opentext™

OpenText™ Exstream™

Switch Reference

Design and Production Documentation

Release 16.6.0

OpenText™ Exstream
Switch Reference
Rev.: 2019-Apr-30

This documentation has been created for software version 16.6.0.

It is also valid for subsequent software versions as long as no new document version is shipped with the product or is published at https://knowledge.opentext.com.

Open Text Corporation
275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111
Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440
Fax: +1-519-888-0677

Support: https://support.opentext.com
For more information, visit https://www.opentext.com

Contents

Chapter 1: Switch Reference

This guide provides information about the switches that you use to run the production engine, customize database administration, or specify packaging options. Some of these switches are required and other are optional.

You can use these switches within a control file or from the command prompt. For information about using a control file to run the production engine, see *Preparing Applications for Production* in the Exstream Design and Production documentation .

The switches in this guide are organized by their purpose. For example, switches related to running the Exstream production engine, can be found in the Engine Switches section. Each of the following sections contains a list of relevant switches:

- Database Administrator switches
- Designer switches
- Design Manager switches
- Exstream Application Manager switches
- Engine switches
- LiveEditor Switches
- Logic Designer switches
- Packager switches
- Web application switches

You can also view an alphabetical list of available switches.

Note: The information provided in this guide might not be complete and is subject to change. Not all switches are available for all versions of Exstream Design and Production.

Chapter 2: Building a Control File

A control file is a reusable format file that helps you increase productivity and save on costs. A control file lets you control the behavior of the engine by specifying required and optional engine switches in a centralized location. The format in which you write a control file varies depending on the platform you use. Any engine switch can be placed in a control file. When you use a control file, the only command necessary at the prompt is `CONTROLFILE=command`. Since you must run the production engine executable from the command prompt, options in the control file override options placed directly on the command line.

To build a control file, you must complete the following tasks:

1. Writing a Control File to Manage Application Production
2. Validating Your License Key in the Control File to Avoid Engine Errors
3. Executing a Production Engine Run with a Control File

2.1 Writing a Control File to Manage Application Production

Writing a control file is the most optimal method you can use to manage application production. You can use a control file for packaging or for a production run. Enter the commands necessary for the production or packaging run into the control file. A package control file is created the same way as a control file for the engine. The only difference is that packaging switches are used instead of engine switches.

For more information about packaging with a control file, see *Preparing Applications for Production* in the Exstream Design and Production documentation.

You enter switches into a control file using the same format that you use on the command line. All switches must have the following features:

- Preceded by a dash (-) or slash (/), depending on the platform
- In capital letters

> **Caution:** If you include double quotation marks around a file name in a control file, You receive errors in the message file, that the specified file could not be found or created.

When you write a control file, you must select the engine switches necessary for the production run and add them to the file. The engine switches you use in your control file depends on the needs of your application and performance factors related to the engine.

To write an engine control file for use on the Windows, UNIX, or Linux platforms:

1. Open a text editor program, such as Windows Notepad.
2. Create a file for the control file. For example, `CTRL1.txt`. You can use any extension you prefer.
3. On the first line, enter the mandatory PACKAGEFILE switch with the argument specifying the package file you want to compose.
4. Specify any additional switches you want to include in the control file, listing each on its own line.

   Caution: If you include double quotation marks (`""`) around a file name in a control file, you receive errors in the message file, that the specified file could not be found or created.

5. Save the file.

   For a list of engine switches, see *Switch Reference* in the Exstream Design and Production documentation.

2.1.1 Writing Control Files for the z/OS Production Engine

Writing a control file for the z/OS production engine differs from that of a control file for the Windows or UNIX production engine. If you are operating the production engine on the z/OS platform and you want to write a control file, you must complete the following z/OS-specific tasks:

- Specifying the z/OS Name in the Control File
- Using a Fully Qualified Name in a z/OS Control File
- Using Rules to Specify a File Name in a z/OS Control File

The following example of a z/OS control file, used with dynamic images, can help you better conceptualize the examples used in the z/OS-specific tasks in this section.

```
-OUTPUTFILE=DD:EXOUTPUT

-PACKAGEFILE=DD:PACKAGE

-MESSAGEFILE=DD:EXMSGS

-REPORTFILE=DD:EXREPORT

-RUNMODE=PRODUCTION
```

```
-TRACKIN=DISABLE

-TRACKOUT=FILE

-REPORT=CUSTOMER

-ENGINEDSN=P390LOC

-IMPORTDIRECTORY=EXSTREAM.TIFF
```

This control file sets the import directory using the IMPORTDIRECTORY switch to `EXSTREAM.TIFF`.

Note: You must license the Dynamic Content Import module to use the IMPORTDIRECTORY engine switch.

Specifying the z/OS Name in the Control File

To specify the z/OS name in the control file, you can use a combination of a Data Definition (DD) name and member name.

1. Set the DD to a Persistant Data Store (PDS) without specifying the member name.

   For example:

   ```
   //TIFFIMPO DD DSN=P390A.EXSTREAM.TIFF,DISP=SHR
   ```

2. Set the DD as the IMPORTDIRECTORY name and use the original data file.

   For example:

   ```
   -IMPORTDIRECTORY= DD:TIFFIMPO
   ```

   Note: The z/OS engine is distributed with a control file. It has a built-in DD name:

   `DD:TIFFIMPO(SOUTHERN)`

   This is a valid z/OS file name and can help you get started.

Using a Fully Qualified Name in a z/OS Control File

To use a fully qualified name, you must put single quotation marks around the entire name. If the full name is `MYHLQ.EXSTREAM.TIFF (member)`, you must put the first quotation mark in the control file. For example: `-IMPORTDIRECTORY= 'MYHLQ.EXSTREAM.TIFF`

The trailing quote mark must be in the data file:

C Robert Stevens 123456 9/25/2000 (ASCII1)

T ABC (SOUTHERN)' (ASCII1)'

T DEF (SOUTH)' (ASCII2)'

z/OS automatically appends the user's High Level Qualifier (HLQ) to the front of the data set. You do not need to use a fully qualified name because z/OS will automatically add the current HLQ.

Using Rules to Specify a File Name in a z/OS Control File

You can also build the entire name using a rule to specify a file name (including the leading and trailing quote), where needed. In this case, you would not use the `-IMPORTDIRECTORY` switch at all. The following are options you can use to build the file name:

- Get the name of the fully qualified PDS from an initialization file.
- Get the member name only from the data file, but not parentheses or quotation marks.
- Build the fully qualified name using a rule by adding the following:
  - Leading quote
  - Open parenthesis
  - Trailing parenthesis and quotation marks

2.2 Validating Your License Key in the Control File to Avoid Engine Errors

To avoid production engine errors, you must validate your license key in the control file you use. Specifying the license key in your control file lets the engine confirm that all features used in the application are valid. If you use an expired key, or a key that does not carry all of the licensing features that appear in your application, then you receive error messages and the engine immediately stops processing.

2.2.1 Inserting the License Key on Windows and UNIX Platforms

If you are using a package file with an expired license key, you must insert a valid license key into the control file or the engine does not compose the package file.

Insert the following switch in your control file:

```
-KEY=value
```

The value is the text string of your license key. This value overrides the license key in the package file. You can copy the key characters from your Exstream Key File (EKF) by double-clicking on the license key, which, in the Windows environment, opens the license key in a program of your choice. In other environments, you can open the file using a text editor.

```
-KEY=value
```

When you run the engine with the control file, the new license key is used with the old package file.

When you copy this license key into your control file it must be an exact match. To prevent errors, copy and paste the license key into the control file you are using.

Using the CONTROLFILE Switch on Windows and UNIX Platforms

You can call other control files from within a control file. This option is available to make license key management easier. For example, you can have a control file using only the KEY switch. When you need to change the license key, change it in this control file and then you can link all your control files to the KEY control file.

2.2.2 Inserting the License Key on the z/OS Platform

In z/OS, you must use uppercase characters. However, the license key must consist of the original mixed case characters. For example:

Your key: XxxxXxXXXxxxxXXxXxXxXxxXXxxXXxxxXXxxxxXxXxxxXXXxxx

The license key must be specified exactly like this in the control file.

If you edit your z/OS control file on z/OS, the characters you enter for the license key are converted to upper case. For example:

```
-KEY=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

This is not a valid license key, as the key must be mixed case. To ensure your license key is mixed case, set the z/OS command CAPS OFF. If you do not use this command, all text is automatically converted to uppercase.

Using the KEYPART Switch with Large Keys on the z/OS Platform

On z/OS systems you can use the KEYPART switch to split a key string into multiple parts. The KEYPART switch is used for keys that are too large to fit on a single line of Job Control Language (JCL). When the engine processes multiple KEYPART switches, it combines the parts in the order supplied to form a single key string.

2.3 Specifying Multiple Package File Operations in the Control File

You must follow a specific syntax to ensure you specify multiple package file operations in the control file. In normal engine processing, or in step one for output sorting, all the packages for an application can be specified in a list in the control file.

The application package file must be specified first. For example:

```
-packagefile=App_master.pub
```

```
-packagefile=4_4_2002__Doc.pub
```

```
-packagefile=New_Phase_4_Camp_and_docs.pub
```

```
-packagefile=One_changed_doc_and_one_new_doc.pub
```

You can also specify a text file containing the preceding list. For example:

```
-APP_PACKAGEFILES=MyPackageList.txt
```

In Post-Sort mode, available with the High-Volume Delivery module only, package files can be specified in the same manner, but if you are using the Application Consolidator module to consolidate more than one application, you must use one `APP_PACKAGEFILES` switch per application.

For more information about output sorting, Post-Sort mode, and the Application Consolidator module, see *Creating Output* in the Exstream Design and Production documentation.

2.4 Executing a Production Engine Run with a Control File

To execute an engine run with a control file, you must enter the CONTROLFILE switch on the command line to specify the location of your control file.

To run the engine using a control file:

1. Working from the Exstream executable's directory, enter one of the following commands to start the engine run from the command prompt:
   - In Windows, the command is `prodengine`.
   - In UNIX, the command is `Engine`.
2. Enter a space after the `Engine` or `prodengine` command.

3. After the space, enter the `CONTROLFILE` command to specify the location of the control file.
4. Press `ENTER` to start the engine run.

For more information about executing production engine runs, see *Preparing Applications for Production* in the Exstream Design and Production documentation.

Chapter 3: Database Administrator switches

The following switches are for use with the Database Administrator utility (`DBAdmin.exe`).

Chapter 4: Designer switches

The following switches are for use with `Designer.exe`.

Chapter 5: Design Manager switches

The following switches are for use with `DesignManager.exe`.

5.1 Database Connection and Login

DSN (DBAdmin)

ACCESSDB (DBAdmin)

EXSTREAMUSER

EXSTREAMPASSWORD

5.2 Load

XOBLOAD

XOBFOLDER

XOBPASSWORD

XOBLOADENVIRONMENTAL

XOBLOADSYSTEMSETTINGS

XOBKEEPAPPROVEDSTATUS

XOBLOG

XOBALLVERSIONS

5.3 Unload

XOBUNLOAD

XOBUNLOADTARGET

XOBUNLOADAPPEND

XOBUNLOADAPPROVEDONLY

Chapter 6: Exstream Application Manager switches

The following switches are for use with `EAM.exe`.

EAMKEY

EAM_PORT_NUMBER

EAM_SERVER_INSTALL

6.1 Database Connection and Login

DSN (DBAdmin)

ACCESSDB (DBAdmin)

EXSTREAMUSER

EXSTREAMPASSWORD

Chapter 7: Engine switches

The following switches are for use with `Engine.exe` or `Engine_DBCS.exe`.

7.1 Backward compatibility switches

The following switches can be used to revert behavior to a previous version of Design and Production. You can use the ENABLE_BACKWARD_COMPAT engine switch to collectively apply all of the backward compatibility switches that have been implemented since a previous version of Exstream Design and Production that you specify.

DONT_CORRECT_HTML_HEADER_FOOTER_POSITIONS_AND_COLOR

DONT_FORCE_NEW_PARAS_WITH_STYLES_IN_HTML

DONT_MAKE_DOCX_TOC_ADJUSTMENTS

DONT_POPULATE_RTF_CELL_BORDERS

DONT_RECALC_SIZE_TO_END_SECTION

DONT_REMOVE_EMPTY_LINES_WITH_SPECIAL_RETURNS

DONT_RESET_COMPONENTS_IN_SECTION_DOCS

DONT_USE_CUSTOMER_LANGUAGE_TO_PARSE_DATES

DONT_USE_REPEAT_ROW_NUMBER_FOR_EMBEDDED_OBJS

DONT_USE_RESPONSIVE_DESIGN_IN_HTML

HTML_IGNORE_UNSET_FONT_COLOR

HTML_IMPORT_IGNORE_16_4_UPDATES

HTML_IMPORT_IMAGE_PADDING_BEFORE_BORDER

HTML_USE_OLD_TABLE_WIDTH_CALC

IGNORE_HTML_TABLE_CELL_FRAME_SEGMENTS

KEEP_WRONG_DOCX_IMPORT_FONT_COLOR_HANDLING

NO_SMOOTH_FILLS

ONLY_USE_REPEATING_HEADER_ROWS_FOR_RTF_TABLES

OVERRIDE_HTML_TD_DEFAULT_PADDING

PCL_FULL_COLOR_HIGHLIGHT

REVERT_SECTION_TABLE_MULTICOPY_FIX

SKIP_MULTIPLE_HEADER_ROWS_IN_RTF

USE_7BIT_EMAIL

USE_OLD_AXIS_SCALING_FOR_SAME_MAX_MIN

USE_OLD_EMAIL_LISTS

USE_OLD_FONT_LABEL_CHART_SYMBOLS

USE_OLD_HTML_EMAIL_TABLE_WIDTHS

USE_OLD_HTML_NUMBERING_FONT_AFTER_BULLETS

USE_OLD_MERGED_CELLS_HEIGHT_CALC

USE_SHORT_TABLE_HEADER_ACCUMULATORS

7.2 Customer processing switches

CUSTOMERLIST

CUSTOMERLISTFILE

DISABLE_RECIPIENT_COPIES

DONT_USE_CUSTOMER_LANGUAGE_TO_PARSE_DATES

DRIVERLISTFILE

END

NTH

START

7.3 Data switches

AGGREGATE_DATA

ALLOW_EMPTY_REF_FILES

CONVERT_AS_IS_TO_PPD

COPYINPUT

DSNMAP

FILEMAP

FORCE_ONDEMAND_SHUTDOWN_ON_INPUT_ERROR

IGNORE_SPEC_ARRAY_ELEM_ON_XML_DATA_AREAS

LIVE_HONOR_EXCLUSION_RULES

LOADREFFILE

MSG or MSGCHANGE

REFLOOKUPALWAYS

SECUREOUTPUT

SISOINSERTION

TWO_DIGIT_CENTURY_LIMIT

VARSCOPE

7.4 DBCS switches

CONTROLFILEENCODING

DONT_RIGHT_ALIGN_EMBEDS_IN_RTL_LAYERS

MSGENCODE

PDF_DBCS_STD_FONTS_ANSI

RTL_VAR_BREAK

SISOINSERTION

7.5 Design switches

The following sections provide information about switches related to elements in your design. Many of these switches change the appearance of these elements in the output. Some of these switches allow you to make specific design alterations outside of Designer, and others allow you to revert to previous default behavior following an upgrade. For information on using switches to revert to previous default behavior after an upgrade, see Backward compatibility switches.

7.5.1 Charts (Traditional)

The following switches affect the way that traditional charts are included in the output:

3D_BAR_CHART_SUPPRESS_ZERO

BAR_CHART_WHITESPACE

BAR_DRAW_XAXIS_AFTER_UPGRADE

BAR_USE_FONT_AXES_FOR_LABEL

BAR_ZERO_AXIS_ON_TOP

CENTER_BOTTOM_CHART_LABEL

CENTER_TOP_CHART_LABEL

CHART_DRAW_FRAME_AFTER_UPGRADE

DEFAULT_LABEL_TICKS_TO_NUMSERIES

FORCE_LABEL_COLLISION_AVOIDANCE

NO_FILL_ROTATION

NO_SMOOTH_FILLS

USE_OLD_AXIS_SCALING_FOR_SAME_MAX_MIN

USE_OLD_FONT_LABEL_CHART_SYMBOLS

7.5.2 Images

The following switch affects the way that images are included in the output:

HTML_IMPORT_IMAGE_PADDING_BEFORE_BORDER

7.5.3 Layout

The following switches affect the way that content is laid out in the design:

DEFAULT_LAYER_FLOW_FRAMES_RESPECT_LANGUAGE

DONT_RIGHT_ALIGN_EMBEDS_IN_RTL_LAYERS

FORCE_MASTER_TEXTDRAW_MODE

HONOR_FULL_JUSTIFICATION_WITH_LINE_FEED

OLD_FRAME_LANGUAGE_CHECKS

SPLIT_MODE

TEXT_TRUNC_COMPAT_27435

USE_LEGACY_SUPERSCRIPT_SUBSCRIPT_IMPORT

VERSION_SIX_LINE_BREAKS

7.5.4 Tables

The following switches affect the way that tables are included in the output:

AUTOSIZE_RTF_TABLES_FOR_VARIABLES

BREAK_PAGE_OTHER_ROWS_PLACED

COMBINE_ADJACENT_POLY_FILLS

CREATE_RTF_TABLE_OBJECTS

FORCE_ZERO_HEADER_VAR_INDEX

HONOR_DASH_DOT_LINE_STYLES

HTML_USE_OLD_TABLE_WIDTH_CALC

IGNORE_HTML_TABLE_CELL_FRAME_SEGMENTS

OVERRIDE_HTML_TD_DEFAULT_PADDING

RETAIN_EMPTY_SECTION_START

SECTIONBREAKCOUNT

SET_COLUMN_BALANCE_DEVIATION

SKIP_MULTIPLE_HEADER_ROWS_IN_RTF

USE_OLD_HTML_EMAIL_TABLE_WIDTHS

7.6 Dynamic content import switches

The following switches are related to content imported dynamically at run time. To use these switches, you must have licensed the Dynamic Content Import module.

7.6.1 All Import Formats

CREATE_RTF_TABLE_OBJECTS

CUSTOMER_RESET_IMPORTS

DISABLE_DRAWIMAGESFIRST

DONT_RESOLVE_VARIABLES_IN_PLAIN_TEXT_IMPORT

FLUSH_TIFF_FOR_PS

GS_PAUSE

IMPORTDIRECTORY

PRESERVE_TAGGED_PARA_FORMAT

OTDSRESOURCEID

OTDSURL

PLACE_LATE_NONFLOWING_OBJECTS_LAST

RESMAN_PLAIN_TEXT

TEST_TIFFS

USE_LEGACY_SUPERSCRIPT_SUBSCRIPT_IMPORT

USE_OLD_SORTADD_FOR_VAR_IMPORTS

USE_RTF_PAGE_MARGINS

USE_RTF_PARA_PROPS

7.6.2 Connecting to CAS

MGWAPPDOMAIN

MGWPASSWORD

MGWTENANT

MGWURL

MGWUSER

OTDSRESOURCEID

OTDSURL

7.6.3 DOCX Import

ALLOW_PLACEHOLDER_CONTENT_TO_FLOW

DOCX2DXF_LOCATION (correct subcategory? or create one for DXF Export?)

7.6.4 DXF Import

ALLOW_PLACEHOLDER_CONTENT_TO_FLOW

7.6.5 HTML Import

CLEAR_FLOATS_FROM_HTML_IMPORT

HTML_IMPORT_IMAGE_PADDING_BEFORE_BORDER

7.6.6 PDF Import

DSB_PDF_ACC_TAG_IMPORT

PDFIMPORT_USE_CROPBOX_DIMENSIONS

PDFTOPDF_LANDSCAPE_FIT_COUNTERCLOCKWISE

PDFTOPS_SKIP_SPLASH_RASTER

PDF_IMPORT_RASTER_PDF

PDF_PASSTHROUGH_ACMA_CONVERT

PDF_PASSTHROUGH_AS_EPS

PDF_PASSTHROUGH_AS_LZW

PDF_PASSTHROUGH_AS_PDF

PDF_PASSTHROUGH_AS_RGB_JPEG

PDF_PASSTHROUGH_USE_HIGH_PRECISION

PDFTOPDF_HONOR_ROTATIONS

PDFTOPS_IGNORE_PAGEROTATION

SPLIT_LARGE_PDF_IMPORTS

7.6.7 Troubleshooting

CAPTURETESTDATA

MSGENCODE

MSGLANGUAGE

MSGRESOURCE

GSTEMPDIR

REPORT

REPORTFILE

RULEANALYSISREPORT

7.7 Dynamic data access switches

ALLOW_PLACEHOLDER_CONTENT_TO_FLOW

CLOSE_OUTPUT_ON_BREAK

CONNECTORMAP

COPYINPUT

DDAFILEMAP

DDALLOC

DDAMESSAGE

DDAOUTPUT

DDAOUTPUTFILE

EBCDIC_CODEPAGE

FORCE_POST_PROCESSOR_BLOCK_WRITES

LICENSE_PATH

LICENSE_WAIT

LOGDDA

LOGDDANOHEADER

LOGERROR

LOGFILTER

NODDAOUTPUTHEADER

NOREPORTS

NORTHSOUTHINDEX

NUM_CUSTOMERS_HOLD_MUP_IMPORTS

ONDEMAND

ONDEMAND_CLOSE_OUTPUTS_ON_SHUTDOWN_ONLY

OVERRIDE_DEMO_WATERMARK_POSITION

PDL

REALTIME

REALTIME_CLOSE_OUTPUTS_ON_SHUTDOWN_ONLY

REALTIME_LICENSE_MODE

SET_RTF_DEFAULT_LANGUAGE

SINGLECUSTOMER

SPLIT_LARGE_PDF_IMPORTS

SUPPRESS_NEWLINE_ON_DDA_WRITE

WRITEOUTPUTHEADER

7.8 General engine switches

CONTROLFILE

DISABLE_SEARCH_KEY_CONVERT

ENABLE_BACKWARD_COMPAT

ENABLE_SHELL_COMMANDS

FONTDIRECTORIES

KEY

KEYFILE

LANGUAGE

LICENSE_PATH

LICENSE_WAIT

LOCALE

MAX_DECOMPRESSED_PACKAGE_BYTES

MSG or MSGCHANGE

OUTPUTFILES

PACKAGEFILE

RUN

RUNDATE

RUNMODE

TEMPORARY_DIRECTORY

TWO_DIGIT_CENTURY_LIMIT

USE_LOCAL_CONTEXT_BASED_MISSING_NODE_CHECK

USE_UNPREFIXED_INTERNAL_HTML_IDS

7.8.1 Mainframe

BLOCK_OUTPUT_MVS

DDALLOC

EBCDIC_CODEPAGE

KEYPART

MVS_DATACLASS

TAPEUNIT

VMCHECK

ZOS_USE_X15_NEWLINE_CHAR

7.8.2 Performance

MEMORYCACHE

MEMORYSAVE

MEMORYSTATS

7.8.3 Troubleshooting

ALWAYS_COMPOSE_ADDINS_DURING_ENGINE_RUN

CAPTURETESTDATA

MESSAGEFILE (Engine)

MESSAGELEVEL

READABILITY

RULEANALYSISREPORT

RUNSUMMARY

VERBOSE

XMLMESSAGEFILE

7.9 High-volume switches

The following switches are related to high-volume production and document delivery. To use these switches, you must have licensed the High-Volume Delivery module.

7.9.1 General

APP_PACKAGEFILES

BYPASS_BUNDLE_COVER_RULE_ON_INITIALIZE

BYPASS_QUEUE_RULE_ON_INITIALIZE

CODE128_MAP_SPECIAL_CHAR

DISABLE_MIXEDPLEX_LOCK

DISABLE_TLE_MUP_PAGE_BREAKS

DISABLEPRINT

DISABLESORT

DO_NOT_FORCE_POSTSORT_MUP_BREAK

DO_NOT_RUN_LATE_PAGE_RULES

NORTHSOUTHINDEX

OPENFILES

PRESERVE_LATE_COUNTS_FOR_REPRINTS

REMOVE_USER_BACKS_WITH_BLANKS

REVERT_SYS_BUNDLING_COUNTS

SETPRINTERBINCONTENTS

SETQUEUEINSERTER

SET_QUEUE_VARS_FROM_LAST_QUEUE

OUTPUTFILES

USEQUEUE

7.9.2 Barcodes

BARCODE_PLACEMENT_HONOR_PAGE_ORIENT

MAILMARK

7.9.3 Campaigns and Messages

CHECK_POSTAGE_FOR_POOL_PAGES

DYNAMSG/DYNAMESSAGE

IGNORE_PCM_FOR_MARKETING_PAGE_LIMITS

INCLUDE_JURISDICTION_EXPIRATION_DATE

MARK_LAST_SKIPPED_FRAME_ARTICLE_FRAME_FILL_MODE

RECALCULATE_INSERT_WEIGHT_PER_CUSTOMER

TIMESTAMP_MESSAGES

TRACK_OVERRIDE_VERSION_CHECK

TRACKCOMMITINTERVAL

TRACKDSN

TRACKFILE

TRACKIN

TRACKOUT

TRACKPWD

TRACKSCHEMA

TRACKSTATUSINTERVAL

TRACKUID

TRK_CAMP_AND_MSG_USED_ONLY

7.9.4 Output

See Output switches.

7.9.5 Sorting and Bundling

DISABLE_BUNDLE_INCREMENT_FOR_NON_QUALIFIED_QUEUES

POSTSORTQUEUE

SORTDATA

SORTID

SORTINDEX

SORTINDEXLENGTH

7.9.6 Performance

ALWAYS_SET_SYS_TABLECOLUMN

CACHETABLE

DEFAULT_PAGE_ALLOCATIONS

STOPERROR

USE_OLD_SORTADD_FOR_VAR_IMPORTS

7.10 Output switches

The following sections provide information about switches related to output from your application. To use these switches, you must have licensed the appropriate output driver

module. For more information about each output type, see *Creating Output* in the Exstream Design and Production documentation.

7.10.1 General Output

The following switches can be used with any output type.

CHECK_NEW_LINE_FONT_SPACING

COMPOSE_FOOTNOTES_ON_ALL_QUEUES

DRAWIMAGESFIRST

FOOTNOTE_MARGIN

FORCE_MIXED_CASE_PANTONE_NAMES

MAXOVERLAYS

OUTPUTDIRECTORY

OUTPUTFILE

OVERRIDE_DEMO_WATERMARK_POSITION

PDL

PNG_IGNORE_RESOLUTION

REMOVE_USER_BACKS_WITH_BLANKS

RTL_VAR_BREAK

STDDELIVERYFILE

TRIM_LINE_END_SPACES

7.10.2 AFP

You must have licensed the AFP pDriver module to use the following switches. For more information about AFP output, see *Creating Output* in the Exstream Design and Production documentation.

AFP_BASIC_DBCS

AFP_DYNAMIC_IMG_SCALETOFILL

AFP_PURE_BLACK_FORCE_OCA

AFP_REMOVE_DEFAULT_NOPS

AFP_SEPARATE_DBCS_CODEPAGES

AFP_SEPARATE_INSERTER_BARCODES

AFP_TILE_RASTER_FILLS

AFP_USE_VARIABLE_LENGTH_BNG

AFP_WORKFLOW_BUNDLE

AFPFONT or AFPFONTSTART

AFPMCF1

COPYGROUPEACHPAGE

FULL_NAME_AFP_BMO_EMO

IGNORE_LOC_SEARCH_KEY_LEN

NO_AFP_DRAWRULE

PDF_PASSTHROUGH_ACMA_CONVERT

PDF_PASSTHROUGH_AS_PDF

PDF_PASSTHROUGH_USE_HIGH_PRECISION

USE_MCF_FOR_AFP_REF_FONTS_FOR_ANY_NAME

7.10.3 DLF

DISABLE_FLOW_TARGETS_FOR_LIVE

FORCE_RESET_TIMING_OF_DLF_IMPORT_SECTION_VARS

IGNORE_SPEC_ARRAY_ELEM_ON_XML_DATA_AREAS

7.10.4 HTML and HTML (Email)

You must have licensed an HTML eDriver module to use the following switches. For more information about HTML and HTML (email) output, see *Creating Output* in the Exstream Design and Production documentation

ALLOW_MERGE_PARAS_IN_HTML

ALWAYS_HONOR_DESIGN_PARAGRAPHS_FOR_HTML

CONSTANT_HTML_BASESHIFTS

DONT_ENSURE_DIV_FOR_EMBED_IN_HTML

DONT_FORCE_NEW_PARAS_WITH_STYLES_IN_HTML

DONT_MOVE_BLANK_PARA_NEWLINE

DONT_USE_RESPONSIVE_DESIGN_IN_HTML

HIDE_STYLE_CLASS_IN_HTML_OUTPUT

HTML_DTD_LOCATION

HTML_FORCE_DESIGN_IMAGE_NAMING

HTML_KEEP_EMPTY_TRAILING_PARAS_31274

HTML_USE_COMPLETE_DOCTYPE

INCLUDE_HEIGHT_FOR_HTML5_TABLES

REMOVE_MCXML_EMBED_SPACERS

USE_7BIT_EMAIL

USE_NATIVE_ELEMENTS_FOR_ACCESSIBLE_HTML_LISTS

USE_OLD_EMAIL_LISTS

USE_OLD_HTML_NUMBERING_FONT_AFTER_BULLETS

USE_UNPREFIXED_INTERNAL_HTML_IDS

METAPAGE

7.10.5 Microsoft Word (DOCX)

You must have licensed the DOCX pDriver module to use the following switches. For more information about DOCX output, see *Creating Output* in the Exstream Design and Production documentation.

DISABLE_DOCX_PAGENUM_RESET

EXPOIWRAPPER_CLASSPATH

FORCE_NO_DOCX_TEXT_FRAMES

USE_DOCX_NUMPAGES_VAR

7.10.6 PCL

You must have licensed the PCL pDriver module to use the following switches. For more information about PCL output, see *Creating Output* in the Exstream Design and Production documentation.

PCL_EDGE_TO_EDGE_OFFSETS

PCL_FULL_COLOR_HIGHLIGHT

7.10.7 PDF

You must have licensed the PDF pDriver module to use the following switches. For more information about PDF output, see *Creating Output* in the Exstream Design and Production documentation.

DONT_READ_BULLETS_AND_NUMBERS

DSB_PDF_ACC_TAG_IMPORT

ENCODING_VECTOR_OVERRIDES

FORCE_OLD_PDF_FONTBOX

PDF_DBCS_STD_FONTS_ANSI

PDF_QUOTERIGHTFIX

PDF_QUOTESINGLEFIX

PDF_RESOLVE_RESOURCE_RECURSION

PDFTOPDF_HONOR_ROTATIONS

PDFTOPDF_LANDSCAPE_FIT_COUNTERCLOCKWISE

REDRAW_BOTTOM_CELL_BORDER_FOLLOWED_BY_FILLED_CELL

USE_FONT_FONT_MAPPING_FOR_EMBEDDED_PDF_FONTS

USE_REPEATING_DOC_TO_INDEX_BOOKMARK

7.10.8 Rich Text Format (RTF)

You must have licensed the RTF pDriver module to use the following switches. For more information about RTF output, see *Creating Output* in the Exstream Design and Production documentation.

CREATE_RTF_TABLE_OBJECTS

ONLY_USE_REPEATING_HEADER_ROWS_FOR_RTF_TABLES

RTF_IGNORE_TRAILING_PAR

SET_RTF_DEFAULT_LANGUAGE

SKIP_MULTIPLE_HEADER_ROWS_IN_RTF

USE_LEGACY_SUPERSCRIPT_SUBSCRIPT_IMPORT

7.10.9 XML (Content)

You must have licensed the XML (content) eDriver module to use the following switches. For more information about XML (content) output, see *Creating Output* in the Exstream Design and Production documentation.

FORCE_OLD_CONTENT_XML_NATIVE_FORMAT

7.10.10 XML (Multi-Channel)

You must have licensed the XML (multi-channel) eDriver module to use the following switches. For more information about Multi-Channel XML output, see *Creating Output* in the Exstream Design and Production documentation.

ALWAYS_HONOR_DESIGN_PARAGRAPHS_FOR_HTML

USE_OLD_MULTI_XML_IMAGE_SIZE

7.11 Resource management switches

CREATE_RESMANAGERFILE

CREATE_REUSABLE_RESMANAGERFILE

DYNAMICIMGNAMEMAP

MAX_DECOMPRESSED_PACKAGE_BYTES

REPORT_RESMAN

RESMANAGERFILE

RES_ONLY_OVERRIDE

REUSABLE_RESMANAGERFILE

TEMPORARY_DIRECTORY

7.12 Variable switches

ALWAYS_COUNT_FIRST_ENVELOPE

ALWAYS_SET_SYS_TABLECOLUMN

DEFAULT_LAYER_FLOW_FRAMES_RESPECT_LANGUAGE

DEFAULT_REF_VARS

DISABLE_BUNDLE_INCREMENT_FOR_NON_QUALIFIED_QUEUES

FORCE_READ_ALL_DFI_FROM_IMPORTDIR

FORMAT_TAGGED_TEXT

INITVARSET

NOVARDEL

PRESERVE_BLANK_ELEMENTS_IN_JOIN_FUNCTION

PRINT_PAGE_RANGE_USE_PCM

RTL_VAR_BREAK

VARDELLIST

VARSCOPE

VARSET

ZERO_DATE_IS_ERROR

Chapter 7: Web application switches

The following switches are for use with Exstream web applications.

7.12.1 Exstream Content Author switches

ALL_THEMES_DISQUALIFIED

Chapter 8: LiveEditor Switches

The following switches are for use with `LiveEditor.exe`.

8.1 LiveEditor

DSNMAP

FILEMAP (LiveEditor)

INITVARSET

USE_PRINT_ORDER_FOR_DLF_INITIAL_ENGINE_TABLES

8.2 Live Fulfillment

HONOR_LIVE_SUBST_TIMING_IN_FULFILL

8.3 Live Processing

CREATE_SECTION_DATA_FOR_REF_VARS_IN_LIVE

DRIVERLISTFILE

IGNORE_SPEC_ARRAY_ELEM_ON_XML_DATA_AREAS

LIVE_HONOR_EXCLUSION_RULES

SECTION_DOCS_HONOR_EXCLUSION_FOR_LIVE

Chapter 9: Logic Designer switches

The following switches are for use with `LogicDesigner.exe`.

Chapter 10: Packager switches

The following switches are for use with `Packager.exe`.

10.1 Packaging

MGWURL

OTDSRESOURCEID

OTDSURL

PACKAGEFILE

PACKAGEPROFILE

PACKAGETYPE

STARTDATE

UPLOAD_PACKAGE_TO_CAS

WRITE_JSON

10.2 Troubleshooting

MESSAGEFILE (Packager)

Appendix A: Reference

This reference section includes all of the available engine, database, and packaging switches, listed in alphabetical order.

A.1 Switches 0-9

A.1.1 3D_BAR_CHART_SUPPRESS_ZERO

Used with: Engine

Use the 3D_BAR_CHART_SUPPRESS_ZERO switch to suppress the bar when the data represented has a value of zero. This switch is valid only for 3D traditional bar charts.

Syntax

```
-3D_BAR_CHART_SUPPRESS_ZERO
```

This switch does not have any arguments.

Supported in

7.0.608 and later

A.1.2 5_0_OVERLAPPING_HEADER_BEHAVIOR

For backward compatibility after you upgrade from version 5.0 to versions 6.1 and later, use the 5_0_OVERLAPPING_HEADER_BEHAVIOR switch to maintain legacy overlapping header behavior, which contained enhancements that affected the placement of currency symbols in overlapping headers.

Syntax

```
-5_0_OVERLAPPING_HEADER_BEHAVIOR
```

This switch does not have any arguments.

Supported in

6.1.007 and later

A.2 Switches A-D

A.2.1 ACCEPT_PRIVACY_STATEMENT

The ACCEPT_PRIVACY_STATEMENT switch acknowledges that you have read and accept the Exstream Design and Production anonymous data collection and usage statement provided at https://analytics.exstream.com/privacy/en-US/index.html.

This switch is required for upgrading or creating a database from the command line for use with Exstream Design and Production versions 9.5.301 and later, or Exstream Application Manager (EAM) versions 3.5.301 and later.

> **Note:** Beginning with Exstream version 16.2.0, which provides a 64-bit design environment, EAM is no longer supported.

Syntax

```
-ACCEPT_PRIVACY_STATEMENT
```

This switch does not have any arguments.

Supported in

9.5.301 and later

A.2.2 ACCESSDB

Use the ACCESSDB switch to open an existing Microsoft Access database without creating a data source name (DSN) for the database. The file name can use either the .mdb or .accdb extension but keep in mind that Microsoft Access versions earlier than 2007 support only the .mdb extension.

You must use either the ACCESSDB switch or the DSN switch to run database maintenance from the command line. You can optionally use the DBAUTHENTICATION switch to specify the database authentication method.

You can also use the ACCESSDB switch in conjunction with the COPYDB switch and the COPYTOACCESSDB switch to copy content from one Microsoft Access database to another.

Syntax

```
-ACCESSDB=<databaseName>
```

Argument	Required	Supported values
`databaseName`	Yes	The fully qualified file name of the Microsoft Access database that you want to open

Example

```
-ACCESSDB=C:\Temp\Sample.accdb
```

A.2.3 ACCESSDB (DBAdmin)

The ACCESSDB switch opens an existing Microsoft Access database without requiring you to create a DSN. The file name can use either the `.mdb` or `.accdb` extension.

Requirements

Either the ACCESSDB switch or theDSN switch is required to run database maintenance from the command line.

Syntax

