearch provided by Netcellent in the following ways:
1. User definable searching filtering condition
2. User definable displaying columns for returning results.
3. User definable ordering sequence for returning results.

PowerSearch is not enabled by default. To enable PowerSearch, you need to go to Global Setup -> System -> Comp. Specialized Control, and change the flag "5. Enable PowerSearch" to "Y". The twin flag "Use Def Search Only" determines if the user will see all user definable searches or only the default search. For more information, please refer to the following KB article: https://support.elliott.com/knowledgebase/articles/1988338-powersearch-setup-and-configuration

Avalara Sales Tax Interface

Avalara Sales Tax solution is a cloud-based US national sales tax solution. If you need to collect sales tax nationwide, this solution greatly simplifies your tasks for (1) calculating the right sales tax amount; (2) reporting sales tax for various taxing jurisdictions.

In Elliott 8.6, we are phasing out Vertex Sales Tax in favor of the Avalara Sales Tax solution. It is an add-on solution. Please discuss with Netcellent for the details first if you are interested. For more information, please find relevant KB documents in the following URL: https://support.elliott.com/knowledgebase/topics/151125-avalara-integration

Document Notes

Document Notes allow users to document operation procedures for a particular task. For example, when you are at the Customer Maintenance screen and before you have selected a particular customer, pressing F6 will bring up a list of notes associated with Customer Maintenance. Document Notes are stored in the <ElliottRoot> (e.g. M:\Elliott7) folder and apply to all companies in your Elliott installation.

Document Notes are not the same as Help information (pressing F1). Help information is maintained by Netcellent and is considered as the operator's manual. Document Notes is maintained by you and expand the Help information by adding operation-specific notes that are relevant to you on specific subject matter. To get started using Document Notes, please refer to the following KB article: https://support.elliott.com/knowledgebase/articles/1959244-feature-document-notes

Optionally Eliminate ASCII DATA Files

Due to legacy reasons, there are certain ASCII data files that reside in Elliott's data folder. They are primarily used to store setup and counter information, such as the next order or invoice number. You can optionally implement NTFS security protection to guard all Elliott BTR files against unauthorized access. But the ASCII data files are still vulnerable and may be subject to the attack of, say, ransomware. The damage caused by a ransomware attack is relatively low if you implement the recommended NTFS security. For more information, please see the following KB article: https://support.elliott.com/knowledgebase/articles/654601-elliott-8-directory-structure-and-ntfs-rights

In Elliott 8.6, users can now optionally convert all ASCII DATA files to BTR files to tie up the DAT file loose ends related to NTFS security. This step may also help if you experience duplicate invoice numbers in your system. Note that you should not run the procedure to convert DAT to BTR until you are using Elliott V8.6 for all your users. Once you convert, Elliott is no longer backward compatible with prior Elliott versions which require DAT files to work. Please be aware of the following:
1. Your database needs to be converted to support alphabetic document number first. Please see 8.5 documentation on how to do that.
2. If you are using web services, you must use 8.6 specific web services with proper configuration.
3. Vertex interface may not work after the conversion.
4. F9 users require special procedures. Please speak to Netcellent.
5. No Elliott users should be in the system during this conversion.
6. To convert, use EL860CF.Exe in your <ElliottRoot>\Bin86 folder. Go to the database tab and right click on the company to convert and choose to change database information. Click on the conversion check box to convert. We suggest that you have your DATA_?? folder backed up before conversion. After a successful conversion, you will see two subfolders under the corresponding DATA_?? folder: (1) DAT2BTR: it contains a backup of DAT files that had been converted to BTR; (2) DATSAVE: it contains all other DAT files that have no use in Elliott. We move them to DATSAVE just in case you still need them for any reason.

7. TBL files under DATA_?? folders are work files used by Saturn DYO (Design Your Own). After the conversion, they are moved to <ElliottRoot>\Log\<CompNo> folder. This is a critical step. Please contact Netcellent support if you are not sure how to proceed.

Support Multiple Email Servers and Email Throttling

Even though users can use 8.6 with prior versions side by side, the changes with email servers' settings are significant enough that 8.6 cannot share the same registry configuration settings. Therefore, when users startup Elliott 8.6 for the first time, a popup message will show that the prior version's email server settings are copied to 8.6. From that point, the changes users make in 8.6 are decoupled from the prior version.

Elliott 8.6 now supports a third email server for each user:
- The default email server
- User-specific email server 1
- **User-specific email server 2**

Currently, many people are moving away from in-house hosted email servers and use third party servers instead (like Microsoft-hosted email service, Outlook 365). Generally speaking, the "default email server" option does not work well when your email server is hosted outside. This is because it is difficult to authenticate with one single email account to serve all your Elliott users. Also, hosted email servers generally do not support email relaying. Therefore, it is likely you will use a "user-specific email server" option so each user authenticates differently with the email server.

Starting with Elliott 8.6, we support an additional user-specific email server. The intention is that one will be used to authenticate with a reliable email server like Microsoft to send your regular emails and PDF PostOffice documents. The other one could be used to for mass marketing email through something like SendGrid.

A Microsoft-hosted email server makes a reliable solution, but it restricts you, for example, from sending more than 30 emails per minute. This makes it difficult to use Microsoft email servers for mass email purposes. To avoid this throttling requirement, Elliott 8.6 now allows you to set a delay time (like 2 seconds) per email server so you can still use it for mass email by waiting a little bit between emails.