```
-ACCESSDB=<Access database path and file name>
```

Use the following arguments with the ACCESSDB switch:

Argument	Required
`<Access database path and file name>`	Yes
`DBAUTHENTICATION`	No

Example

```
-ACCESSDB=C:\Temp\Sample.accdb
```

A.2.4 AFP_BASIC_DBCS

Use the AFP_BASIC_DBCS switch to produce regular DBCS output that matches IBM specifications and recommendations. Applications that use the switch produce coded fonts that are referenced in pages that use MCF1 structures. You must make the following selections on the **Resource Management** tab of the AFP output object:

- From the **Font type** list, select **Bitmap**.
- From the **Resource inclusion** list, select **Used resources only**.

Syntax

```
-AFP_BASIC_DBCS
```

This switch does not have any arguments.

Supported in

8.0.305 and later

A.2.5 AFP_DYNAMIC_IMG_SCALETOFILL

Use the AFP_DYNAMIC_IMG_SCALETOFILL engine switch to optimize AFP devices when your application imports images into an empty frame at run time and you do not want the engine to retain the aspect ratio of the images. The AFP_DYNAMIC_IMG_SCALETOFILL engine switch lets the engine fill the frame without using the image's aspect ratio.

Syntax

```
-AFP_DYNAMIC_IMG_SCALETOFILL
```

This switch does not have any arguments.

Supported in

6.1.030 and later

A.2.6 AFP_FORCE_OLD_TEXT_LINE_COLOR_PROCESSING

When you create AFP OCA B/W output, all of the objects in the output are set to one of 16 OCA colors, based on a standard color conversion. Prior to Exstream Design and Production version 5.0.058, objects such as text and simple lines were limited to the 8-color OCA range. This difference in OCA color processing can cause gray text and lines to appear lighter in prior versions (with the 8-color OCA range) and can cause the same content to appear darker in Exstream Design and Production version 5.0.058 and later (with the 16-color OCA range).

The AFP_FORCE_OLD_TEXT_LINE_COLOR_PROCESSING switch forces the engine to process output using the old behavior, and applies color to objects in the output by using the 8-color OCA range.

Syntax

```
-AFP_FORCE_OLD_TEXT_LINE_COLOR_PROCESSING
```

This switch does not have any arguments.

Supported in

5.0.058 and later

A.2.7 AFP_GOCA_SET_LINE_END

Use the AFP_GOCA_SET_LINE_END engine switch to specify an AFP set line end command on all Graphics Object Content Architecture (GOCA) objects in the output. When you run the engine to produce AFP output that contains barcodes and that includes the NO_AFP_DRAWRULE engine switch in the control file, the engine generates the barcodes as GOCA lines. The resulting output produces incorrect barcodes. Using the AFP_GOCA_SET_LINE_END engine switch, you can specify a draw rule value for the AFP set line command that specifies the line ends on barcodes in the output.

Syntax

```
-AFP_GOCA_SET_LINE_END=<line_end_command>
```

Arguments	Required	Supported values
`<line_end_command>`	Yes	• `default`—A value of 0 for the line end. • `flat`—A value of 1 for the line end. • `square`—A value of 2 for the line end. • `round`—A value of 3 for the line end.

Example

```
-AFP_GOCA_SET_LINE_END=flat
```

Supported in

6.1.035 and later

A.2.8 AFP_NO_EXTERNAL_ALGORITHM_SPECIFICATION

Use the AFP_NO_EXTERNAL_ALGORITHM_SPECIFICATION engine switch to ensure that PDF images display as expected in JPEG-based AFP output. When you run an application that imports DLF content at run time that includes PDF images and produces JPEG-based AFP output, some of the PDF images may appear blank in the resulting AFP output.

Syntax

```
-AFP_NO_EXTERNAL_ALGORITHM_SPECIFICATION
```

This switch does not have any arguments.

Supported in

- 8.0.328 and later maintenance releases
- 8.6.104 and later versions

A.2.9 AFP_NO_INVERT_RESMAN_TIFF

For backward compatibility after you upgrade from Exstream Design and Production version 7.0, use the AFP_NO_INVERT_RESMAN_TIFF engine switch when you produce AFP output with the following conditions:

- The application contains dynamic black-and-white TIFF images.
- Encoding for the black-and-white TIFF images is set to 0-is-black.
- **Used resources only** is selected from the **Resource inclusion** list on the **Resource Management** tab of the AFP output object.

The colors in the images appear to be reversed in the output.The AFP_NO_INVERT_RESMAN_TIFF engine switch prevents Exstream from automatically reversing the images at run time.

Syntax

```
-AFP_NO_INVERT_RESMAN_TIFF
```

This switch does not have any arguments.

Supported in

8.0.323 and later

A.2.10 AFP_PURE_BLACK_FORCE_OCA

Use the AFP_PURE_BLACK_FORCE_OCA switch to avoid color management on pure black objects in full-color AFP output. This switch forces the production environment to use the AFP OCA black color specification for pure black graphic and text objects instead of process color pure black.

Syntax

```
-AFP_PURE_BLACK_FORCE_OCA
```

This switch does not have any arguments.

Supported in

6.1.009 and later

A.2.11 AFP_REMOVE_DEFAULT_NOPS

Use the AFP_REMOVE_DEFAULT_NOPS switch to remove the Exstream Design and Production standard informational no-operation (NOP) identifiers at the top of an AFP file. The AFP_REMOVE_DEFAULT_NOPS switch excludes the following NOPs from appearing in AFP output:

- Version
- Date
- Registered key owner
- Application

Removing the Exstream Design and Production standard informational NOPs from the top of the AFP file can prevent errors when using AFP files with other external systems or architectures, such as the IBM Content Manager OnDemand architecture.

Syntax

```
-AFP_REMOVE_DEFAULT_NOPS
```

This switch does not have any arguments.

Supported in

8.0.310 and later

A.2.12 AFP_REMOVE_MDR_DOC_LEVEL

For backward compatibility, after you upgrade from version 7.0 or earlier, use the AFP_REMOVE_MDR_DOC_LEVEL switch when you produce AFP output if you selected the **Use AFP color management architecture** check box on the **Resource Management** tab of the AFP output object.

Using the AFP_REMOVE_MDR_DOC_LEVEL switch prevents the mapped data resource (MDR) objects at the document level from causing the printer to stop responding due to a memory overload. The AFP_REMOVE_MDR_DOC_LEVEL switch removes the MDR objects from the document level of the output.

Syntax

```
-AFP_REMOVE_MDR_DOC_LEVEL
```

This switch does not have any arguments.

Supported in

8.0 and later

A.2.13 AFP_REMOVE_MPO_DOC_LEVEL

For backward compatibility after you upgrade from Exstream Design and Production version 7.0 or earlier, use the AFP_REMOVE_MPO_DOC_LEVEL switch when you produce AFP output if you selected the **Use AFP color management architecture** check box on the **Resource Management** tab of the AFP output object.

Using the AFP_REMOVE_MPO_DOC_LEVEL switch prevents the map page overlay (MPO) records at the document level from causing the printer to stop responding due to a memory overload. The AFP_REMOVE_MPO_DOC_LEVEL switch removes the optional MPO records from the document level of the output.

Syntax

```
-AFP_REMOVE_MPO_DOC_LEVEL
```

This switch does not have any arguments.

Supported in

- 7.0.635 and later maintenance releases
- 8.0.319 and later versions

A.2.14 AFP_SEPARATE_DBCS_CODEPAGES

Use the AFP_SEPARATE_DBCS_CODEPAGES switch to use multiple code pages in DBCS applications where DBCS text exceeds the 256-character limit in a single code page. Additionally, you can use the AFP_SEPARATE_DBCS_CODEPAGES switch when you want to retain the textual meaning of the DBCS characters used in your application.

This switch lets you select the naming convention you want to use for multiple code pages. Some systems integrated with Exstream Design and Production require separate code pages for DBCS applications to allow for greater flexibility with text retrieval in AFP and/or font conversion. For example, a DBCS output archived in an external system such as the IBM OnDemand environment might require that AFP fonts be mapped to fonts that are available in another format. Separate code pages allow the user to define font mappings in post-production.

The AFP_SEPARATE_DBCS_CODEPAGES switch is valid only when the following two options have been enabled on the **Resource Management** tab of the AFP output object properties:

- From the **Font type**list, select **Bitmap**.
- From the **Resource inclusion**, select **Used resources only**.

Syntax

```
-AFP_SEPARATE_DBCS_CODEPAGES=<character_naming_convention>
```

Arguments	Required	Supported values
`<character_naming_convention>`	Yes	• U—use a Unicode-based naming conventiom. • IBM—use the IBM standard naming convention.

Example

```
-AFP_SEPARATE_DBCS_CODEPAGES=U
```

Supported in

8.0.305 and later

A.2.15 AFP_SEPARATE_INSERTER_BARCODES

Use the AFP_SEPARATE_INSERTER_BARCODES switch to place each inserter barcode into a separate presentation text (PTX) record along with a no-operation (NOP) locator description of the barcode.

Syntax

```
-AFP_SEPARATE_INSERTER_BARCODES
```

This switch does not have any arguments.

Supported in

3.5 and later

A.2.16 AFP_TILE_RASTER_FILLS

Use the AFP_TILE_RASTER_FILLS switch to optimize AFP file size for output containing raster images as backgrounds in table rows. The AFP_TILE_RASTER_FILLS switch reduces the file size by enabling tiling in raster images. When using the AFP_TILE_RASTER_FILLS switch, always make sure to use the minimum size for tiles.

Use this switch in conjunction with the COMBINE_ADJACENT_POLY_FILLS switch.

Syntax

```
-AFP_TILE_RASTER_FILLS
```

This switch does not have any arguments.

Supported in

7.0.617 and later

A.2.17 AFP_USE_VARIABLE_LENGTH_BNG

Use the AFP_USE_VARIABLE_LENGTH_BNG switch to override the default 8-byte Begin Named Group (BNG) name and allow BNG names of up to 255 bytes.

Syntax

```
-AFP_USE_VARIABLE_LENGTH_BNG
```

This switch does not have any arguments.

Supported in

5.0.008 and later

A.2.18 AFP_WORKFLOW_BUNDLE

Use the AFP_WORKFLOW_BUNDLE switch to create Begin Named Group/End Named Group (BNG/ENG) pairs with the same value for each customer in a bundle for consumption by InfoPrint Workflow. There is no default value for this switch.

If you use both bundle and customer search keys with the AFP_WORKFLOW_BUNDLE switch, the output does not contain nested BNG/ENG pairs. This is due to a limitation in InfoPrint Workflow when processing nested BNG/ENGs.

Syntax

```
-AFP_WORKFLOW_BUNDLE=<bundle_number>
```

This switch requires the starting bundle number as its argument.

Example

```
-AFP_WORKFLOW_BUNDLE=10
```

Supported in

3.5 and later

A.2.19 AFPFONT or AFPFONTSTART

Use the AFPFONT switch or the AFPFONTSTART switch to generate Mapped Coded Font (MCF) records. You can use these switches to overlay data using StreamWeaver.

Syntax

```
-AFPFONT=<number>
```

or

```
-AFPFONTSTART=<number>
```

The only argument for these switches is a number (between 1 and 110) to use as the starting number in all MCF records. The default value is 1.

Example

```
-AFPFONT=1
```

or

```
-AFPFONTSTART=1
```

Supported in

4.5.201 and later

A.2.20 AFPMCF1

Use the AFPMCF1 switch to use old MCF type 1 font records in AFP instead of the newer MCF2 records. Type 1 has been retired according to AFP specifications, but many composition tools continue to support it.

Syntax

```
-AFPMCF1
```

This switch does not have any arguments.

Supported in

6.0 and later

A.2.21 AGGREGATE_DATA

Use the AGGREGATE_DATA switch if you are using any data section built-in functions to perform data aggregation. You are not required to use this switch if you use the data aggregation features in Design Manager and Designer.

You also use the switch if you are using an XML input file that requires a specific data hierarchy to define relationships between data. By default, the engine does not recognize that a third level tag is a child of a second level tag. Restructuring the data file might require the addition of redundant data sections. Instead of restructuring, use the AGGREGATE_DATA engine switch.

When you use the switch, the engine reads all of the data for a customer, including any sections, and holds the data in memory until it reaches the next customer. The switch requires additional memory and processing time, but it lets you use the original data file hierarchy.

Syntax

```
-AGGREGATE_DATA
```

This switch does not have any arguments.

Supported in

- version 7.0 and later

A.2.22 ALL_THEMES_DISQUALIFIED

Use the ALL_THEMES_DISQUALIFIED switch to tell the Exstream engine how to respond when a customer does not qualify for any available Content Author themes.

Syntax

```
-ALL_THEMES_DISQUALIFIED=<customerAction>
```

The only required argument for this switch is `<customerAction>`. This switch has no default value.

Supported `<customerAction>` values:

- `SKIP`—The customer is not included in the output and the engine issues a warning in the message file.
- `WARNING`—The customer is included in the output, but the output will not include any content that was created in Content Author, and the engine issues a warning in the message file.
- `SEVERE`—The engine stops processing the application when it encounters a customer that does not qualify for Content Author themes, and issues a severe error in the message file.

Example

```
-ALL_THEMES_DISQUALIFIED=WARNING
```

Supported in

16.4.0 and later

A.2.23 ALLOW_EMPTY_REF_FILES

If you must include an empty reference file, use the ALLOW_EMPTY_REF_FILES switch to prevent the engine from issuing an error.

Syntax

```
-ALLOW_EMPTY_REF_FILES
```

This switch does not have any arguments.

Supported in

3.5.082 and later

A.2.24 ALLOW_MERGE_PARAS_IN_HTML

In Exstream Design and Production versions earlier than 16.3.0, blank text paragraphs in designs were merged with the following text paragraph. This resulted in both of them being placed within the same parent tag (such as <p> or <td>) in HTML and HTML (email) output. For example:

```
<table>
<tr>
<td>
   <span></br></span>
   <span>Text</span>
</td>
...
```

This produced differences in HTML and HTML (email) output when compared to the design. The issue was corrected in Exstream Design and Production version 16.3.0. In the resulting HTML, the blank text paragraph is placed within a separate parent tag (such as <p> or <td>). For example:

```
<table>
<tr>
<td>
   <span></br></span>
</td>
<td>
   <span>Text</span>
</td>
...
```

For backward compatibility after you upgrade to 16.3.0, use the ALLOW_MERGE_PARAS_IN_HTML switch to merge blank text paragraphs with the text paragraph that immediately follows. This switch will be automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 16.2.999 or earlier.

> **Note:** This switch is related to DONT_MOVE_BLANK_PARA_NEWLINE.

Syntax

```
-ALLOW_MERGE_PARAS_IN_HTML
```

This switch does not have any arguments.

Supported in

16.3.0 and later

A.2.25 ALLOW_PLACEHOLDER_CONTENT_TO_FLOW

Use the ALLOW_PLACEHOLDER_CONTENT_TO_FLOW switch to allow the engine to repaginate DOCX or DXF files that contain variables when they are imported at run time. When the DOCX or DXF file is imported at run time, the imported document can then flow to accommodate the additional text that is added by the variables within the document. This switch can be used only for DOCX or DXF files that are imported at run time using a placeholder document.

Syntax

```
-ALLOW_PLACEHOLDER_CONTENT_TO_FLOW=<repaginate>
```

Arguments	Required	Supported values
`<repaginate>`	Yes	• `YES`—the engine to repaginate the imported DOCX or DXF file.

Supported in

- 8.6.114 and later maintenance releases
- 9.0.109 and later versions

A.2.26 ALWAYS_COMPOSE_ADDINS_DURING_ENGINE_RUN

Use the ALWAYS_COMPOSE_ADDINS_DURING_ENGINE_RUN engine switch to force all add-ins to be composed at run time.

If you do not use this switch, Exstream add-ins that appear in HTML output are not automatically composed when the Exstream engine runs, and are instead composed when the HTML output is opened in a web browser. However, if an add-in is not composed during the

engine run, then no messages related to the add-in are written to the engine message file, which could make it harder to troubleshoot if there is an issue.

Keep in mind that using this switch can impact engine performance, and might increase the amount of time it takes to process your application.

Syntax

```
-ALWAYS_COMPOSE_ADDINS_DURING_ENGINE_RUN
```

This switch does not have any arguments.

Supported in

16.3.0 and later

A.2.27 ALWAYS_COUNT_FIRST_ENVELOPE

Use the ALWAYS_COUNT_FIRST_ENVELOPE engine switch when you produce output from an application that includes banner pages and uses application consolidation to force the engine to increment system variable banner counts for the first bundle, even though the application does not include a banner in the first bundle.

Syntax

```
-ALWAYS_COUNT_FIRST_ENVELOPE
```

This switch does not have any arguments.

Supported in

7.0.637 and later

A.2.28 ALWAYS_HONOR_DESIGN_ PARAGRAPHS_FOR_HTML

To improve spacing between paragraphs in HTML output, use the ALWAYS_HONOR_ DESIGN_PARAGRAPHS_FOR_HTML switch so that the engine produces new HTML paragraph tags for each paragraph in Designer.

Syntax

```
-ALWAYS_HONOR_DESIGN_PARAGRAPHS_FOR_HTML
```

This switch does not have any arguments.

Supported in

- 9.0.108 and later maintenance releases
- 9.5.101 and later versions

A.2.29 ALWAYS_SET_SYS_TABLECOLUMN

Use the ALWAYS_SET_SYS_TABLECOLUMN switch to always set the SYS_TableColumn variable. The default behavior without this switch is to set the SYS_TableColumn variable only if it is used within the table cells. If you use the SYS_TableColumn variable in a rule, for example, you must use this switch to set the variable.

> Using this switch can decrease performance speed.

Syntax

```
-ALWAYS_SET_SYS_TABLECOLUMN
```

This switch does not have any arguments.

Supported in

6.1.013 and later

A.2.30 APP_PACKAGEFILES

Use the APP_PACKAGEFILES switch to specify multiple package files with output sorting.

Syntax

```
-APP_PACKAGEFILES=<package_file_paths>
```

The argument for this switch is the fully-qualified path and file name of a flat text file that lists the paths and file names of the package files. The application file must be listed first in the flat text file. This switch has no default value.

Example

```
-APP_PACKAGEFILES=C:\Packages\2009_02\Packlist.txt
```

The `Packlist.txt` example contains the following package files:

```
C:\Packages\2009_02\Application Packages\Application.pub
```

```
C:\Packages\2009_02\Document Packages\Documents.pub
```

```
C:\Packages\2009_02\Campaign Packages\Campaigns.pub
```

A.2.31 APPLICATION

Use this packaging switch to specify the name and location of the application that you want to package, and whether to target campaigns and documents for packaging.

This switch is required when you package applications from the command prompt.

Syntax

```
-APPLICATION=<applicationName>,[<targetObject>],[<folderPath>]
```

Arguments	Required	Supported values
`applicationName`	Yes	The name of the application that you want to package
`targetObject`	No	The specific package object that you want to package If you do not specify a value for this argument, the switch uses `Application` as the default argument value.
`folderPath`	No	The path the specified Exstream application, relative to the root folder **Tip:** If the application is in the root folder (that is, in the `Exstream` folder in the Library in Design Manager), then you do not need to specify this argument. However, if the application is in a subfolder, you must specify the folder path without the root folder, for example, `Samples/Demonstration.`

Example

```
-APPLICATION=BankStatement,,Savings/Statement
```

A.2.32 APPLICATION_MODE (DBAdmin)

The APPLICATION_MODE switch sets the application mode of Exstream Design and Production 8.0 or later databases.

Syntax

```
-APPLICATION_MODE=<mode>
```

Arguments	Required	Supported values
`<mode>`	Yes	• `SBCS`—SBCS application mode • `DBCS`—DBCS application mode. This is the default if no value is specified. • `ALL`—SBCS/DBCS application mode

Example

```
-APPLICATION_MODE=DBCS
```

A.2.33 APPLICATION_MODE

This packaging switch specifies the application mode for the application being packaged.

Syntax

```
APPLICATION_MODE=<mode>
```

Arguments	Required	Supported values
mode	Yes	• SBCS—Specifies that the application being packaged contains SBCS objects • DBCS—Specifies that the application being packaged contains DBCS objects.

Example

```
APPLICATION_MODE=SBCS
```

A.2.34 ASCII_OUTPUT_MVS

Deprecated: This switch has been deprecated, and is no longer required when producing output on the z/OS platform. All ASCII PDL output is always produced as if this switch were used. If you use this switch, the engine issues an informational message indicating that the switch is no longer necessary.

On the z/OS platform, use the ASCII_OUTPUT_MVS switch to produce ASCII PDL output that is byte-by-byte similar to that available on the Windows platform. You can download the ASCII in binary mode for the output device, which eliminates problems that can arise when converting EBCDIC to ASCII during the download.

Syntax

```
-ASCII_OUTPUT_MVS
```

This switch does not have any arguments.

A.2.35 AUTOCREATE

Use the AUTOCREATE switch to create an Exstream Design and Production database from the command line or from a batch file using the Database Administration (DBAdmin) utility. This

switch automatically creates the structure for a database with no interaction. It uses the default Exstream Design and Production tablespace names, unless you specify alternate tablespace file names, or use the TABLESPACE switch.

Requirements

If you use the AUTOCREATE switch, you must also use the following switches:

- QUERYPATH
- One of the following: ACCESSDB, EMPTYACCESSDB, or DSN
- ACCEPT_PRIVACY_STATEMENT (Required only if you are creating a database for use with Exstream Design and Production version 9.5.301 and later, or Exstream Application Manager (EAM) versions 3.5.301 and later.)

> **Note:** EAM is not supported in Exstream Design and Production version 16.2.0 and later.

Syntax

```
-AUTOCREATE=<databaseType>,<databaseLanguage>,[<primaryTablespace>],
[<indexTablespace>],[<LOBTablespace>],[<tempTablespace>],[<databaseNameDB2>]
```

Argument	Required	Supported values
`databaseType`	Yes	• `DESIGN` • `TRACKING` • `EAM`
`databaseLanguage`	Yes	• `de-de` (German) • `en-us` (English) • `es-mx` (Spanish) • `fr-fr` (French) • `ja-jp` (Japanese) • `pt-br` (Portuguese Brazilian) • `zh-cn` (Simplified Chinese)
`primaryTablespace`	No	The file name of the primary tablespace This argument is used for DB2 and Oracle databases on Linux, UNIX, and Windows platforms.
`indexTablespace`	No	The file name of the index tablespace This argument is used for DB2 and Oracle databases on Linux, UNIX, and Windows platforms.

Argument	Required	Supported values
`lobTablespace`	No	The file name of the LOB tablespace This argument is used for DB2 and Oracle databases on Linux, UNIX, and Windows platforms.
`tempTablespace`	No	The file name of the temporary tablespace This argument is used for DB2 and Oracle databases on Linux, UNIX, and Windows platforms.
`databaseNameDB2`	No	The name of your DB2 database on the Z/OS platform. The database structure is created in the default database unless you use this argument to specify another database name.

Example

```
-AUTOCREATE=DESIGN,en-us,"C:\tablespaces\ExstreamPrimaryTS.dat","C:\tablespac
es\ExstreamIndexTS.dat","C:\tablespaces\ExstreamLOBTS.dat"
-QUERYPATH="C:\Program Files\OpenText\Exstream\Exstream 16.3.0\Query Files"
-DSN=MyDSN,,MyUserName,MyPassword
-ACCEPT_PRIVACY_STATEMENT
```

A.2.36 AUTODROP

The AUTODROP switch automatically drops all tables and associated objects for the specified role (design or tracking) with no interaction, but does not drop the database instance.

Requirements

If you use the AUTODROP switch, you must also use the following switches:

- QUERYPATH
- One of the following: ACCESSDB or DSN

Syntax

```
-AUTODROP=<database_type>,<DB2_z/OS_database_
name>,DROPTABLESPACES,DROPBUFFERPOOL
```

Argument	Required	Supported values
`<database_type>`	Yes	• `DESIGN` • `TRACKING`
`<DB2_z/OS_database_name>`	No	The structure is created in the Exstream Design and Production default database unless you use this argument to specify another database name.

Argument	Required	Supported values
DROPTABLESPACES	No	Use this argument to drop the default Exstream Design and Production tablespaces. For DB2 on z/OS, the structure is dropped in the Exstream default database unless you specify another database name.
DROPBUFFERPOOL	No	For DB2 on Linux, UNIX, and Windows databases, use this argument to drop the default Exstream Design and Production buffer pool.

Example

```
-AUTODROP=TRACKING
-QUERYPATH=C:\Temp
-DSN=MyDSN,,MyUserName,MyPassword
```

A.2.37 AUTOSIZE_RTF_TABLES_FOR_VARIABLES

Use the AUTOSIZE_RTF_TABLES_FOR_VARIABLES switch to allow table rows to autosize after the engine inserts RTF variable data into the table. Using this switch prevents text from being cut off when there are long string variables in a table cell. Using the AUTOSIZE_RTF_TABLES_FOR_VARIABLES switch affects all imported tables in a design, and prevents you from creating statically-sized tables with variable content.

Syntax

```
-AUTOSIZE_RTF_TABLES_FOR_VARIABLES
```

This switch does not have any arguments.

A.2.38 AUTOUPDATE (DBAdmin)

The AUTOUPDATE switch is used to update the structure of a database from the command line or from a batch file using the Database Administrator (DBAdmin) utility. This switch automatically updates the structure of a database with no interaction. If required, the default Exstream tablespaces are used. If the default Exstream Design and Production tablespaces do not exist, they will be created.

Requirements

If you use the AUTOUPDATE switch, you must also use the following switches:

- QUERYPATH
- One of the following: ACCESSDB, EMPTYACCESSDB, or DSN
- ACCEPT_PRIVACY_STATEMENT (Required only if you are creating a database for use

with Exstream Design and Production versions 9.5.301 and later, or Exstream Application Manager (EAM) versions 3.5.301 and later.)

Note: Beginning with Exstream Design and Production version 16.2.0, which provides a 64-bit design environment, EAM is no longer supported.

Syntax

```
-AUTOUPDATE=<database_type>,<database_language>,<DB2_z/OS_database_name>
```

Use the following arguments:

Argument	Required	Supported values
`<database_type>`	Yes	• `DESIGN` • `TRACKING`
`<database_language>`	Yes	• `de-de` (German) • `en-us` (English) • `es-mx` (Spanish) • `fr-fr` (French) • `ja-jp` (Japanese) • `pt-br` (Portuguese Brazilian) • `zh-cn` (Simplified Chinese)
`<DB2_z/OS_database_name>`	No	The structure is updated in the Exstream Design and Production default database unless you use this argument to specify another database name.

Example

```
-AUTOUPDATE=DESIGN,en-us
-QUERYPATH=C:\Temp
-DSN=MyDSN,,MyUserName,MyPassword
-ACCEPT_PRIVACY_STATEMENT
```

A.2.39 BAR_CHART_WHITESPACE

Use the BAR_CHART_WHITESPACE engine switch to reduce the amount of white space that appears between the first and last bars and the edges of a traditional bar chart. The BAR_CHART_WHITESPACE engine switch lets you set an integer value to represent a percentage is used to reduce the amount of white space.

Syntax

```
-BAR_CHART_WHITESPACE=integer
```

The argument for this switch is an integer value between 0 and 100. The default value for this switch is 100.

Example

```
-BAR_CHART_WHITESPACE=100
```

Supported in

7.0.635 and later

A.2.40 BAR_DRAW_XAXIS_AFTER_UPGRADE

For backward compatibility after you upgrade from Exstream Design and Production version 5.0, use the BAR_DRAW_XAXIS_AFTER_UPGRADE engine switch when you repackage an application for output that contains a traditional bar chart, so that the x-axis line is drawn on the chart in the output.

Syntax

```
-BAR_DRAW_XAXIS_AFTER_UPGRADE
```

This switch does not have any arguments.

Supported in

6.0.016 and later

A.2.41 BAR_USE_FONT_AXES_FOR_LABEL

For backward compatibility after you upgrade to Exstream Design and Production 7.0 or later, use the BAR_USE_FONT_AXES_FOR_LABEL switch if traditional bar chart label fonts appear differently in the output that you produce following the upgrade. This switch sets the label fonts to the same font as the x-axis font.

As an alternative to using this switch, you can open the page in Designer, manually change the font settings for both the x-axis and the y-axis, save the page, and repackage the application.

Syntax

```
-BAR_USE_FONT_AXES_FOR_LABEL
```

This switch does not have any arguments.

Supported in

6.1.025 and later

A.2.42 BAR_ZERO_AXIS_ON_TOP

For backward compatibility after you upgrade to Exstream Design and Production version 7.0, use the BAR_ZERO_AXIS_ON_TOP engine switch when you produce output from an application that contains a traditional bar chart in which the zero axis line is incorrectly being drawn over the bars in the chart.

Syntax

```
-BAR_ZERO_AXIS_ON_TOP
```

This switch does not have any arguments.

Supported in

5.0 and later

A.2.43 BARCODE_PLACEMENT_HONOR_PAGE_ORIENT

Use the BARCODE_PLACEMENT_HONOR_PAGE_ORIENT engine switch when you produce output from an application that contains inserter barcodes, so that the engine honors page orientation settings when determining the barcode placement. The BARCODE_PLACEMENT_HONOR_PAGE_ORIENT engine switch prevents the engine from using a page's dimensions as the determining factors for inserter barcode placement.

Syntax

```
-BARCODE_PLACEMENT_HONOR_PAGE_ORIENT
```

This switch does not have any arguments.

Supported in

6.1.026 and later

A.2.44 BLOCK_OUTPUT_MVS

Use the BLOCK_OUTPUT_MVS switch to enable byte-by-byte output, similar to that of a PC. This switch honors the blocking options on output object settings for the mainframe platform.

This switch is used when output is made on a mainframe and then sent to a third party.

Syntax

```
-BLOCK_OUTPUT_MVS
```

This switch does not have any arguments.

A.2.45 BREAK_PAGE_OTHER_ROWS_PLACED

Use the BREAK_PAGE_OTHER_ROWS_PLACED switch to force a table page break when you place rows from another section or level.

Syntax

```
-BREAK_PAGE_OTHER_ROWS_PLACED
```

This switch does not have any arguments.

A.2.46 BUILDFONTS

Use this packaging switch to specify which fonts to package.

Note: When packaging resources, the PACKAGETYPE switch must use `RESOURCESONLY` as the argument.

Syntax

```
-BUILDFONTS=<fonts_to_package>
```

Arguments	Required	Supported values
`<fonts_to_package>`	Yes	• `ALL`—Include all fonts. • `WITHNAMES`—Include only fonts with names. • `WITHOUTNAMES`—Include only fonts without names.

Example

```
-BUILDFONTS=ALL
```

A.2.47 BUILDIMAGES

Use this packaging switch to specify whether to package images.

Note: When packaging resources, the PACKAGETYPE switch must use `RESOURCESONLY` as the argument.

Syntax

```
-BUILDIMAGES
```

None

A.2.48 BYPASS_BUNDLE_COVER_RULE_ON_INITIALIZE

Use the BYPASS_BUNDLE_COVER_RULE_ON_INITIALIZE switch to prevent the execution of the qualifying rule on a queue bundle cover/trailer during initialization.

Syntax

```
-BYPASS_BUNDLE_COVER_RULE_ON_INITIALIZE
```

This switch does not have any arguments.

A.2.49 BYPASS_QUEUE_RULE_ON_INITIALIZE

Use the BYPASS_QUEUE_RULE_ON_INITIALIZE switch to prevent the execution of the qualifying rule on a queue.

Syntax

```
-BYPASS_QUEUE_RULE_ON_INITIALIZE
```

This switch does not have any arguments.

A.2.50 CACHETABLE

Use the CACHETABLE switch to change the default threshold (250,000 rows) for theMEMORYCACHE switch. After all rows for a customer are processed, the rows that exceed the threshold are written to the file specified by the MEMORYCACHE engine switch, freeing the memory used. Use the CACHETABLE switch when producing applications on the 32-bit version of the production engine.

Important: You must specify MEMORYCACHE before CACHETABLE.

Syntax

```
-CACHETABLE=<numberOfRows>
```

The required argument for this switch is the number of table rows to cache. This switch has no default value.

Example

```
-MEMORYCACHE

-CACHETABLE=100000
```

A.2.51 CAPTURETESTDATA

If you have licensed the Test Data Capture module, you can use the CAPTURETESTDATA switch to collect a minimal set of test data that fully exercises all lines of the named and unnamed rules that are included in an application.

This switch generates a new customer driver file and an optional test data capture report. Test Data Capture supports columnar, delimited, or XML-formatted customer driver files. On two-pass engine runs, Test Data Capture can collect data only for rules that are executed on the first pass.

The CAPTURETESTDATA switch supports multiple-byte encoding.

Important: The CAPTURETESTDATA switch is not compatible with sorting and bundling. If your application uses sorting and bundling, either remove the CAPTURETESTDATA switch from your control file, or disable sorting and bundling before using this switch.

Syntax

```
CAPTURETESTDATA=<captureFileName>,[<statementCustomers>],[<reporting>],
[<reportFileName>],[<optimization>]
```

Argument	Required	Supported values
`captureFileName`	Yes	The name of the file to which the production environment writes the captured data The production environment generates the same type of data file as the customer driver file included in the application.
`statementCustomers`	No	The number of customers to capture per statement This value also specifies the number of customers you want the production environment to capture based on their inclusion or exclusion from the outcome after executing the rule statement. The default value of this argument is 1.

Argument	Required	Supported values
`reporting`	No	• `NONE`—No report • `SUMMARY`—Includes the rule names and identifies whether each was fully tested • `DETAILED`—Includes each statement of each rule and identifies how many customers were captured for each statement • `AUDIT`—Includes each statement of each rule and identifies which customers were captured for each statement The `DETAILED` and `AUDIT` levels of reporting are available only if you have licensed both the Test Data Capture and the Rule Analyzer modules.
`reportFileName`	No	The name of the file to which the production environment writes the Test data capture report The default argument value is `TestDataCaptureReport.txt` or `DD:TDCRPT` for z/OS.
`optimization`	No	• `LOW`—Tracks the minimum number of customers possible, uses the least memory, and runs the fastest of all three settings • `NORMAL`—Tracks customers, memory usage, and speed run at a moderate level. This is the default option. • `HIGH`—Tracks a large number of customers, uses the most memory, and runs the slowest of all three settings Higher optimization levels increase memory usage and time to run, but can produce smaller customer driver files with test data.

Example

```
-CAPTURETESTDATA=CaptureFile.dat,1,DETAILED,CaptureReport.txt,HIGH
```

A.2.52 CASCREATENEWPACKAGE

Use the CASCREATENEWPACKAGE switch to specify that a new package file object should be created in the CAS for the package file that you are uploading, even if a package file object from the same application already exists in the CAS.

If you do not use this switch, a new package file object is created only if a file from the same application does not already exist. If a file from the same application exists, that file is updated and the version number for the file is incremented.

This switch is used with the UPLOAD_PACKAGE_TO_CAS switch to upload a package file to the CAS.

Tip: When you use the CASCREATENEWPACKAGE switch, you can upload multiple package files with the same name to the CAS, even if they are created from the same application. To avoid confusion, OpenText recommends that you use the CASPACKAGENAME switch to specify a unique name when you upload a new package file to the CAS.

Syntax

```
-CASCREATENEWPACKAGE
```

This switch does not have any arguments.

Supported in

- 16.3.3 and later maintenance releases
- 16.4.0 Update 1 and later versions

A.2.53 CASPACKAGEDESCRIPTION

Use the CASPACKAGEDESCRIPTION packaging switch to specify the description of the package file that you want to upload to the common asset service (CAS).

This switch is used with the UPLOAD_PACKAGE_TO_CAS switch to upload a package file to the CAS.

Syntax

```
-CASPACKAGEDESCRIPTION=<description>
```

Arguments

The argument is the description of the package file to upload to the CAS. You must specify a description with 255 or fewer characters. If the description has spaces in it, and if you are using the CASPACKAGEDESCRIPTION switch at the command prompt, you must enclose the argument value in double quotation marks (" "). This switch has no default value.

Example

```
-CASPACKAGEDESCRIPTION="Package file to produce customer output"
```

Supported in

- 16.3.3 and later maintenance releases
- 16.4.0 Update 1 and later versions

A.2.54 CASPACKAGENAME

Use the CASPACKAGENAME packaging switch to specify the name of the package file that you want to create in the common asset service (CAS).

This switch is used with the UPLOAD_PACKAGE_TO_CAS switch to upload a package file to the CAS.

Tip: You can upload multiple package files with the same name to the CAS. To avoid confusion, OpenText recommends that you use the CASPACKAGENAME switch to specify a unique name when you upload a new package file to the CAS.

Syntax

```
-CASPACKAGENAME=<packageName>
```

Arguments

The argument is the name of the package file that you want to create in the CAS. You must specify a name with 255 or fewer characters. If the name has spaces in it, and if you are using the CASPACKAGENAME switch at the command prompt, you must enclose the argument value in double quotation marks (" "). This switch has no default value. If you do not use this switch, the name of the Exstream application is used.

Example

```
-CASPACKAGENAME=MyPackageName
```

Supported in

- 16.3.3 and later maintenance releases
- 16.4.0 Update 1 and later versions

A.2.55 CASPACKAGEUPLOADTIMEOUT

Use the CASPACKAGEUPLOADTIMEOUT packaging switch to specify the number of seconds to wait for a package file to finish uploading to the common asset service (CAS) before

the upload times out.

This switch is used with the UPLOAD_PACKAGE_TO_CAS switch to upload a package file to the CAS.

Note: The use of the CASPACKAGEUPLOADTIMEOUT switch overrides any value provided using the **CAS upload timeout (seconds)** option in Design Manager.

Syntax

```
-CASPACKAGEUPLOADTIMEOUT=<seconds>
```

The argument for this switch is the number of seconds to wait before the upload times out. The default value is 600 (10 minutes).

Example

```
-CASPACKAGEUPLOADTIMEOUT=300
```

Supported in

- 16.3.3 and later maintenance releases
- 16.4 Update 1 and later versions

A.2.56 CASSTOPLEVEL

Use the CASSTOPLEVEL packaging switch to specify the level of error that stops the package file from being uploaded to the common asset service (CAS).

This switch is used with the UPLOAD_PACKAGE_TO_CAS switch to upload a package file to the CAS.

Syntax

```
-CASSTOPLEVEL=<errorLevel>
```

Arguments	Required	Supported values
`<errorLevel>`	No	• `SEVERE`—The package will not be uploaded in the case of a severe error. This is the default. • `ERROR`—The package will not be uploaded in the case of an error. • `WARNING`—The package will not be uploaded in the case of a warning.

Example

```
-CASSTOPLEVEL=ERROR
```

Supported versions

- 16.3.3 and later maintenance releases
- 16.4.0 Update 1 and later versions

A.2.57 CENTER_BOTTOM_CHART_LABEL

Use the CENTER_BOTTOM_CHART_LABEL switch to center the bottom tick label in a traditional chart, instead of using the arbitrary offset to keep the bottom tick label above the chart bottom.

Syntax

```
-CENTER_BOTTOM_CHART_LABEL
```

This switch does not have any arguments.

Supported in

5.0 and later

A.2.58 CENTER_TOP_CHART_LABEL

Use the CENTER_TOP_CHART_LABEL switch to center the top tick label instead of using an arbitrary offset to keep the top tick label below the chart top.

Syntax

```
-CENTER_TOP_CHART_LABEL
```

This switch does not have any arguments.

Supported in

5.0 and later

A.2.59 CHART_DRAW_FRAME_AFTER_UPGRADE

For backward compatibility after you upgrade to versions 7.0 and later, use the CHART_DRAW_FRAME_AFTER_UPGRADE engine switch when you produce output from an existing application that contains a traditional chart, and you want to draw a frame around the chart using the chart composition behavior from Exstream Design and Production version 6.1.

Syntax

```
-CHART_DRAW_FRAME_AFTER_UPGRADE
```

This switch does not have any arguments.

Supported in

7.0.410 and later

A.2.60 CHECK_CAS_IMG_UPDATES

Note: This switch is used when running `packager.exe` from the command prompt, a package control file, or a batch file.

If you use common asset service (CAS) image resources in your application, you can use the CHECK_CAS_IMG_UPDATES switch during packaging to return a list of CAS images with updates available in the packaging message file.

Syntax

```
-CHECK_CAS_IMG_UPDATES
```

This switch does not have any arguments.

Supported in

16.3.0 and later

A.2.61 CHECK_NEW_LINE_FONT_SPACING

Use the CHECK_NEW_LINE_FONT_SPACING engine switch to prevent extra line spacing when you produce output from an application in which an empty variable causes a blank line in a text paragraph that is ignored in the output. When you do not use the CHECK_NEW_LINE_ FONT_SPACING switch, the line after the ignored line still includes the line spacing from the ignored line, which results in the extra line spacing.

Syntax

```
-CHECK_NEW_LINE_FONT_SPACING
```

This switch does not have any arguments.

Supported in

8.0.314 and later

A.2.62 CHECK_POSTAGE_FOR_POOL_PAGES

Use the CHECK_POSTAGE_FOR_POOL_PAGES switch when your application is set to exclude campaigns or teaser messages. The CHECK_POSTAGE_FOR_POOL_PAGES switch implements postage breaks for pool pages.

Syntax

```
-CHECK_POSTAGE_FOR_POOL_PAGES
```

This switch does not have any arguments.

Supported in

6.1.024 and later

A.2.63 CLEAR_FLOATS_FROM_HTML_IMPORT

In versions earlier than 16.2.0, text paragraphs that followed HTML placeholder variables in Designer did not wrap around floated objects in the imported HTML content.

This default behavior was changed in 16.2.0 so that text wraps around non-text objects when `float` settings are included in imported HTML content and there are no `clear` settings to create a break between the floated object and the following text paragraph. This can affect the appearance of the design in the output.

For backward compatibility after you upgrade to 16.2.0, use the CLEAR_FLOATS_FROM_HTML_IMPORT switch. This switch will be automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 9.5.303 or earlier.

Syntax

```
-CLEAR_FLOATS_FROM_HTML_IMPORT
```

This switch does not have any arguments.

Supported in

16.2.0 and later

A.2.64 CLOSE_OUTPUT_ON_BREAK

When using WSMQ Connector, use the CLOSE_OUTPUT_ON_BREAK switch to require the engine to write a new output file for each EOF call WSMQ Connector returns.

Syntax

```
-CLOSE_OUTPUT_ON_BREAK
```

This switch does not have any arguments.

A.2.65 CODE128_MAP_SPECIAL_CHAR

If you want to use special characters in your Code 128 barcode, you can use the CODE128_MAP_SPECIAL_CHAR switch to override the default character mappings in the barcode character set.

Syntax

```
-CODE128_MAP_SPECIAL_CHAR=<characterSet>,<characterValue>,<codeValue>
```

Arguments

Argument	Required	Supported values
`characterSet`	Yes	A, B, or C, which specifies the barcode character set
`characterValue`	Yes	The ASCII character value that you want to use for a custom mapping (0 to 255)
`codeValue`	Yes	The barcode set character value that you want to override (0 to 125) To override multiple characters, you can specify multiple values in succession.

Example

```
-CODE128_MAP_SPECIAL_CHAR=C,52,20
```

A.2.66 COMBINE_ADJACENT_POLY_FILLS

Use the COMBINE_ADJACENT_POLY_FILLS switch when delivering output that contains table cells with the same fill to reduce output file size. The COMBINE_ADJACENT_POLY_FILLS switch identifies matching properties for poly fills in table cells and optimizes the print file by drawing a single raster for a table row.

When using AFP output, use this switch in conjunction with the AFP_TILE_RASTER_FILLS switch to optimize your output file size.

Syntax

```
-COMBINE_ADJACENT_POLY_FILLS
```

```
-AFP_TILE_RASTER_FILLS
```

This switch does not have any arguments.

A.2.67 COMPOSE_FOOTNOTES_ON_ALL_QUEUES

Use the COMPOSE_FOOTNOTES_ON_ALL_QUEUES switch to compose footnotes per queue instead of only on the primary queue. This switch allows documents that have been excluded from the primary queue by a rule to have their footnotes composed if they are included on a later queue.

Syntax

```
-COMPOSE_FOOTNOTES_ON_ALL_QUEUES
```

This switch does not have any arguments.

A.2.68 COMPRESSPACKAGEFILE

Use the COMPRESSPACKAGEFILE packaging switch to specify that the package file should be compressed. Compression takes place at the end of the packaging process, and compresses the contents that are located between the package header and the AQS manifest. This switch is recommended for package files that will be uploaded to OpenText cloud applications.

Keep in mind the following considerations when using package file compression:

- The APPLICATION packaging switch must target an application for packaging.
- If you use the PACKAGETYPE packaging switch, compression is supported only if all print resources are included (`PACKAGETYPE=ALL`).
- Compressed package files can be decompressed only by an Exstream production engine that installed and running in a Windows, Linux, or AIX environment.
- Although you should always compress Design and Production package files that are used with OpenText cloud applications, package file compression can also be used to reduce the amount of resources required to store package files locally or transmit them to other platform resources such as Common Asset Service (CAS).
- Package files that contain mostly images might not achieve a high rate of compression, since images are already in a compressed format.

Tip: The Exstream engine decompresses compressed package files at run time. The default maximum size for a decompressed package file is 104857600 bytes (approximately 100 MB). You can change the default maximum size using theMAX_DECOMPRESSED_PACKAGE_BYTES engine switch. Because the engine run will halt if there is not a temporary directory available to store a decompressed package that exceeds the maximum size, it is a good idea to also use the TEMPORARY_DIRECTORY engine switch any time you use package compression.

Syntax

```
-COMPRESSPACKAGEFILE
```

This switch does not have any arguments.

Related switches

MAX_DECOMPRESSED_PACKAGE_BYTES

TEMPORARY_DIRECTORY

Supported in

- 16.4.6 and later versions

A.2.69 CONNECTORMAP

Use the CONNECTORMAP engine switch to dynamically change connector object properties at run time without repackaging the application.

Syntax

```
-CONNECTORMAP=<ConnectorName>,<ProgramType>,<Module>,<Function>,<BufferSize>,
<OpenParams>
```

Argument Name	Required	Supported values
`<ConnectorName>`	Yes	Name of the connector object that you want to modify at run time
`<ProgramType>`	Yes	The language used to create the module: • `ASM` • `COBOL` • `DLL` (for C and C++) • `PL1`

Argument Name	Required	Supported values
`<Module>`	Yes	The location of the DLL file
`<Function>`	Yes	The function name in the DLL/routine
`<BufferSize>`	No	The size of the buffer (in bytes), which must accommodate the size of the output
`<OpenParams>`	No	Parameters specific to the module

Example

```
-CONNECTORMAP=ConnectorName,DLL,C:\CustDLL\myDLL.dll\,myRoutine,5000,myParams
```

A.2.70 CONSTANT_HTML_BASESHIFTS

Prior to Exstream Design and Production version 9.5.201, the default baseline shift for superscript or subscript in HTML output was 0.25 em. In 9.5.201 and later, the default baseline shift is the appropriate number of pixels.

For backward compatibility after you upgrade to 9.5.201, use the CONSTANT_HTML_BASESHIFTS switch. This switch will be automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 9.5.199 or earlier.

Syntax

```
-CONSTANT_HTML_BASESHIFTS
```

This switch does not have any arguments.

Supported in

9.5.201 and later

A.2.71 CONTROLFILE

Used with: Engine, Packager

Use the CONTROLFILE switch to specify the name of the control file that you want to use for packaging or engine processing.

Syntax

```
-CONTROLFILE=<fileName>
```

The argument is the fully qualified file name of the control file that you want to include. This switch has no default value.

Example

```
-CONTROLFILE=C:\path\to\ControlFile.txt
```

A.2.72 CONTROLFILEENCODING

Use the CONTROLFILEENCODING switch to specify which encoding to use in your control file.

Syntax

```
-CONTROLFILEENCODING=<encoding>
```

Arguments	Required	Supported values
`<encoding>`	Yes	• `ASCII` • `BIG5` • `EBCDIC` • `ISO88592` (default value) • `JIPS` • `LATIN1` • `LATIN2` • `SJIS` • `UTF16UTF16BEUTF16LEUTF8`

Example

```
-CONTROLFILEENCODING=UTF8
```

A.2.73 CONVERT_AS_IS_TO_PPD

Use the CONVERT_AS_IS_TO_PPD switch to take existing search keys of the **As is** type (that is, search keys where no COMMENT or no operation (NOP) characters are added), and move them inside a page-piece dictionary entry that can be read by IBM Content Manager On Demand.

This switch can convert only **As is** search keys where the **Placement** list is set to **Before each page.** Optionally, you can enter the names of specific search keys that you want to convert to page-piece dictionary entries. However, if you do not specify any search key names, all **As is** search keys with a **Placement** option of **Before each page** will be moved.

Note: If you do enter a search key name as an argument, you must enter the search key name as it appears in the Library. You cannot use unnamed search keys with this switch.

The text of your search key or keys must be formatted as shown in the following example, which uses three search keys:

```
/DocumentNbr(1)

/CustomerName(Andrew Nixon)

/CustomerZip(46234)
```

The CONVERT_AS_IS_TO_PPD switch will put each of the specified search keys inside the following structure:

```
/PieceInfo <</IBM-ODIndexes <</Private

<<

[Your search keys will be placed here]

>>

/LastModified(D:201604150000000Z)

>>

>>/LastModified(D:201604150000000Z)

>>
```

The “LastModified” dates will be set to the time the engine was started. These dates cannot be changed.

Syntax

```
-CONVERT_AS_IS_TO_PPD
```

or

```
-CONVERT_AS_IS_TO_PPD=<search_key_name, search_key_name2,…search_key_nameN>
```

The argument for this switch is an optional list of search key names.

Example

```
-CONVERT_AS_IS_TO_PPD
```

In this example, all **As is** search keys where the **Placement** list is set to **Before each page** will be moved inside the page structure.

```
-CONVERT_AS_IS_TO_PPD=searchKey_1,searchKey2
```

In this example, only the identified search keys are converted, leaving all other application search keys unaffected.

Supported in

- 9.0.113 and later maintenance releases
- 9.5.301 and later versions

A.2.74 CONVERTCCSID

If you are using the WSMQ Connector and you need to convert messages that contain accented characters into the format required by the receiving system before transmission, use the CONVERTCCSID switch to specify the coded character set identification number that the messaging queue manager requires to convert the messages and read the accented characters correctly.

> **Note:** If you are using the CONVERTCCSID switch, you must also specify `Y` for the CONVERT parameter in your WSMQ Connector initialization file. If you set the CONVERT parameter to `N`, you receive a warning message and the CONVERT parameter is ignored. For more information about the CONVERT parameter for the WSMQ Connector, see *Configuring Connectors* in the Exstream Design and Production documentation.

Syntax

```
-CONVERTCCSID=<codedCharacterSetIDnumber>
```

The argument for this switch is the number that represents the coded character set identification number. For example, if you want to convert UTF-8 characters, you must specify `1208` as the argument. The default setting for this switch is `MQCCSI_DEFAULT`. For a complete listing of argument values, see the IBM MQ documentation on the IBM website.

Example

```
-CONVERTCCSID=1208
```

A.2.75 COPYDB

Use the COPYDB switch to copy database content from one database to another.

Requirements

If you use the COPYDB switch, you must also use the following switches:

- To specify the source database: DSN or ACCESSDB
- To specify the destination database: COPYTODSN or COPYTOACCESSDB

Syntax

```
-COPYDB=[<designQueryFilePath>],[<trackingQueryFilePath>],
[<EAMQueryFilePath>],[<copyOption>]
```

Argument	Required	Supported values
designQueryFilePath	No	The file path to the design query file Use this argument if the database has the design role. Otherwise, leave the argument blank. If omitted, DBAdmin will use its standard algorithm for locating the query files.
trackingQueryFilePath	No	The file path to the tracking query file Use this argument if the database has the tracking role. Otherwise, leave the argument blank. If omitted, DBAdmin will use its standard algorithm for locating the query files.
EAMQueryFilePath	No	The file path to the EAM query file Use this argument if the database has the EAM role.Otherwise, leave the argument blank. If omitted, DBAdmin will use its standard algorithm for locating the query files.
copyOption	No	COPYONLYSYSTEMTBL Use this argument to copy only the system settings that are contained in SYSTEMTBL table.

Example

```
-COPYDB="C:\Program Files\OpenText\Exstream\Exstream 16.3.0\Query Files","C:\
Program Files\OpenText\Exstream\Exstream 16.3.0\Query Files"
-DSN=DevDB,development02,jsmith,mypassword
-COPYTODSN=TestDB,test01,jsmith,mypassword
```

A.2.76 COPYGROUPEACHPAGE

Use the COPYGROUPEACHPAGE switch to force pages to be placed on the backs of other pages in AFP output, even when different paper types are used in the output. This switch suppresses Invoke Media Map (IMM) records before each new paper type if the IMM appears on the back of a page. This is useful for printers that use advanced multiple-up form definition (formdef) files.

To use this switch, you must have set up AFP copygroups in Design Manager using the **AFP copygroup name** option on the paper type properties. The COPYGROUPEACHPAGE switch forces a copygroup to be called out in the AFP for each page.

Syntax

```
-COPYGROUPEACHPAGE
```

This switch does not have any arguments.

Supported in

3.0 and later

A.2.77 COPYINPUT

Use the COPYINPUT switch to generate a file that contains a copy of each record from a driver file, initialization file, or reference that the engine reads at run time. You can use this switch with the SECUREOUTPUT switch to produce a file that contains all records, but has confidential data removed.

The COPYINPUT switch supports multiple-byte encoding.

> **Note:** You cannot use the COPYINPUT switch with XML or Print Miner data files.

Syntax

```
-COPYINPUT=<inputDataFile>,<outputFile>
```

Argument	Required	Supported Values
`inputDataFile`	Yes	The file name of the input data file, as specified in the package file
`outputFile`	Yes	The file name of the output file that you want to generate

Example

```
-COPYINPUT=sampleDriverFile,sampleOutputFile
```

A.2.78 COPYTOACCESSDB

When you use the COPYDB switch to copy content from one database to another, use the COPYTOACCESSDB switch to specify the name of the destination Microsoft Access database. The file name can use either the .mdb or the .accdb extension, but keep in mind that Microsoft Access versions earlier than 2007 support only the .mdb extension.

Syntax

```
-COPYTOACCESSDB=<destinationDatabaseName>
```

Argument	Required	Supported values
`destinationDatabaseName`	Yes	The fully qualified file name of the destination Access database

Example

```
-COPYDB="C:\Program Files\OpenText\Exstream\Exstream 16.3.0\Query Files","C:\
Program Files\OpenText\Exstream\Exstream 16.3.0\Query Files
-ACCESSDB=C:\Temp\Sample.mdb
-COPYTOACCESSDB=C:\Temp\Example.accdb
```

A.2.79 COPYTODSN

When you use the COPYDB switch to copy content from one database to another, use the COPYTODSN switch to specify the data source name (DSN) of the destination database file.

Syntax

```
-COPYTODSN=<destinationDSN>,[<schema>],[<databaseUsername>],
[<databasePassword>]
```

Argument	Required	Supported values
`destinationDSN`	Yes	The DSN name set up in the Windows ODBC Administrator
`schema`	No	The schema name used
`databaseUsername`	No	The database server user name
`databasePassword`	No	The database server user password

Example

```
-COPYDB="C:\Program Files\OpenText\Exstream\Exstream 16.3.0\Query Files",
"C:\Program Files\OpenText\Exstream\Exstream 16.3.0\Query Files"
-DSN=DevDB,development02,jsmith,mypassword
-COPYTODSN=TestDB,test01,jsmith,mypassword
```

A.2.80 CORRECT_PS_TOTALPAGES_WITH_BANNER

Use the CORRECT_PS_TOTALPAGES_WITH_BANNER engine switch to ensure accurate page numbering on banner pages when you package and run an XOB file with a control file.

Syntax

```
-CORRECT_PS_TOTALPAGES_WITH_BANNER
```

This switch does not have any arguments.

Supported in

7.0 and later

A.2.81 CREATE_RESMANAGERFILE

If you have licensed the Output Sorting and Bundling module, you can use the CREATE_RESMANAGERFILE switch to automatically create a Resource Manager (ResManager) file that lets you control which resources the engine places at the top of the print stream when you use external content in an application. This file lets you fine-tune an application for better performance and file size. It is especially useful if you have dynamically imported image files and page, message, or template overlays.

For more information about using a resource manager file, see *Creating Output* in the Exstream Design and Production documentation.

Syntax

```
-CREATE_RESMANAGERFILE=<fileName>
```

Argument	Required	Supported values
`fileName`	Yes	The fully qualified name of the ResManager file. This switch has no default value. This argument is used to provide a name for the new file being created.

Example

```
-CREATE_RESMANAGERFILE=newResManagerFile
```

A.2.82 CREATE_REUSABLE_RESMANAGERFILE

Use the CREATE_REUSABLE_RESMANAGERFILE switch to create a reusable ResManager file based on an existing ResManager file (specified for use in later engine runs). You must use the RESMANAGERFILE switch to specify the existing ResManager file.

For supported output devices, external resources that are specified in the existing ResManager file are written along with the reusable ResManager file, which contains information about those resources. As a result, subsequent engine runs do not require access to the image source files on the disk.

Syntax

```
-CREATE_REUSABLE_RESMANAGERFILE=<fileName>
```

The only argument is the name of the ResManager file name that you are creating. This switch has no default value.

Example

```
-CREATE_REUSABLE_RESMANAGERFILE=NewResManagerFile
```

A.2.83 CREATE_RTF_TABLE_OBJECTS

In RTF output, the RTF format does not allow tables to be ordered or layered with other objects in the design. This behavior can cause unexpected results in some designs.

For example, suppose that your original design contains a background image and a table. Due to the limitations of the RTF format, the background image appears in front of the table in the RTF output.To make sure that table objects are arranged as expected in your RTF output, you can use the CREATE_RTF_TABLE_OBJECTS engine switch to have the engine draw the table inside a text shape object in the RTF output. This placement allows the table to be ordered or layered with other objects in the design and prevents objects (such as background images) from incorrectly overlapping the table object in the final output.

Syntax

```
-CREATE_RTF_TABLE_OBJECTS
```

This switch does not have any arguments.

Supported in

- 8.0.336 and later maintenance releases
- 8.6.109 and later maintenance releases
- 9.0.104 and later maintenance releases
- 9.5.101 and later versions

A.2.84 CREATE_SECTION_DATA_FOR_REF_VARS_IN_LIVE

Use the CREATE_SECTION_DATA_FOR_REF_VARS_IN_LIVE engine switch when you generate a DLF file that contains section-driven data to ensure that the application processes section data as expected. When fulfilling a DLF file, if the engine does not process the section-driven data as expected, you will need to generate a new DLF file with this engine switch to resolve the issue. This engine switch is only needed during the initial engine run when the DLF file is generated and is not needed for fulfillment.

Syntax

```
-CREATE_SECTION_DATA_FOR_REF_VARS_IN_LIVE
```

This switch does not have any arguments.

Supported in

- 8.0.337 and later maintenance releases
- 8.6.110 and later maintenance releases
- 9.0.104 and later maintenance releases
- 9.5.101 and later versions

A.2.85 CUSTOMER_RESET_IMPORTS

Use the CUSTOMER_RESET_IMPORTS switch to reset memory for images imported at run time between customers and to prevent memory issues when you import a large number of unique files.

You can use the NUM_CUSTOMERS_HOLD_MUP_IMPORTS switch together with the CUSTOMER_RESET_IMPORTS switch to specify a greater number of customers to hold in memory at one time.

> This switch does not support using a ResManager file (accessed using the RESMANAGERFILE engine switch) or the **Used resources only** option on the **Resource Inclusion** list (located on the **Resource Management** tab of the of the output object).

Syntax

```
-CUSTOMER_RESET_IMPORTS=<resetInterval>
```

Argument	Required	Supported values
`resetInterval`	No	• ALL—Resets memory between every customer or file import. This is the default value. • UNIQUE—Resets the memory only when a new image is encountered. • NONE—Does not reset the memory between imports.

Example

```
-CUSTOMER_RESET_IMPORTS=UNIQUE
```

Supported in

7.0 and later

A.2.86 CUSTOMERLIST

Use the CUSTOMERLIST switch to specify the individual customers and/or range or ranges of customers that you want to process in the current engine run. This switch allows you to specify a non-contiguous set of customer numbers for processing, unlike theSTART and END switches, which allow you to specify only a contiguous range of customers.

Alternatively, you can use the CUSTOMERLISTFILE switch to point to an external text file that specifies the customers to be processed. Using an external text file to designate customers for processing can be more efficient than using the CUSTOMERLIST switch if the list of customers that you want to process is long.

> **Note:** You do not have to enter customer numbers in numerical order when using this switch. Customers will be processed in numerical order and the output will be automatically sorted by customer number, regardless of the order in which you enter the customer numbers. Also, if you enter duplicate customer numbers, the engine ignores the duplicates and produces only one instance of the output for each customer. If you use this switch in conjunction with the CUSTOMERLISTFILE switch, the specified numbers will be merged.

Keep in mind that if you use the CUSTOMERLIST or CUSTOMERLISTFILE switch with the START or END switches, the CUSTOMERLIST or CUSTOMERLISTFILE switch overrides the START or END switches.

Syntax

```
-CUSTOMERLIST=<list of customer numbers or ranges of numbers>
```

The argument for the CUSTOMERLIST switch is a comma-separated list of individual customer numbers or hyphenated range or ranges of customer numbers.

Example

Each customer number or range of customer numbers must be separated from the next number or range with a comma. For example, to process customer numbers 2, 7, and 14 through 50, the argument should be written in the following way:

```
-CUSTOMERLIST=2,7,14-50
```

In cases where the range covers the same record twice, that record will be run only once. In the following example, record 17 will be run only once though it is included twice:

```
-CUSTOMERLIST=1,2,10,17,15-20
```

Supported in

- 8.0.349 and later maintenance releases
- 8.6.118 and later maintenance releases
- 9.0.112 and later maintenance releases
- 9.5.201 and later versions

A.2.87 CUSTOMERLISTFILE

Use the CUSTOMERLISTFILE switch to specify the location of a text file that designates the individual customers or range or ranges of customers that you want to process in the current engine run. This switch allows you to specify a non-contiguous set of customer numbers for processing, unlike the START and END switches, which allow you to specify only a contiguous range of customers.

Alternatively, you can use the CUSTOMERLIST switch to specify the customers for processing without using an external text file. Using the CUSTOMERLIST switch can be more efficient if the list of customers that you want to process is short.

The specified text file can contain individual customer numbers as well as hyphenated ranges of customer numbers. Each customer number or range of customer numbers must be separated from the next number or range with a comma. For example, to process customer numbers 2, 7, and 14 through 50, the text file should be written in the following way: 2, 7, 14-50.

> You are not required to enter customer numbers in numerical order when using these switches. Customers will be processed in numerical order and the output will be automatically sorted by customer number, regardless of the order in which you enter the customer numbers. Also, if you enter duplicate customer numbers, the engine ignores the duplicates and produces only one instance of the output for each customer. If you use this switch in conjunction with the CUSTOMERLIST switch, the specified numbers will be merged.

Keep in mind that if you use the CUSTOMERLIST or CUSTOMERLISTFILE switch with the START or END switches, the CUSTOMERLIST or CUSTOMERLISTFILE switch overrides the START or END switches.

Syntax

```
-CUSTOMERLISTFILE=<file_path>
```

The argument for this switch is the absolute or relative path to the text file that contains the comma-separated list of individual customer numbers or hyphenated range or ranges of customer numbers.

Example

```
-CUSTOMERLISTFILE=C:\Program Files\Exstream\CustomerList.txt
```

Supported in

- 8.0.349 and later maintenance releases
- 8.6.118 and later maintenance releases
- 9.0.112 and later maintenance releases
- 9.5.201 and later versions

A.2.88 DBPASSWORD

Summary

This packaging switch specifies the password to use if it is not specified in the `DSN`. The `DBPASSWORD` packaging switch is required only if the database used in the application requires a password.

Arguments

As an argument to the switch, you must specify the password.

Syntax

```
-DBPASSWORD=password
```

Example

```
-DBPASSWORD=secretdbpasswd
```

A.2.89 DBSCHEMA (packager)

Summary

This packaging switch specifies the database schema to use if it is not specified in the DSN. The DBSCHEMA packaging switch is required only if the database used in the application requires a schema.

Arguments

The switch requires the name of the database schema as an argument.

Syntax

```
-DBSCHEMA=schema
```

Example

```
-DBSCHEMA=MySchema
```

A.2.90 DBUSER

Used with: DBAdmin, Packager

Summary

This packaging switch specifies the database user to use if it is not specified in the DSN. The DBUSER packaging switch is required only if the database used in the application requires a schema.

Arguments

The argument for this switch is the username.

Syntax

```
-DBUSER=user name
```

Example

```
-DBUSER=admin
```

A.2.91 DDAFILEMAP

Used with: Engine, LiveEditor

Use the DDAFILEMAP switch to change the data file target to a DDA routine. When the file opens, the engine or LiveEditor checks for the mapping, overrides it, and then opens the correct file.

Syntax

```
-DDAFILEMAP=<fileName>,<programType>,<module>,<function>,[<bufferSize>],
[<openParams>]
```

Arguments	Required	Supported values
`FileName`	Yes	The name of the data file
`ProgramType`	Yes	The language used to create the module: • `ASM` • `COBOL` • `DLL` (for C and C++) • `PL1`
`module`	Yes	The location of the DLL file
`function`	Yes	The function name in the DLL/routine
`bufferSize`	No	The size of the buffer (in bytes), which must accommodate the size of the output
`openParams`	No	Parameters specific to the module

Example

```
-
DDAFILEMAP=DriverFile.dat,COBOL,C:\CustomDLL\MyDLL.dll,MyRoutine,5000,MyParams
```

A.2.92 DDALLOC

Use the DDALLOC engine switch to optimize engine performance on a mainframe platform. The DDALLOC engine switch lets you specify the type of memory allocation that you want to use for

a specified Data Description (DD) and dataset name (DSN).

Syntax

```
-DDALLOC=<ddname>,<dsname>,<dataset_
status>,<normalDISP>,<conditionalDISP>,<space_type>,<primary_
space>,<secondary_space_>,<record_organization>,<logical_record_
length>,<blocksize>>,<record_format>,<dataset_
organization>,DSNtype>,<RLSE>,<CLOSE>,
```

Specify the following required and optional arguments:

Arguments	Required	Supported values
`ddname`	Yes	The DD name that you want to use.
`dsname`	Yes	The dataset name that you want to use.
`dataset_status`	No	The `dataset status` argument replaces either the normal DISP value or the conditional DISP value and specifies one of the following disposition values: • `NEW` • `OLD` • `MOD` • `SHR`
`normalDISP`	No	The `normalDISP` argument specifies a normal disposition that uses one of the following disposition values: • `CATLG` • `UNCATLG` • `DELETE` • `KEEP`
`conditionalDISP`	No	The `conditionalDISP` argument specifies a conditional disposition: • `CATLG` • `UNCATLG` • `DELETE` • `KEEP`

Arguments	Required	Supported values
space_type	Yes	The space_type argument specifies the type of space allocation that you want to use: • TRK • TRACK • CYL • CYLINDER • Any integer that specifies the amount of space to allocate in blocks
primary_space	Yes	The primary space argument specifies a positive integer that indicates the number of units to allocate. The value of the integer that you specify must be less than 2^31.
secondary_space	Yes	The secondary space argument specifies a positive integer that indicates the number of units to allocate when more space is required. The value of the integer that you specify must be less than 2^31.
record_ organization	Yes	The record organization argument specifies the type of VSAM command that you want to use.
logical_record_ length	Yes	The logical record length argument specifies an integer to determine logical record length. The integer must be no more than 2^16-1.
blocksize	Yes	The blocksize argument specifies an integer that indicates the block size that you want to use. The integer must be no more than 2^15-1.
record_format	Yes	The record format argument specifies the mainframe flat file format that you want to use.
dataset_ organization	Yes	The dataset organization argument specifies whether a dataset is a sequential dataset, a partitioned dataset, or a VSAM file: • SEQ • PS • PO • Library • VSAM
DSNtype	Yes	The DSNtype argument specifies the type of dataset that you want to use. Select one of the following values: • HFS • LIBRARY • PDS • PIPE
RLSE	Yes	The RLSE argument specifies that the engine release any unused allocated space when the file closes.
CLOSE	Yes	The CLOSE argument specifies that the engine de-allocate the dataset when the file closes.

Example

```
-DDALLOC=ddname,dsname,CTLG,CTLG,TRK,65535,65535,F,SEQ,PIPE,RLSE,CLOSE
```

Supported in

7.0.626 and later

A.2.93 DDAMESSAGE

Use the DDAMESSAGE switch when DDAOUTPUT specifies how messages are to be handled to control messages.

Syntax

```
-DDAMESSAGE=<message_option>
```

Arguments	Required	Supported values
`<message_option>`	Yes	• `APPEND`—If you select `APPEND`, the engine places the message in the DDA output buffer. • `NONE`—If you select `NONE`, the engine deletes the message. • `FILE`—If you select `FILE`, the engine sends the message to a message file as usual. This is the default.

Example

```
-DDAMESSAGE=APPEND
```

A.2.94 DDAOUTPUT

Use the DDAOUTPUT switch to specify a DDA routine for receiving the buffered output, with one output write per record.

Syntax

```
-DDAOUTPUT=<ProgramName>,<Module>,<Function>,<BufferSize>,<OpenParams>
```

Argument Name	Required	Description
<ProgramType>	Yes	The language used to create the module. Select from the following options: • ASM • COBOL • DLL (for C and C++) • PL1
<Module>	Yes	The location of the DLL file
<Function>	Yes	The function name in the DLL/routine
<BufferSize>	Yes	The size of the buffer (in bytes), which must accommodate the size of the output
<OpenParams>	No	Parameters specific to the module

Example

```
-DDAOUTPUT=DLL,C:\Path\CustomDDA.dll,MyFunction,5000,MyParams
```

A.2.95 DDAOUTPUTFILE

Use the DDAOUTPUTFILE switch to specify the fully-qualified directory in which to place temporary output files.

Syntax

```
-DDAOUTPUTFILE=<directory>
```

The argument for this switch is the fully-qualified directory. The default directory for this switch is `C:\DDAOutputfile`.

Example

```
-DDAOUTPUTFILE=C:\MyCustomDirectory\
```

A.2.96 DEFAULT_LABEL_TICKS_TO_NUMSERIES

Deprecated: This switch has been deprecated.

Use the DEFAULT_LABEL_TICKS_TO_NUMSERIES switch to correct grid lines in traditional area and bar charts by setting the default number of ticks based on the number of data series in

use. The DEFAULT_LABEL_TICKS_TO_NUMSERIES engine switch applies only to the data label method, and enables deprecated chart functionality from version 3.5.105.

Syntax

```
-DEFAULT_LABEL_TICKS_TO_NUMSERIES
```

This switch does not have any arguments.

Supported in

4.5.401 and later

A.2.97 DEFAULT_LAYER_FLOW_FRAMES_RESPECT_LANGUAGE

Use the DEFAULT_LAYER_FLOW_FRAMES_RESPECT_LANGUAGE switch to force flow frames set on a default language layer to honor language settings. This switch does not override the options on the **Language** tab of the page properties.

Syntax

```
-DEFAULT_LAYER_FLOW_FRAMES_RESPECT_LANGUAGE
```

This switch does not have any arguments.

A.2.98 DEFAULT_PAGE_ALLOCATIONS

Use the DEFAULT_PAGE_ALLOCATIONS engine switch to optimize engine performance for applications that use large amounts of memory. The DEFAULT_PAGE_ALLOCATIONS engine switch lets you specify the number of pages that you want to allocate for processing when the engine adds a new document to the document list.

Syntax

```
-DEFAULT_PAGE_ALLOCATIONS=<number of pages>
```

The argument for this switch is any number between 1 and 50 that represents the number of pages that you want to allocate.

Example

```
-DEFAULT_PAGE_ALLOCATIONS=5
```

Supported in

7.0.619 and later

A.2.99 DEFAULT_REF_VARS

Use the DEFAULT_REF_VARS switch to reset all variables in a lookup file to their default values when the reference key is empty.

Syntax

```
-DEFAULT_REF_VARS
```

This switch does not have any arguments.

Supported in

4.0.058 and later

A.2.100 DISABLE_ADDING_META_CLASS_TO_HTML_EMAIL_SPANS

After you upgrade from Exstream Design and Production version 16.3.3 to a later version, `class` information for CSS styles based on metadata is added to `<span>` tags in HTML (email) output.

For backward compatibility, use the DISABLE_ADDING_META_CLASS_TO_HTML_EMAIL_SPANS engine switch to revert to the original behavior. This switch is automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 16.3.2 or earlier.

Syntax

```
-DISABLE_ADDING_META_CLASS_TO_HTML_EMAIL_SPANS
```

This switch does not have any arguments.

Supported in

16.3.3 and later

A.2.101 DISABLE_BUNDLE_INCREMENT_FOR_NON_QUALIFIED_QUEUES

For backward compatibility after you upgrade to version 8.0.303 or later, use the DISABLE_BUNDLE_INCREMENT_FOR_NON_QUALIFIED_QUEUES engine switch with applications that include multiple queues in order to cause the sequence numbers that are reported by the 'SYS_BundleInQueue1' through 'SYS_BundleInQueue5' system variables to increment only for bundles that match the rule defined for the current output queue (qualified queues).

This switch will be automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 8.0.302 or earlier.

Syntax

```
-DISABLE_BUNDLE_INCREMENT_FOR_NON_QUALIFIED_QUEUES
```

This switch does not have any arguments.

Supported in

- 7.0.642 and later maintenance releases
- 8.0.335 and later maintenance releases
- 8.6.108 and later maintenance releases
- 9.0.102 and later maintenance releases
- 9.5.101 and later versions

A.2.102 DISABLE_DOCX_PAGENUM_RESET

Use the DISABLE_DOCX_PAGENUM_RESET engine switch when you run an application that uses the SYS_PageInQueue system variable and you do not want the page number to reset between customers. If you use the SYS_PageInQueue system variable to produce DOCX output, the page count will automatically reset between customers due to how Microsoft Word paginates a document. The DISABLE_DOCX_PAGENUM_RESET engine switch will prevent the page number from resetting between customers, even if your design includes the SYS_PageInQueue system variable.

Syntax

```
-DISABLE_DOCX_PAGENUM_RESET
```

This switch does not have any arguments.

Supported in

- 8.0.330 and later maintenance releases
- 8.6.104 and later maintenance releases
- 9.0.101 and later versions

A.2.103 DISABLE_DRAWIMAGESFIRST

Use the DISABLE_DRAWIMAGESFIRST switch to prevent images from being drawn before other design elements on a page. When importing external content, such as a PDF, the DISABLE_DRAWIMAGESFIRST switch forces the engine to compose the external content first, which allows images to appear correctly on a page.

Syntax