On the other hand, if you need to send thousands of marketing emails, the 30 emails per minute limit is not a desirable situation. In that case, you may want to use an email server like SendGrid, which does not have a throttling limitation. However, non-throttling servers may have a lesser reputation and some of your emails may go to recipients' junk mail folders. While that may be OK for mass marketing, for PDF PostOffice documents like invoices, that's not okay. In Elliott 8.6, you can now precisely control the type of email server to use according to the different types of email. For more details, please see the following KB article: https://support.elliott.com/knowledgebase/articles/1976928-feature-8-6-email-improvements-multiple-email-s

Just as a side note, we now have the ability to send email with wildcard and multiple attachments. See the following KB article for details:

https://support.elliott.com/knowledgebase/articles/2001877-feature-use-wildcards-and-multiple-paths-in-email

Defer Process Supports PDF PostOffice

In Elliott 8.6, you can now defer all PDF PostOffice documents – a function that was not supported previously. The main reason is because there could be some documents that need to be printed on a physical printer if there's no eContact, no email, or no fax to which you can send the copy. Since deferred processing does not support printing to the physical printer, this creates a dilemma.

In Elliott 8.6, we resolved this problem by adding a special node in Spooled Reports Manager called "ToBePrinted". Any PDF PostOffice document that needs to be printed as a hard copy will be re-routed to this "ToBePrinted" folder during defer processing. Users will need to monitor this folder and print out the hard copy to send it through post mail. For more information, please see the following KB article:
https://support.elliott.com/knowledgebase/articles/1939006-deferred-and-postoffice-processing-in-elliott-v8-6

Invoice Printing Improvements

Invoice printing are improved in the following areas:
1. You can defer invoice printing including PDF PostOffice Support. See previous explanation.
2. Multiple users can perform batch invoice printing without locking. Previously, multiple users can print invoice at the same time as long as it is one invoice at a time. We now support multiple users printing ranges of invoices at the same time.
3. Prevent duplicate invoice number among unposted invoices.

Century Year Cutoff Is Changed from 30 to 50

Elliott internally stores 8-digit dates in the YYYYMMDD format. However, on the screen we only enter a 2-digit year in the format of MM/DD/YY for the sake of convenience. Prior Elliott version used a 30-year century cut-off. That means if you enter 12/31/30, it is equivalent to 12/31/2030. On the other hand, if you enter 1/1/31, it is equivalent to 1/1/1931. With the year 2030 now approaching, we have changed the century year cutoff to 50 which means the last date users can enter is now 12/31/50, which is equivalent to 12/31/2050. If users wish to enter a date outside of the 1/1/1951 to 12/31/2050 range like a birthday, for example -- they can right click on the date field and choose the "Calendar" selector to override.

Requirements

Elliott V8.6 requires the Pervasive® PSQL 13 or 15 database engine to run. Elliott does not support PSQL 14. You may obtain a copy of PSQL from your Elliott reseller if you do not already own a copy.

Client Operation Systems Support

Elliott V8.6 no longer supports Windows XP, Vista, Windows 7, Windows Server 2003 and all versions of Windows 2008. Elliott V8.6 will run on the following client operating systems:

Windows 8 (all versions except Windows RT)
Windows 10
Windows 11

Server Operating Systems Support

Elliott V8.6 will run on the following server operating systems:
Windows 2012 Server
Windows 2016 Server
Windows 2019 Server
Windows 2022 Server
For PSQL 13, you will need to download the latest services pack from Actian to support the
Windows 2019 Server & up.

Unless you login to your Windows server as "Administrator" when running Elliott directly on a
Windows server 2016 and up, you may encounter an NTFS permission issue due to elevated
UAC security precautions. This can be a problem even for an admin equivalent user. We
recommend you follow the Elliott NTFS Security Implementation guidelines to address this
issue. Please see the following KB article for more details:
https://support.elliott.com/knowledgebase/articles/654601-elliott-8-directory-structure-and-ntfs-rights.

Running ElliottV8.6 directly on most of the modern Windows servers (like a terminal server)
requires adding Elliott EXE files to the DEP (Data Execution Prevention) list. Starting with
Windows 2016, the DEP list no longer supports defining EXE for a mapped drive, which is
needed for Elliott. In that case, you have three options:
1. In DEP setup, choose "turn on DEP for essential Windows programs and services only."
   In essence, this turns off the DEP checking for application programs. This is the simplest
   solution to address DEP issues if you intend to use the server for Elliott Database engine
   purposes only. But it may present potential security issues for a remote desktop server if
   you use other applications like Internet browsers or emails.
2. Use the Run Local Mode feature, which we recommend. If you do not want to turn off
   DEP for security considerations, then this is your best solution. Your users can use the
   provided EL860.EXE utility to create the DEP list automatically when creating the
   desktop Run Local icon.
3. Use the Run Hybrid Mode feature. This is like Run Local, but all users on the server
   share the same local copy of Elliott. It saves disk space, but you will encounter Elliott
   update confliction issues. If you have enough disk space on the server, we recommend
   that you use the Run Local Mode.

Microsoft®.NET Framework V4.5

This version of Elliott requires Microsoft.NET Framework V4.5 or above. If it is not already
installed on your computer, you may download it from www.microsoft.com . All new PCs come
preinstalled with Microsoft.NET Framework V4.5 or higher.

Known Limitation

Generally speaking, you should avoid exporting to or importing from a file name or path that contains spaces. But export processor and mass email programs are special cases. The export processor and mass email programs do support exporting to file names and paths with spaces.

Areas Requiring Attention When Upgrading to V8.6

A database conversion to the 8.5/8.6 relational database is required when upgrading to version 8.6 if:

1. You wish to start using alphabetic document numbers and are not already doing so under version 8.5.
2. Your database is not relational compatible (segmented keys) and you want to use Report Desk.
3. You optionally want to convert all ASCII DATA files to BTR files to tie up any loose ends related to NTFS security.

If you have already started to use alphabetic document numbers under version 8.5, you do not need to convert your database to start using Report Desk. However, if you also want to convert all ASCII DATA files to BTR files you will need to run the DAT to BTR Conversion using the Elliott 8.6 configuration utility.

Converting Database Format

Since Elliott's document numbers were indexed as numeric in the past, if you wish to start using the alphabetic document number feature, you will have to go through a conversion process to re-index these columns as strings. This feature is supported starting with Elliott 8.5. If you already converted your database to support alphabetic document numbers, you can skip this section.

If you choose to store alphabetic document numbers in these fields, there will be no reversing back to the Elliott V8.2 database format. It is unlikely that you will need to go back to the Elliott V8.2 database format due to any Elliott V8.6 issues. But it is likely for you to encounter issue with other third-party applications -- like Crystal Reports or web applications -- that access Elliott's data through ODBC or Web Services. These applications treated Elliott's document number columns as numeric in the past, so some sort of conversion is required. You will have to handle these conversions on your own. Please refer to http://support.elliott.com for more help in this area.

The main risk of converting from the Elliott V8.2 to the V8.6 database format is that your external applications may stop working. To minimize your risk, we suggest you take the following steps:

Step 1 - Convert External Applications First
If you use external applications like Crystal Reports or web applications, you should start converting them to become Elliott V8.6 compatible. By default, Elliott V8.6 will create the database name like ELI86DATA. This is based on the new DDF files in the <ElliottRoot>\Bin86\DDF40 folder. We presume that you will copy your existing applications to another area when performing this conversion and point to the ELI86DATA database.

While you perform this conversion, you will continue to let your production Crystal Reports or web applications access your V8.2 database (e.g., ELIDATA) and Web Services.

We tested Starship, and it appears to be working with alphabetic order numbers. We have users who use FedEx or UPS shipping manifest systems, and they report that they had issues when they started using alphabetic numbers. This included error messages like invalid column 'FILLER_001', due to using the new 8.6 DDF, and invalid column 'AA0238' where 'AA0238' is the order number. Their applications may have treated Elliott's order numbers as numeric in the past and used a SQL statement like the following to retrieve Elliott's order data:

       SELECT * FROM CPORDHDR WHERE ORDER_NO = AA0238

If the order number was still numeric, then this would work. But since the ORDER_NO is alphabetic now, they get an error message from PSQL like:

```
<<<<<<<<<<<<<<<<<<<<<<<<
SELECT * FROM CPORDHDR WHEREORDER_NO = AA0238
[LNA][PSQL][SQL Engine]Error in expression: AA0238
[LNA][PSQL][SQL Engine]Invalid column name: 'AA0238'.
>>>>>>>>>>>>>>>>>>>>>>>>
```

To fix this error, they need to put the order number in single quotes as follows:

       SELECT * FROM CPORDHDR WHERE ORDER_NO = 'AA0238'

This type of integration with Elliott is at the third party's discretion, and we can't really predict if they will continue to work when you start using alphabetic order numbers in 8.6. The only way to find out is to test before you convert.

Step 2 –Convert Database, Test or Live

After you are done with your conversion, to begin using theV8.6 database, like ELI86DATA, you will need to convert your V8.2 data to the Elliott V8.6 format. This involves rebuilding the index of many Elliott tables. For example, Invoice Number used to be a numeric field. It is a string field in Elliott V8.6. For all tables that have the invoice number field and are part of the table's index, you will need to convert those indexes to treat invoice numbers as string fields now. You can do so by logging on to the PSQL server as an administrator, bringing up the command prompt and going to the folder <ElliottRoot>\Bin86, then typing the following command:

       DDF2BTR <ElliottRoot>\DATA*.BTR

Substitute<ElliottRoot>with the root directory where your Elliott is installed. For example, use "M:\Elliott7". The DATA is the corresponding DATA folder, so it can be DATA, DATA_02…etc. which corresponds to each company. If you have a custom modification with your database, you will specify the path of your custom DDF with the DDF2BTR command. Use DDF2BTR /? to find out the proper command to do so.

When you perform this conversion on the PSQL server, you can estimate about 100 MB per minute. Do not perform this conversion on the client side since the performance will be significantly slower and less reliable. The actual time may depend on many factors. Only convert your Elliott data when no one is using Elliott, including external applications that access Elliott data like Report Writer or Web Services.

To be safe, you can optionally copy your production company's data to a new company and perform your database conversion in the new company first, then continue your test with external applications that point to this new company. If the external application tests are successful, then convert your production company data. If you have multiple companies, you may perform this conversion one at a time. You don't need to convert all of them at the same time. You cannot convert the V8.6 database back to theV8.2 format once it is converted.

Step 3 –Finally, Start Using Alphabetic Document Numbers
This step is optional. But if you do need to use alphabetic document numbers, you will go to Global Setup -> System -> Comp. Specialized Control. Answer "Y" to "Use Alpha Document Numbers?" Then you will indicate the document for which you intend to use alphabetic values. For example, you may choose to use an alphabetic value for an invoice or sales order number, but not for a purchase order number. Once you are done with this, you will go to the corresponding counter area to set the starting value. For example, if you choose to use alphabetic invoice numbers, you may go to A/R Setup and set the starting invoice number as, say, AAA001. This value is up to you. The important thing to know is if the digit in the starting value is alphabetic, then it will stay alphabetic. If it is numeric, then it will stay numeric. So the digit of value "A" can have a value of A-Z. The numeric digit will have the value of 0-9. Therefore, this format can accommodate a maximum of $26 * 26 * 26 * 10 * 10 * 10 = 17,576,000$ invoices. It is up to you to decide your invoice number format. Alphabetic values can be in the beginning, the middle, or the end. You can also make them all alphabetic. The format will stay the same until you change the starting counter value. Once you have alphabetic values stored in any document number field, you will not be able to go back to Elliott V8.2.