```
-DISABLE_DRAWIMAGESFIRST
```

This switch does not have any arguments.

A.2.104 DISABLE_FLOW_TARGETS_FOR_LIVE

For backward compatibility after you upgrade to version 8.0, use the DISABLE_FLOW_ TARGETS_FOR_LIVE engine switch to disable targeted flow on DLF output when you produce output from an application with the following conditions and settings:

- The application contains multiple output queues.
- The **Enable flow frame targeting** check box is selected on the **Documents** tab of the application properties.

Syntax

```
-DISABLE_FLOW_TARGETS_FOR_LIVE
```

This switch does not have any arguments.

Supported in

8.0.326 and later

A.2.105 DISABLE_MIXEDPLEX_LOCK

Use the DISABLE_MIXEDPLEX_LOCK switch to prevent “locking” the duplex mode of the last non-flow page for subsequent flow pages. Using this switch allows a duplex flow page to print duplex, even if it follows a simplex page.

This switch does not have any arguments.

Syntax

```
-DISABLE_MIXEDPLEX_LOCK
```

A.2.106 DISABLE_RECIPIENT_COPIES

Use the DISABLE_RECIPIENT_COPIES switch to suppress all recipient copies of customers' documents in the output. This switch lets you create output that includes only the original customer documents, without including any copies for other recipients that might be included in the application.

Syntax

```
-DISABLE_RECIPIENT_COPIES
```

This switch does not have any arguments.

A.2.107 DISABLE_SEARCH_KEY_CONVERT

Use the DISABLE_SEARCH_KEY_CONVERT switch to override the use of the native PDL format to display search keys. For example, if you specify that the search key be included in a **Specified PDL**, such as **PDF output**, and set the **Type** to **As is**, the search key appears in PDF format unless you use the DISABLE_SEARCH_KEY_CONVERT switch.

Syntax

```
-DISABLE_SEARCH_KEY_CONVERT
```

This switch does not have any arguments.

Supported in

5.0.056 and later

A.2.108 DISABLE_TLE_MUP_PAGE_BREAKS

Use the DISABLE_TLE_MUP_PAGE_BREAKS switch to disable forced multiple-up page breaks to handle correct placement of customer TLE search keys.

Syntax

```
-DISABLE_TLE_MUP_PAGE_BREAKS
```

This switch does not have any arguments.

Example

```
-DISABLE_TLE_MUP_PAGE_BREAKS
```

Supported in

- 3.5.079 and later maintenance releases
- 4.5.411 and later maintenance releases
- 5.0.004 and later versions

A.2.109 DISABLEPRINT

Use the DISABLEPRINT switch to override the use of output queues in package files.

Syntax

```
-DISABLEPRINT=<option>
```

Arguments	Required	Supported values
<option>	Yes	• YES—Override output queues. • YES_AND_OMIT_REMINDER—Override output queues, and do not issue a message. • NO—Turn the switch off. This is the default value.

Example

```
-DISABLEPRINT=YES
```

A.2.110 DISABLESORT

Use the DISABLESORT switch to omit creating a sort data file and a sort index file, even though the package file is created for sorted queues.

Syntax

```
-DISABLESORT=<option>
```

Arguments	Required	Supported values
<option>	Yes	• YES—Create a sort data and sort index file. • NO—Do not create a sort data and sort index file. This is the default value.

Example

```
-DISABLESORT=YES
```

A.2.111 DO_NOT_DEFAULT_REF_VARS

Use the DO_NOT_DEFAULT_REF_VARS switch to prevent variables from being reset to their default values before they are read from a lookup file.

Syntax

```
-DO_NOT_DEFAULT_REF_VARS
```

This switch does not have any arguments.

Supported in

16.4.0 and later

A.2.112 DO_NOT_FORCE_POSTSORT_MUP_BREAK

Use the DO_NOT_FORCE_POSTSORT_MUP_BREAK switch to disable forced multiple-up page breaks when you use dynamic imports, post-sort processing, and multiple-ups. Using these features together causes multiple-up page breaks at each customer.

Syntax

```
-DO_NOT_FORCE_POSTSORT_MUP_BREAK
```

This switch does not have any arguments.

A.2.113 DO_NOT_RUN_LATE_PAGE_RULES

Use the DO_NOT_RUN_LATE_PAGE_RULES switch to bypass page rules that are set on the output queue object of a document. When you specify the DO_NOT_RUN_LATE_PAGE_RULES switch, the engine processes all of the page rules one time during the normal stage of page processing. If you set a page rule on the output queue object of a document to run at the end of an engine run, the DO_NOT_RUN_LATE_PAGE_RULES switch prevents page rule execution.

Syntax

```
-DO_NOT_RUN_LATE_PAGE_RULES
```

This switch does not have any arguments.

Example

Suppose that you created an application in Exstream Design and Production 6.0 that contains multiple section-driven documents. Your application includes one document that contains page rules that exclude some output during run time, and a second document that contains a document rule that has “At end” specified as the rule run time. The “At end” rule is used to hold outputs that are composed during processing until the end of a run. At the end of the run, the engine runs pending rules and discards any documents that are excluded by another rule.

When you use Exstream Design and Production 6.0 to run the application, the engine produces the output as expected. However, if you use Exstream Design and Production 8.0 to run the same application, you find that there are pages missing from the finished output. In this case, the engine is honoring an exclusion rule during processing of the document that contains the page rules, and then executing the page rules again when the “At end” document rule is fired. When the exclusion rule is applied the second time, it causes the engine to erroneously discard a page that was created after the rule was executed the first time.

To prevent the engine from running the page rules more than once, you can use the engine switch, DO_NOT_RUN_LATE_PAGE_RULES. This switch tells the engine to run the page rules only once – at the normal time during document processing – and not to run the page rules again when the “At end” rule is triggered.

Supported in

- 3.5.079 and later maintenance releases
- 4.0.085 and later maintenance releases
- 4.5.413 and later maintenance releases
- 5.0.006 and later maintenance releases
- 5.5.101 and later versions

A.2.114 DOCCACHE

Use the DOCCACHE switch with On Demand Delivery production (on UNIX, Windows, and z/OS) to limit the number of documents (and their objects) held at any one time in memory.

The DOCCACHE switch is used for applications containing hundreds of documents to complete large runs with lower memory requirements. For example, if you enter 50 here, a run uses only enough memory to support the first 50 customer documents. The addition of the 51st document removes the first document in memory; inclusion of the 52nd document removes the second document in memory; and so on.

Syntax

```
-DOCCACHE=<# of documents to cache>
```

The required argument is the number of documents to cache in memory. This switch has no default value.

Example

```
-DOCCACHE=50
```

A.2.115 DOCUMENT

Use this packaging switch to package a particular document in a folder, or to package all of the documents in the specified folder. Documents in subfolders are not included.

Syntax

```
-DOCUMENT= <documentName>,[<folderPath>],[<excludeErrors>]
```

Argument	Required	Supported values
`documentName`	Yes	• Document name—Packages the indicated document in the specified folder • An asterisk (*)—Packages all of the documents in the specified folder
`folderPath`	No	The folder path with different folder levels separated by a dot (.)
`excludeErrors`	No	`EXCLUDEOK`—Changes the severity of error messages that are generated if a document will not be packaged (because of the version status, expiration date and so on) to informational messages **Note:** This argument is valid only if you package all the documents in a folder. Specify `EXCLUDEOK` as an argument if you use other switches, such as the WIP switch, that limit the object versions that will be packaged. When you use this parameter, the normally severe error that is caused by a document being excluded is changed to an informational message to ensure that packaging is successful.

Example

```
-DOCUMENT=*,MyApps.Statements.June,EXCLUDEOK
```

A.2.116 DOCX2DXF_LOCATION

Use the DOCX2DXF_LOCATION switch to specify the file path or file name of the XSL that will be used to process DOCX import files. When importing DOCX files at run time, the engine uses the XSL to parse the DOCX that is being imported.

Syntax

```
-DOCX2DXF_LOCATION=<file_path or file_name>
```

If you specify only a file path, then the engine references the specified directory to locate the XSL file that will be used for processing the DOCX import files. If a file name is specified, then the engine references the specific XSL file name that will be used for processing the DOCX import files. By default, the engine expects the XSL file to be located in the same directory as the engine.

Example

```
-DOCX2DXF_LOCATION=C:\Path\docx2dxf.xsl
```

A.2.117 DONT_COLLAPSE_PARAGRAPH_SPACING_IN_HTML

For backward compatibility after you upgrade to version 9.5.201, use the DONT_COLLAPSE_PARAGRAPH_SPACING_IN_HTML switch to preserve how paragraph spacing is handled in HTML and HTML (email) output. This switch is automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 9.5.199 or earlier.

In Exstream Design and Production versions earlier than 9.5.201, margins that were defined in an HTML import file (versions 9.5.1x only) or on paragraphs in Designerwere translated to padding in HTML and HTML (email) output. This resulted in some visual differences in the paragraph spacing between HTML output and other output types.

For example, suppose that you have two paragraphs in your design. Paragraph 1 has a 50-pixel bottom margin and paragraph 2 has a 50-pixel top margin. When these values are applied to the margins in the HTML output, the two margin values are collapsed, resulting in 50 pixels of space between the paragraphs. However, when these values are applied to the padding, then the padding values are combined, resulting in 100 pixels of space between the paragraphs.

This issue was corrected in Exstream Design and Production 9.5.201.

Syntax

```
-DONT_COLLAPSE_PARAGRAPH_SPACING_IN_HTML
```

This switch does not have any arguments.

Supported in

9.5.201 and later

A.2.118 DONT_CORRECT_HTML_HEADER_FOOTER_POSITIONS_AND_COLOR

In versions of Exstream Design and Production earlier than the versions listed below, the following issues were sometimes seen in HTML output:

- Headers and footers within text boxes or tables were improperly repeated.
- Header and footer paragraphs were misplaced if the text box flowed to a new page.
- Text color used in a header or footer was applied to the text of the main body.

The default behavior has changed such that the headers and footers are not repeated, and any text color within them does not change the color of the text in the main body.

For backward compatibility after you upgrade to any of the Exstream Design and Production versions listed below, use the DONT_CORRECT_HTML_HEADER_FOOTER_POSITIONS_AND_COLOR engine switch to revert to the original behavior.

Syntax

```
-DONT_CORRECT_HTML_HEADER_FOOTER_POSITIONS_AND_COLOR
```

This switch does not have any arguments.

Supported in

- 9.5.308 and later maintenance releases
- 16.2.4 and later maintenance releases
- 16.3.3 and later maintenance releases
- 16.4 Update 1 and later versions

A.2.119 DONT_ENSURE_DIV_FOR_EMBED_IN_HTML

Prior to Exstream Design and Production 9.5.2, if you had two paragraphs that had the same formatting, and the first paragraph did not have embedded content but the second one did, the HTML output would create a line break for the second paragraph rather than a new paragraph. For example, the output might appear as:

```
<p> Paragraph 1

<br> Paragraph <div> embedded content </div>

</p>
```

In 9.5.2 or later, this issue has been corrected so that the output will appear as:

```
<p> Paragraph 1 </p>
```

```
<div> Paragraph 2 <div> embedded content </div> </div>
```

For backward compatibility after you upgrade to 9.5.2, use the DONT_ENSURE_DIV_FOR_EMBED_IN_HTML switch.

The DONT_ENSURE_DIV_FOR_EMBED_IN_HTML switch will appear in the output when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 9.5.199 or earlier.

You will not encounter this issue if you use the ALWAYS_HONOR_DESIGN_PARAGRAPHS_FOR_HTML switch, which forces a new <p> or tag for each paragraph in the HTML output.

Syntax

```
-DONT_ENSURE_DIV_FOR_EMBED_IN_HTML
```

This switch does not have any arguments.

Supported in

9.5.201 and later

A.2.120 DONT_FORCE_NEW_PARAS_WITH_STYLES_IN_HTML

When creating HTML or HTML (email) output in Exstream Design and Production versions earlier than 16.2.0, some text paragraphs were combined with the preceding paragraph (<p> tag) when they both had the same styles applied to them. This included the following styles: first line indent, space above, space below, a frame style, or clear. This change in HTML structure resulted in visual differences between the design and the output.

This issue was corrected in Exstream Design and Production version 16.2.0, so that a new <p> tag is created for each of the text paragraphs with those styles applied to them.

For backward compatibility after you upgrade to Exstream Design and Production 16.2.0, use the DONT_FORCE_NEW_PARAS_WITH_STYLES_IN_HTML switch. This switch will be automatically included when theENABLE_BACKWARD_COMPAT switch is used and the version specified is 9.5.999 or earlier.

Syntax

```
-DONT_FORCE_NEW_PARAS_WITH_STYLES_IN_HTML
```

This switch does not have any arguments.

Supported in

16.2.0 and later

A.2.121 DONT_MAKE_DOCX_TOC_ADJUSTMENTS

In versions of Exstream Design and Production earlier than the versions listed below, if your design included a table of contents, DOCX output sometimes contained an extra flow page with misplaced footers. This issue was corrected in the versions listed below.

For backward compatibility after you upgrade to any of the Exstream Design and Production versions listed below, use the DONT_MAKE_DOCX_TOC_ADJUSTMENTS engine switch to revert to the original behavior.

Syntax

```
-DONT_MAKE_DOCX_TOC_ADJUSTMENTS
```

This switch does not have any arguments.

Supported in

- 9.5.309 and later maintenance releases
- 16.2.4 and later maintenance releases
- 16.3.3 and later maintenance releases
- 16.4 Update 2 and later versions

A.2.122 DONT_MOVE_BLANK_PARA_NEWLINE

In Exstream Design and Production versions earlier than 16.3.0, for designs that included a blank text paragraph, formatting placed on the following text paragraph were also applied to the blank text paragraphs. In the resulting HTML, both text paragraphs were included in the same parent tag (such as `<p>` or `<td>`), so the CSS styles applied to both paragraphs. For example:

```
<table>
<tr>
<td class="bold">
   <span></br></span>
   <span>Text</span>
</td>
```

```
...
```

This produced differences in HTML and HTML (email) output when compared to the design. The issue was corrected in Exstream Design and Production. In the resulting HTML, the blank text paragraph is placed in a separate parent tag (such as <p> or <td>), each with its own CSS styles applied. For example:

```
<table>
<tr>
<td>
   <span></br></span>
</td>
<td class="bold">
   <span>Text</span>
</td>
...
```

For backward compatibility after you upgrade to 16.3.0, use the DONT_MOVE_BLANK_PARA_NEWLINE switch merge blank text paragraphs with the text paragraph that immediately follows. This switch will be automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 16.2.999 or earlier.

> **Note:** This switch is related to ALLOW_MERGE_PARAS_IN_HTML.

Syntax

```
-DONT_MOVE_BLANK_PARA_NEWLINE
```

This switch does not have any arguments.

Supported in

16.3.0 and later

A.2.123 DONT_POPULATE_RTF_CELL_BORDERS

In versions of Exstream Design and Production earlier than the versions listed below, some table borders did not render correctly in dynamically imported RTF content. This issue was corrected in the versions listed below.

For backward compatibility after you upgrade to any of the Exstream Design and Production versions listed below, use the DONT_POPULATE_RTF_CELL_BORDERS engine switch to revert to the original behavior.

Syntax

```
-DONT_POPULATE_RTF_CELL_BORDERS
```

This switch does not have any arguments.

Supported in

- 16.3.3 and later maintenance releases
- 16.4 Update 2 and later versions

A.2.124 DONT_READ_BULLETS_AND_ NUMBERS

Use the DONT_READ_BULLETS_AND_NUMBERS switch to prevent a screen reader from reading bullets and numbers in tagged lists contained in accessible PDF output.

For more information about creating accessible PDF output and working with lists in accessible PDF output, see *Designing Customer Communications* in the Exstream Design and Production documentation.

Syntax

```
-DONT_READ_BULLETS_AND_NUMBERS
```

This switch does not have any arguments.

Supported in

- 9.5.308 and later maintenance releases
- 16.2.4 and later maintenance releases
- 16.3.3. and later maintenance releases
- 16.4.0 update 1 and later versions

A.2.125 DONT_RECALC_SIZE_TO_END_SECTION

In versions of Exstream Design and Production earlier than the versions listed below, if a design included flow frames and a table that was set to split and flow across the frames, some tables extended outside the frame and did not split properly. This issue was corrected in the versions listed below.

For backward compatibility after you upgrade to any of the Exstream Design and Production versions listed below, use the DONT_RECALC_SIZE_TO_END_SECTION engine switch to revert to the original behavior.

Syntax

```
-DONT_RECALC_SIZE_TO_END_SECTION
```

This switch does not have any arguments.

Supported in

- 16.3.3 and later maintenance releases
- 16.4 Update 2 and later versions

A.2.126 DONT_REMOVE_EMPTY_LINES_WITH_SPECIAL_RETURNS

In Exstream Design and Production version 8.6.103, line feed characters alone and column breaks are not recognized as line separators in variables, and the **Remove empty variable lines** check box in the **Text Properties** dialog box in Designer is not honored when either of these characters is used. This behavior was changed by CR 60210, and now the **Remove empty variable lines** setting is honored for line feed characters alone and column breaks.

For backward compatibility after you upgrade from version 8.6.103, use the DONT_REMOVE_EMPTY_LINES_WITH_SPECIAL_RETURNS switch to keep empty lines that contain special return characters.

Syntax

```
-DONT_REMOVE_EMPTY_LINES_WITH_SPECIAL_RETURNS
```

This switch does not have any arguments.

Supported in

8.6.103 and later

A.2.127 DONT_RESET_COMPONENTS_IN_SECTION_DOCS

For backward compatibility after you upgrade to version 9.0.106, use the DONT_RESET_COMPONENTS_IN_SECTION_DOCS engine switch to disable the reset of components when re-flowing section documents. When you generate output from an application that contains a section-driven table that uses output sorting and bundling, the last section-driven table for each customer may be missing in the final output due to the table not being properly reset.

This switch will be automatically included when the ENABLE_BACKWARD_COMPAT engine switch is used and the version specified is 9.01.01.

Syntax

```
-DONT_RESET_COMPONENTS_IN_SECTION_DOCS
```

This switch does not have any arguments.

Supported in

- 9.0.106 and later maintenance releases
- 9.5.101 and later versions

A.2.128 DONT_RESOLVE_VARIABLES_IN_PLAIN_TEXT_IMPORT

If you want to prevent variables from resolving in plain text imports, use the DONT_RESOLVE_VARIABLES_IN_PLAIN_TEXT_IMPORT switch. In Exstream Design and Production version 8.0, the **Text files container variables** check box was removed from the **Placeholder** tab of a placeholder variable. When the check box was removed, the default behavior was changed to always resolve variables in order to make the behavior consistent with other import types that supported referencing Exstream Design and Production variables by name. Use the DONT_RESOLVE_VARIABLES_IN_PLAIN_TEXT_IMPORT switch if you want backward compatibility to applications that previously did not resolve variables in plain text imports.

Syntax

```
-DONT_RESOLVE_VARIABLES_IN_PLAIN_TEXT_IMPORT
```

This switch does not have any arguments.

Supported in

8.0 and later

A.2.129 DONT_RIGHT_ALIGN_EMBEDS_IN_RTL_LAYERS

For backward compatibility after you upgrade to version 9.5.301, use the DONT_RIGHT_ALIGN_EMBEDS_IN_RTL_LAYERS engine switch. This switch will be automatically included when the ENABLE_BACKWARD_COMPAT engine switch is used and the version specified is 9.5.299 or earlier. In versions earlier than version 9.5.301, embedded objects were left-aligned in right-to-left (RTL) language layers by default. In version 9.5.301, this issue was corrected.

Syntax

```
-DONT_RIGHT_ALIGN_EMBEDS_IN_RTL_LAYERS
```

This switch does not have any arguments.

Supported in

9.5.301 and later

A.2.130 DONT_USE_CUSTOMER_LANGUAGE_TO_PARSE_DATES

For backward compatibility, after you upgrade to Exstream Design and Production 9.5 or later, use the DONT_USE_CUSTOMER_LANGUAGE_TO_PARSE_DATES engine switch if you want the engine to use only the default language for the application—and not the language specified in the driver file—when parsing the names of months. For example, if the default language of your application is English, but the driver file specifies Spanish month names for customers in Spanish-speaking locales, then adding this switch causes the engine to ignore the information in the driver file and use English month names instead.

Syntax

```
-DONT_USE_CUSTOMER_LANGUAGE_TO_PARSE_DATES
```

This switch does not have any arguments.

Supported in

9.5 and later

A.2.131 DONT_USE_REPEAT_ROW_NUMBER_FOR_EMBEDDED_OBJS

Prior to Exstream version 16.4 Update 2, if your output contained automated tables with embedded objects that were controlled by an array variable, each repeating row of the table used the first element in the array.

Starting with version 16.4 Update 2, for empty image objects or object hyperlinks that use array variables, the array index is based on the number of the repeating row that contains that object. That is, an array variable in the nth instance of a repeating row will use the nth element in the array.

For backward compatibility after you upgrade to version 16.4 Update 2, use the DONT_USE_REPEAT_ROW_NUMBER_FOR_EMBEDDED_OBJS switch to revert to the previous behavior for empty image objects and object hyperlinks.

This switch will be automatically included when the ENABLE_BACKWARD_COMPAT engine switch is used and the version specified is 16.4.009 or earlier.

Syntax

```
-DONT_USE_REPEAT_ROW_NUMBER_FOR_EMBEDDED_OBJS
```

This switch does not have any arguments.

Supported in

16.4 Update 2 and later

A.2.132 DONT_USE_RESPONSIVE_DESIGN_IN_HTML

In Exstream Design and Production versions earlier than 16.3.0, responsive design features, such as hiding content or changing its layout on small screens, were supported only in HTML (email) output. In version 16.3.0, this functionality was extended to HTML5 output created from a container design.

For backward compatibility after you upgrade to Exstream Design and Production 16.3.0, use the DONT_USE_RESPONSIVE_DESIGN_IN_HTML switch to disable responsive design features when you produce HTML5 output from a container design. This switch is automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 16.2.999 or earlier.

Keep in mind that as long as you use this switch, the responsive features will not be included in HTML output, even if you repackage.

Syntax

```
-DONT_USE_RESPONSIVE_DESIGN_IN_HTML
```

This switch does not have any arguments.

Supported in

16.3.0 and later

A.2.133 DRAWIMAGESFIRST

Use the DRAWIMAGESFIRST switch to draw all images to the page first. This switch prevents images from overlapping variable data.

> **Note:** You cannot use the DRAWIMAGESFIRST switch with Metacode output.

Syntax

```
-DRAWIMAGESFIRST
```

This switch does not have any arguments.

A.2.134 DRIVERLISTFILE

Used with: Engine

Use the DRIVERLISTFILE switch to identify the location of a text file that identifies a list of driver files. The file specified by the DRIVERLISTFILE switch must contain the absolute or relative path to each Live document to be used as a driver file (with each path on a separate line).

Sample text file containing a list of Live documents

```
C:\Program Files\Exstream\CustomerWelcome003423.DLF
C:\Program Files\Exstream\CustomerWelcome003553.DLF
C:\Program Files\Exstream\CustomerWelcome003801.DLF
C:\Program Files\Exstream\CustomerWelcome004463.DLF
C:\Program Files\Exstream\CustomerWelcome004493.DLF
C:\Program Files\Exstream\CustomerWelcome004528.DLF
C:\Program Files\Exstream\CustomerWelcome004687.DLF
C:\Program Files\Exstream\CustomerWelcome005923.DLF
C:\Program Files\Exstream\CustomerWelcome006702.DLF
C:\Program Files\Exstream\CustomerWelcome006723.DLF
```

In order to be processed in a single batch run, all the Live documents must have the same layout. If the engine encounters a Live document that does not match the layout specified in the

fulfillment application, you will receive an error. When the Live documents are processed, they inherit the data file properties of the primary driver file that provides the XML layout in the fulfillment application.

Syntax

```
-DRIVERLISTFILE=<fileName>
```

The only argument for this switch is the fully qualified file name of the text file that contains the list of driver files.

Example

```
-DRIVERLISTFILE=C:\Program Files\OpenText\DriverList.txt
```

A.2.135 DSB_PDF_ACC_TAG_IMPORT

By default, PDFs that contain accessibility tags will retain those tags during run-time import into accessible PDF output. However, in some cases Exstream Design and Production might reorder those tags or your accessibility tool might not read the content correctly. The DSB_PDF_ACC_TAG_IMPORT switch lets you remove those tags from the imported PDF. This switch does not affect any accessibility tags created within the Exstream environment.

For backward compatibility after you upgrade to the Exstream Design and Production versions listed below, use the DSB_PDF_ACC_TAG_IMPORT engine switch to disable all the accessibility tags in a PDF that is dynamically imported into an application at run time.

Syntax

```
-DSB_PDF_ACC_TAG_IMPORT
```

This switch does not have any arguments.

Supported in

- 9.0.118 and later maintenance releases
- 9.5.304 and later maintenance releases
- 16.2.1 and later maintenance releases
- 16.3.0 and later versions

A.2.136 DSN

The DSN switch opens the Database Administrator utility and selects the specified database DSN. The DSN switch is also used to specify the database on which to run database

maintenance. Either the DSN switch or the ACCESSDB switch is required to run database maintenance from the command line.

Syntax

```
-DSN=<databaseDSN>,<schema>,<database_username>,<database_password>
```

Argument	Required
`<databaseDSN>`	Yes
`<schema>`	No
`<database_username>`	No
`<database_password>`	No
`DBAUTHENTICATION`	Optional (If you use this argument, then the database user name and database password arguments are ignored.)

Example

```
-DSN=MyDSN,,MyUserName,MyPassword
```

A.2.137 DSN (DBAdmin)

The DSN switch opens the Database Administrator utility and selects the specified database DSN. The DSN switch is also used to specify the database on which to run database maintenance. Either the DSN switch or theACCESSDB (DBAdmin) switch is required to run database maintenance from the command line.

Syntax

```
-DSN=<databaseDSN>,<schema>,<database_username>,<database_password>
```

Argument	Required
`<databaseDSN>`	Yes
`<schema>`	No
`<database_username>`	No
`<database_password>`	No

Argument	Required
DBAUTHENTICATION	Optional (If you use this argument, then the database user name and database password arguments are ignored.)

Example

```
-DSN=MyDSN,,MyUserName,MyPassword
```

A.2.138 DSN (Packager)

This packaging switch specifies the name of the DSN to use. The DSN packaging switch is required.

Syntax

```
-DSN=<dsn_name>
```

The only argument required by the switch is the DSN name.

Example

```
-DSN=ExstreamDB
```

A.2.139 DSNMAP

Use the DSNMAP switch to change the data source at engine run time.

For LiveEditor, use the DSNMAP switch to change the data source when loading LiveEditor. When an end user selects **Live > Data file >Read >Data file name** from the LiveEditor Menu bar, LiveEditor checks for mapping in the default file, overrides it, and then opens the file specified.

Syntax

```
-DSNMAP=<originalDSN>,<newDSN>[,<user_name>,<password>,<schema>]
```

This switch has no default value. Use the following arguments with the DSNMAP switch:

Argument Name	Required	Supported values
<originalDSN>	Yes	The original data source name (DSN) of the data source
<newDSN>	Yes	The new DSN of the data source
<user_name>	No	The user name, if required, to log in to the data source

Argument Name	Required	Supported values
<password>	No	The password, if required, to log in to the data source
<schema>	No	The schema of the data source

Example

```
-DSNMAP=OriginalDSN,NewDSN
```

A.2.140 DYNAMSG/DYNAMESSAGE

Use either the DYNAMESSAGE or DYNAMSG switch to import messages without repackaging.

Syntax

```
-DYNAMSG=<xml_file>,<array_string_variable>,SORT
```

or

```
-DYNAMESSAGE=<xml_file>,<array_string_variable>,SORT
```

Arguments	Required	Supported values
<xml_file>	Yes	The XML file containing the messages in RTF format
<array_ string_ variable>	Yes	The name of the array string variable that contains the messages
SORT	No	If you specify SORT, then the dynamic messages appear in the output in the order they appear in the control variable, rather than in the order in which they are read from the input XML file.

Example

```
-DYNAMESSAGE=message.xml,ArrayVariableName,SORT
```

Supported in

6.0 and later

A.2.141 DYNAMICIMGNAMEMAP

Use the DYNAMICIMGNAMEMAP switch to assign reference names to top-of-print-stream images. Each image requires a separate instance of the switch. To use this switch, your application must contain top-of-print-stream images. If you use the DYNAMICIMGNAMEMAP

switch with the Output Sorting and Bundling module, this switch should be included in your control file for Step Two processing.

Syntax

```
-DYNAMICIMGNAMEMAP=<image>,<ReferenceName>,<output_format>
```

Argument Name	Required	Supported values
`<image>`	Yes	This argument refers to the image name or the absolute location of the image for which you want to create a reference name.
`<reference_name>`	Yes	This argument refers to the reference name you want to assign to an image. You can use any special characters in the reference name except commas, which are used to separate arguments in the switch. Reference names with a number of characters above the limit for an output are truncated. Character limits for image reference names for the following outputs are: • 8 characters for AFP output • 6 characters Metacode output • 15 characters for PostScript, PPML, VIPP, and VPS output You can use two different resource names for the same source image only if you assign each resource name to a different output. Make sure that no two images are assigned the same reference name. This can cause errors in your output.
`<output_format>`	No	With this argument, you can specify the output in which the production environment applies the reference name. If no output is specified, all supported outputs use the same reference name. Only one output can be specified per switch. Outputs that support reference names for dynamic images are as follows and must be entered in the same case-sensitive format: • `AFP` • `Metacode` • `PostScript` • `PPML` • `VIPP` • `VPS`

Example

```
-DYNAMICIMGNAMEMAP=C:\PATH\MyImage.jpg,Image1,PostScript
```

A.3 Switches E-H

A.3.1 EAM_PORT_NUMBER

The EAM_PORT_NUMBER command line switch for silent installation is an optional Exstream Design and Production switch that lets you specify the communication port number that EAM uses for the EAM services.

Note: Beginning with Exstream version 16.2.0, which provides a 64-bit design environment, EAM is no longer supported.

Syntax

```
-EAM_PORT_NUMBER=EAM port number
```

The value for this switch is the port number that EAM uses for the EAM services. By default, if you do not specify a value for the EAM_PORT_NUMBER command line switch, then the installer sets a default value of `7050`.

Example

The following example uses InstallShield and Microsoft standard command line switches, and the EAM_SERVER_INSTALL , EAM_HOST_NAME , and EAM_PORT_NUMBER Exstream Design and Production command line switches to install the English language version of Exstream Design and Production 9.0.101 on an EAM server, to specify the EAM host name and EAM port number, and to write the installation log file to a specific directory.

```
Exstream_setup_9.0.101.exe /s /L1033 /v”/qn
-EAM_SERVER_INSTALL=YES
-EAM_HOST_NAME=EAM-PC
-EAM_PORT_NUMBER=7050
/l*v C:\install_log.txt”
```

Supported in

9.0.101 and later

A.3.2 EAM_SERVER_INSTALL

The EAM_SERVER_INSTALL command line switch for silent installation is an optional Exstream switch that lets you specify whether you want to install the design environment on an EAM server. If you specify that you want to install the design environment on an EAM server, then the installer updates the EAM design service configuration file and installs the EAM design service.

By default, if you do not specify the EAM_SERVER_INSTALL command line switch, then the installer does not install the EAM design service or update the EAM design service configuration file.

Note: Beginning with Exstream Design and Production version 16.2.0, which provides a 64-bit design environment, EAM is no longer supported.

If you specify the EAM_HOST_NAME or EAM_PORT_NUMBER command line switches, then you must specify the EAM_SERVER_INSTALL command line switch.

Syntax

```
-EAM_SERVER_INSTALL=<option>
```

Arguments	Required	Supported values
`<option>`	Yes	• YES—Specifies that you want to install the design environment on an EAM server. • NO—Specifies that you do not want to install the design environment on an EAM server. This is the default value for the EAM_SERVER_INSTALL command line switch.

Example

```
-EAM_SERVER_INSTALL=YES
```

Supported in

9.0.101 and later

A.3.3 EAMKEY

This switch can be used to automatically set a new key in an EAM database. DBAdmin will exit without displaying a dialog. Any error messages will be suppressed. Can be used at the same time as AUTOCREATE or AUTOUPDATE (DBAdmin).

Note: Beginning with Exstream Design and Production version 16.2.0, which provides a 64-bit design environment, EAM is no longer supported.

Syntax

```
-EAMKEY=<key string>
```

The only argument required is the new key.

A.3.4 EBCDIC_CODEPAGE

Use the EBCDIC_CODEPAGE engine switch to specify how ISO 8859-1 encoded characters are mapped when you produce output from an application using the engine on a z/OS platform.

Exstream uses IBM-1047 as the default code page. Exstream cannot automatically detect the file encoding, if your data files on the mainframe are encoded with a different EBCDIC code page, such as IBM-037 or IBM-285, you must use this switch to specify the EBCDIC code page that you are using in the data files.

Syntax

```
-EBCDIC_CODEPAGE=<mappingMethod>
```

The only argument for this switch is the method that you want to use to map ISO 8859-1 encoded characters, and supports the following values:

- `<codePage>`—Uses the specified EBCDIC code page, which should be one of those expected by the `iconv` utility. For example, `IBM-037`.
- `design`—Uses the EBCDIC to ASCII conversion table that is specified in Design Manager, in the **System Settings** properties, on the **Text and Fonts** tab
- `6.1`—Maps ISO 8859-1 encoded characters to EBCDIC as they were mapped in Exstream Design and Production version 6.1 and earlier (for backward compatibility)

Example 1

```
-EBCDIC_CODEPAGE=IBM-037
```

Example 2

```
-EBCDIC_CODEPAGE=design
```

Example 3

```
-EBCDIC_CODEPAGE=6.1
```

Supported in

- 7.0.625 and later maintenance releases
- 8.0.308 and later versions

The `design` argument is available in these versions:

- 7.0.627 and later maintenance releases
- 8.0.309 and later versions

The `6.1` argument is available in the following versions:

- 7.0.627 and later maintenance releases
- 8.0.310 and later versions

A.3.5 EFFECTIVE

Use the EFFECTIVE packaging switch to specify the type of effectivity used in a package file.

Depending on the argument that you use with this packaging switch, you might also need to use the STARTDATE and ENDDATE switches.

Syntax

```
-EFFECTIVE=<effectivityType>
```

The only argument for this switch is the effectivity type of the package file, indicated by the following supported values:

- NOW—The package file is effective as of the current packaging date. This is the default option. For example, if you are creating a package file that contains approved objects, then those objects must have an approval date on or before the current date.
- ASOF—The package file is effective as of the date specified in the ENDDATE switch. For example, if you are creating a package file that contains approved objects, then those objects must have an approval date on or before the specified end date.
- RANGE—The package file is effective in the date range specified using the STARTDATE and ENDDATE switches. For example, if you are creating a package file that contains approved objects, then those objects must have an approval date within the specified start and end dates.

Example 1

```
-EFFECTIVE=NOW
```

Example 2

```
-EFFECTIVE=ASOF
-ENDDATE=12/31/2018
```

Example 3

```
-EFFECTIVE=RANGE
-STARTDATE=12/31/2018
-ENDDATE=12/31/2020
```

Related switches

ENDDATE

STARTDATE

A.3.6 EMBED_NO_SPACE_BEFORE_AFTER

For backward compatibility after you upgrade from version 8.0 or earlier to version 8.6 or later, use the EMBED_NO_SPACE_BEFORE_AFTER engine switch to restore the original position of embedded tables that split and flow to a new page.

After you upgrade, the engine might crash when an embedded table splits and flows to a new page. Additionally, the positioning of embedded tables that split and flow to a new page might change due to two defect fixes in version 8.6 and later.

This engine switch will be automatically included when theENABLE_BACKWARD_COMPAT engine switch is used and the version specified is 8.5.999 or earlier.

> **Note:** The EMBED_NO_SPACE_BEFORE_AFTER engine switch might cause the engine to ignore spacers between embedded tables in a container design.

Syntax

```
-EMBED_NO_SPACE_BEFORE_AFTER
```

This switch does not have any arguments.

Supported in

- 8.6.111 and later maintenance releases
- 9.0.105 and later versions

A.3.7 EMPTYACCESSDB

The EMPTYACCESSDB switch creates and opens an Access database using the specified file name. No tables are created. The file name can use either the .mdb or .accdb extension. Keep in mind that Access versions earlier than 2007 support only the .mdb extension.

Syntax

```
-EMPTYACCESSDB=<Access database path and name>
```

Use the following argument:

Argument	Required
`<Access database path and name>`	Yes
`DBAUTHENTICATION`	No

Example