Set Up to Run Local

By default, Elliott V8.6 runs from a local folder even though Elliott V8.6 is installed in the network share. To setup a workstation to run local, you will use Windows Explorer to browse to the folder <ElliottRoot>\Bin86 and run on the EL860.EXE application. This application will ask you which Elliott application to create. By default, Elliott V8.6 (EL860CC.EXE) will be checked. Accept the default and click "Create" to create the icon and Windows Program Menu. Before creating the icon on your desktop, this program will check if the Elliott application programs are already setup in your local folder. If not, it will copy the necessary files from the network to your local folder. Your Elliott programs are copied to your local user profile folder as follows:

   C:\users\<username>\appdata\local\Netcellent\M\Elliott7\bin86

"M" is the network drive where your Elliott V8.6 is installed. "Elliott7" is the folder where your Elliott is installed on the network drive. The startup executable for the Elliott V8.6 Control Center is EL860CC.EXE. Since this path is long, in the future, when you try to run Elliott from your local, simply click on the Elliott V8.6 icon on your desktop to run.

If you are running Elliott on the server, you should select the "Create DEP Exclusions (for Running on Server)" option within the EL860.EXE application. If you do not, it is likely you will receive Error 114 when you start up Elliott. Please refer to the section "Fix DEP Issue If You Get Error 114 When Starting Up Elliott" below for more details on this issue.

Each Run Local program folder is about 300MB in size. Imagine if you have 20 users on a terminal server, and for each user you create a Run Local folder in the user profile. That can take up to 300MB * 20 = 6GB of disk space on your C: drive (the default volume where your user profiles reside.) If this additional disk space consumption is not an issue, then Run Local should be the best solution for you because it has minimal potential for software update conflicts.

If you choose to run Elliott 8.6 from a local folder, then you can update the Elliott 8.6 network folder programs anytime during the day. Once the update is done, the next time the run local Elliott user starts up Elliott 8.6, their local copy will be synchronized automatically. This brings up an important point. Some of your automated processes like deferred processing tend to stay logged in for a long time. To update the local programs used by deferred processing, you need to close and exit the deferred processing programs and start up the program again after your Elliott 8.6 update has been installed. Otherwise, you will continue running on the older version of Elliott 8.5 with your deferred processing.

The Elliott 8.6 installation utility will update DDF files (database schema) for the following folders: <ElliottRoot>\Bin86\DDF40 & <ElliottRoot>\Bin86\DDFRROOT. It does not update <ElliottRoot>\Bin86\DDF40\STDDDF and <ElliottRoot>\Bin86\DDFROOT\STDDDF folders. This is because STDDDF are used to create DSN like ELI86DATA and ELI86ROOT which may be used by the PSQL relational engine during the day when Elliott 8.6 users are in the system. You can login to SUPERVISOR in the evening which will initiate the process of updating DDF to the STDDDF folders. For more information, please refer to the following KB article: https://support.elliott.com/knowledgebase/articles/1993777-updating-ddf-files-after-elliott-v8-6-update

Set Up to Run in Hybrid Mode

Hybrid mode can be used if you run Elliott on the terminal server and you have disk space concerns with the Run Local option. Also, if you have difficulties in resolving DEP issues with Run Local, then you can consider using the Run Hybrid mode. Elliott's Hybrid mode is similar to the Local mode, where the Elliott programs are copied to the local folder. But in Local mode, each user will have their own copy of Elliott programs in their profile folder. With Hybrid mode, all users on the server will share the same programs in a central directory. For example, you may setup the following folder on your server:

     D:\Netcellent\M\Elliott7\Bin86

The easiest way to setup Hybrid mode is to use the EL860.EXE application. By default, the Local Folder will be similar to the following:

     C:\users\<username>\appdata\local\Netcellent\M\Elliott7\bin86.

Click on the button "Change" to choose the drive or a folder. The system will automatically append a path similar to \Netcellent\M\Elliott7\Bin86 to the drive and folder you choose. The key thing to remember about the Hybrid mode path is that it must have the folder name "Netcellent." After that comes the drive letter, the Elliott folder name, and Bin86. You can have whatever you'd like preceding the "Netcellent" part. So, in the case of:

     D:\Netcellent\M\Elliott7\Bin86

it implies the corresponding Elliott V8.6 folder on the network is at "M:\Elliott7\Bin86" and the local Hybrid folder will synchronize with that network folder. The icon for Hybrid mode will be setup for you by the EL860.EXE application. If this is a terminal server, it is a good idea to

choose the option of creating a shortcut for "Everyone on this machine" so you don't need to create Elliott shortcuts for other users on the same terminal server. If you do not want to do that, you can manually create the icon on each user's desktop by using the following sample values:

> Target: D:\Netcellent\M\Elliott7\Bin86\EL860CC.EXE
> Start in: D:\Netcellent\M\Elliott7\Bin86

When you run Elliott on most of the modern servers, you are likely to receive error 114 due to DEP issues. To resolve a DEP problem, you should make sure the check box "Create DEP Exclusions (for Running on Server)" is selected. If you did not do that, you will define the Elliott executable in the DEP exclusion list. Please refer to the section "Fix DEP Issue If You Get Error 114 When Starting Up Elliott" for more details on the DEP issue.

Since all users running in the Hybrid mode can cause updates from the network to local, you should not implement NTFS security to restrict your Elliott users from updating the local Elliott folder. Alternatively, the system administrator should take on the responsibility of immediately going to each server running in the Hybrid mode and starting up Elliott to initiate the synchronization. With Hybrid mode, you should avoid updating Elliott during the day since the next user that brings up Elliott in Hybrid mode will initiate the synchronization process and may run into an update conflict.