```
-EMPTYACCESSDB=C:\Temp\Sample.mdb
```

A.3.8 ENABLE_BACKWARD_COMPAT

Use the ENABLE_BACKWARD_COMPAT engine switch to collectively apply the backward compatibility switches that have been implemented since a previous version of Exstream Design and Production that you specify. When you use the ENABLE_BACKWARD_COMPAT switch, the output that you produce approximates the output that would be produced if you were using the version of the engine that you specify for the switch.

For example, suppose that you are using Exstream Design and Production version 9.0.102, but you are producing output using an application that was designed in Exstream Design and Production version 7.0.601. If you notice differences in the output and you want the output to appear the same as the output that you produced using Exstream Design and Production 7.0.601, you can specify `-ENABLE_BACKWARD_COMPAT=7.0.601`. Then, the engine applies the specific backward compatibility switches that were implemented in versions and maintenance releases later than Exstream Design and Production version 7.0.601.

Furthermore, you can specify a second version number in order to exclude backward compatibility switches that were implemented in versions or maintenance releases that are later than that version. That is, the engine applies only the backward compatibility switches that were implemented between the two versions that you specify.

The engine also produces a list of all of the specific backward compatibility switches that apply to the version or versions that you specify, as well as a list of a subset of those switches that actually apply to your application. The lists appear either in the engine message file or in the output file that you specify as an argument.

> **Note:** Using the ENABLE_BACKWARD_COMPAT switch or specific backward compatibility switches might roll back software fixes that were applied later than the version that you specify. You should use this switch only for testing output and for temporary use when you want to maintain the appearance of output from existing applications after an upgrade. For best results, make the updates to your application that are necessary to discontinue using backward compatibility switches as your time and business requirements permit.

For a list of the specific backward compatibility switches that might be applied when you use the ENABLE_BACKWARD_COMPAT switch, see Backward compatibility switches.

Syntax

This switch has no default value. Use one of the following syntaxes with the ENABLE_BACKWARD_COMPAT switch:

Syntax	Description
`-ENABLE_BACKWARD_COMPAT=<version1>`	Specify an earlier version of Exstream Design and Production with which you want your output to be backward compatible. When you produce output, the engine applies all backward compatibility switches that were implemented since the version that you specify. The engine adds the lists of applicable switches to the engine message file. Do not specify a version earlier than 3.0.000.
`-ENABLE_BACKWARD_ COMPAT=<version1>,<version2>`	Specify two earlier versions of Exstream Design and Production, between which you want to apply relevant backward compatibility switches. When you produce output, the engine applies all backward compatibility switches that were implemented since the first version that you specify, but excludes switches that were implemented after the second version that you specify. The engine adds the lists of applicable switches to the engine message file.
`-ENABLE_BACKWARD_ COMPAT=<version1>,<filepath>`	Specify an earlier version of Exstream Design and Production with which you want your output to be backward compatible, and the path of an output file in which you want to record the applicable backward compatibility switches. When you produce output, the engine applies all backward compatibility switches that were implemented since the version that you specify. The engine adds the lists of applicable switches to the output file that you specify.
`-ENABLE_BACKWARD_ COMPAT=<version1>,<version2>,<filepath>`	Specify two earlier versions of Exstream Design and Production, between which you want to apply relevant backward compatibility switches, and the path of an output file in which you want to record the applicable backward compatibility switches. When you produce output, the engine applies all of the backward compatibility switches that were implemented since the first version that you specify, but excludes switches that were implemented after the second version that you specify. The engine adds the lists of applicable switches to the engine message file. The engine adds the lists of applicable switches to the output file that you specify.

For the first version, you can specify any existing version of Exstream Design and Production between version 3.0.000 and the version you are using. However, if you specify a second version, you can specify only one of the following:

- 95101
- 90102
- 90101
- 86107
- 86106
- 86105
- 86103
- 86101
- 86999
- 80333

- 80331
- 80330
- 80325
- 80324
- 80323
- 80322
- 80317
- 80316
- 80312
- 80311
- 80309
- 80304
- 80301
- 80300
- 82999
- 80200
- 80100
- 79999
- 70638
- 70629
- 70619
- 70617
- 70612
- 70608
- 70606
- 70604
- 70601
- 70600
- 70599
- 70408

- 70406
- 70403
- 70402
- 70400
- 70300
- 70004
- 62003
- 62001
- 61999
- 61032
- 61029
- 61028
- 61025
- 61015
- 61014
- 61010
- 61008
- 61006
- 61005
- 61004
- 60999
- 60017
- 60016
- 60015
- 60014
- 60013
- 60012
- 60009
- 60008
- 60000

- 55408
- 55406
- 55305
- 55303
- 55302
- 55207
- 54999
- 50075
- 50074
- 50064
- 50054
- 50048
- 50045
- 50022
- 50020
- 45400
- 45102
- 40120
- 40086
- 40075
- 40039
- 40009

You can specify versions numbers with or without periods. For example, 8.6.102 is equivalent to 86102.

Example

```
-ENABLE_BACKWARD_COMPAT=8.6.102
```

```
-ENABLE_BACKWARD_COMPAT=8.6.102,9.0.102
```

```
-ENABLE_BACKWARD_COMPAT=8.6.102,C:\logs\switchoutput.txt
```

```
-ENABLE_BACKWARD_COMPAT=8.6.102,9.0.102,C:\logs\switchoutput.txt
```

Supported in

- 8.0.335 and later maintenance releases
- 8.6.108 and later maintenance releases
- 9.0.102 and later versions

A.3.9 ENABLE_EPS_PDF_IMPORT_AUTOROTATE

Use the ENABLE_EPS_PDF_IMPORT_AUTOROTATE engine switch to allow Exstream Design and Production to automatically rotate PDF or EPS images that are imported into a placeholder object or empty image object at run time by 90 degrees when needed to better fit the dimensions of the placeholder object or empty image object.

For example, if the horizontal dimension of an empty image object is greater than the vertical dimension, and the reverse is true for the imported image, then the image rotates to better fit the empty image object. By default, the image rotates clockwise when needed. To automatically rotate the image counterclockwise, you can use the ENABLE_EPS_PDF_IMPORT_AUTOROTATE_COUNTERCLOCKWISE engine switch.

The ENABLE_EPS_PDF_IMPORT_AUTOROTATE engine switch applies only to placeholder objects or empty image objects in designs created or edited in Designerversion 8.5.105 or later. PDF or EPS images that are imported into placeholder objects or empty image objects in designs created in versions earlier than 8.5.105 might automatically rotate without using the switch.

The ENABLE_EPS_PDF_IMPORT_AUTOROTATE engine switch is ignored when the rotation of the placeholder object or empty image object is set to a value other than 0.

The ENABLE_EPS_PDF_IMPORT_AUTOROTATE switch was renamed from the ENABLE_PDFEPSIMPORT_AUTOROTATE engine switch, which is available in Exstream Design and Production version 8.6.101 and version 8.6.102.

Syntax

```
-ENABLE_EPS_PDF_IMPORT_AUTOROTATE
```

This switch does not have any arguments.

Supported in

8.6.103 and later

A.3.10 ENABLE_EPS_PDF_IMPORT_AUTOROTATE_COUNTERCLOCKWISE

If you want Exstream Design and Production to automatically rotate PDF files that are imported at run time by 90 degrees counterclockwise (instead of the default 90 degrees clockwise) when needed to better fit the dimensions of a placeholder document, use the ENABLE_EPS_PDF_IMPORT_AUTOROTATE_COUNTERCLOCKWISE engine switch.

Additionally, you can use the ENABLE_EPS_PDF_IMPORT_AUTOROTATE_COUNTERCLOCKWISE engine switch together with theENABLE_EPS_PDF_IMPORT_AUTOROTATE engine switch to have Exstream Design and Production automatically rotate EPS and PDF images that are imported into a placeholder object or an empty image object when needed to better fit the dimensions of the placeholder object or empty image object.

The ENABLE_EPS_PDF_IMPORT_AUTOROTATE switch applies only to placeholder objects or empty image objects in designs created or edited in Designer version 8.5.105 or later. PDF or EPS images that are imported into placeholder objects or empty image objects in designs created in versions earlier than 8.5.105 might automatically rotate without using the switch.

The ENABLE_EPS_PDF_IMPORT_AUTOROTATE_COUNTERCLOCKWISE switch was renamed from the PDFTOPDF_LANDSCAPE_FIT_COUNTERCLOCKWISE engine switch, which is available in certain versions of Exstream Design and Production prior to version 8.6.103.

Syntax

```
-ENABLE_EPS_PDF_IMPORT_AUTOROTATE_COUNTERCLOCKWISE
```

This switch does not have any arguments.

Supported in

8.6.103 and later

A.3.11 ENABLE_PDFEPSIMPORT_AUTOROTATE

Use the ENABLE_PDFEPSIMPORT_AUTOROTATE engine switch to allow Exstream Design and Production to automatically rotate PDF or EPS images that are imported into a placeholder object or empty image object at run time by 90 degrees when needed to better fit the dimensions of the placeholder object or empty image object.

For example, if the horizontal dimension of an empty image object is greater than the vertical dimension, and the reverse is true for the imported image, then the image rotates to better fit the empty image object. By default, the image rotates clockwise when needed. To automatically

rotate the image counterclockwise, you can also use thePDFTOPDF_LANDSCAPE_FIT_COUNTERCLOCKWISE engine switch.

The ENABLE_PDFEPSIMPORT_AUTOROTATE engine switch applies only to placeholder objects or empty image objects in designs created or edited in Designer version 8.5.105 or later. PDF or EPS images that are imported into placeholder objects or empty image objects in designs created in versions earlier than 8.5.105 might automatically rotate without using the switch. The ENABLE_PDFEPSIMPORT_AUTOROTATE engine switch is ignored when the rotation of the placeholder object or empty image object is set to a value other than 0.

The ENABLE_PDFEPSIMPORT_AUTOROTATE switch is deprecated in the latest version of Exstream Design and Production. In Exstream Design and Productionversion 8.6.103 or later, use theENABLE_EPS_PDF_IMPORT_AUTOROTATE engine switch instead.

Syntax

```
-ENABLE_PDFEPSIMPORT_AUTOROTATE
```

This switch does not have any arguments.

Supported in

8.6.101 and 8.6.102

A.3.12 ENABLE_SHELL_COMMANDS

Use the ENABLE_SHELL_COMMANDS switch to enable the direct execution of command line scripts from functions, rules, or variables within Exstream Design and Production.

Important: By default, shell commands are disabled during engine runs to mitigate potential security risks. Shell commands can be enabled by including this switch in a control file or from the command prompt. Review all command line scripts prior to using this switch.

Syntax

```
-ENABLE_SHELL_COMMANDS
```

This switch does not have any arguments.

Supported in

16.2.0 and later

A.3.13 ENCODING_VECTOR_OVERRIDES

Use the ENCODING_VECTOR_OVERRIDES switch to override the default PDF encoding vector with the specified name at the specified code.

Syntax

```
-ENCODING_VECTOR_OVERRIDES=<code>,<name>
```

Arguments	Required	Supported values
`<code>`	Yes	Number of the encoding between 32 and 256
`<name>`	Yes	Encoding vector name

A.3.14 END

Use the END switch to specify the number of the last customer to process. The production environment does not allow the END switch to truncate a bundle, so the switch is ignored when sorting and bundling.

Syntax

```
-END=<customer number>
```

The argument for this switch is the customer number. This switch has no default value.

Example

```
-END=999999
```

A.3.15 ENDDATE

Use this packaging switch to specify the date as of which a package file is effective, or to specify the end date of the effective date range.

If you use this switch to specify the end date of a range, then you must also use the STARTDATE packaging switch to specify the start date of the range.

This packaging switch is required only if you are using theEFFECTIVE switch, and if you have set the argument for that switch to `ASOF` or `RANGE`.

Syntax

```
-ENDDATE=<date>
```

The only argument for this switch is the end date, in `mm/dd/yyyy` format.

Example 1

```
-EFFECTIVE=ASOF
-ENDDATE=12/31/2018
```

Example 2

```
-EFFECTIVE=RANGE
-STARTDATE=12/31/2018
-ENDDATE=12/31/2020
```

Related switches

EFFECTIVE

STARTDATE

A.3.16 ERRORLOG

The ERRORLOG switch creates a log that contains all of the errors and messages that normally appear in interactive mode. If you enter an existing file name, the file is overwritten.

Important: If you use the ERRORLOG switch, you must specify it before all other switches.

Syntax

```
-ERRORLOG=<logFile>
```

Argument	Required	Supported values
`logFile`	Yes	The file name of the error log file

Example

```
-ERRORLOG=SAMPLE
-AUTOUPDATE=DESIGN,en-us
-QUERYPATH=C:\Temp
-DSN=MyDSN,,MyUserName,MyPassword
```

A.3.17 EXPOIWRAPPER_CLASSPATH

Use the EXPOIWRAPPER_CLASSPATH engine switch to specify the location and name of the Exstream Design and Production JAR file, `expoiwrapper.jar`. When generating DOCX output, the engine uses the JAR file to parse the DOCX file.

By default, the engine expects the JAR in the same directory as the engine. If the JAR is in another directory, or if the JAR has a different name, you must specify the location of the JAR, the name of the JAR, or both using the EXPOIWRAPPER_CLASSPATH engine switch.

Syntax

```
-EXPOIWRAPPER_CLASSPATH=<path to the expoiwrapper.jar file>
```

The only argument for this switch is the path to the `expoiwrapper.jar` file.

Example

```
-EXPOIWRAPPER_CLASSPATH=C:\Exstream\expoiwrapper.jar
```

A.3.18 EXSTREAMPASSWORD

Used with: DBAdmin, Designer, Design Manager, EAM, Logic Designer, Packager

Use the EXSTREAPASSWORD switch to specify the password for an Exstream user when signing in to a design database.

You must use this switch in conjunction with the EXSTREAMUSER switch.

The EXSTREAMUSER switch and the EXSTREAMPASSWORD switch are required to run database maintenance from the command line. You must also use this switch when you use a command prompt to perform an action that requires you to sign in to a design database.

Syntax

```
-EXSTREAMPASSWORD=<password>
```

The argument for this switch is the password for the user name specified in the EXSTREAMUSER switch.

Example

```
-EXSTREAMUSER=admin
-EXSTREAMPASSWORD=xxx
```

A.3.19 EXSTREAMUSER

Used with: DBAdmin, Designer, Design Manager, EAM, Logic Designer, Packager

This switch lets you specify the Exstream user that you want to use to sign in to the design database.

When you use the EXSTREAMUSER switch, you must also use the EXSTREAMPASSWORD switch to specify the password for the Exstream user.

The EXSTREAMUSER switch and the EXSTREAMPASSWORD switch are required to run database maintenance from the command line. You must also use this switch when you use a command prompt to perform an action that requires you to sign in to a design database.

Syntax

```
-EXSTREAMUSER=<userName>
```

The argument for this switch is the user name of the Exstream user that you want to use to sign in to the design database.

Example

```
-EXSTREAMUSER=admin
-EXSTREAMPASSWORD=xxx
```

A.3.20 FILEMAP

Use the FILEMAP switch to change a symbolic file name specified in the data file properties to a valid path and file name at engine run time.

Syntax

```
-FILEMAP=<symbolic file name>,<file path>
```

Arguments	Required	Supported values
`<symbolic file name>`	Yes	The symbolic file name
`<file path>`	Yes	The fully-qualified file path and name

Example

```
-FILEMAP=DATASOURCE,DD:MYDATA
```

```
-FILEMAP=DATASOURCE,C:\Data Files\MyData.dat
```

A.3.21 FILEMAP (LiveEditor)

Use the FILEMAP switch to specify a different location for a reference file, initialization file, or report file that is available for use in a DLF file. For example, if you have an updated version of a reference file, you can use the FILEMAP switch to direct LiveEditor to use the updated file. Before using the FILEMAP switch on a data file, however, you must make the file available in the Live document by adding it as an authorized data file on the Live setting object.

When using the FILEMAP switch, keep in mind the following behaviors:

- The FILEMAP switch cannot be used to change the location of driver files.
- Any updates that result from applying the FILEMAP switch to an initialization file will be visible only in the Debugger, not in the Live document itself. The updated data does not appear in the Live document because the content from the initialization file is already composed prior to the FILEMAP switch being applied.
- Similarly, if you have selected the **Initial** open check box in the IO times area on the **Live** tab of the properties of a reference file or a report file, any changes that result from applying the FILEMAP switch will not appear in the Live document the first time it is opened after the switch is applied. The updated data does not appear initially because the content from the reference file or report file is already composed prior to the FILEMAP switch being applied. After the Live document is saved and reopened, however, the updated data will appear in the Live document.

Syntax

```
-FILEMAP=<DataFileName>,<NewFileLocation>
```

Use the following arguments with the FILEMAP switch:

Argument Name	Required	Supported values
`<DataFileName>`	Yes	The name of the data file
`<NewFileLocation>`	Yes	The new location of the data file

Example

```
-FILEMAP=MyData.dat,C:\Program Files\Exstream\MyData.dat
```

A.3.22 FLUSH_TIFF_FOR_PS

Use the FLUSH_TIFF_FOR_PS switch to remove extra data at the end of some TIFF G4 images when sent to PostScript as output, so the output will print or distill.

Syntax

```
-FLUSH_TIFF_FOR_PS
```

This switch does not have any arguments.

A.3.23 FONTDIRECTORIES

Important: This switch cannot be used with the Exstream z/OS engine.

Use the FONTDIRECTORIES engine switch to specify the directory path of TrueType fonts that are dynamically imported into a design at run time. At run time, if the application uses a font that is not included on the **Font Resource** tab of the application properties in Design Manager, then the engine searches for the font using the paths that are listed in the FONTDIRECTORIES switch. There is no limitation on the number of paths that you can add to this switch. This switch does not search sub-directories.

The engine uses the closest font available from either the packaged fonts or the fonts that are located in the specified paths. If an acceptable alternative is not found, then the engine uses the Times New Roman font.

Note: This switch cannot be used to search for TrueType Symbol fonts, vertical fonts, end-user-defined characters (EUDCs), or with applications that use IJPDS print stream optimization. Subscript and superscript formatting is also not supported.

This switch supports the following output types:

- AFP *
- DLF
- DOCX
- Empower
- IJPDS *
- PDF
- PostScript
- TIFF
- ZPL

* The following resource management limitations apply to AFP and IJPDS output:

- **SBCS applications**—The **Resource inclusion** setting cannot be set to `Default`.
- **DBCS applications**—A **Resource inclusion** setting of `Default` cannot be used in conjunction with a **Font type** setting of `Bitmap`.

Resource management settings are located on the **Resource Management** tab of the output object properties in Design Manager.

Syntax

```
-FONTDIRECTORIES=<filepath1>,<filepath2>,...,<filepathN>
```

The argument for this switch is the file path. This switch has no default value.

Example

```
-FONTDIRECTORIES=C:\Windows\Fonts\,C:\myfonts\
```

```
-FONTDIRECTORIES=/usr/Johns/fonts/
```

Supported in

16.4.0 and later

A.3.24 FOOTNOTE_MARGIN

Use the FOOTNOTE_MARGIN switch to specify the amount of space, in inches, the engine places between footnotes.

Syntax

```
-FOOTNOTE_MARGIN=<footnote_margin>
```

Arguments	Required	Supported values
`<footnote_margin>`	Yes	A numerical value that defines the space between footnotes. The default value is 0.1 inches.

Example

```
-FOOTNOTE_MARGIN=0.75
```

A.3.25 FORCE_LABEL_COLLISION_ AVOIDANCE

Use the FORCE_LABEL_COLLISION_AVOIDANCE engine switch to prevent labels in traditional charts from overlapping. You can use the FORCE_LABEL_COLLISION_ AVOIDANCE engine switch with the following traditional chart types:

- Line charts
- Area charts
- Label charts

Syntax

```
-FORCE_LABEL_COLLISION_AVOIDANCE
```

This switch does not have any arguments.

Supported in

7.0.401 and later

A.3.26 FORCE_MASTER_TEXTDRAW_MODE

For backward compatibility after you upgrade from 7.0 to 8.6, use the FORCE_MASTER_ TEXTDRAW_MODE engine switch to force the engine to compose all of the sections and paragraphs in an application at once. When the engine composes all of the sections and paragraphs in an application at once, the flow, numbering, and widow and orphan control settings are implemented as if the text of the sections and paragraphs were designed in a single text box.

If you use the FORCE_MASTER_TEXTDRAW_MODE engine switch, you may experience a reduction in engine performance, especially if the content spans multiple pages.

Syntax

```
-FORCE_MASTER_TEXTDRAW_MODE
```

This switch does not have any arguments.

Supported in

6.0.014 and later

A.3.27 FORCE_MIXED_CASE_PANTONE_NAMES

Use the FORCE_MIXED_CASE_PANTONE_NAMES switch to specify Hexachrome® colors as mixed case if your output device does not recognize uppercase PANTONE® Color names.

The default is to uppercase HEXACHROME in PANTONE® Color names.

This switch does not have any arguments.

Syntax

```
-FORCE_MIXED_CASE_PANTONE_NAMES
```

Version

8.0 and later

A.3.28 FORCE_NO_DOCX_TEXT_FRAMES

To replace all of the frames with text boxes in DOCX output, use the FORCE_NO_DOCX_TEXT_FRAMES engine switch when you produce DOCX output from an application. Text boxes in DOCX files are compatible with Microsoft Word 2003 and later.

Syntax

```
-FORCE_NO_DOCX_TEXT_FRAMES
```

This switch does not have any arguments.

Supported in

8.0.324 and later

A.3.29 FORCE_OLD_CONTENT_XML_NATIVE_FORMAT

For backward compatibility after you upgrade to Exstream Design and Production version 7.0 and later, use the FORCE_OLD_CONTENT_XML_NATIVE_FORMAT engine switch when you produce output from an application and you want the engine to process XML (content) in native format as opposed to the default UTF-8 format. The FORCE_OLD_CONTENT_XML_NATIVE_FORMAT engine switch forces the engine to bypass UTF-8 conversion.

Syntax

```
-FORCE_OLD_CONTENT_XML_NATIVE_FORMAT
```

This switch does not have any arguments.

Supported in

7.0.610 and later

A.3.30 FORCE_OLD_PDF_FONTBOX

When you upgrade to Exstream Design and Production 8.0 or later, and you produce PDF content from an application that uses the redaction feature in Adobe Acrobat, the blacked-out area in the PDF output might appear wider than the text you selected for redaction. For backward compatibility after you upgrade to Exstream Design and Production 8.0 or later, use the FORCE_OLD_PDF_FONTBOX engine switch.

This engine switch will be automatically included when the ENABLE_BACKWARD_COMPAT engine switch is used and the version specified is 8.0.335 or earlier.

> **Note:** This engine switch might reintroduce errors in the font metrics of the PDF output that were fixed in Exstream Design and Production version 8.0.

Syntax

```
-FORCE_OLD_PDF_FONTBOX
```

This switch does not have any arguments.

Supported in

- 8.0.339 and later maintenance releases
- 8.6.110 and later maintenance releases
- 9.0.105 and later maintenance releases
- 9.5.101 and later versions

A.3.31 FORCE_ONDEMAND_SHUTDOWN_ON_INPUT_ERROR

For backward compatibility after you upgrade to version 7.0 or later, use the FORCE_ONDEMAND_SHUTDOWN_ON_INPUT_ERROR engine switch when you produce on-

demand (real-time) output from an application and you want the engine to stop processing in the event that the engine encounters severe errors from customer input data.

Note: The FORCE_REALTIME_SHUTDOWN_ON_INPUT_ERROR engine switch is an alternate version of the FORCE_ONDEMAND_SHUTDOWN_ON_INPUT_ERROR engine switch. You can use either switch in your control file. Both switches perform the same function.

Syntax

```
-FORCE_ONDEMAND_SHUTDOWN_ON_INPUT_ERROR
```

This switch does not have any arguments.

Supported in

6.0.999 and later

A.3.32 FORCE_POST_PROCESSOR_BLOCK_WRITES

Use the FORCE_POST_PROCESSOR_BLOCK_WRITES switch to specify that the DDA postprocessor receive data in 1024 byte chunks, instead of a whole file, when used in combination with an output processor.

Syntax

```
-FORCE_POST_PROCESSOR_BLOCK_WRITES
```

This switch does not have any arguments.

A.3.33 FORCE_POSTSCRIPT_CUSTOMER_JOG_RESET

Use the FORCE_POSTSCRIPT_CUSTOMER_JOG_RESET engine switch when you produce PostScript output from an application that includes multiple-up (MUP) pages and you want the engine to force customer-level jogging resets to PostScript. The FORCE_POSTSCRIPT_CUSTOMER_JOG_RESET engine switch resets customer-level jogging to resolve issues with MUP jogging in the PostScript output queue.

Syntax

```
-FORCE_POSTSCRIPT_CUSTOMER_JOG_RESET
```

This switch does not have any arguments.

Supported in

6.0.005 and later

A.3.34 FORCE_READ_ALL_DFI_FROM_IMPORTDIR

Use the FORCE_READ_ALL_DFI_FROM_IMPORTDIR switch to ignore path information on placeholder variables and instead use the import directory to find resources.

> You must use the IMPORTDIRECTORY switch with the FORCE_READ_ALL_DFI_FROM_IMPORTDIR switch.

The FORCE_READ_ALL_DFI_FROM_IMPORTDIR switch helps you avoid the time-consuming practice of hard-coding file paths into your placeholder variables, especially if your resources change often or must be moved to a different location. If the path information is specified in the initialization file or using the VARSET switch, this switch is unnecessary.

Syntax

```
-FORCE_READ_ALL_DFI_FROM_IMPORTDIR
```

This switch does not have any arguments.

Example

```
-FORCE_READ_ALL_DFI_FROM_IMPORTDIR
-IMPORTDIRECTORY=C:\Imports\DynamicImages
```

A.3.35 FORCE_REALTIME_SHUTDOWN_ON_INPUT_ERROR

For backward compatibility after you upgrade to version 7.0 and later, use the FORCE_REALTIME_SHUTDOWN_ON_INPUT_ERROR engine switch when you produce real-time (on-demand) output from an application and you want the engine to stop processing in the event that the engine encounters severe errors from customer input data.

> **Note:** The FORCE_REALTIME_SHUTDOWN_ON_INPUT_ERROR engine switch is an alternate version of the FORCE_ONDEMAND_SHUTDOWN_ON_INPUT_ERROR engine switch.You can use either switch in your control file. Both switches perform the same function.

Syntax

```
-FORCE_REALTIME_SHUTDOWN_ON_INPUT_ERROR
```

This switch does not have any arguments.

Supported in

6.2 and later

A.3.36 FORCE_RESET_TIMING_OF_DLF_IMPORT_SECTION_VARS

Use the FORCE_RESET_TIMING_OF_DLF_IMPORT_SECTION_VARS engine switch to override the variable substitution time that is specified for variables that determine the inclusion of a section-based document and force the substitution of variables during the first and last engine runs and during editing in LiveEditor. Using the FORCE_RESET_TIMING_OF_DLF_IMPORT_SECTION_VARS switch is equivalent to selecting a substitution time of Initial and Final engine and LiveEditor (Exstream Design and Production version 8.6 or earlier) or Initial Engine, Interactive Editor, and Final Engine (Exstream Design and Production version 9.0 and later).

Syntax

```
-FORCE_RESET_TIMING_OF_DLF_IMPORT_SECTION_VARS=option
```

Select one of the following options:

Argument	Required	Supported values
TRUE	No	Select TRUE if you want to override the variable substitution time.
FALSE	No	Select FALSE if you do not want to override the variable substitution time.
NEVER	No	Select NEVER if you want to disable all variable substitution time overrides and use the variable's original substitution time instead.

Supported in

- 8.0.337 and later maintenance releases
- 8.6.110 and later maintenance releases
- 9.0.105 and later maintenance releases
- 9.5.101 and later maintenance releases

A.3.37 FORCE_TIFF_NEWSUBFILETYPE_FIELD

Use the FORCE_TIFF_NEWSUBFILETYPE_FIELD switch to create multiple-page TIFF output using the **NewSubFileType** field, recognized by newer TIFF archiving systems, instead of the outdated **SubFileType** field.

Syntax

```
-FORCE_TIFF_NEWSUBFILETYPE_FIELD
```

This switch does not have any arguments.

A.3.38 FORCE_TOP_ALIGN_FOR_SPLIT_ ROWS

Use the FORCE_TOP_ALIGN_FOR_SPLIT_ROWS engine switch when you produce output from an application to ensure that vertically centered text within a table is placed correctly in the output when the row splits. If you have text centered vertically within a row within a table, when the row splits, the text appears below the table and below the deleteID of the page instead of being placed as expected in the design.

Syntax

```
-FORCE_TOP_ALIGN_FOR_SPLIT_ROWS
```

This switch does not have any arguments.

Supported in

- 8.0.348 and later maintenance releases
- 8.6.117 and later maintenance releases
- 9.0.112 and later maintenance releases
- 9.5.103 and later maintenance releases
- 9.5.201 and later versions

A.3.39 FORCE_ZERO_HEADER_VAR_INDEX

Use the FORCE_ZERO_HEADER_VAR_INDEX switch to force the index of rows to count by subsection rather than section.

For example, if you had a table section that contained ten array variables with ten entries each, by default the table rows will count from zero to 99. When you use the FORCE_ZERO_ HEADER_VAR_INDEX switch, the table rows count the values zero to nine for each of the ten array variables.

Syntax

```
-FORCE_ZERO_HEADER_VAR_INDEX
```

This switch does not have any arguments.

A.3.40 FORMAT_TAGGED_TEXT

Use the FORMAT_TAGGED_TEXT engine switch when you produce output from an application that contains a tagged text variable, and an option from the **Special Formatting** list is selected. In Designer, if you select an option from the **Special Formatting** list, the properties of the tagged text variables do not change the format of the variable when you produce output.

You must specify the special formatting for an instance of the variable in Designer; the **Output Format** property is not available for a tagged text variable in Design Manager.

Syntax

```
-FORMAT_TAGGED_TEXT
```

This switch does not have any arguments.

Supported in

- 8.0.339 and later maintenance releases
- 8.6.110 and later maintenance releases
- 9.0.105 and later maintenance releases
- 9.5.101 and later versions

A.3.41 FULL_NAME_AFP_BMO_EMO

Use the FULL_NAME_AFP_BMO_EMO switch to use the actual overlay name in the Begin Map Overlay/End Map Overlay (BMO/EMO) fields.

Syntax

```
-FULL_NAME_AFP_BMO_EMO
```

This switch does not have any arguments.

Supported in

4.5.301 and later

A.3.42 GS_PAUSE

Use the GS_PAUSE switch to eliminate Ghostscript debugging issues with the engine when you use the PDF_PASSTHROUGH_AS_LZW switch in your control file. Ghostscript encounters memory issues when a user imports PDF files as images to AFP output using the PDF_PASSTHROUGH_AS_LZW switch.

Requirements

The GS_PAUSE switch must be used only in conjunction with the PDF_PASSTHROUGH_AS_LZW switch.

Syntax

```
-GS_PAUSE
```

This switch does not have any arguments.

A.3.43 GSTEMPDIR

Summary

Use the GSTEMPDIR switch to specify the directory in which Ghostscript temporarily stores its output before the engine reads it into a buffer.

Important: All other temporary files that Ghostscript creates are not redirected by this switch. By default, these files are written to `/tmp`. To change the location of these files, you must set the `TEMP` or `TEMPDIR` environment variable. For more information, see http://ghostscript.com/doc/current/Use.htm#Temp_files.

Syntax

```
-GSTEMPDIR=<directory path>
```

Arguments

The argument for this switch is the temp directory for Ghostscript output. There is no default value.

Example

```
-GSTEMPDIR=C:\PostScriptFiles\
```

A.3.44 HIDE_STYLE_CLASS_IN_HTML_OUTPUT

If you produced HTML or HTML (email) output from designs that used styles and style sheets to format text in Exstream Design and Production versions earlier than 16.3.0, the resulting `class` attributes were automatically generated and did not include the style name.

In versions 16.3.0 and later, the default behavior changed to use the style names as part of the `class` attribute values to allow for easier implementation of custom cascading style sheets. For example, if you create a paragraph style named "Heading," then `Heading` is included in the `class` attribute value.

For backward compatibility after you upgrade to 16.3.0, use the HIDE_STYLE_CLASS_IN_HTML_OUTPUT switch to exclude `class` attributes for styles added in Designer. This switch is automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 16.2.999 or earlier.

Syntax

```
-HIDE_STYLE_CLASS_IN_HTML_OUTPUT
```

This switch does not have any arguments.

Supported in

16.3.0 and later

A.3.45 HONOR_DASH_DOT_LINE_STYLES

Use the HONOR_DASH_DOT_LINE_STYLES switch when you produce PDF output from an application that includes a dash-dot alternated line format. By default, Exstream Design and Production converts all dash-dot alternated lines to dashed lines in PDF output.

Syntax

```
-HONOR_DASH_DOT_LINE_STYLES
```

This switch does not have any arguments.

Supported in

- 8.0.335 and later maintenance releases
- 8.6.108 and later maintenance releases
- 9.0.102 and later maintenance releases
- 9.5.101 and later versions

A.3.46 HONOR_FULL_JUSTIFICATION_WITH_LINE_FEED

Use the HONOR_FULL_JUSTIFICATION_WITH_LINE_FEED switch to ensure that the engine applies full justification to any text that is located within a text box and when the user enters a soft line break.

Syntax

```
-HONOR_FULL_JUSTIFICATION_WITH_LINE_FEED
```

This switch does not have any arguments.

Supported in

7.0.624 and later

A.3.47 HONOR_LIVE_SUBST_TIMING_IN_FULFILL

Use the HONOR_LIVE_SUBST_TIMING_IN_FULFILL engine switch when you use a Live document in a fulfillment process that contains a variable in a section-driven paragraph. Using this engine switch will ensure that the production engine honors the substitution times specified for variables that appear in section-driven paragraphs.

Additionally, if you use the RUN_SECTION_FORMULAS_IN_DLF_FULLFILLMENT engine switch, variables that appear in section-driven paragraphs may not reset as expected. The HONOR_LIVE_SUBST_TIMING_IN_FULFILL engine switch will correctly reset the variables when the RUN_SECTION_FORMULAS_IN_DLF_FULLFILLMENT engine switch is used.

Syntax

```
-HONOR_LIVE_SUBST_TIMING_IN_FULFILL
```

This switch does not have any arguments.

Supported in

- 8.0.339 and later maintenance releases
- 8.6.110 and later maintenance releases
- 9.0.105 and later maintenance releases
- 9.5.101 and later versions

A.3.48 HP_INDIGO_PPML_DUPLEX

Use the HP_INDIGO_PPML_DUPLEX switch to print duplex PPML documents on HP Indigo Digital Press printers. You can use this switch only when all the paper types used in the application are the same size, including the pages included in multiple-ups and banner pages.

Syntax

```
-HP_INDIGO_PPML_DUPLEX
```

This switch does not have any arguments.

A.3.49 HTML_DTD_LOCATION

Use the HTML_DTD_LOCATION switch to specify the file path or the file name of the XHTML DTD. You must have the XHTML DTD to transform HTML output.

Requirements

The following files must be available when transforming HTML output:

Default file name	Contents of file
`xhtml1-transitional.dtd`	Extensible HTML version 1.0 Transitional DTD
`xhtml-lat1.ent`	Latin-1 character entity set (referenced by the DTD)
`xhtml-special.ent`	Special character entity set (referenced by the DTD)
`xhtml-symbol.ent`	Symbol character entity set (referenced by the DTD)

These files are available from the W3C Web site.

Syntax

```
-HTML_DTD_LOCATION=<PathOrFileName>
```

If you specify only a file path, then the engine references the specified directory to locate the DTD file, using the file name xhtml1-transitional.dtd. If you specify a file name, then the engine references that file name for the XHTML DTD. If you do not use this switch, then you must include the DTD file in the same directory as the engine.

The location and file names of the required character entity files are specified within the DTD. When you are using the Extensible HTML version 1.0 Transitional DTD, the character entity files listed in the table must be in the same folder as the DTD, using the default file names. If you specify a different DTD file, you must include any character entity files referenced by that DTD in the location specified by the DTD.

Example

```
-HTML_DTD_LOCATION=C:\Path\xhtml.dtd
```

A.3.50 HTML_FORCE_DESIGN_IMAGE_NAMING

Use the HTML_FORCE_DESIGN_IMAGE_NAMING switch to maintain control over static image names in your HTML output. The HTML_FORCE_DESIGN_IMAGE_NAMING switch forces the engine to use image design names to create and reference HTML static images.

When you use the HTML_FORCE_DESIGN_IMAGE_NAMING switch, make sure you assign unique names to all the images in your design database. If you have more than one static image with the same design name in the application, the engine overwrites previously-created static image resources. If you do not assign unique design names to your images, the engine uses the default naming convention to name static image resources (for example, `HTMLImage1.jpg`, `HTMLImage2.jpg`, and so on).

Syntax

```
-HTML_FORCE_DESIGN_IMAGE_NAMING
```

This switch does not have any arguments.

A.3.51 HTML_IGNORE_UNSET_FONT_COLOR

After you upgrade from Exstream Design and Production version 16.3.1 to a later version, if a font color has not been specified for the text in your design, then in HTML and HTML (email) output, that text will be set to the default color `#000000` (black).

For backward compatibility after you upgrade to 16.3.1, use the HTML_IGNORE_UNSET_FONT_COLOR switch to revert to the original behavior. This switch is automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 16.3.0 or earlier.

Syntax

```
-HTML_IGNORE_UNSET_FONT_COLOR
```

This switch does not have any arguments.

Supported in

16.3.1 and later

A.3.52 HTML_IMPORT_IGNORE_16_4_UPDATES

Exstream Design and Production version 16.4.0 introduced several new options for HTML import to help support customers who import content from OpenText xPression projects into Design and Production applications at run time. The new options include:

- Support for importing Microsoft Word HTML files that use CSS `page-break-before` and `page-break-after` properties to control page breaks in print output.
- Support for importing HTML content that uses the `rowspan` attribute. Because Design and Production does not natively support the `rowspan` attribute, when the Exstream engine encounters spanned rows in imported content, it adds enough empty cloned cells to mimic the number of spanned rows.
- Support for importing content that uses the CSS `text-indent` property.
- Support for importing content that contains nested embeds (specifically, nested tables inside of tables).

For backward compatibility after you upgrade to version 16.4.0 or later, use the HTML_IMPORT_IGNORE_16_4_UPDATES switch. This switch will be automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 163999 or earlier.

Syntax

```
-HTML_IMPORT_IGNORE_16_4_UPDATES
```

This switch does not have any arguments.

Supported in

16.4.0 and later

A.3.53 HTML_IMPORT_IMAGE_PADDING_BEFORE_BORDER

In Exstream Design and Production versions earlier than 9.5.303, image borders in imported HTML content were applied using default HTML behavior. That is, the border was drawn outside of the image edges.

In Exstream Design and Production version 9.5.303, the default behavior was changed so that border settings in imported HTML content are applied using the default behavior for borders in Designer. That is, the engine draws a custom border that overlaps the image edges, which can affect the appearance of any applied padding.

For example, if your imported HTML content contains the following image:

```
<img src="image.jpg" style="border: 5px black; padding: 7px;">
```

With the switch, the image appears in the output with a 7-pixel space between the image and the border on all four sides. Without the switch, the padding is honored differently. Instead of the padding being applied between the image and the border, it is applied between the image and other objects. This causes the 5-pixel border to overlap the edges of the image.

For backward compatibility after you upgrade to 9.5.303 or later, use the HTML_IMPORT_IMAGE_PADDING_BEFORE_BORDER switch. This switch will be automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 9.5.302 or earlier.

> If you use the HTML_IMPORT_IMAGE_PADDING_BEFORE_BORDER switch, the imported HTML content will not be supported in Live or Empower output.

Syntax

```
-HTML_IMPORT_IMAGE_PADDING_BEFORE_BORDER
```

This switch does not have any arguments.

Supported in

9.5.303 and later

A.3.54 HTML_KEEP_EMPTY_TRAILING_PARAS_31274

Use the HTML_KEEP_EMPTY_TRAILING_PARAS_31274 engine switch when you produce HTML output from an application so that the engine produces a < br> HTML tag for each instance of an empty trailing paragraph that follows a non-empty paragraph.

Syntax

```
HTML_KEEP_EMPTY_TRAILING_PARAS_31274
```

This switch does not have any arguments.

A.3.55 HTML_USE_COMPLETE_DOCTYPE

Use the HTML_USE_COMPLETE_DOCTYPE engine switch to improve the rendering of HTML 4.01 output in Internet browsers when running an application that imports DOCX content (DOCX pass-through) and produces HTML output. This switch outputs HTML with a DOCTYPE Declaration reference that prevents the truncation of text at the bottom of table cells and text boxes.

Syntax

```
-HTML_USE_COMPLETE_DOCTYPE
```

This switch does not have any arguments.

Supported in

8.0.321 and later

A.3.56 HTML_USE_OLD_TABLE_WIDTH_CALC

In Exstream Design and Production versions earlier than version 9.5.301, table cell widths and alignment in HTML output were based on the Internet Explorer "quirks mode" box model layout. When viewing HTML 4 or HTML 5 output produced from an earlier version of Exstream Design and Production in Internet Explorer (version 9 or later) or other browsers, tables might appear misaligned.

This issue was corrected in Exstream Design and Production version 9.5.301.

For backward compatibility after you upgrade to 9.5.301, use the HTML_USE_OLD_TABLE_WIDTH_CALC switch. This switch will be automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 9.5.299 or earlier.

Syntax

```
-HTML_USE_OLD_TABLE_WIDTH_CALC
```

This switch does not have any arguments.

Supported in

9.5.301 and later

A.4 Switches I-L

A.4.1 IGNORE_HTML_TABLE_CELL_FRAME_SEGMENTS

In Exstream Design and Production versions earlier than version 9.5.301, if your design included table cells with different borders defined for each side of the cell, these were not honored in HTML output. Instead, the border properties applied to the right side of the cell were applied on all sides of the cell. For example, if you applied a blue border that was 0.5" on the left side of the cell, but a black border that was 0.25" on the right side of the cell, the 0.25" black border would be applied to all sides of the cell.

In version 9.5.301, this issue was corrected.

For backward compatibility after you upgrade to 9.5.301, use the IGNORE_HTML_TABLE_CELL_FRAME_SEGMENTS switch. This switch will be automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 9.5.299 or earlier.

Syntax

```
-IGNORE_HTML_TABLE_CELL_FRAME_SEGMENTS
```

This switch does not have any arguments.

Supported in

9.5.301 and later

A.4.2 IGNORE_LOC_SEARCH_KEY_LEN

Use the IGNORE_LOC_SEARCH_KEY_LEN switch to ignore the AFP location search key **Length** setting specified in the variable properties in Designer. Without this key, the production environment truncates the variable value in the output according to the **Length** value specified in the search key properties.

Syntax

```
-IGNORE_LOC_SEARCH_KEY_LEN
```

This switch does not have any arguments.

Supported in

6.1.010 and later

A.4.3 IGNORE_PCM_FOR_MARKETING_PAGE_LIMITS

For backward compatibility after you upgrade from 6.1 to 7.0 and later, use the IGNORE_PCM_FOR_MARKETING_PAGE_LIMITS switch to prevent documents and applications from honoring the **Duplex page counting method** setting when counting extra pages that are added to a document for campaigns and messages, according to the **Maximum message-driven pages** setting. When you enable this switch, the engine counts only design pages, regardless of the specified **Duplex page counting method** setting.

Syntax

```
-IGNORE_PCM_FOR_MARKETING_PAGE_LIMITS
```

This switch does not have any arguments.

Supported in

6.1.033 and later

A.4.4 IGNORE_SPEC_ARRAY_ELEM_ON_XML_DATA_AREAS

For backward compatibility after you upgrade to 7.0 or later, use the IGNORE_SPEC_ARRAY_ELEM_ON_XML_DATA_AREAS engine switch when you produce output from an application that contains only one array element (instead of multiple array elements) in DLF output.

When you use this switch, the engine treats the XML element in the DLF layout file or in the XML output file as mapped to the entire array and not just to the specified array element. As an alternative to using this switch, you can clear the **Specific array element** check box on the **Data Area Properties** dialog box for each array variable that is mapped in your XML output data file.

Syntax

```
-IGNORE_SPEC_ARRAY_ELEM_ON_XML_DATA_AREAS
```

This switch does not have any arguments.

Supported in

7.0 and later

A.4.5 IJPDS_FORCE_PHR_RELATIVE_POSITION

For IJPDS output, use the IJPDS_FORCE_PHR_RELATIVE_POSITION switch if you use multiple printheads in the same RIP to create printhead relative offsets based on printhead sizes within the RIP.

Syntax

```
-IJPDS_FORCE_PHR_RELATIVE_POSITION
```

This switch does not have any arguments.

A.4.6 IMPORTDIRECTORY

Use the IMPORTDIRECTORY switch to specify the directory for files referenced by placeholder variables.

Syntax

```
-IMPORTDIRECTORY=pathname
```

The argument for this switch is the fully qualified directory. This switch has no default.

For z/OS, the argument is the fully-qualified PDS name that holds your dynamic import files.

If you use a DD statement, you must include the fully qualified PDS name in your JCL. With either method, the value of your placeholder variables must contain the import file name enclosed by parentheses.

Example

General:

```
-IMPORTDIRECTORY=C:\DynamicContentFiles\
```

z/OS:

```
-IMPORTDIRECTORY=HLEV1.HLEV2.IMAGELIB
```

```
-IMPORTDIRECTORY=DD:IMAGE
```

A.4.7 INCLUDE_HEIGHT_FOR_HTML5_TABLES

In Exstream Design and Production versions earlier than 16.3.0, when producing HTML5 output from a container design that included tables with borders in a grid cell, the table height was placed on the `<div>` tag that contained the table structure. Since browsers calculate table height

differently depending on borders and padding, the height specified in the tag was sometimes less than the table height in the design. This caused visual differences, in some cases clipping content from the output and in others changing the spacing between the table and the next element.

The default behavior was changed in Exstream Design and Production version 16.3.0 to better accommodate placing a maximum height on grid cells that contain a table with borders.

For backward compatibility after you upgrade to 16.3.0, use the INCLUDE_HEIGHT_FOR_HTML5_TABLES switch to place the table height on the tag that contains the table structure in HTML5 output from a container design. This switch will be automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 16.2.999 or earlier.

Syntax

```
-INCLUDE_HEIGHT_FOR_HTML5_TABLES
```

This switch does not have any arguments.

Supported in

16.3.0 and later

A.4.8 INCLUDE_JURISDICTION_EXPIRATION_DATE

Use the INCLUDE_JURISDICTION_EXPIRATION_DATE engine switch when you run an application that contains jurisdictions and you want the engine to include objects with an effective date that are equal to objects with an expiration date. By default, in an application that includes jurisdictions, the engine includes only objects prior to the expiration date. To include objects prior to (and on) the expiration date, you must include the INCLUDE_JURISDICTION_EXPIRATION_DATE engine switch in your control file.

Syntax

```
-INCLUDE_JURISDICTION_EXPIRATION_DATE
```

This switch does not have any arguments.

Supported in

8.0.317 and later

A.4.9 INCLUDE_TOP_FRAME_HEAD_SIZE_IN_FIRST_PAGE_CALC

Use the INCLUDE_TOP_FRAME_HEAD_SIZE_IN_FIRST_PAGE_CALC engine switch when you produce output from an application that contains a table with multiple headers to resolve an issue where the table splits between two pages when there appears to be enough room.

Syntax

```
-INCLUDE_TOP_FRAME_HEAD_SIZE_IN_FIRST_PAGE_CALC
```

This switch does not have any arguments.

Supported in

- 9.0.114 and later maintenance releases
- 9.5.202 and later maintenance releases
- 9.5.301 and later versions

A.4.10 INITVARSET

Use the INITVARSET switch to set customer or system variables to a specific value before the initialization or postsort initialization files are read. You can use this switch as many times as needed to set the value of multiple variables. The INITVARSET switch is ignored in LiveViewer.

> **Tip:** Using this switch is the same as using the VARSET switch with a `variableTiming` value of `PREINIT`.

Syntax

```
-INITVARSET=<variableName>,<variableValue>
```

Argument name	Required	Supported values
`variableName`	Yes	The name of the variable that you want to set
`variableValue`	Yes	The value of the variable

Examples

```
-INITVARSET=AccountNumber,00001
```

```
-INITVARSET=CompanyState,KY
```

A.4.11 INSTALL_DATABASES

The INSTALL_DATABASES command line switch is an optional switch for silent installation that lets you specify whether you want to install or overwrite the Exstream Sample and Tracking databases.

If you use the INSTALL_DATABASES command line switch during a command line database removal, then the installer ignores the INSTALL_DATABASES switch and any value that you specify for the switch.

Syntax

```
INSTALL_DATABASES=<option>
```

Argument	Required	Supported values
`<option>`	Yes	• `YES`—Specifies that you want to install the Exstream Sample and Tracking databases. If you have previously installed the Exstream Sample and Tracking databases, then specifying the `YES` value leaves the previously installed databases intact. • `NO`—Specifies that you do not want to install the Exstream Sample and Tracking databases. If you have previously installed the Exstream Sample and Tracking databases, then specifying the `NO` value leaves the databases intact. • `OVERWRITE`—Specifies that you want to install the Exstream Sample and Tracking databases and that you want the installer to overwrite previously installed Exstream Sample and Tracking databases if they exist.

Example

The following example uses InstallShield and Microsoft standard command line switches and the INSTALL_DATABASES command line switch to install the Exstream Sample and Tracking databases on the English language version of Exstream Design and Production 9.0.101 and to write the installation log file to an specified directory.

```
>Exstream_setup_9.0.101.exe /s /L1033 /v”/qn INSTALL_DATABASES=YES /l*v
C:\install_log.txt”
```

Supported in

9.0.101 and later

For more information about InstallShield standard command line switches, go to the Flexera web site. For more information about Microsoft standard command line switches, go to the Microsoft web site.

A.4.12 INSTALL_TYPE

The INSTALL_TYPE command line switch is an optional switch for silent installation that lets you specify a value for the type of installation that you want to use. By default, if you do not specify the INSTALL_TYPE command line switch, then the installer replaces the current version of Exstream Design and Production if one exists.

Syntax

```
INSTALL_TYPE=PRESERVE_VERSION
```

Argument	Value	Supported values
`<installation_type>`	Yes	• `PRESERVE_VERSION`—Preserves the current version of Exstream Design and Production and installs the newer version of Exstream Design and Production to a separate installation directory. • `REPLACE_VERSION`—Replaces the current version of Exstream Design and Production with the newer version.

Example

The following example uses Microsoft and InstallShield command line switches and the INSTALL_TYPE optional Exstream Design and Production command line switch to install the English language version of Exstream Design and Production 8.0.325, to keep currently installed versions of the software, and to write the installation log file to a specific directory.

```
>Exstream_setup_8.0.325.exe /s /L1033 /v"/qn INSTALL_TYPE=PRESERVE_VERSION /l
C:\install_log.txt"
```

Supported in

8.0.325 and later

For more information about InstallShield command line switches, go to the Flexera web site. For more information about Microsoft command line switches, go to the Microsoft web site.

A.4.13 INSTALLDIR

The INSTALLDIR command line switch is an optional switch for silent installation that lets you specify the installation directory you want to use.

Syntax

```
INSTALLDIR=directory file path
```

The value of this switch is the file path of the directory.

Example

The following example uses Microsoft and InstallShield command line switches and the INSTALLDIR optional command line switch to install the English language version of 8.0.325, to install the 8.0.325 version of the software to a specific directory, and to write the installation log file to a specific directory.

```
>Exstream_setup_8.0.325.exe /s /L1033 /v"/qn INSTALLDIR=\"C:\Program
Files\Exstream\Exstream 8.0.325\" /l C:\install_log.txt"
```

Supported in

8.0.325 and later

For more information about InstallShield standard command line switches, go to the Flexera web site. For more information about Microsoft standard command line switches, go to the Microsoft web site.

A.4.14 KEEP_SPLIT_WHITESPACE

Use the KEEP_SPLIT_WHITESPACE switch to keep blank lines in a paragraph when the paragraph splits across pages. Without this switch, the production environment removes blank lines from splitting text.

Syntax

```
-KEEP_SPLIT_WHITESPACE
```

This switch does not have any arguments.

A.4.15 KEEP_WRONG_DOCX_IMPORT_FONT_COLOR_HANDLING

In versions of Exstream Design and Production earlier than the versions listed below, some font, color, or style changes made to a section of text were either ignored, or were applied to the entire paragraph in DOCX content that was imported at run time. In newer versions of Design and Production, the default behavior has been changed to honor the formatting of sections of text in imported DOCX content.

For backward compatibility after you upgrade to any of the Exstream Design and Production versions listed below, use the KEEP_WRONG_DOCX_IMPORT_FONT_COLOR_HANDLING engine switch to revert to the original behavior.

Syntax

```
-KEEP_WRONG_DOCX_IMPORT_FONT_COLOR_HANDLING
```

This switch does not have any arguments.

Supported in

- 8.6.123 and later maintenance releases
- 9.0.120 and later maintenance releases
- 9.5.308 and later maintenance releases
- 16.2.4 and later maintenance releases
- 16.3.3 and later maintenance releases
- 16.4 Update 1 and later versions

A.4.16 KEY

Use the KEY switch to override the key in the package file. Use this switch to apply a specific key to an engine run, as in the case where you need an expired key to run package files with a prior date. This switch is required to run the Exstream Batch Compare Utility.

If you specify a floating license key, and if your floating license file is not present in the engine directory, then you must also use the LICENSE_PATH switch to specify the location of the floating license file.

Syntax

```
-KEY=<keyString>
```

The only argument for this switch is the text string of your key. The length of this string varies depending on the Design and Production version:

- 155 characters (with no spaces) in version 6.0 and above
- 50 or 155 characters (with no spaces) between version 5.0 and 6.0
- 50 characters (with no spaces) between version 3.5 and 5.0
- 25 characters in version 3.0

A.4.17 KEYFILE

Use the KEYFILE switch to specify a key from a Design and Production license key file (*.ekf) to override the key that is specified in the package file. Use this switch to apply a specific key to an engine run, as in the case where you need an expired key to run package files with a prior date.

If you specify a floating license key, and if your floating license file is not present in the engine directory, then you must also use the LICENSE_PATH switch to specify the location of the floating license file.

Syntax

```
-KEYFILE=<path>
```

The only argument for this switch is the fully-qualified path to the key file.

Example

```
-KEYFILE=C:\keys\MyKeyName.ekf
```

A.4.18 KEYPART

Use the KEYPART switch to split a key string into multiple parts. This switch is used primarily with z/OS for keys that are too large to fit on a single line of JCL. You use this switch multiple times in the same control file. When the engine processes the KEYPART switch, it combines the parts in the order supplied to form a single key string. This overrides the key in the package file. This switch is used to apply a specific key to an engine run, as in the case where you need an expired key to run package files with a prior date.

Syntax

```
-KEYPART=<partialStringOne>
-KEYPART=<partialStringTwo>
...
-KEYPART=<partialStringEnd>
```

The only argument for this switch is the partial key string. You can use as many instances of the KEYPART switch as needed to enter all the partial strings that make up the key string. This switch has no default value.

Supported in

5.0 and later

A.4.19 LANGUAGE

Used with: DBAdmin, Engine

Use the LANGUAGE switch to change the default language in the package file (when used with the engine) or to set the language for the user interface (when used with the Database Administration utility).

Syntax

```
-LANGUAGE=<language>
```

The argument for this switch depends on the application with which you are using it.

- If you are using the switch with the engine, the argument for this switch is the name of the language object as defined in the design database. The default value is the value specified on the **Workflow** tab in **System Settings**.
- If you are using the switch with the Database Administrator utility, you must use one of the following language codes:
  - `de-de` (German)
  - `en-us` (English)—This is the default.
  - `es-mx` (Spanish)
  - `fr-fr` (French)
  - `ja-jp` (Japanese)
  - `nl-nl` (Dutch)
  - `pt-br` (Portuguese Brazilian)
  - `zh-cn` (Simplified Chinese)

Example

```
-LANGUAGE=ENGLISH
```

```
-LANGUAGE=fr-fr
```

A.4.20 LICENSE_PATH

Use the LICENSE_PATH switch to specify the workstation license file location. If the switch is not specified, the engine looks in the current directory.

Syntax

```
-LICENSE_PATH=<path>
```

The only argument for this switch is the fully-qualifed path to the workstation license file.

Example

```
-LICENSE_PATH=C:\Users\Public\Documents\Exstream\Licenses\
```

A.4.21 LICENSE_WAIT

Use the LICENSE_WAIT switch to specify whether to exit, wait and continue checking for an available floating license indefinitely, or wait and continue checking for an available floating license for the specified time when all floating licenses are in use.

If this switch is not specified and a floating license is not available, then the program continues to check for an available license every 15 seconds indefinitely. Users must stop the program if they decide to discontinue waiting on an available license.

Syntax

```
–LICENSE_WAIT=<wait_option>,<wait_time>
```

Arguments	Required	Supported values
`<wait_option>`	Yes	• `EXIT` — If a floating license is not available, the program closes. • `WAIT`— If a floating license is not available, the program continues to check for an available license indefinitely based on the value of the `<wait_time>` argument. • `<SILENT_WAIT`— If a floating license is not available, the program continues to check for an available license indefinitely based on the value of the `<wait_time>` argument. If you specify `SILENT_WAIT`, no status message is printed to the console.
`<wait_time>`	No	If you specify `WAIT` or `SILENT_WAIT`, you can specify a whole number to select the number of minutes to wait before checking for an available license. If no value is specified, the default value is 15 seconds.

Example

```
-LICENSE_WAIT=WAIT,20
```

A.4.22 LIVE_HONOR_EXCLUSION_RULES

By default, any document or page is included in a DLF file regardless of engine usage rules or engine timing because it allows the inclusion of resources into the DLF that otherwise would not be available if the document or page was removed. This behavior results in a larger DLF file size for each document or page that is included. Use the LIVE_HONOR_EXCLUSION_RULES switch to exclude pages and documents from a DLF file that have been disabled by initial engine-only rules and to decrease the DLF file size, as pages and documents will not be included in the file.

Syntax

```
-LIVE_HONOR_EXCLUSION_RULES
```

This switch does not have any arguments.

Supported in

8.0.310 and later

A.4.23 LOADREFFILE

Use the LOADREFFILE switch to specify a reference file the engine preloads onto a mainframe system to reduce production time.

Syntax

```
-LOADREFFILE=<reference file>
```

Arguments	Required	Supported values
`<reference file>`	Yes	• The name of the reference file to preload. • `ALL`—Preload all reference files. This switch has no default value.

Example

```
-LOADREFFILE=DD:MYDATA
```

A.4.24 LOCALE

Use the LOCALE switch to change the default locale in the package file. In Design Manager, you can set the default locale, and define additional locales, on the **Workflow** tab in **System Settings**.

Syntax

```
-LOCALE=<localeName>
```

The only argument for this switch is the locale to be used. The locale name that you specify in the switch must match the name of a locale object that you have previously defined in Design Manager

Example

```
-LOCALE=EUROPE
```

A.4.25 LOGDDA

Use the LOGDDA switch to send error messages to a DDA.

Syntax

```
-LOGDDA=<ProgramType>,<Module>,<Function>,<BufferSize>,<OpenParams>
```

Argument Name	Required	Supported values
<ProgramType>	Yes	The language used to create the module. Select from the following options: • ASM • COBOL • DLL (for C and C++) • PL1
<Module>	Yes	The location of the DLL file
<Function>	Yes	The function name in the DLL/routine
<BufferSize>	No	The size of the buffer (in bytes), which must accommodate the size of the output
<OpenParams>	No	Parameters specific to the module

Example

```
-LOGDDA=MyDriverFile,DLL,C:\Path\CustomDDA.dll,MyFunction,5000,MyParams
```

A.4.26 LOGDDANOHEADER

Use the LOGDDANOHEADER switch to suppress a DDA output header on messages placed in the LOGDDA DDA routine.

Syntax

```
-LOGDDANOHEADER
```

This switch does not have any arguments.

A.4.27 LOGERROR

Use the LOGERROR switch with the LOGDDA switch to specify the severity of errors that are written to the logging connector.

Syntax

```
-LOGERROR=<error_severity>
```

Arguments	Required	Supported values
<error_severity>	Yes	• WARNING — Log all warnings. • ERROR — Log all errors and warnings. This is the default. • SEVERE — Log only severe errors.

Example

```
-LOGERROR=ERROR
```

A.4.28 LOGFILE

The LOGFILE switch specifies the file used to log the create, update, copy, and drop processes.

Syntax

```
-LOGFILE=<Log file path and name>,OVERWRITE
```

Argument	Required	Supported values
<Log file path and name>	Yes	The fully-qualified path and file name of the log file.
OVERWRITE	No	Unless you specify the <OVERWRITE > argument, the <LOGFILE> switch appends to the file, if it already exists. This argument is not valid with copies. When you copy a database, the log file is always overwritten.

Example

```
-LOGFILE=C:\temp\DBAdmin.log,OVERWRITE
```

A.4.29 LOGFILTER

Use the LOGFILTER switch with the LOGDDA switch to pass only specific messages to the DDA routine.

Syntax

```
-LOGFILTER=<MessageNumber>
```

or

```
-LOGFILTER=<MessageNumberRange>
```

Arguments	Required	Supported values
<MessageNumber>	No	The engine error number to log. There is no default value.
<MessageNumberRange>	No	The engine error number range to log. There is no default value.

Example

```
-LOGFILTER=2377
```

or

```
-LOGFILTER=2300-2400
```

A.5 Switches M-P

A.5.1 MAILMARK

Use the MAILMARK engine switch when you want to produce output with a 2D Datamatrix barcode that conforms to the Royal Mail Mailmark technical requirements. The engine switch enforces C40 encoding for the Mailmark barcode.

In addition to using the engine switch, you must set up a 2D Datamatrix barcode object in Design Manager as you normally would. The size of the Royal Mail Mailmark barcode varies depending on the Mailmark barcode type. There are three types of Mailmark barcodes with parameters that are specific to each type. The following table lists the three Royal Mail Mailmark barcode types and gives the physical measurements that each type will have in the output. The table also lists the appropriate height and width settings that you must apply in Design Manager on the **Basic** tab of the barcode object (in the **Dimension HxW (pixels)** list).

Barcode type	Output Measurement (millimeters)	Dimensions HxW (pixels)
Type 7	12x12	24x24
Type 9	16x16	32x32
Type 29	8x24	16x48

Additionally, you must apply the following settings on the **Basic** tab of the barcode object:

- From the **Error correction level** down list, select **ecc 200**.
- In the **Pixel block size** box, select values between 0.05 cm. and 0.07 cm, depending on your output resolution, to produce the correct barcode when printed.
- From the **Data type** list, select **Full 128 ASCII set**. Note that including characters outside of this character set will cause the barcode to unlatch from the C40 encoding.

Syntax

```
-MAILMARK=<barcode_name>
```

The only argument required is the barcode object name as it appears in Design Manager.

Supported in

8.0 and later

A.5.2 MARK_LAST_SKIPPED_FRAME_ARTICLE_FRAME_FILL_MODE

Use the MARK_LAST_SKIPPED_FRAME_ARTICLE_FRAME_FILL_MODE switch to override default behavior for marking skipped frames when **Articles** is selected for the **Frame fill method** on the **Composition** tab of the document properties. When using the article frame fill method, default behavior is to mark the first skipped frame, so that message content cannot flow out of it. When using the MARK_LAST_SKIPPED_FRAME_ARTICLE_FRAME_FILL_MODE switch, the last skipped frame is marked so that message content cannot flow out of it. Using this switch can prevent messages from appearing out of a logical order in the document.

Syntax

```
-MARK_LAST_SKIPPED_FRAME_ARTICLE_FRAME_FILL_MODE
```

This switch does not have any arguments.

Supported in

6.1.018 and later

A.5.3 MAX_DECOMPRESSED_PACKAGE_BYTES

When a package file has been compressed during the packaging process, you can use the MAX_DECOMPRESSED_PACKAGE_BYTES switch to specify the maximum size allowed

for the decompressed package to be stored in memory when the Exstream engine decompresses the package file.

If the size of the decompressed package exceeds the maximum set by this switch, the package contents must be stored in a temporary directory. If no temporary directory has been specified, then the decompression process stops. To specify a temporary directory, use the TEMPORARY_DIRECTORY switch.

> **Note:** Keep in mind that compressed package files can be decompressed only by an Exstream production engine that is installed and running in a Windows, Linux, or AIX environment.

Syntax

```
-MAX_DECOMPRESSED_PACKAGE_BYTES=<numberOfBytes>
```

Arguments

The argument for this switch is the maximum number of bytes allowed for the size of the decompressed package file. The default value for this argument is 104857600 (100 MB).

Example

```
-MAX_DECOMPRESSED_PACKAGE_BYTES=75000000
```

Related switches

COMPRESSPACKAGEFILE

TEMPORARY_DIRECTORY

Supported in

- 16.4.6 and later versions

A.5.4 MAXOVERLAYS

Use the MAXOVERLAYS switch to specify the maximum number of inline overlays that can be created and used (up to 99999). The engine renders the remaining overlays as images on pages at the end of the document.

Syntax

```
-MAXOVERLAYS=<maximum number of overlays>
```

The only argument is the maximum number of overlays. This switch has no default value.

Example

```
-MAXOVERLAYS=10000
```

A.5.5 MEMORYCACHE

Use the MEMORYCACHE engine switch to save memory by writing table row data to a file or to extra memory. Use the MEMORYCACHE switch as your first option for caching because it saves the most memory. However, keep in mind that because of the large amount of memory it saves, it processes data more slowly than other caching options. The engine stores all subsequent row data for the table to a file when the row count for any one table exceeds one of the following choices:

- 250,000 (the default)
- The number you specify with CACHETABLE

OpenText recommends that you use the MEMORYSAVE switch when producing applications on the 32-bit version of the production engine.

Syntax

```
-MEMORYCACHE=<option>
```

The only argument for this switch is the option that you want to use to save memory and supports the following values:

- `<fileName>`—The name of an ESDS VSAM file on z/OS. The file you specify must be able to accommodate row data for your largest customer.
- `HIPERSPACE`—For improved performance, you can specify `HIPERSPACE` instead of a VSAM file name. Hiperspace™ is extra memory on z/OS. Your system administrator can specify the number of hiperspaces allowed per application and their maximum size (up to two gigabytes for each hiperspace). With this switch, very large applications can be cached in hiperspace instead of to a VSAM file.

As the engine composes the table, it sends the stored row data into the print stream along with pagination, header/deleteID, and page design information. The engine then resets the file to prepare for the next large table in the run.

With extremely large tables, you can use multiple MEMORYCACHE switches with different file names. The engine writes row data to the next specified file when one file fills up.

A.5.6 MEMORYSAVE

Use the MEMORYSAVE switch to manage memory dynamically. When a transaction or page count exceeds the number you specify, the engine flushes previously used variables and

reallocates memory at the start of new sections or customers. Use the MEMORYSAVE switch when producing applications on the 32-bit version of the production engine.

Syntax

```
-MEMORYSAVE=<number of transactions>
```

The required argument for this switch is the number of transactions. This switch has no default value.

Example

```
-MEMORYSAVE=5000
```

A.5.7 MEMORYSTATS

Use the MEMORYSTATS engine switch to include detailed memory statistics at the end of a report in a message file.

Syntax

```
-MEMORYSTATS
```

This switch does not have any arguments.

Supported in

6.1 and later

A.5.8 MESSAGEFILE (Engine)

Use the MESSAGEFILE switch to specify a text-based format for the message reporting file.

Syntax

```
-MESSAGEFILE=<MyMessageFile.dat>
```

The text file containing the message reporting file is the required argument.

If no value is specified, the default value for PC and UNIX is `ExstreamMessages.dat`. For z/OS, it is `DD:MESSAGE`.

Example

```
-MESSAGEFILE=MyMessageFile.dat
```

A.5.9 MESSAGEFILE (Packager)

This packaging switch lets you specify the fully-qualified file name and path to the location to which you want Packager to write the message file.

Syntax

```
-MESSAGEFILE=<filename>
```

The only argument for this switch is the fully-qualified file name and path to the location to which you want Packager to write the message file.

If no value is specified, the default value is `PackagerMessages.dat`.

Example

```
-MESSAGEFILE=path/to/packagermessages.txt
```

A.5.10 MESSAGELEVEL

Use the MESSAGELEVEL switch with the MESSAGEFILE (Engine) switch to specify the lowest level of engine messages to be logged.

Syntax

```
-MESSAGELEVEL=<level>
```

Arguments	Required	Supported values
`<level>`	Yes	• `INFO`—Displays information, warnings, errors, and severe errors. This is the default. • `WARNING`—Displays warnings, errors, and severe errors • `ERROR`—Displays errors and severe errors

Example

```
-MESSAGELEVEL=INFO
```

A.5.11 META_DOT_OFFSET

Use the META_DOT_OFFSET switch to set the number of dots you want to offset content from the top of the page. Exstream Design and Production automatically offsets at 100 dots from the top of the page as this is the standard for a letter size paper type.

Syntax

```
-META_DOT_OFFSET=<number of dots>
```

Arguments	Required	Supported values
`<number of dots>`	Yes	The value of the offset. The default is 100.

A.5.12 META_SIMPLEXDUPLEX_WITH_NUFRONT

Use the META_SIMPLEXDUPLEX_WITH_NUFRONT switch with the Metacode command SIDE=NUFRONT during Metacode simplex and duplex mode processing if you want to produce simplex pages as output without changing the printer to simplex.

Using the META_SIMPLEXDUPLEX_WITH_NUFRONT switch can result in faster printing, but the printer counts the blank backs of pages as clicks. If you do not use this switch during Metacode simplex and duplex mode, the default is that the printer counts clicks only for front pages and the backs of pages that are not blank. However, the default can result in slower printing.

Syntax

```
-META_SIMPLEXDUPLEX_WITH_NUFRONT
```

This switch does not have any arguments.

A.5.13 METAPAGE

Use the METAPAGE switch to specify whether 0x8B or 0x89 indicates a new page for printing Metacode based on your consumer requirements.

Syntax

```
-METAPAGE=<page control value>
```

Arguments	Required	Supported values
`<page control value>`	Yes	The page control value. The default value is `8B`.

Example

```
-METAPAGE=89
```

A.5.14 MGWAPPDOMAIN

Used with: Engine, Packager

Use the MGWAPPDOMAIN switch to specify the application domain for connecting to the CAS repository that contains your CAS resources. This must be a domain within the tenant that is specified on the **Integration** tab or in the MGWTENANT switch.

> **Caution:** Using this switch will override the application domain specified in the **Application domain** box on the **Integration** tab in **System Settings** in Design Manager.

Syntax

```
-MGWAPPDOMAIN=<domain>
```

The argument for this switch is the name of the application domain. This switch has no default value.

Example

```
-MGWAPPDOMAIN=yourDomain
```

Supported in

16.3.0 and later

A.5.15 MGWPASSWORD

Use the MGWPASSWORD switch to specify the password for the OTDS user that you want to use to connect to the CAS repository that contains your CAS resources.

Syntax

```
-MGWPASSWORD=<password>
```

The argument for this switch is the password for the OTDS user that you want to use.

Supported in

16.3.0 and later

A.5.16 MGWTENANT

Used with: Engine, Packager

Use the MGWTENANT switch to specify the management gateway tenant that corresponds to the CAS repository that contains your CAS resources.

Caution: Using this switch will override the management gateway tenant that is specified in the **Tenant name** box on the **Integration** tab in **System Settings** in Design Manager.

Syntax

```
-MGWTENANT=<tenantName>
```

The argument for this switch is the name of the management gateway tenant. This switch has no default value.

Example

```
-MGWTENANT=yourTenant
```

Supported in

16.3.0 and later

A.5.17 MGWURL

Used with: Engine, Packager

Use the MGWURL switch to specify the URL for the management gateway server that you are using to connect to the CAS repository that contains your CAS resources.

Caution: Using this switch will override the management gateway URL that is specified in the **Management Gateway URL** box on the **Integration** tab in **System Settings** in Design Manager.

Syntax

```
-MGWURL=<url>
```

The argument for this switch is the base URL for the management gateway server. This switch has no default value.

The URL must be provided in the following format:

```
https://<hostname>:<port>
```

Example

```
-MGWURL=https://sampleserver.example.com:28600
```

Supported in

16.3.0 and later

A.5.18 MGWUSER

Use the MGWUSER switch to specify the OTDS user name that you want to use to connect to the CAS repository that contains the CAS resources that are referenced by the placeholder variables in your design.

Syntax

```
-MGWUSER=<userName>
```

The argument for this switch is the OTDS user name that you want to use.

Supported in

16.3.0 and later

A.5.19 MSG or MSGCHANGE

Use the MSG (or MSGCHANGE) switch to modify an engine message. You can use this switch multiple times in your control file.

Note: You cannot change the severity of the following messages:

- `1035 FILEFORMATNOTAUTHORIZED`
- `1033 DDANOTAUTHORIZED`
- `1037 ONDEMAND_NOTAUTHORIZEDSEVERE`
- `2608 UNLICENSEDFEATURE`

Syntax

```
-MSG=<messageNum>,<severity>,[<displayNum>],[<errorLevel>],[<suppress>]
```

or

```
-MSGCHANGE=<messageNum>,<severity>,[<displayNum>],[<errorLevel>],[<suppress>]
```

Argument	Required	Supported values
`messageNum`	Yes	The message number of the engine message that you want to modify

Argument	Required	Supported values
`severity`	Yes	The severity level that you want to assign to the modified message: • `S`—Severe • `W`—Warning • `E`—Error • `I`—Informational
`displayNum`	No	The number of times that the engine can display the message during the run (no practical limits)
`errorLevel`	No	The invalid data error level that you want to set for each customer: • `0`—No invalid data • `1`—Invalid data, continue, or set to default • `2`—Invalid data, error • `10`—Invalid data, skip customer This argument sets the value of the SYS_CustInvalidDataLevel system variable. If the invalid data level for a customer exceeds the maximum value, the customer is skipped and the engine continues to run.
`suppress`	No	• `TRUE`—Prevents the modified message from being displayed • `FALSE`—Allows the modified message to be displayed The default value is `FALSE`.

Example

```
-MSG=3004,E,3
```

This changes the message `EX003004W` into `EX003004E` and lets it repeat up to 3 times.

A.5.20 MSGENCODE

Use the MSGENCODE switch to set the encoding for the message, debug, or reporting file. The name must match an encoding name in the database from which the package file was created.

Syntax

```
-MSGENCODE=<encoding_name>
```

Arguments	Required	Supported values
`<encoding_name>`	Yes	The encoding name: • `ASCII` • `BIG5` • `EBCDIC` • `ISO88592` • `JIPS` • `LATIN1` • `LATIN2` • `SJIS` • `UTF16` (default) • `UTF16BE` • `UTF16LE` • `UTF8`

Example

```
-MSGENCODE=ASCII
```

A.5.21 MSGLANGUAGE

Use the MSGLANGUAGE switch to specify a language for engine messages that is different from the language that was used when the package file was created. Using this switch tells the engine which encoding to use for the message resource file and changes the default message resource file name to the name of the corresponding language-specific message resource file. To override the default message resource file name, use the MSGRESOURCE switch.

Note: You must specify this switch before any other switches on the command line, and you must list it as the first switch if used in a control file.

Syntax

```
-MSGLANGUAGE=<languageCode>
```

The only argument for this switch is the language code for the language that you want to use for engine messages:

- `de-de` (German)
- `en-us` (English)—This is the default.
- `es-mx` (Spanish)

- `fr-fr` (French)
- `ja-jp` (Japanese)
- `nl-nl` (Dutch)
- `pt-br` (Portuguese Brazilian)
- `zh-cn` (Simplified Chinese)

Example