Continue to Run in Network Mode

Running Elliott in Network mode is the traditional method and is still supported, but not recommended. In Elliott V8.6, Report Desk may not work correctly when you run from Network. You will have to set up the Elliott icon manually yourself. For example, if your Elliott V8.6 resides in network folder "M:\Elliott7\Bin86," then you will setup the icon with the following sample value:

> Target: M:\Elliott7\Bin86\EL860CC.EXE
> Start in: M:\Elliott7\Bin86

Fix DEP Issue If You Get Error 114 When Starting Up Elliott

When you start up Elliott and immediately receive an Error 114 message, you have a DEP issue. If you are running Elliott on a workstation, chances are you won't encounter DEP issues. On the other hand, if you are running Elliott on the server and DEP is not turned off, you should add DEP Exclusions for Elliott's EXE. Otherwise, you are likely to receive Error 114 when you start up Elliott.

With Elliott 8.6, you can add DEP Exclusions easily with the EL860.EXE utility when you use it to create Elliott Run Local or Hybrid folders and desktop icons. The "Create DEP Exclusions" option will be checked by default. As a result, you don't need to manually define them as in previous Elliott releases. Under most situations, this utility will solve your DEP issue. But in some situations, if the DEP problem still persists, we suggest you use the following solutions:
   (1) Add DEP entries in the Windows DEP Tab. The EL860.EXE can add the DEP Exclusions by adding to the registry the HKLM area, which mimics the Windows DEP tab behavior. To do so requires admin rights. If your user does not have admin rights, then you can add the DEP entries for that user manually through your login. See the section entitled "Add DEP Entries in Windows DEP Tab" below.

(2) In some situations, we have noticed that the DEP entries need to be added to the Registry HKCU area when the user needs to run Elliott from the user profile folder. The EL860.EXE will do so automatically. But if the process fails for security or other reasons, then the user can add to the HKCU registry manually. Look for the KB article entitled "Manage Data Execution Prevention (DEP) Through Registry Editing" on http://support.elliott.com for more details.

(3) If both (1) and (2) fail, implement the Run Hybrid method above. Look for the KB article entitled "The Run Local or Hybrid Feature in Elliott 8.6" on http://support.elliott.com for more details.

Add DEP Entries in Windows DEP Tab

Bring up Computer Properties and go to "Advanced System Settings." Under "Performance," click on the "Settings" button. Go to the "Data Execution Prevention" tab. If you are running Elliott on the server only for the admin user or deferred processing, we suggest that you select "Turn on DEP for essential Windows programs and services only." This will require rebooting your server in order for this change to take effect.

If this is a terminal server, we suggest that you define Elliott V8.6 programs in the exclusion list. Depending on whether you are running in Local, Hybrid or Network mode, browse to the corresponding Elliott programs folder and exclude the following Elliott V8.6 EXE programs from DEP:

    DDF2BTR.EXE
    EL860CC.EXE
    EL860DP.EXE
    EL860EV.EXE
    EL860GE.EXE
    EL860ME.EXE
    EL860P.EXE
    EL860RP.EXE
    EL860TK.EXE
    EL860US.EXE

You may search using keyword DEP or 114 on http://support.elliott.com for more details.

Global User Security Flags

The following flag is new in Password Setup -> User Global Security -> Screen 6:
    Allow to Modify Report Desk's User Def Report
This flag will determine if a user can modify a Report Desk report. You may only want to give admin users this ability. For users who do not have privileges to modify Report Desk, you can allow them to use Report Desk without modifying rights through a new flag in Password Setup -> User Global Security -> Screen 7:
    Allow to Run Report Desk's Reports

Report Desk Databases

By default, Report Desk is not enabled in Elliott V8.6. To enable it, you will go to Global Setup -> System -> Comp. Specialized Control and answer "Y" to the "Use Report Desk" flag. Report Desk will only work with the PSQL relational engine. Your database index must be converted to

be relational compatible. If you are already using alphabetic document numbers, then you must have done the conversion already. If you previously converted your database with the DDF2BTR.EXE utility to make your data compatible with third party relational applications like Crystal Reports or Web Services, then there's no need to convert it again. If you are not sure if you have previously done so, call Netcellent to confirm.

Conversion should take place when there are no other Elliott users in the system, such as in the evening or on a weekend. If you are not using alphabetic document numbers, to convert your database to be relational compatible, do not convert in the Bin86 folder because it will convert your database document numbers to alphabetic format. Instead, go to a folder like Bin82 to convert. For example, bring up a command prompt and change your current folder to <ElliottRoot>\Bin82. Type the following command:

> M:\Elliott7\Bin82> **DDF2BTR.EXE M:\Elliott7\DATA*.BTR**

Substitute the above folder to what's applicable to you. Please go to http://support.elliott.com and search for DDF2BTR to find the article on how to use this utility.

If you use alphabetic document numbers, you should configure your Report Desk to use databases like ELI86DATA??, where ?? is the Elliott company number. You have the opportunity to create these databases during the installation of Elliott V8.6. ELI86DATA?? database is based on V8.6 DDF, which has the document number defined as a string. On the other hand, if you choose numeric document numbers, you should configure your Report Desk to use databases like EliData??. For Elliott database naming conventions, please see KB article: https://support.elliott.com/knowledgebase/articles/1825993-elliott-database-naming-convention

Deferred Processing

If you start using alphabetic document numbers, certain deferred processing tasks setup previously in Elliott 8.2 may not work correctly after converting to Elliott 8.6. These are the deferred processing tasks that prompt for starting and ending document numbers. For example, the COP Invoice Posting task prompts for a starting invoice and an ending invoice number. When you setup deferred processing for this task in Elliott 8.2, you most likely hit the Enter key on the "Starting Invoice Number" field, which causes the system to default the starting invoice number to 000000 and the ending invoice number to 999999. After you start using alphabetic invoice numbers, this task that was setup in 8.2 will not post any alphabetic invoice numbers. The solution is to delete this deferred processing task and set it up again in Elliott 8.6. Review your deferred processing tasks and decide which ones need to be setup again in Elliott 8.6 after you start using alphabetic document numbers.