```
-MSGLANGUAGE=EN-US
```

A.5.22 MSGRESOURCE

Use the MSGRESOURCE switch to override the default message resource file name and specify the file name of the message resource file that you want to use. The default name is determined by the language that is used when the package file was created, or by theMSGLANGUAGE switch.

- For Windows and UNIX platforms, the default message resource file name is `MsgResource_<languageCode>.dat`, for example, `MsgResource_en-us.dat`.
- For z/OS platforms, the default message resource file name is `DD:DLMSGRES`, for all languages

The MSGRESOURCE switch is necessary on z/OS only if you do not want to use the default DD name for the message resource file.

> **Note:** The MSGRESOURCE switch must be specified as the first switch on the command line or control file, or it must follow the MSGLANGUAGE switch if both switches are used.

Syntax

```
-MSGRESOURCE=<fileName>
```

The only argument for this switch is the file name of the message resource file that you want to use. If the message resource file that you specify is not in the engine directory, you must use the fully qualified path and file name.

Example 1

For Windows platforms:

```
-MSGRESOURCE=C:\path\to\myMsgResourceFile
```

Example 2

For z/OS platforms:

```
-MSGRESOURCE=DD:DMSGCUST
```

In JCL:

```
// DLMSGRES DD DSN=P390A.DMSGCUST,DISP=SHR
```

A.5.23 MVS_DATACLASS

Use the MVS_DATACLASS switch to set an SMS data class on the z/OS platform.

Using this switch also sets the following:

- VOLSER to `null`
- Primary and secondary allocation quantities to `0`

Syntax

```
-MVS_DATACLASS=<dataclass>
```

The only argument for this switch is the data class that you want to set.

Example

```
-MVS_DATACLASS=SMS
```

A.5.24 NEW_FLOW_LEFT_RIGHT_RELATIVE_METHOD

Use the NEW_FLOW_LEFT_RIGHT_RELATIVE_METHOD engine switch when you produce output from an application that includes two side-by-side automated tables that are set to be relative to one another to ensure that the tables flow as expected to multiple pages. When you have two tables that meet the previous conditions within a design, the table on the right will flow as expected in the resulting output, but the table on the left will flow from the first page to the third page. This engine switch corrects this behavior and the output will appear as expected.

Syntax

```
-NEW_FLOW_LEFT_RIGHT_RELATIVE_METHOD
```

This switch does not have any arguments.

Supported in

- 8.0.336 and later maintenance releases
- 8.6.110 and later maintenance releases
- 9.0.104 and later maintenance releases
- 9.5.101 and later versions

A.5.25 NO_AFP_DRAWRULE

Because the engine uses Graphics Object Content Architecture (GOCA) fill patterns, you can use the NO_AFP_DRAWRULE switch to override draw rules for rectangular shading.

Syntax

```
-NO_AFP_DRAWRULE
```

Arguments

This switch does not have any arguments.

Supported in

- 3.0.020 and later
- 3.5.017 and later

A.5.26 NO_FILL_ROTATION

For backward compatibility after you upgrade from version 3.0 or earlier, use the NO_FILL_ROTATION engine switch when you produce TIFF, Metacode, or AFP output from an application that contains charts with the **Convert charts and shapes to images** check box selected on the **Resource Management** tab of the **Output Object Properties** in Designer. The NO_FILL_ROTATION engine switch does not correct the rotation of fills for landscape pages.

Syntax

```
-NO_FILL_ROTATION
```

This switch does not have any arguments.

Supported in

2.5 and later

A.5.27 NO_META_FORMFEED_AFTER_INTERLEAVED

Use the NO_META_FORMFEED_AFTER_INTERLEAVED switch to override \x8B\x20\x01 records (form feeds) after page interleaved graphics. \x8B\x20\x01 records (form feeds) cause blank pages on some Metacode printers.

Syntax

```
-NO_META_FORMFEED_AFTER_INTERLEAVED
```

This switch does not have any arguments.

A.5.28 NO_SMOOTH_FILLS

For backward compatibility after you upgrade from Exstream Design and Production 4.0 and earlier, use the NO_SMOOTH_FILLS engine switch to ensure that smooth fills are not used when you produce TIFF, Metacode, or any other output driver from an application that contains images or traditional charts with the **Convert charts and shapes into images** option selected on the **Resource Management** tab in the output object properties in Design Manager.

Syntax

```
-NO_SMOOTH_FILLS
```

This switch does not have any arguments.

Supported in

4.0 and later

A.5.29 NODDAOUTPUTHEADER

Use the NODDAOUTPUTHEADER switch to skip the header generated when using MQ Connector, JMS Connector, or any user-written DDAOUTPUT. Use the NODDAOUTPUTHEADER switch if you do not want headers to be generated based on data section lengths.

Syntax

```
-NODDAOUTPUTHEADER
```

This switch does not have any arguments.

A.5.30 NOREPORTS

Use the NOREPORTS switch to suppress all report files, including those that are defined in a DDA routine.

Syntax

```
-NOREPORTS
```

This switch does not have any arguments.

A.5.31 NORTHSOUTHINDEX

Use the NORTHSOUTHINDEX switch to specify the location in which to write and read the sort index file. By default, when ordering customers north to south, the engine deletes the temporary sort index file from memory. When you use this switch, the engine writes the file to the location specified in the switch. Then the engine reads the sort index file from that location.

Syntax

```
-NORTHSOUTHINDEX=<sort index file>
```

The argument is the fully-qualified sort index file name. This switch has no default value.

Example

```
-NORTHSOUTHINDEX=C:\PostSort\SortIndexStorage\SortIndexName.txt
```

A.5.32 NOVARDEL

Use the NOVARDEL switch to suppress the optimization that removes unreferenced variables.

Syntax

```
-NOVARDEL
```

This switch does not have any arguments.

A.5.33 NTH

Use the NTH switch to process only every nth customer.

Syntax

```
-NTH=<starting customer number>
```

The argument for this switch is the starting customer. The default value for this switch is 1.

Example

In the following example, only customers 2, 4, 6, 8, and so on are processed:

```
-NTH=2
```

A.5.34 NUM_CUSTOMERS_HOLD_MUP_IMPORTS

When you produce multiple-up (MUP) output from an application that imports files (such as images) at run time and you are using the CUSTOMER_RESET_IMPORTS engine switch to reset imports between customers, use the NUM_CUSTOMERS_HOLD_MUP_IMPORTS engine switch to specify the number of customers for which you want to hold imported files in memory during an engine run. Using this switch can help keep necessary imported files in memory if you are using multiple-up pages that contain data from multiple customers. You must use the NUM_CUSTOMERS_HOLD_MUP_IMPORTS engine switch in conjunction with the CUSTOMER_RESET_IMPORTS engine switch.

Syntax

```
-NUM_CUSTOMERS_HOLD_MUP_IMPORTS=<numberofcustomers>
```

The argument for this switch is the number of customer import pages that you want to hold in memory during an engine run.

Example

```
-NUM_CUSTOMERS_HOLD_MUP_IMPORTS=64
```

Supported in

6.1.022 and later

A.5.35 OLD_FRAME_LANGUAGE_CHECKS

For backward compatibility after you upgrade to version 6.0 or later, use the OLD_FRAME_LANGUAGE_CHECKS engine switch when you produce output from an application that includes message frames on language layers and the message frames do not appear correct in the output. The OLD_FRAME_LANGUAGE_CHECKS engine switch uses old software behavior for determining the language eligibility of message frames.

As an alternative to using this switch, you can change your selections for the **Use default language as background for other languages** check box or the **Send default language if customer language does not exist** check box on the language layer properties.

Syntax

```
-OLD_FRAME_LANGUAGE_CHECKS
```

This switch does not have any arguments.

Supported in

5.0.075 and later

A.5.36 ONDEMAND

Use the ONDEMAND switch to set the engine into transaction (continuous operation) mode for on-demand production.

When running in on-demand mode, the engine continuously processes data, even if it encounters any severe errors on input data. If the customer driver file contains invalid data for a customer, the engine can do any of the following:

- Issue an error message
- Ignore the remaining data for the customer
- Skip to the end of the file

To keep the engine running despite errors not related to input, use the MSG or MSGCHANGE switch.

Syntax

```
-ONDEMAND
```

This switch does not have any arguments.

A.5.37 ONDEMAND_CLOSE_OUTPUTS_ON_SHUTDOWN_ONLY

Use the ONDEMAND_CLOSE_OUTPUTS_ON_SHUTDOWN_ONLY switch to instruct the engine to keep output files open until the engine is commanded to shut down. The ONDEMAND_CLOSE_OUTPUTS_ON_SHUTDOWN_ONLY switch prevents file breaks in a multiple customer on-demand run.

Syntax

```
-ONDEMAND_CLOSE_OUTPUTS_ON_SHUTDOWN_ONLY
```

This switch does not have any arguments.

A.5.38 ONLY_USE_REPEATING_HEADER_ROWS_FOR_RTF_TABLES

In Exstream Design and Production versions earlier than 16.2.0, only header rows with the row type **Repeating header (H+)** were marked as headers in RTF output. Other types of header rows were not marked as headers.

This issue was corrected in Exstream Design and Production version 16.2.0.

For backward compatibility after you upgrade to Exstream Design and Production 16.2.0, use the ONLY_USE_REPEATING_HEADER_ROWS_FOR_RTF_TABLES switch. This switch will be automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 9.5.999 or earlier.

Syntax

```
-ONLY_USE_REPEATING_HEADER_ROWS_FOR_RTF_TABLES
```

This switch does not have any arguments.

Supported in

16.2.0 and later

A.5.39 OPENFILES

Use the OPENFILES switch to open all files or only files as needed while the queue is processed.

Syntax

```
-OPENFILES=<files>
```

Arguments	Required	Supported values
`<files>`	Yes	• `ASNEEDED`—Opens queue files only where data is written • `ALL`—Opens all queues and writes a blank file, even if nothing is sent to the queue. This is the default

Example

```
-OPENFILES=ASNEEDED
```

A.5.40 OTDSRESOURCEID

Used with: Engine, Packager

Use the OTDSRESOURCEID switch to specify the OTDS resource identifier that is associated with the resource for Exstream, as it appears in the OTDS server that you are using for user authentication.

Caution: Using this switch will override the OTDS resource identifier that is specified in the **Resource ID** box on the **Integration** tab in **System Settings** in Design Manager.

Syntax

```
-OTDSRESOURCEID=<identifier>
```

The argument for this switch is the resource identifier as it appears in the OTDS server. You can obtain the resource identifier using the **Resources** section in the OpenText Directory Services Administration web browser interface.

Example

```
-OTDSRESOURCEID=abd87460-39d4-4f2c-b30d-f4a2ff83727a
```

Supported in

16.3.0 and later

A.5.41 OTDSURL

Used with: Engine, Packager

Use the OTDSURL switch to specify the URL for the OTDS server that you are using for user authentication.

Caution: Using this switch will override the OTDS URL that is specified in the **OTDS base URL** box on the **Integration** tab in **System Settings** in Design Manager.

Syntax

```
-OTDSURL=<url>
```

The argument for this switch is the base URL for the OTDS server.

In a single-tenant OTDS system, or if you are connecting to the default tenant in a multi-tenant OTDS system, the URL must be provided in the following format:

```
https://<hostname>:<port>
```

In a multi-tenant OTDS system, if you are connecting to a tenant other than the default tenant, the URL must also specify the tenant name:

```
https://<hostname>:<port>/otdstenant/<tenantname>
```

Example

```
-OTDSURL=https://sampleserver.example.com:8443
```

```
-OTDSURL=https://sampleserver.example.com:8443/otdstenant/yourtenant
```

Supported in

16.3.0 and later

A.5.42 OUTPUTDIRECTORY

Use the OUTPUTDIRECTORY switch to override the directory specified in the printer or queue and write output to the specified directory.

Syntax

```
-OUTPUTDIRECTORY=<path>
```

The only argument is the output directory path name. This switch has no default value.

Syntax

```
-OUTPUTDIRECTORY=C:\Exstream\Output\
```

A.5.43 OUTPUTFILE

Use the OUTPUTFILE switch to specify the output file to use if this package file was made for a specific output.

Note: Do not use this switch only if you are using output queues.

Syntax

```
-OUTPUTFILE=<file name>
```

The only argument for this switch is the fully-qualified file name. This switch does not have a default value.

Example

```
-OUTPUTFILE=C:\Output\test.pdf
```

A.5.44 OUTPUTFILES

Use the OUTPUTFILES switch to specify whether to create all the output files as specified in each queue.

Syntax

```
-OUTPUTFILES=<create output>
```

Arguments	Required	Supported values
`<create output>`	Yes	• YES—Creates the output files. This is the default. • NO—Does not create the output files. This lets you test an application without filling up disk space with output files.

A.5.45 OVERRIDE_DEMO_WATERMARK_POSITION

Use the OVERRIDE_DEMO_WATERMARK_POSITION switch to change the placement of the "Demonstration Powered by Exstream" watermark that appears at the top of pages when you produce output without using a production license key.

Note: This switch is not supported on RTF output. Instead, use DOCX output.

Syntax

```
-OVERRIDE_DEMO_WATERMARK_POSITION=<position>
```

Arguments	Required	Supported values
`<line_end_command>`	Yes	• CENTER—The watermark appears at the center of a page. • BOTTOM—The watermark appears at the bottom of a page. If you do not specify an argument, the watermark will appear at the top of a page.

Example

```
-OVERRIDE_DEMO_WATERMARK_POSITION=CENTER
```

A.5.46 OVERRIDE_HTML_TD_DEFAULT_PADDING

When creating HTML or HTML (email) output in Exstream Design and Production versions 9.5.301 and earlier, if the design contained a table cell with a padding value of zero, the resulting HTML did not include the padding value. For example, instead of `<td style="padding: 0px">`, the resulting HTML was `<td>`. Without the padding value, the browser applied the default HTML padding value of 1 pixel. This resulted in visual differences between the design and the output.

In version 9.5.301, this issue was corrected.

For backward compatibility after you upgrade to Exstream Design and Production 9.5.301, use the OVERRIDE_HTML_TD_DEFAULT_PADDING switch. This switch will be automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 9.5.999 or earlier.

Syntax

```
-OVERRIDE_HTML_TD_DEFAULT_PADDING
```

This switch does not have any arguments.

Supported in

9.5.301 and later

A.5.47 PACKAGEFILE

Used with: Engine, Packager

Use the PACKAGEFILE switch to create a package file (when used for packaging), or to identify the package file that will be used in an engine run (when used with the engine).

This is switch is required to run the engine. You can use it more than once in a control file. However, if you have multiple packages in a two-pass engine run, the first package that is listed in the control file must be the main package file, and other package files that are listed must contain only campaigns or documents.

Do not use the PACKAGEFILE switch for output sorting and bundling. Instead, use the APP_PACKAGEFILES switch.

Syntax

```
-PACKAGEFILE=<fileName>
```

The argument for this switch is the fully qualified file name of the package file. If the file path has spaces in it, and if you are using the PACKAGEFILE switch at the command prompt, you must enclose the argument value in double quotation marks (" ").

Keep in mind that if you are using the switch with the engine, the full path is required only if the package file that you want to use is located in a directory other than the directory in which Exstream Design and Production is installed.

Example

```
-PACKAGEFILE="C:\path\to\MyPackageName.pub"
```

A.5.48 PACKAGEPROFILE

This packaging switch configures packaging settings based on the package profile. Packaging settings defined in the package profile override settings specified by packaging switches prior to the PACKAGEPROFILE switch.

Requirements

You must specify the APPLICATION before you specify a package profile with the PACKAGEPROFILE switch.

Syntax

```
-PACKAGEPROFILE=<profile name>
```

The only argument for this switch is the name of the packaging profile you want to use.

Example

```
-APPLICATION=MyApp
```

```
-PACKAGEPROFILE=MyPackagingProfile
```

A.5.49 PACKAGETYPE

Use the PACKAGETYPE switch to specify which print resources to include in the package file.

Syntax

```
-PACKAGETYPE=<resourceOption>
```

Arguments	Required	Supported values
resourceOption	Yes	• ALL—Includes all print resource files that are available in the application, even if they are not used, and creates a package file. This is the default. • RESOURCESONLY—Creates a print resource file that includes only print resources that are used in the application. • WITHOUTRESOURCES—Creates a package file that does not include print resource files.

Note: If you want to create a package file that only contains print resources, you must use the RESOURCESONLY as the argument.

Example

```
-PACKAGETYPE=RESOURCESONLY
```

A.5.50 PCL_EDGE_TO_EDGE_OFFSETS

Use the PCL_EDGE_TO_EDGE_OFFSETS switch to manually specify the edge-to-edge offset. This switch corrects PCL edge-to-edge printing for rotated coordinate systems.

Syntax

```
-PCL_EDGE_TO_EDGE_OFFSETS=<x offset>,<y offset>
```

Required arguments include the X offset and Y offset (in pixels). This switch has no default value.

A.5.51 PCL_FULL_COLOR_HIGHLIGHT

For backward compatibility after you upgrade to version 5.0 or later, use the PCL_FULL_COLOR_HIGHLIGHT engine switch to force the engine to use full-color commands to change the color of the primary palette between highlight, black and white, and color, instead of using the default PCL4 raster graphics commands (0, 1, or 2) to represent the palette that is used. This switch creates vector graphics instead of raster graphics.

Syntax

```
-PCL_FULL_COLOR_HIGHLIGHT
```

This switch does not have any arguments.

Supported in

5.0.065 and later

A.5.52 PDF_DBCS_STD_FONTS_ANSI

Use the PDF_DBCS_STD_FONTS_ANSI engine switch when you produce PDF output from an application that includes DBCS Unicode characters to translate the DBCS Unicode characters to WIN ANSI. When you produce PDF output from an application that includes DBCS Unicode characters, and you do not use this engine switch, they may not appear as expected or be replaced with other characters in the final output. This engine switch allows you to use DBCS Unicode characters to produce PDF output without enabling the inclusion of standard fonts. If your application currently includes standard fonts, the DBCS Unicode characters should appear in the output as expected without using this engine switch.

Syntax

```
-PDF_DBCS_STD_FONTS_ANSI
```

This switch does not have any arguments.

Supported in

8.0 and later

A.5.53 PDF_IMPORT_RASTER_PDF

Use the PDF_IMPORT_RASTER_PDF engine switch to create a raster image of PDF content that is dynamically imported at run time (PDF pass-through) for PDF output.

Note: This switch differs from using the **Image each page** option for a PDF output object in that it allows for full color support.

Syntax

```
-PDF_IMPORT_RASTER_PDF=<rasterType>
```

Argument	Required	Supported values
`rasterType`	Yes	• RGBJPEG—Converts imported PDF files to JPEG files with the RGB color profile. This option is best suited for online delivery. • CMYKJPEG—Converts imported PDF files to JPEG files with the CMYK color profile. This option is best suited for print delivery. • BWTIFF—Converts imported PDF files to TIFF files with a black and white color profile. This option provides the greatest reduction in file size, and is suitable for both print and online delivery.

Example

```
-PDF_IMPORT_RASTER_PDF=CMYKJPEG
```

Supported in

9.5.309 and later maintenance releases

16.2.4 and later maintenance releases

16.3.3 and later maintenance releases

16.4.10 and later versions

A.5.54 PDF_PASSTHROUGH_ACMA_CONVERT

For backward compatibility after you upgrade from version 8.0 or earlier, use the PDF_PASSTHROUGH_ACMA_CONVERT engine switch when you produce AFP output from an application that imports PDF content at run time, to ensure that the PDF content appears correctly in the resulting AFP output.

Syntax

```
-PDF_PASSTHROUGH_ACMA_CONVERT
```

This switch does not have any arguments.

Supported in

8.0.332 and later

A.5.55 PDF_PASSTHROUGH_AS_EPS

Use the PDF_PASSTHROUGH_AS_EPS switch to dynamically import PDF to AFP on z/OS. This switch converts the PDF to EPS and then inserts it into the print stream.

Note: This type of AFP prints only on certain printers. To make the PDF_PASSTHROUGH_AS_EPS switch work properly, you must run Print Services Facility (PSF) or InfoPrint Manager on a distributed platform with some special Adobe libraries.

Syntax

```
-PDF_PASSTHROUGH_AS_EPS
```

This switch does not have any arguments.

A.5.56 PDF_PASSTHROUGH_AS_LZW

Use the PDF_PASSTHROUGH_AS_LZW switch to use the LZW compression method to compress imported PDF files that contain images. The LZW compression method compresses PDF files that contain images by converting each page of the PDF file to an LZW-compressed CMYK TIFF file instead of only converting the images that are contained in the PDF file. The LZW compression method reduces the size of the PDF file.

The PDF_PASSTHROUGH_AS_LZW switch is used only when importing a PDF file that contains CMYK color spaces to AFP output.

If you are using the PDF_PASSTHROUGH_AS_LZW switch with PDF files that contain high-resolution images, or if you are importing a large file size PDF, you could encounter memory issues with the engine. Additionally, file size reduction is not guaranteed if the PDF file contains a large number of colors, or if the PDF file contains a large amount of full color images.

Syntax

```
-PDF_PASSTHROUGH_AS_LZW
```

This switch does not have any arguments.

A.5.57 PDF_PASSTHROUGH_AS_PDF

Use the PDF_PASSTHROUGH_AS_PDF engine switch when you produce AFP output from an application that imports dynamic PDF content and run-time image imports, and you do not want the dynamic PDF content processed with the default settings. This engine switch allows your compatible AFP device to process the dynamic PDF content in object containers, while the image imports are processed with the default settings.

Syntax

```
-PDF_PASSTHROUGH_AS_PDF
```

This switch does not have any arguments.

Supported in

8.0.332 and later

A.5.58 PDF_PASSTHROUGH_AS_RGB_JPEG

For backward compatibility after you upgrade from version 6.1 to version 7.0, use the PDF_PASSTHROUGH_AS_RGB_JPEG engine switch to force the engine to process CMYK images as RGB JPEG images when you produce AFP output from an application that imports PDF content at run time.

Syntax

```
-PDF_PASSTHROUGH_AS_RGB_JPEG
```

This switch does not have any arguments.

Supported in

7.0.635 and later

A.5.59 PDF_PASSTHROUGH_USE_HIGH_PRECISION

Use the PDF_PASSTHROUGH_USE_HIGH_PRECISION engine switch when you produce output from an application that uses PDF placeholders to dynamically import PDF content with a lower resolution into your design that will be generated as output with a higher resolution. When you produce high resolution PDF output from dynamically imported low resolution PDF files, the dynamically imported content may appear stretched or cut short. This engine switch ensures that the lower resolution PDF content displays as expected in the final output.

Syntax

```
-PDF_PASSTHROUGH_USE_HIGH_PRECISION
```

This switch does not have any arguments.

Supported in

- 8.6.111 and later maintenance releases
- 9.0.105 and later maintenance releases
- 9.5.101 and later versions

A.5.60 PDF_QUOTERIGHTFIX

For backward compatibility after you upgrade from version 7.0, to ensure that the engine uses the quoteright fixed encoding vector to compose an apostrophe character instead of the default encoding vector, use the PDF_QUOTERIGHTFIX engine switch when you produce PDF output from an application that includes text that contains apostrophes.

Syntax

```
-PDF_QUOTERIGHTFIX
```

This switch does not have any arguments.

Supported in

8.0.331 and later

A.5.61 PDF_QUOTESINGLEFIX

To use the quotesingle fixed encoding vector instead of the default encoding vector, use the PDF_QUOTESINGLEFIX engine switch when you produce PDF output from an application that includes quote characters in text. The quotesingle encoding vector ensures that the engine uses the appropriate value for quote characters during composition.

You must make sure that the **Build Type 1 fonts** is not selected from the **Font type** list on the **Resource Management** tab of the PDF output object properties.

Syntax

```
-PDF_QUOTESINGLEFIX
```

This switch does not have any arguments.

Supported in

4.0.079 and later

A.5.62 PDF_RESOLVE_RESOURCE_ RECURSION

If you use page overlays in your application, use the PDF_RESOLVE_RESOURCE_ RECURSION switch to create a separate resource dictionary for the page overlay objects. Without this switch, PDF parsing tools that cannot handle recursion might fail while loading resources. For example, your PDF printer or viewer might get stuck in a loop while repeatedly attempting to load the page resources.

Syntax

```
-PDF_RESOLVE_RESOURCE_RECURSION
```

This switch does not have any arguments.

A.5.63 PDFIMPORT_USE_CROPBOX_ DIMENSIONS

Use the PDFIMPORT_USE_CROPBOX_DIMENSIONS engine switch when you produce output from an application that dynamically imports PDF content. This switch lets you use the

PDF CropBox commands, instead of the MediaBox commands, to determine the page size.

Depending on the way in which the original PDF content was set up, when it is dynamically imported into the application, the imported pages may change in size due to the differences in the page size that is specified in the CropBox and MediaBox commands in the original PDF file.

Syntax

```
-PDFIMPORT_USE_CROPBOX_DIMENSIONS
```

This switch does not have any arguments.

Supported in

- 9.0.105 and later maintenance releases
- 9.5.101 and later versions

A.5.64 PDFTOPDF_HONOR_ROTATIONS

Use the PDFTOPDF_HONOR_ROTATIONS engine switch to prevent conflicts between the PDF page rotation settings and the auto-rotate settings that the engine can apply when you produce PDF output from an application that imports multiple-page PDF content with pages that include multiple rotation settings. When you include the PDFTOPDF_HONOR_ROTATIONS engine switch, page rotations that are set in the PDF appear correctly in the PDF output.

Syntax

```
-PDFTOPDF_HONOR_ROTATIONS
```

This switch does not have any arguments.

Supported in

- 7.0.641 and later maintenance releases
- 8.0.329 and later maintenance releases
- 8.6.104 and later maintenance releases
- 9.0.101 and later versions

A.5.65 PDFTOPDF_LANDSCAPE_FIT_COUNTERCLOCKWISE

Deprecated: This switch has been deprecated.

If you want Exstream Design and Production to automatically rotate PDF files that are imported at run time by 90 degrees counterclockwise (instead of the default 90 degrees clockwise) when needed to better fit the dimensions of a placeholder document, use the PDFTOPDF_LANDSCAPE_FIT_COUNTERCLOCKWISE engine switch.

Additionally, in version 8.6.101 and version 8.6.102, you can use the PDFTOPDF_LANDSCAPE_FIT_COUNTERCLOCKWISE engine switch together with theENABLE_EPS_PDF_IMPORT_AUTOROTATE engine switch to have Exstream Design and Production automatically rotate EPS and PDF images that are imported into a placeholder object or an empty image object when needed to better fit the placeholder object or empty image object.

The ENABLE_PDFEPSIMPORT_AUTOROTATE engine switch applies only to placeholder objects or empty image objects in designs created or edited in Designer version 8.5.105 or later. PDF or EPS images that are imported into placeholder objects or empty image objects in designs created in versions earlier than 8.5.105 might automatically rotate without using the switch.

This switch is deprecated in the latest version of Exstream Design and Production. In version 8.6.103 or later, use the ENABLE_EPS_PDF_IMPORT_AUTOROTATE_COUNTERCLOCKWISE engine switch instead of the PDFTOPDF_LANDSCAPE_FIT_COUNTERCLOCKWISE switch.

Syntax

```
-PDFTOPDF_LANDSCAPE_FIT_COUNTERCLOCKWISE
```

This switch does not have any arguments.

Supported in

- 6.0.015 and later maintenance releases
- 6.1.020 and later maintenance releases
- 7.0.408 to 8.6.102

A.5.66 PDFTOPS_IGNORE_PAGEROTATION

Use the PDFTOPS_IGNORE_PAGEROTATION switch to force PDF imports to PostScript outputs to ignore the imported PDF's page-level rotation.

Syntax

```
-PDFTOPS_IGNORE_PAGEROTATION
```

This switch does not have any arguments.

A.5.67 PDFTOPS_SKIP_SPLASH_RASTER

For backward compatibility after you upgrade to version 8.0 or later, use the PDFTOPS_SKIP_SPLASH_RASTER engine switch to bypass the splash rasterization engine functionality when you produce PostScript and PostScript-derivative outputs from an application that imports PDF content at run time.

Versions 8.0 and later use splash raster by default, as there was better support for transparency through the splash raster code when XPDF was upgraded.

Syntax

```
-PDFTOPS_SKIP_SPLASH_RASTER
```

This switch does not have any arguments.

Supported in

8.0.101 and later

A.5.68 PDFTOPS_XPDF_L3

Use the PDFTOPS_XPDF_L3 engine switch when producing PostScript output from an application that uses imported PDF content (PDF pass-through). Without the engine switch, Adobe Acrobat Distiller may fail to convert a PostScript file output to a PDF file.

Syntax

```
-PDFTOPS_XPDF_L3
```

This switch does not have any arguments.

Supported in

8.0327 and later

A.5.69 PDL

Use the PDL switch to select the output type to create at run time.

Syntax

```
-PDL=<outputType>
```

Arguments	Required	Supported values
outputType	Yes	• 3211LD • AFP • Composed XML • CONTENTXML • EDGAR HTML • Empower • HTML • HTML EMAIL • IJPDS • LIVE • METACODE • MIBF • MULTI-CHANNEL XML • PCL • PDF • PDF/A • PDF/VT • POSTSCRIPT • POWERPOINT • PPML • PPTX • RTF • TIFF • TOP • VDX • VIPP • VPS • Word (*.docx) • ZPL

Example

```
-PDL=POSTSCRIPT
```

A.5.70 PLACE_LATE_NONFLOWING_ OBJECTS_LAST

Use the PLACE_LATE_NONFLOWING_OBJECTS_LAST engine switch when you produce output from an application that imports a full page image at run time, to force late, non-flowing objects to be composed last.

Syntax

```
-PLACE_LATE_NONFLOWING_OBJECTS_LAST
```

This switch does not have any arguments.

Supported in

- 8.0.335 and later
- 8.6.108 and later
- 9.0.102 and later
- 9.5.101 and later

A.5.71 PNG_IGNORE_RESOLUTION

Use the PNG_IGNORE_RESOLUTION switch to ignore any native resolution information in a PNG-formatted image and instead assume that the image resolution is the same as the resolution set in the output. If this switch is omitted, images with resolutions different from the output resolution will appear at a different size to match the output resolution.

For example, when creating HTML output, this switch causes PNG images to be displayed at their actual pixel dimensions on the page. If this switch is omitted, the height and width attributes for each image are adjusted to maintain relative positions and sizes from the design and match printed output.

Syntax

```
-PNG_IGNORE_RESOLUTION
```

This switch does not have any arguments.

A.5.72 POSTSORTQUEUE

Use the POSTSORTQUEUE switch to specify an output queue to create postsort output when there are multiple queues in the application. List the main (primary) queue object before any postsort queues in the Library. The document sorting and bundling properties and bundling control variables have the following characteristics:

- They must be identical to those on the primary sorting and bundling queue.
- They can have different cover or trailer pages, or no cover or trailer pages.

All page counts for bundles and customers are taken from the primary sorting and bundling queue.

The POSTSORTQUEUE switch overrides the queue rule when you specify an output queue.

Syntax

```
-POSTSORTQUEUE=<output queue>
```

Arguments	Required	Supported values
`<output queue>`	Yes	• <*queue name*>—Creates post-sort output for a specific queue. The queue name is case-sensitive. • `ALL`—Creates postsort output for all queues that have **For post-sort processing only** selected in the **Use** list on the **Basic** tab of the output queue object properties can receive output. This switch has no default value.

Example

```
-POSTSORTQUEUE=MyMainQueue
```

A.5.73 PRECOMPOSE_PARAS_FOR_NONDATA_DOCS

Use the PRECOMPOSE_PARAS_FOR_NONDATA_DOCS switch to tell the engine to compose any data-driven content that appears in documents that are not driven by data files before composing those documents.

Syntax

```
-PRECOMPOSE_PARAS_FOR_NONDATA_DOCS
```

This switch does not have any arguments.

Supported in

16.4.0 and later

A.5.74 PRESERVE_BLANK_ELEMENTS_IN_JOIN_FUNCTION

Use the PRESERVE_BLANK_ELEMENTS_IN_JOIN_FUNCTION switch to force the Join function to display blank array elements in the output string. For example, an array variable contains [a,,c] and you specify a comma delimiter in the Join function. If the Join function does not preserve the blank elements, the output string is "a,c." If the Join function does preserve blank elements, the output string is "a,,c", which includes an extra delimiter for the empty array element.

Syntax

```
-PRESERVE_BLANK_ELEMENTS_IN_JOIN_FUNCTION
```

This switch does not have any arguments.

A.5.75 PRESERVE_LATE_COUNTS_FOR_REPRINTS

Use the PRESERVE_LATE_COUNTS_FOR_REPRINTS switch if you want to reprint a page range from a prior run and you want the page counts to match those of the original output. The PRESERVE_LATE_COUNTS_FOR_REPRINTS switch prevents the engine from re-computing the page counts used in late compose objects, such as barcodes.

Syntax

```
-PRESERVE_LATE_COUNTS_FOR_REPRINTS
```

This switch does not have any arguments.

A.5.76 PRESERVE_SECTION_DESIGN_ORDER

When you produce output from an application that contains data-driven sections, use the PRESERVE_SECTION_DESIGN_ORDER engine switch to force the engine to add data-driven sections to the output in the same order in which they appear in the application design, instead of appending them to the section content.

Syntax

```
-PRESERVE_SECTION_DESIGN_ORDER
```

This switch does not have any arguments.

A.5.77 PRESERVE_TAGGED_PARA_FORMAT

Use the PRESERVE_TAGGED_PARA_FORMAT switch to force new paragraphs created using tagged text to use the style and formatting of the previous paragraph.

Syntax

```
-PRESERVE_TAGGED_PARA_FORMAT
```

This switch does not have any arguments.

A.5.78 PRINT_PAGE_RANGE_USE_PCM

Use the PRINT_PAGE_RANGE_USE_PCM switch to force the SYS_PagePrintStart and SYS_PagePrintEnd variables to honor the page count method instead of the physical page values. For two-step processing, this switch must be used in both passes.

Syntax

```
-PRINT_PAGE_RANGE_USE_PCM
```

This switch does not have any arguments.

Supported in

6.1.018 and later

A.5.79 PS_CONVERT_CHARTS_AND_SHAPES_TO_IMAGE

Use the PS_CONVERT_CHARTS_AND_SHAPES_TO_IMAGE switch to convert charts and shapes to images in PS, VPS, PPML, VIPP, and VDX outputs. When you convert a shape or chart into an image, the shape or chart retains its design appearance (such as patterns) that might otherwise not be supported by the output format. However, the engine cannot convert charts and shapes that are inside overlays for PostScript output. If you want to convert charts and shapes that are within overlays into images, you must select the **Do not create overlays** option from the **Overlay processing** list in the printer settings for PostScript output.

Syntax

```
-PS_CONVERT_CHARTS_AND_SHAPES_TO_IMAGE
```

This switch does not have any arguments.

Supported in

This switch is available in Exstream Design and Production 4.0 only. In versions 4.5.4 and later, this switch was changed to the **Convert charts and shapes to images** check box on the **Resource Management** tab of the output object properties.

A.5.80 PS_DOCUMENTMEDIA

Use the PS_DOCUMENTMEDIA switch to print %%DocumentMedia commands to PostScript.

Syntax

```
-PS_DOCUMENTMEDIA=INCLUDE_FORM
```

There is no default value for this switch. This switch may be used with or without the `<INCLUDE_FORM>` argument.

If you specify `INCLUDE_FORM` as the argument, the engine prints the PostScript media name in the preprinted form name field of the Document Structuring Convention (DSC) command. This is the last parameter, which appears in parentheses in the output.

If you do not specify the argument, the engine does not print anything in the preprinted form name field.

A.6 Switches Q-T

A.6.1 QUERYPATH

The QUERYPATH switch specifies the query path to use for the create, update, and drop processes.

Syntax

```
-QUERYPATH=<query file path>
```

Argument	Required	Supported values
`<query file path>`	Yes	The path to use for the create, update, and drop processes. When entering the path, do not include a backlash at the end.

Example

```
-QUERYPATH=C:\Temp
```

A.6.2 READABILITY

Use the READABILITY switch to access Flesch statistics in a report file. Flesch statistics are derived from a Flesch rating, which measures the readability of a document. Using the READABILITY switch allows you to measure the readability of the final output for each customer (including variable content, and so on).

Syntax

```
-READABILITY=<text to include>
```

Arguments	Required	Supported values
`<text to include>`	Yes	• NONE—No readability information is generated in the report. • BODY—A readability report for each customer is generated, but text in headers and deleteIDs is not included. • HEADER—Body text and header text are included in the readability report. • FOOTER—Body text and deleteID text are included in the readability report. • ALL—All text is used to generate the report. This switch has no default value.

Example

```
-READABILITY=BODY
```

A.6.3 REALTIME

Deprecated: This switch has been deprecated.

Use the REALTIME switch to set the engine into transaction (continuous operation) mode for on-demand production. This switch has been deprecated in versions 7.0 and later. Use ONDEMAND instead.

Syntax

```
-REALTIME
```

This switch does not have any arguments.

A.6.4 REALTIME_CLOSE_OUTPUTS_ON_SHUTDOWN_ONLY

Deprecated: This switch has been deprecated.

Use the REALTIME_CLOSE_OUTPUTS_ON_SHUTDOWN_ONLY switch to instruct the engine to keep output files open until the engine is commanded to shut down. The REALTIME_CLOSE_OUTPUTS_ON_SHUTDOWN_ONLY switch prevents file breaks in a multiple customer on-demand run.

This switch has been deprecated in 7.0 and up. Use the ONDEMAND_CLOSE_OUTPUTS_ON_SHUTDOWN_ONLY switch instead.

Syntax

```
-REALTIME_CLOSE_OUTPUTS_ON_SHUTDOWN_ONLY
```

This switch does not have any arguments.

A.6.5 REALTIME_LICENSE_MODE

Use this switch to specify when to check out and check in licenses for the real-time engine.

Syntax