EL860U.CFG File

If you have the file EL800U.CFG in your existing <ElliottRoot>\Bin or EL850U.CFG in the Bin85 folder, then you should copy it to the <ElliottRoot>\Bin86 folder and rename it as EL860U.CFG. This file contains override information pertaining to EL860S.CFG. The installation utility does not copy or create the EL860U.CFG file for you. Most people do not use this file. But if you use it, you will need to handle this step manually.

Add Read/Execute NTFS Right to Bin86 Folder

When you install Elliott 8.6 on top of an existing Elliott folder, a Bin86 folder is created to store your Elliott 8.6 programs. If you previously implemented NTFS security on your current Elliott folder, you should assign NTFS security to the new Bin86 folder to match what you did to the Bin or Bin85 folder. Typically, you will give the user group "Everyone" the "Read and Execute" rights to the Bin86 folder.

What's New since Elliott V8.5

Below is a list of new features that have been implemented since the Elliott V8.5 general release. There are numerous changes and not all of them are documented in this release note. For complete detailed information about new features added, please visit http://support.elliott.com. Please browse each subject category. Typically, the new features are added at the end of each category. If you are upgrading from an earlier Elliott version, please refer to the "What's New since Elliott V8.2, V8.1, V8.0, V7.5, V7.4, V7.3, V7.2, V7.1 and V7.0" sections for more information.

Elliott Support Portal at https://support.elliott.com

You may find out the details of all features discussed in this section on https://support.elliott.com by doing a keyword search. This is Elliott's support portal, where you can search for documents and Knowledge Base articles. You may also suggest new ideas of how Elliott can improve using this portal. All new features since 2015 have been documented here. We encourage you to visit this website as your first stop to find more information about Elliott Business Software.

Attribute Improvements

Date Field: In the past, attribute date fields are entered through the calendar date selector. While this interface offers advantages in some situations, it can slow down users in other scenarios. Starting with Elliott 8.6, the attribute date field is a free-form field with optional calendar date selector support. See the following KB article for details: https://support.elliott.com/knowledgebase/articles/1940344-feature-manual-date-entry-in-attribute-maintenanc

Read Only Field: In Attribute Code maintenance, the system now offers the option to make a field "Read Only." These are fields that are updated outside of the Attribute user interface in Elliott. This includes Elliott special logic, CSV import and third-party applications. See the following KB article for details: https://support.elliott.com/knowledgebase/articles/2003647-feature-read-only-attribute-fields

Event Improvements

We added new events in the following areas: (1) In COP -> Utility -> Void Invoice; (2) Print Transfer Ticket in Inventory Transfer Management; (3) Post Transfer Ticket in Inventory Transfer Management; (4) COP Sales Order Import to trigger price mismatch event; (5) Added SERCHG Event by item to Serial Number Correction Tool.

In addition, we now have the ability to send an email to the offending user who created the locking event, in addition to the system admin. See KB article for details: https://support.elliott.com/knowledgebase/articles/2001814-feature-send-an-email-to-the-user-locking-a-recor

View Database Activity Log Improvement

The Elliott Database activity log is a very useful debugging tool to find what went wrong by displaying the last 1,000 IO entries. User can view it with the "Detail" button when they receive an IO error message in Elliott. If a user has the privilege, he/she can also see it in Elliott when the system is waiting for users to enter data by pressing the CTL-SHIFT-D. This feature is now improved so that (1) Users can see it even if there's no IO error or waiting for screen input (for example, if a program falls into an infinite IO loop, the user can press CTL-SHIFT-D to see what caused the loop); (2) If a user does not have the privilege, he/she can ask the system admin to enter the SUPERVISOR's password to access the Database Activity Log. See KB article: https://support.elliott.com/knowledgebase/articles/1975899-feature-view-database-activities-through-control

Improved EDI Integration with SPS

Various new features are added to improve EDI interface with SPS. This includes: (1) Allow to export serial number with EDI ASN; (2) Support ASN export with Ship-Via Group option; (3) The ASN export will now perform an integrity check between line item quantity and box quantity; (4) A preview option is added for ASN export so users may see if there are any potential issues before proceeding with export; (5) Tracking numbers are added to the ASN preview and journal to help users spot potential issues; (6) Support EDI 846 (inventory advise); (7) Support importing of routing numbers with Amazon (ARN - Amazon Routing Number) or other trading partners to enforce the exporting of ASN (Advanced Shipping Notice) to include routing number; (8) Allow to export canceled or closed orders and line items to send 855 order acknowledgement; (9) Optionally map bill-to address in ESS to support importing EDI orders from eCommerce site.

Country Validation

Designed to support Avalara Sales Tax Interface, users can now optionally validate the country field in customer, ship-to and orders even if not using Avalara. Please see the following KB article for more details: https://support.elliott.com/knowledgebase/articles/1994359-feature-country-validation

Payment Preparation Improvement

We made the following improvements: (1) In Generalized Payment Selection, the user is allowed to include or exclude up to five vendor types; (2) In the past, a negative voucher (credit memo) amount had to be used in its entirety when performing payment selection. We now support using partial negative voucher amounts. This is important as a means of handling a large prepaid amount that will be used to pay over a period of time for multiple invoices.

Cash Receipt Detail Import to Support Discount and Reference

A/R Cash Receipt Detail Import can now support cash discount and reference information. See KB article: https://support.elliott.com/knowledgebase/articles/1987624-feature-support-for-discount-amount-and-referenc

Stock Status Inquiry 12 Months History Improvement

Added TOR (Turn Over Ratio), QtyOH, and OH-Value columns to the Quote/OH screen in the Location History Information of Stock Status Inquiry. Users are now able to see past or projected future Qty OH and value in Location History drill down. See the following KB article: https://support.elliott.com/knowledgebase/articles/1954687-feature-addition-of-tor-qtyoh-ohvalue-to-stock

UPC Code Assignment

Elliott supports sequential UPC code assignment. In the past, it was assumed that users would not exhaust all 99,999 UPC codes that were assigned to the users. Thus, the system would not function correctly when the sequence number reached 99,999. It is now changed so the system can recycle and use previously assigned UPC codes starting from 0 as long as the old UPC codes are removed from the obsoleted items. See the following KB article: https://support.elliott.com/knowledgebase/articles/1964479-feature-next-upc-code-assignment-enhancement

User Defined CSV Export (Inventory Report Writer)

We added the following features: (1) Added QI(Qty Invoice – from Invoice History) data type to the Item User Defined CSV Export. QI is different from the QS (Qty Sold) column in that the QI is from invoice history line items and QS is from item location history. If you choose to post a kit by component per Global Setup. QS only shows the kit's component quantity sold. To find out kit parent's quantity sold, you will need to use QI. See the following KB article: https://support.elliott.com/knowledgebase/articles/1956892-feature-added-qi-qty-sold-in-invoice-history-d; (2) Added parameter 23, Include Web Items Only. The primary purpose is to export EDI 846 items only where the web item flag should be "Y". Also, the web item flag is now allowed to be blank which means default to the product category's web flag. See the following KB article: https://support.elliott.com/knowledgebase/articles/1965542-feature-allow-to-select-by-include-web-items-onl; (3) We added flexibility to the CSV column heading by supporting variables like <MON>, <MONTH>, <MM>, <YR>, <YEAR> and <TODAY>. See KB article: https://support.elliott.com/knowledgebase/articles/1998769-feature-add-flexibility-to-user-defined-csv-colu

Create Hold Trx When Posting Inventory Trx

Users could previously optionally hold inventory upon warehouse receiving posting. We now add the ability to optionally hold inventory upon posting of receiving or transfer inventory transactions. See the following KB article for details: https://support.elliott.com/knowledgebase/articles/1961836-feature-create-hold-trx-when-posting-inventory-t

---

Item Audit Trail Improvement

The Item Audit Trail now tracks changes to Product Category, User Defined Code, and Material Cost Type. These fields are added to item file audit trail report and new columns in CSV export. See the following KB article for details:
https://support.elliott.com/knowledgebase/articles/1998457-feature-support-product-category-user-defined-c

Product Category and Material Cost Type Location Account Files CSV Import

For those users who use Product Category and Material Cost Type Location Account Files, they will find this feature extremely useful. We now support CSV importing of Product Category Location Account File (for Sales & COGS accounts) and Material Cost Type Location Account File (for Inventory accounts). Hence, you can define these accounts in Excel first, then save it as CSV before importing them to Elliott. See the following KB articles for details:
https://support.elliott.com/knowledgebase/articles/1998523-feature-import-for-product-category-location-a
https://support.elliott.com/knowledgebase/articles/1998538-feature-import-for-material-cost-type-location-a

Inventory Transfer Edit List Improvement

Improved Edit List in I/M Inventory Transfer Management to add support for Starting/Ending Trx Date and Starting/Ending Complete Date, status, user-ID. Also, selection parameters have been added to the report legends. The Edit List has been improved to print more information.

Transfer History Inquiry

In I/M -> Processing ->Transfer Processing, user can now view transfer history. See the following KB article for more details:
https://support.elliott.com/knowledgebase/articles/1990252-feature-transfer-history-inquiry-drill-down-fr

Hold Transaction File Edit List Improvement

Added parameters Starting/Ending Location and Show Inventory Info to the I/M, Maintenance, Hold Trx File, and List. The print layout was changed to accommodate the inventory information for Qty on Hand, Qty on Order, Qty Allocated, and Qty Backordered. We also add parameters to support multi-bin environment. See the following KB articles for more details:
https://support.elliott.com/knowledgebase/articles/1965773-feature-add-new-selection-parameters-to-hold-trx

Invoice Printing with Deferred Processing

Many users have started using Elliott PDF PostOffice to email invoices, so the physical hardcopy was not needed anymore. In Elliott 8.6 invoice printing, the system allows the Invoice Date field to be the system date by default to support deferred processing.

Sales Desk

Introduced "Trial Lock" to prevent locking events when Sales Desk creates a sales order. This can happen when multiple salespeople are placing an order for the same item. Some are placing an order for the item master location and some are placing an order for an alternate inventory location. The new "Trial Lock" logic will prevent a deadlock by locking the Item record and the associated Inventory Location records before creating the order from Sales Desk.

Relax Credit Checking

Added to Global Setup, Cop-Ctl, and Credit Check & Release, in the "Auto Hold Order/Warning" window, a prompt for an option over/under percentage of credit limit. Also, instead of checking maximum amount in period 2 – 4 individually, the system now allows to lump periods 2 – 4 together to compare to a total amount. All programs that perform a credit check have been changed to use the new values. See the following KB article for more details: https://support.elliott.com/knowledgebase/articles/1973832-feature-relaxed-auto-credit-release

Order, Quote and Shipment Acknowledgement Emails Improvement

We have finite control of who will receive PDF PostOffice Document emails per eContact setup. In Elliott 8.6, we are applying this ability to Order, Quote, and RMA acknowledgement e-mails as well. In addition, order, quote and shipment emails can now optionally support HTML format to make them look professional. For more details, please reference the following KB article: https://support.elliott.com/knowledgebase/articles/1988287-feature-add-html-version-of-order-acknowledgemen

Credit Card Receipt in HTML Email Format

We now support PDF PostOffice for credit card receipts in HTML format. The email recipients are based on the eContacts defined in order, customer, ship-to and salesman like PDF PostOffice. This feature is fully supported in Elliott v8.6 and partially supported for v8.5. For more details, please reference the following KB article: https://support.elliott.com/knowledgebase/articles/2002810-feature-html-credit-card-email-receipt-using-pdf

Order/Invoice Inquiry to Drill Down to Serial Number in the Box

If you are using the Shipment Verification and serial number feature, you can now drill down to a list of serial numbers of a box and an item by pressing the Enter key when the box/item is selected in the Shipment Information screen from Order/Invoice History Inquiry. See the following KB article for more details: https://support.elliott.com/knowledgebase/articles/1925311-feature-order-invoice-inquiry-box-serial-number

Pick Ticket Printing

The following features are added: (1) Added the ability to print pick tickets using a Ship Via group instead of one Ship Via code. See the following KB article for more details: https://support.elliott.com/knowledgebase/articles/1947604-feature-print-pick-tickets-using-a-ship-via-grou ; (2) Added an option in Global Setup so the pick ticket is only printed if quantity on hand for the line items is greater than zero. See the following KB article for details:

https://support.elliott.com/knowledgebase/articles/1971609-feature-do-not-print-pick-ticket-if-there-s-insu

Packing List Improvement

Optionally allow printing of one page of packing list per box if Shipment Verification is enabled. Please see the following KB article for more details: https://support.elliott.com/knowledgebase/articles/1927858-feature-packing-list-by-box-one-box-per-page

VICS Bill of Lading Improvements

Various features have been improved for VICS Bill of Lading. This includes: (1) Optional check if all orders in the BOL are processed through Shipment Verification before completing the BOL to prevent integrity issues; (2) Optionally create default notes like sales order number in the BOL instructions area based on either Notes or Attribute of the Sales Order.

Import Shipment Tracking Number

We added the ability to import shipment tracking numbers for sales orders through a CSV file. This is to support third party shipment manifest solutions that cannot integrate with Elliott in real time. Please see the following KB article for more details: https://support.elliott.com/knowledgebase/articles/1945888-feature-add-support-for-import-of-shipment-track

Make COP Popup Notes Read-Only

In COP Order Entry Header, Line Item, Sales Desk and Shipment Verification screens, users can optionally force the matched customer, ship-to or item notes to pop up per Global Setup. We now have the option in Global Setup to make the popup notes be "Read-Only" so users on the COP side cannot modify the corresponding customer or item notes. For more details, please reference the following KB article: https://support.elliott.com/knowledgebase/articles/1960360-allow-auto-popup-customer-or-item-notes-in-order-e and https://support.elliott.com/knowledgebase/articles/1960363-popup-customer-or-ship-to-notes-in-shipment-verifi

Invoice History Archive

Elliott Invoice History is an invaluable database for users to look up history. For this reason, users are hesitating to purge it when it may contain decades of data. On the other hand, when invoice history databases become big, reports (including third party reports like Crystal) may perform poorly. In Elliott 8.6, when you purge COP Invoice History, the data is now moved to the archive history database. You still can view and print invoice history from the archive database. But on the other hand, since your COP Invoice History becomes leaner after the purge, third party invoice history reports can run faster.

Support Colorado Retail Deliver Fee (RDF)

Effective July 1, 2022, Colorado requires merchants with nexus to charge an RDF which is a special retail delivery fee as a separate line item on the order when delivering to addresses in Colorado. Elliott will support this requirement by putting sales orders on hold if the order is

taxable and a ship-to in Colorado without RDF item. See the following KB article for details: https://support.elliott.com/knowledgebase/articles/1992472-feature-colorado-sales-tax-retail-delivery-fee

Support Printing Shipping Label for Attributes

We now support printing up to five additional attribute fields in the COP Shipping Label based on the customer EDI profile. This allows user to select various types of attributes including, customer, item, order, and order line items. This is especially important if the information required to print on the label is from EDI where we imported as order or line-item attributes. For more details, please reference the following KB article: https://support.elliott.com/knowledgebase/articles/1983702-feature-print-attrib-1-through-attrib-5-on-ucc-1

Mass Synchronize PO Line-Item Cost with Current FOB Cost

In the PO Line-Item Screen, F5 Windows, users can now press "F4" to synchronize PO line-item cost with the current FOB Cost. This can be useful in a situation when a vendor is raising a price across the board. For more details, please reference the following KB article: https://support.elliott.com/knowledgebase/articles/1976103-feature-allow-to-mass-update-po-line-item-cost

Enforcing Item Attribute Setup through Warehouse Receiving

We added Global Setup fields to allow an Event to be triggered if an item Attribute does not exist when a Warehouse Receiving is entered. This is to be used when certain attributes are required for an item (like dimension) which you may not have it until the item is received. The event will notify the proper person(s) to get the needed attribute setup for the item. For more details, please reference the following KB article: https://support.elliott.com/knowledgebase/articles/1933636-feature-warehouse-receiving-missing-item-attribu

PO EDI Integration with Vendor EDI Profile

Traditionally, Elliott EDI integration with SPS is through the COP module for customers. We now also support EDI with vendors through the PO module. To make the process automated, we now support vendor EDI profiles. Please see the following KB article: https://support.elliott.com/knowledgebase/articles/1979115-feature-po-export-by-vendor-edi-profile

Operator Inquiry Real Time Monitor

The Operator Inquiry function has been improved to refresh real time data so users can put on a big-screen TV in the shop to show the progress status of each operator: https://support.elliott.com/knowledgebase/articles/1990795-feature-operator-inquiry-auto-refresh-to-show-la

Allow Print Label in Production Order Entry Screens

We added a "Print label" function to the Production Order Entry screens. For more details, please reference the following KB article: https://support.elliott.com/knowledgebase/articles/1964290-feature-added-print-labels-function-to-product