```
-REALTIME_LICENSE_MODE=<mode>
```

Arguments	Required	Supported values
`<mode>`	Yes	• `RUN` — Check out a license when the real-time engine starts. Keep the license checked out for the entire run until the engine is closed. • `PROCESS` — Check out a license at the start of processing data and to check in the license when the data processing is complete. Using this value can potentially result in slower engine process time and increased network traffic. This mode is only recommended when processing is infrequent (for example, processing once every 15 minutes).

Example

```
-REALTIME_LICENSE_MODE=RUN
```

A.6.6 RECALCULATE_INSERT_WEIGHT_PER_CUSTOMER

Use the RECALCULATE_INSERT_WEIGHT_PER_CUSTOMER switch when your insert messages have a variable weight value setting enabled. The RECALCULATE_INSERT_WEIGHT_PER_CUSTOMER switch recalculates the weight of your postage after the message is qualified.

Syntax

```
-RECALCULATE_INSERT_WEIGHT_PER_CUSTOMER
```

This switch does not have any arguments.

Supported in

6.1.023 and later

A.6.7 REDRAW_BOTTOM_CELL_BORDER_FOLLOWED_BY_FILLED_CELL

When producing output from an application that uses a table object with shaded cell rows, use the REDRAW_BOTTOM_CELL_BORDER_FOLLOWED_BY_FILLED_CELL engine switch to prevent the bottom border of the table cells preceding the shaded row from being obscured in the pdf output.

Syntax

```
-REDRAW_BOTTOM_CELL_BORDER_FOLLOWED_BY_FILLED_CELL
```

This switch does not have any arguments.

Supported in

8.0 and later

A.6.8 REFLOOKUPALWAYS

In normal engine operation, the engine performs a reference file lookup only if the key variable value changes from the previous customer. Use the REFLOOKUPALWAYS switch to perform reference file lookups for each customer, regardless of key variable value changes. Use this switch when you want to change reference file variable values in formulas.

Syntax

```
-REFLOOKUPALWAYS
```

This switch does not have any arguments.

Supported in

- 3.5.075 and later maintenance releases
- 4.0.066 and later maintenance releases
- 4.5.312 and later versions

A.6.9 REFLOW_STATIC_TABLE_CELLS_PER_5_0

For backward compatibility after you upgrade from versions 5.0 or earlier, when you produce output from an application that includes table cells with embedded text boxes, use

the REFLOW_STATIC_TABLE_CELLS_PER_5_0 engine switch to position and size table rows as they were positioned and sized in versions 5.0 and earlier.

Syntax

```
-REFLOW_STATIC_TABLE_CELLS_PER_5_0
```

This switch does not have any arguments.

Supported in

6.1.035 and later

A.6.10 REMOVE_MCXML_EMBED_SPACERS

If you generate Multi-Channel XML output in the HTML format, embedded objects can cause unexpected white space in the resulting HTML content. Use the REMOVE_MCXML_EMBED_SPACERS engine switch to remove the spacers in the HTML content that are causing the additional white space.

Syntax

```
-REMOVE_MCXML_EMBED_SPACERS
```

This switch does not have any arguments.

Supported in

8.0 and later

A.6.11 REMOVE_USER_BACKS_WITH_BLANKS

Use the REMOVE_USER_BACKS_WITH_BLANKS switch to remove user backs (blank backs) on secondary queues when the **Remove blank backs if primary queue is duplex** check box is selected on the **Inserter** tab of the output queue object properties.

Syntax

```
-REMOVE_USER_BACKS_WITH_BLANKS
```

This switch does not have any arguments.

A.6.12 REPORT

Use the REPORT switch to specify how much detail to place in a report file. This switch provides information for many objects in the output, including campaigns, documents, pages,

and variables.

Syntax

```
-REPORT=<level of detail>
```

Arguments	Required	Supported values
`<level of detail>`	Yes	• `<CUSTOMER>`—Produces summary information for each customer • `<DETAIL>`—Produces detail information for each item • `<NONE>`—Produces no report • `<SUMMARY>`—Produces a brief overall summary for the results. This is the default.

A.6.13 REPORT_RESMAN

Use the REPORT_RESMAN switch to generate a message in the message file when an image is not found in the ResManager file.

Syntax

```
-REPORT_RESMAN
```

This switch does not have any arguments.

A.6.14 REPORTFILE

Use the REPORTFILE switch to specify the name of the report file to create.

Syntax

```
-REPORTFILE=<data file>
```

The data file containing the report file is the required argument. The default value for PC and UNIX is `ExstreamReport.dat`; for z/OS, it is `DD:REPORT`.

Example

```
-REPORTFILE=MyReport.dat
```

A.6.15 RES_ONLY_OVERRIDE

Use the RES_ONLY_OVERRIDE switch to create output that includes only the resources used by your application. This switch provides the same functionality as selecting the **Used**

resources only option from the **Resource inclusion** list on the **Resource Management** tab of the output object properties.

This switch does not have any arguments.

Syntax

```
-RES_ONLY_OVERRIDE
```

A.6.16 RESET_OUT_OF_SCOPE_VARS_IN_DATA_AGG

For backward compatibility after you upgrade to version 8.6.112, use the RESET_OUT_OF_SCOPE_VARS_IN_DATA_AGG engine switch. In previous versions of Exstream Design and Production, when you produced output from an application and your control file included the AGGREGATE_DATA engine switch, the variables were not generated as expected in the final output.

In versions 8.6.112 and later, this issue was resolved but when you produce output from applications created in earlier versions, the output may not appear as expected. This engine switch reverts to the old default behavior when using the AGGREGATE_DATA engine switch which may re-introduce some incorrect behaviors.

Syntax

```
-RESET_OUT_OF_SCOPE_VARS_IN_DATA_AGG
```

This switch does not have any arguments.

Supported in

- 8.6.112 and later maintenance releases
- 9.0.107 and later maintenance releases
- 9.5.101 and later versions

A.6.17 RESMAN_PLAIN_TEXT

Use the RESMAN_PLAIN_TEXT engine switch to specify that the format of a resource management file is plain single-byte text. To prevent errors in DBCS applications, the engine interprets resource management files in UTF-16. If you are running an SBCS application in Exstream Design and Production and you are using a resource management file generated in plain text, you must include the RESMAN_PLAIN_TEXT switch so that the engine interprets and processes the resource management file correctly.

Syntax

```
-RESMAN_PLAIN_TEXT
```

This switch does not have any arguments.

Supported in

7.0.623 and later

A.6.18 RESMANAGERFILE

Use the RESMANAGERFILE switch to identify which ResManager file to use. You must include the absolute path (unless the file is in the same directory as the engine).

Keep in mind the following considerations for the ResManager file:

- The file that the RESMANAGERFILE switch points to must contain a sequential list of images.
- Included image names must match the value of the placeholder. If the names do not match, the images are placed on the page and not at the top of the print stream because the engine did not find an image file in the list.
- Unless your application is already set up to use a ResManager file, the **Used resources only** option on the output properties is a more efficient way to achieve the same result a ResManager file provides.
- The RESMANAGERFILE file can include a list of dynamic image names to be embedded at the top of the file and referenced on pages (with fully qualified paths).
- This file can also contain a list of overlays to be embedded at the top of the file (exclusive list). If you list any page, message, or template overlays, these are the only overlays of that type that are embedded (all others are omitted).
- The image list file must be a normal new line terminated text file. When referencing an external object, the resource must be named with a complete path. The case must match exactly to the resource called for when the engine runs.

Each line of the file must follow the following format:

```
<resourceName>[,<objectType>]
```

Where:

- `resourceName` is the fully qualified path and file name of the image resource or the name of overlay object
- `objectType` is the object type that is referenced by resourceName

You can specify the following object types:

- DYNAMIC—Specifies a black-and-white; TIFF uncompressed or G4 image file name.
- TIFF_PT—Specifies a file name for a TIFF file that is dynamically imported at run time (TIFF pass-through).
- JPEG_PT—Specifies a file name for a JPEG file that is dynamically imported at run time (JPEG pass-through).
- EPS_PT—Specifies a file name for an EPS file that is dynamically imported at run time (EPS pass-through).
- AFPFSXX_PT—Specifies a file name for an AFP FS45, FS10, or FS11 file that is dynamically imported at run time (AFP FSXX pass-through).
- AFPPSEG_PT—Specifies a file name for an AFP PSEG file that is dynamically imported at run time (AFP PSEG pass-through).
- PNG_PT—Specifies a file name for a PNG file that is dynamically imported at run time (PNG pass-through).
- PDF_PT—Specifies a file name for a PDF file that is dynamically imported at run time (PDF pass-through).
- PAGE—Specifies a Design Manager page name (all images placed at top of file).
- MESSAGE—Specifies a Design Manager message name.
- TEMPLATE—Specifies a Design Manager template name.
- PAGETEMP—Specifies a Design Manager page template name.

For information about the available object types for each output driver, see *Creating Output* in the Exstream Design and Production documentation.

Syntax

```
-RESMANAGERFILE=<fileName>
```

Arguments	Required	Supported values
fileName	Yes	The fully qualified path and file name of the ResManager file. This switch has no default value.

Example

```
-RESMANAGERFILE=C:\path\to\MyResManagerFile.txt
```

A.6.19 RETAIN_EMPTY_SECTION_START

Use the RETAIN_EMPTY_SECTION_START switch when the **Remove empty rows** check box is selected on the **Table** tab of the table properties in Designer and you want to retain a row that contains an empty section that would normally be removed.

Syntax

```
-RETAIN_EMPTY_SECTION_START
```

This switch does not have any arguments.

A.6.20 RETAIN_VARIABLE_RESET_TIME

Use the RETAIN_VARIABLE_RESET_TIME switch to change the engine timing of an array variable that is used in a reference data file.

By default, when an array variable is used in a reference data file, the variable reset time is set to **None**, unless the array variable is the key variable in the reference file. This behavior can cause the values in the array to persist from customer to customer, regardless of any other timing setting. With this default behavior, the only time the array value resets is when the next lookup occurs in the data file. If any customers do not have a key value to look up, those customers start with the results generated for the previous customers. However, if customers have a key value, they get their own results. This behavior occurs only with reference file variables that are not the key variable in the data.

Using this switch will honor the reset time of an array variable instead of overriding it.

Syntax

```
-RETAIN_VARIABLE_RESET_TIME
```

This switch does not have any arguments.

A.6.21 REUSABLE_RESMANAGERFILE

Use the REUSABLE_RESMANAGER switch to specify the reusable Resource Manager (ResManager) file that is produced along with reusable resources when you use theCREATE_REUSABLE_RESMANAGERFILE switch.

To cache dynamic images between runs of the same job, use the CREATE_REUSABLE_RESMANAGERFILE switch and your existing ResManager file to create a reusable ResManager file. You can then load the image resources on your printer and run all subsequent jobs with only the REUSABLE_RESMANAGERFILE switch.

Syntax

```
-REUSABLE_RESMANAGERFILE=<fileName>
```

The only argument is the file name to be created. This switch has no default value.

A.6.22 REVERT_SECTION_TABLE_MULTICOPY_FIX

For backward compatibility after you upgrade to version 8.6, use the REVERT_SECTION_TABLE_MULTICOPY_FIX engine switch when using multiple copies of a section-driven table in your design to ensure the headers and deleteIDs are displayed correctly in each instance of the table in the output. When you produce output from an application that contains a section-driven table that uses a repeating header or deleteID, and the same table occurs more than once in the customer output, the headers and deleteIDs do not appear as expected in each occurrence of the table. This switch will be automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is XXX or earlier.

Syntax

```
-REVERT_SECTION_TABLE_MULTICOPY_FIX
```

This switch does not have any arguments.

Supported in

8.6 and later

A.6.23 REVERT_SYS_BUNDLING_COUNTS

In version 5.0 and earlier, bundling page and sheet counts were not computed correctly. In version 5.5.4 and later, the default behavior was changed to correct bundling page and sheet counts. If you upgrade from version 5.0 or earlier to version 5.5.4 and later, you can use the REVERT_SYS_BUNDLING_COUNTS switch for backward compatibility if page bundling counts are incorrect after you run the engine.

Syntax

```
-REVERT_SYS_BUNDLING_COUNTS
```

This switch does not have any arguments.

Supported in

5.5.4 and later

A.6.24 RTF_IGNORE_TRAILING_PAR

For backward compatibility after you upgrade from version 5.5 to version 6.1, to ignore a paragraph tag if there are no printable characters following the tag, use the RTF_IGNORE_

TRAILING_PAR engine switch when you produce RTF output from an application. The RTF_IGNORE_TRAILING_PAR engine switch removes extra lines in RTF output that can cause alignment issues.

Syntax

```
-RTF_IGNORE_TRAILING_PAR
```

This switch does not have any arguments.

Supported in

6.1.025 and later

A.6.25 RTL_VAR_BREAK

Use the RTL_VAR_BREAK engine switch when you produce output from an application that imports a bi-directional mixture of characters into a variable, such as mixing Roman characters and Arabic characters to create an email address, to maintain a separate text directionality from the surrounding text. When you import a bi-directional mixture of characters into a variable, by default, the text directionality of the variable content is ignored and the characters are placed in the output based on the directionality of the text that surrounds the variable.

The use of this engine switch on complex scripts is currently unsupported in version 9.5.

Syntax

```
-RTL_VAR_BREAK
```

This switch does not have any arguments.

Supported in

- 8.0.340 and later maintenance releases
- 8.6.112 and later maintenance releases
- 9.0.107 and later maintenance releases
- 9.5.101 and later versions

A.6.26 RULEANALYSISREPORT

Use the RULEANALYSISREPORT switch to generate a rule analysis report that includes the rules and parts of rules that are executed and how many times they are executed.

Syntax

```
-RULEANALYSISREPORT=<detail>,<filename>,<sort option>
```

Argument Name	Required	Supported values
`<detail>`	Yes	The amount of detail included in report: • `DETAILED`—Gives all available information in the report. This is the default • `SUMMARY`—Gives minimum information in the report.
`<filename>`	No	The file where report is to be written. The default for the file name is `RuleAnalysisReport.txt` (`DD:RULERPT` for z/OS).
`<sort option>`	No	• `UNEXECUTED`—Shows unexecuted rules first. This is the default. • `INCLUDED`—Shows executed rules first • `EXCLUDED`—Shows excluded rules first • `NAME`—Shows rules sorted by name

Example

```
-RULEANALYSISREPORT=SUMMARY,RuleAnalysisReport.txt,NAME
```

A.6.27 RUN

Use the RUN switch to specify the actions for the engine to perform.

Syntax

```
-RUN=<action>
```

Arguments	Required	Supported values
<action>	Yes	• COMPOSE — Run the engine to compose the documents in the application and create output. This is the default value. • CONTENT — Run the engine to produce an engine message file from which you can extract general information about the package file. This argument does not produce output and does not require a customer driver file. • DATA — Run the engine to analyze the data dictionary to determine whether the data file is mapped properly. • INSERTER — Run the engine to control which inserts to select. • SELECT — Run the engine to determine which campaigns and documents to select for an application. • CONTENT — Run the engine to determine the objects in the package file. • RESOURCES — Run the engine to generate resource files without composing the output files. • LOADREFFILE— Run the engine to load the reference file onto z/OS.

Example

```
-RUN=COMPOSE
```

A.6.28 RUNDATE

Use the RUNDATE switch to specify the date mm/dd/yyyy for use as the current date for the engine run.

Syntax

```
-RUNDATE=<mm/dd/yyyy>
```

The argument is the date to be used, in mm/dd/yyyy format. The default value is the current date.

Example

```
-RUNDATE=03/06/2017
```

A.6.29 RUNMODE

Use the RUNMODE switch to specify the mode in which to run the engine. This switch determines which files are processed.

Syntax

```
-RUNMODE=<engine mode>
```

Arguments	Required	Supported values
`<engine mode>`	Yes	• `PRODUCTION` —Runs the engine in production mode. This is the default value. • `LOCAL` —Runs the engine in local test file mode. This mode prints a test line at the top of each page.

Example

```
-RUNMODE=LOCAL
```

A.6.30 RUNSUMMARY

Use the RUNSUMMARY switch to create a one-line file containing a summary of the engine run. The summary includes the following information:

- Number of customers
- Number of pages
- Number of errors
- The exit status (return code)

Syntax

```
-RUNSUMMARY=<filename>
```

The argument for this switch is the file name to be used. There is no default value for this switch.

A.6.31 SECTION_DOCS_HONOR_EXCLUSION_FOR_LIVE

To exclude section documents in DLF output when a rule is set on the Live object, use the SECTION_DOCS_HONOR_EXCLUSION_FOR_LIVE switch. The **Initial Engine run only** option should be specified from the **Execution time** list on the **Rule** tab of the Live object properties. By default, the engine does not exclude section-based documents in DLF output when a rule is set on the Live object.

Syntax

```
-SECTION_DOCS_HONOR_EXCLUSION_FOR_LIVE
```

This switch does not have any arguments.

Supported in

7.0.612 and later

A.6.32 SECTIONBREAKCOUNT

Use the SECTIONBREAKCOUNT switch to specify the maximum number of records within each section of data files when record indicators define record layout. When the engine reaches the maximum number of records in a section, it creates a section break. The section break splits sections into pieces, each piece containing the maximum number of records.

The SECTIONBREAKCOUNT switch is necessary only when a single section in a data file is extremely large and the engine does not have enough memory to complete the run. Although the number of records needed to consider a single section extremely large is situation-dependent, a section should generally contain at least 500,000 records in order for you to use this switch.

Syntax

```
-SECTIONBREAKCOUNT=<MaximumNumberOfRecords>
```

The required argument is the maximum number of records that are included within each section before creating a section break. This switch has no default value.

Example

If a section in a data file contains 600,000 records and you specify `-SECTIONBREAKCOUNT=250000`, then the engine treats the section as if it contains two sections of 250,000 records and one section of 100,000 records.

A.6.33 SECUREOUTPUT

Use the SECUREOUTPUT switch to prevent sensitive customer data from appearing in test or sample output. Only data areas marked as confidential in the **Data Area Properties** dialog box are replaced by the design sample for the variable to which that data area is mapped.

Syntax

```
-SECUREOUTPUT
```

This switch does not have any arguments.

A.6.34 SET_COLUMN_BALANCE_DEVIATION

Use the SET_COLUMN_BALANCE_DEVIATION switch to force the engine, when balancing columns, to search beyond the target column height for a valid text break. This switch lets you specify a percentage deviation from the target column height. The engine searches until it finds a valid text break or reaches a point at which the next valid break would exceed the deviation.

The SET_COLUMN_BALANCE_DEVIATION affects all text that uses the balance option for columns in the current engine run.

Syntax

```
-SET_COLUMN_BALANCE_DEVIATION=<percentage deviation>
```

Arguments	Required	Supported values
`<percentage deviation>`	Yes	The percentage deviation from the target column height, expressed using an integer between 0 and 100.

A.6.35 SET_QUEUE_VARS_FROM_LAST_QUEUE

When working with multiple queues, use the SET_QUEUE_VARS_FROM_LAST_QUEUE switch to reset the queue variables to the last qualified queue. You use this switch to prevent primary queue variables from being loaded when the engine writes the End of Customer report file, and to make sure all queue variables are correct in the customer report file.

Syntax

```
-SET_QUEUE_VARS_FROM_LAST_QUEUE
```

This switch does not have any arguments.

A.6.36 SET_RTF_DEFAULT_LANGUAGE

Use the SET_RTF_DEFAULT_LANGUAGE switch to specify the language of the spelling dictionary in RTF output. The RTF spelling dictionary language is specified in the `\deflangN` or the `\deflangfe` tag. By default, Exstream Design and Production automatically populates these tags in RTF output using language defined for a customer or the default language for an entire application. You use this when you must specify a spelling dictionary language that is not available in Exstream Design and Production.

The SET_RTF_DEFAULT_LANGUAGE switch overrides any languages specified in the application and you cannot specify a different spelling dictionary for each customer.

Syntax

```
-SET_RTF_DEFAULT=<language name or ID>
```

The argument for this switch is the language name, the hexadecimal ID, or the decimal ID of the spelling dictionary that you want to use. If you use this switch, the language name or ID must match exactly the value accepted in the tag. There is no default value for this switch.

Example

All of the following samples specify the same spelling dictionary language:

```
-SET_RTF_DEFAULT_LANGUAGE=English British

-SET_RTF_DEFAULT_LANGUAGE=0x0809

-SET_RTF_DEFAULT_LANGUAGE=2057
```

A.6.37 SET_TRANSACTION_DATE

Note: You with the real-time engine. This switch is not supported with the engine in batch mode.

Use the SET_TRANSACTION_DATE switch to reset the engine time after each on-demand transaction. This helps make sure the engine date and time are continuously kept current.

Syntax

```
-SET_TRANSACTION_DATE
```

This switch does not have any arguments.

A.6.38 SETPRINTERBINCONTENTS

Use the SETPRINTERBINCONTENTS switch to select a bin contents object for a printer. This switch can be used multiple times.

Syntax

```
-SETPRINTERBINCONTENTS=<printer name>,<bin contents name>
```

This switch has no default value. You must use the following arguments:

Argument Name	Supported values
`<printer name>`	The exact name of the output object in Design Manager
`<bin contents name>`	The exact name of the bin contents object in Design Manager

Example

```
-SETPRINTERBINCONTENTS=MyPrinter,MyBinContents
```

A.6.39 SETQUEUEINSERTER

Use the SETQUEUEINSERTER switch to select the insert for an output queue at run time.

Syntax

```
-SETQUEUEINSERTER=<queue>,<inserter>
```

This switch has no default value. You must use the following arguments:

Argument	Supported values
`<queue>`	The exact name of the output object in Design Manager
`<inserter>`	The exact name of the inserter object in Design Manager

Example

```
-SETQUEUEINSERTER=MyQueue,Inserter007
```

A.6.40 SINGLECUSTOMER

Use the SINGLECUSTOMER switch to override queue file breaks and suppress banners.

Syntax

```
-SINGLECUSTOMER
```

This switch does not have any arguments.

A.6.41 SISOINSERTION

Use the SISOINSERTION switch to include a file with the SI/SO mapping information.

Syntax

```
-SISOINSERTION=<datafile>,<filename>
```

Arguments	Required	Supported values
`<datafile>`	Yes	The name of the data file object as it appears in Design Manager.
`<filename>`	Yes	The fully-qualified path and file name of the data source file.

Example

```
-SISOINSERTION=MyDataFile,C:\path\SISOInsertion.txt
```

A.6.42 SKIP_MULTIPLE_HEADER_ROWS_IN_RTF

In Exstream Design and Production versions earlier than 16.2.0, if a table had multiple header rows, only the first one appeared correctly in RTF output. Other header rows appeared in the output, but were not marked as header rows.

This issue was corrected in Exstream Design and Production version 16.2.0.

For backward compatibility after you upgrade to Exstream Design and Production 16.2.0, use the SKIP_MULTIPLE_HEADER_ROWS_IN_RTF switch. This switch will be automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 9.5.999 or earlier.

Syntax

```
-SKIP_MULTIPLE_HEADER_ROWS_IN_RTF
```

This switch does not have any arguments.

Supported in

16.2.0 and later

A.6.43 SORTDATA

Use the SORTDATA switch to specify files to use as data files in post-sort processing. This switch requires that you also use the SORTINDEX switch. You can use SORTDATA multiple times to specify all the sort data files you need.

Syntax

```
-SORTDATA=<filename>
```

The only required argument is the file name. This switch has no default value.

Example

```
-SORTDATA=C:\Path\MyPostSortDataFile
```

A.6.44 SORTID

The SORTID switch allows you to ensure that the sort index key in the second pass of an engine run is unique. When you use the SORTID switch, you specify a string that replaces the

6-byte default sort index key (time stamp in seconds). The string must be a maximum of six characters (A-Z, a-z, 0-9).

If you specify fewer characters, you receive a message and the string is padded with zeros. If you specify more characters, you receive a message and the string is truncated.

If you have an invalid character when using the SORTID switch, you receive a severe error and the engine stops processing.

Syntax

```
-SORTID=<sort index key>
```

The only required argument is the sort index key. This switch has no default value.

Example

```
-SORTID=AZAZ123
```

A.6.45 SORTINDEX

Use the SORTINDEX switch to specify the postsort index file. This switch requires that you also use the SORTDATA switch. There must be only one sort index file.

Syntax

```
-SORTINDEX=<file name>
```

The required argument for this switch is the fully-qualified file path and name to use for the postsort index. This switch has no default value.

Example

```
-SORTINDEX=C:\Path\MyPostSortIndexFile
```

A.6.46 SORTINDEXLENGTH

Use the SORTINDEXLENGTH switch to indicate that the sort rule index must be treated as a fixed length record file with the specified length, instead of as a CR/LF-terminated set of records.

Syntax

```
-SORTINDEXLENGTH=<length of record file>
```

The required argument for this switch is the length of the record file. This switch has no default value.

Example

```
-SORTINDEXLENGTH=2000
```

A.6.47 SPLIT_LARGE_PDF_IMPORTS

Use the SPLIT_LARGE_PDF_IMPORTS switch to load a large PDF import file into memory in chunks, based on the specified number of pages. This switch improves memory usage but might slow processing time.

The SPLIT_LARGE_PDF_IMPORTS switch is available for PostScript, PPML, VDX, VIPP, and VPS outputs. You can also use this switch with AFP outputs if you use the PDF_PASSTHROUGH_AS_EPS engine switch.

Syntax

```
-SPLIT_LARGE_PDF_IMPORTS=<number of pages>
```

The argument is the number of pages at which to split. This switch has no default value.

Example

```
-SPLIT_LARGE_PDF_IMPORTS=10
```

A.6.48 SPLIT_MODE

In Exstream Design and Production version 7.0.4 and later, you can specify the number of lines of text that must be placed together when a text object splits (such as text in tables, text boxes, messages, and sections and paragraphs). When you specify the number of widow or orphan lines, you specify the number of lines of text that must be together both before and after the split. If the engine cannot place the specified number of lines before and after the split, then the entire text paragraph is moved to the next available frame. For example, if you set a widow/orphan setting of two, at least two lines must be together when the engine places the text. If the text paragraph contains five lines, three lines can appear on one page and two on another.

In Exstream Design and Production versions earlier than 7.0.4, the default behavior was for widow or orphan controls to be automatically set to one line of text. This allowed any text paragraph to break across pages as long as at least one line of text could fit on both pages.

If you upgrade and text splits incorrectly, you can use the SPLIT_MODE switch for backward compatibility or to specify which behavior you want to apply to the application.

Syntax

```
-SPLIT_MODE=<behavior>
```

Arguments	Required	Supported values
<behavior>	Yes	• DEFAULT— Apply the split behavior from 7.0.4 and later to only new text or text on which the widow/orphan settings have been modified. Otherwise, apply the previous split behavior. • FORCE_NEW — Apply the split behavior from 7.0.4 and later to all text, including any legacy applications and packages. • FORCE_OLD — Apply the split behavior from versions earlier than 7.0.4 to all text, including both new text and modified widow/orphan settings, to preserve the original split behavior.

Example

```
-SPLIT_MODE=DEFAULT
```

Supported in

7.0.4 and later

A.6.49 START

Use the START switch to specify the customer number to start processing.

Syntax

```
-START=<customer number>
```

The argument is the customer number with which to start. If no value is specified, the default is 1.

Example

```
-START=2267
```

A.6.50 STARTDATE

Use this packaging switch in conjunction with the ENDDATE switch to specify the effective date range for a package file.

This packaging switch is required only if you are using theEFFECTIVE switch, and if you have set the argument for that switch to RANGE.

Syntax

```
-STARTDATE=<mm/dd/yyyy>
```

The only argument for this switch is the start date, in mm/dd/yyyy format.

Example

```
-EFFECTIVE=RANGE
-STARTDATE=12/31/2018
-ENDDATE=12/31/2020
```

Related switches

EFFECTIVE

ENDDATE

A.6.51 STDDELIVERYFILE

The STDDELIVERYFILE engine switch is used to generate an output manifest (also known as a delivery file) that contains uses a pre-defined schema to provide a comprehensive list of the output composed in an engine run. The manifest file includes information from your Exstream Design and Production application, such as output queue sources, that can be used to deliver output to customers. The output manifest is required to connect Exstream engine output to the output connectors if you are using the Communications Server for orchestration in the Exstream platform environment.

Important: Do not manually include this switch if you are using Communications Server for engine orchestration. The STDDELIVERYFILE switch is passed in automatically when the Communications Server application invokes the Exstream engine.

However, if you are not using the platform orchestration features, you can use the STDDELIVERYFILE switch to generate the manifest file and use the output delivery information for your specific requirements.

Syntax

```
-STDDELIVERYFILE=<fileName>
```

The switch requires the fully qualified file name of the output manifest as the argument.

Example

```
-STDDELIVERYFILE=C:\ExstreamDelivery\outputmanifest.xml
```

Supported in

16.2.0 and later

A.6.52 STOPERROR

Use the STOPERROR switch to specify the level of error at which processing stops and the program ends.

Syntax

```
-STOPERROR=<error level>
```

Arguments	Required	Supported values
`<error level>`	Yes	• `SEVERE`—The engine stops at severe errors. This is the default. • `ERROR`—The engine stops at errors. • `WARNING`—The engine stops at warnings.

Example

```
-STOPERROR=SEVERE
```

A.6.53 SUPPRESS_NEWLINE_ON_DDA_WRITE

Use the SUPPRESS_NEWLINE_ON_DDA_WRITE engine switch to suppress empty new line messages in the engine message file. When you run a real-time application that uses both the Java Enabler DDA routine and includes the DDAFILEMAP switch in your control file, the engine writes an empty message for each expected message in the message file. Adding the SUPPRESS_NEWLINE_ON_DDA_WRITE engine switch prevents these empty new line messages from appearing in the message file.

Syntax

```
-SUPPRESS_NEWLINE_ON_DDA_WRITE
```

This switch does not have any arguments.

Supported in

7.0.630 and later

A.6.54 TABLESPACE

When you use the AUTOCREATE switch to create an Exstream Design and Production database, you can use the TABLESPACE switch to specify the specify alternate tablespace file names that you want the Database Administrator utility to use when it creates a new Oracle or

DB2 database. The tablespace you specify in the switch is used for all of the Exstream Design and Production tables created in the new database.

Syntax

```
-TABLESPACE=<primary>,<index>,<lob>
```

Argument	Required	Notes
`primary`	Yes	The file name of the primary tablespace This argument is required only for Oracle databases. If you leave this argument empty when creating a new DB2 database, the Database Administrator utility uses the default tablespace specified in the query file.
`index`	No	The file name of the index tablespace If you leave this argument empty, the Database Administrator utility uses the default tablespace specified in the query file.
`lob`	No	The file name of the LOB tablespace If you leave this argument empty, the Database Administrator utility uses the default tablespace specified in the query file.

A.6.55 TAPEUNIT

Use the TAPEUNIT switch to specify the unit name for dynamic tape allocation.

Syntax

```
-TAPEUNIT=<MVSUnitName>
```

Arguments	Required	Supported values
`<MVSUnitName>`	Yes	The z/OS tape unit name: • *<filename>* — On z/OS, the file you specify is an ESDS VSAM file. The size of the file must accommodate row data for your largest customer. • `HIPERSPACE` — To improve performance further, you can use HIPERSPACE instead of a VSAM file name. Hiperspace TM is extra memory on z/OS. Your system administrator can specify the number of hiperspaces allowed per application and their maximum size (up to two gigabytes for each hiperspace). With this switch, very large applications can be cached in hiperspace instead of to a VSAM file.

Example

```
-TAPEUNIT=HIPERSPACE
```

A.6.56 TEMPORARY_DIRECTORY

When a package file has been compressed during the packaging process, you can use this switch in conjunction with the MAX_DECOMPRESSED_PACKAGE_BYTES switch to specify a temporary directory to store a decompressed package that is too large to be stored in memory when the Exstream engine decompresses the package file. By default, the files are deleted after use, but an optional parameter can be used to preserve the temporary files.

Note: Keep in mind that compressed package files can be decompressed only by an Exstream production engine that is installed and running in a Windows, Linux, or AIX environment.

Syntax

```
-TEMPORARY_DIRECTORY=<directoryName>,[filePurge]
```

Arguments

Argument	Required	Supported values
`directoryName`	Yes	The fully-qualified path name of the temporary directory
`filePurge`	No	• `PURGE`—Deletes the files in the temporary directory (the default setting) • `NO-PURGE`—Preserves the files in the temporary directory

Examples

```
-TEMPORARY_DIRECTORY=C:\temp\EmployeeLetters
```

```
-TEMPORARY_DIRECTORY=C:\temp\EmployeeLetters,NO-PURGE
```

Related switches

MAX_DECOMPRESSED_PACKAGE_BYTES

COMPRESSPACKAGEFILE

Supported in

- 16.4.6 and later versions

A.6.57 TEST_TIFFS

Use the TEST_TIFFS switch to control whether compressed TIFF images are verified before being added to the print stream. If this switch is in use you receive a message.

The TEST_TIFFS switch is generally used to test large batches of TIFFs from a new source, such as to check images from a new scanning system. After you know the scanning system produces valid TIFFs, you should disable this switch to increase performance.

Syntax

```
-TEST_TIFFS
```

This switch does not have any arguments.

A.6.58 TEXT_TRUNC_COMPAT_27435

For backward compatibility after you upgrade from version 3.5 or earlier, use the TEXT_TRUNC_COMPAT_27435 engine switch when you produce output from an application that includes text that exceeds the boundaries of a text box and you do not want the engine to truncate the text, although the engine reports in the message file that the text is truncated.

Syntax

```
-TEXT_TRUNC_COMPAT_27435
```

This switch does not have any arguments.

Supported in

4.0.092 and later

A.6.59 TIMESTAMP_MESSAGES

Use the TIMESTAMP_MESSAGES switch to add a timestamp prior to each issued message. The timestamp uses the format: mm/dd/yyyy hh:mm:ss. The time is always set to the time the message is issued.

Syntax

```
-TIMESTAMP_MESSAGES
```

This switch does not have any arguments.

A.6.60 TRACK_OVERRIDE_VERSION_CHECK

Use the TRACK_OVERRIDE_VERSION_CHECK switch to allow the use of a newer tracking database version than the current supported version.

Syntax

```
-TRACK_OVERRIDE_VERSION_CHECK
```

This switch does not have any arguments.

A.6.61 TRACKCOMMITINTERVAL

Use the TRACKCOMMITINTERVAL switch to specify the frequency with which the engine commits processed records.

Syntax

```
-TRACKCOMMITINTERVAL=<number of records>
```

The required argument is the number of records the engine processes before committing. The default value is `1000`.

To improve the performance of the tracking capabilities of the engine, use a higher number of records for less frequent transaction record commitments. Trying to do too much in one transaction, however, might exceed the capacity of your database server.

Example

```
-TRACKCOMMITINTERVAL=50000
```

A.6.62 TRACKDSN

Use the TRACKDSN switch to supply ODBC parameters to access the tracking database DSN. You must use TRACKIN orTRACKOUT with this switch.

Syntax

```
-TRACKDSN=<data source name>
```

The argument for this switch is the DSN.

A.6.63 TRACKFILE

Use the TRACKFILE switch to specify the name of the Campaign Management tracking file to create. This is a flat text file that can be loaded into the tracking database.

Syntax

```
-TRACKFILE=<data file name>
```

The argument is the name of the text file to be created. The default for PC and UNIX is `ExstreamTrack.dat`; for z/OS, the default is `DD:TRACK`.

Example

```
-TRACKFILE=MyCustomTrack.dat
```

A.6.64 TRACKIN

Use the TRACKIN switch to specify whether or not to use the tracking database data for campaign targeting.

Syntax

```
-TRACKIN=<option>
```

Arguments	Required	Supported values
`<option>`	Yes	• `DB`—Use the Knowledgebase for tracking. This is the default value for the TRACKIN engine switch. • `DISABLE`—Disable activity-based campaigns. • `ERROR`—The engine to issue an error for activity-based campaigns

Supported in

2.5 and later

A.6.65 TRACKOUT

Use the TRACKOUT switch to specify whether to track campaign results and whether to store the results in a flat file or database. If you are using a file, then you must specify the name of the file in the TRACKFILE switch.

Syntax

```
-TRACKOUT=<tracking option>
```

Arguments	Required	Supported values
`<tracking option>`	Yes	• `NONE`—Do not track campaign results • `FILE`— Use a text file for tracking. This is the default value for this switch. • `DB`—Use the Knowledgebase for tracking. • `FILE_DELAYED_LOAD`— Write the tracking file during an engine run. Upon completion of the engine run, the engine loads the tracking file into the tracking database.

Supported in

3.0 and later

A.6.66 TRACKPWD

Use the TRACKPWD switch to provide a password for the tracking database. This switch requires you to use theTRACKDSN andTRACKUID switches.

Syntax

```
-TRACKPWD=<DSN password>
```

The argument for this switch is the DSN password.

Supported in

8.0.301 and later

A.6.67 TRACKSCHEMA

Use the TRACKSCHEMA switch to provide the tracking database schema name. This switch requires you to use the TRACKDSN switch.

Syntax

```
-TRACKSCHEMA=<schema name>
```

The only argument is the schema name. There is no default value for this switch.

Supported in

8.0.301 and later

A.6.68 TRACKSTATUSINTERVAL

Use the TRACKSTATUSINTERVAL switch when you want to specify the number of records to be processed by the engine before a status appears. The information for this switch is displayed on the console instead of being written to the engine message file. The default setting for this switch is 0. When the value is set to 0, the console does not display messages.

Syntax

```
-TRACKSTATUSINTERVAL=<number of records>
```

The only argument for this switch is the number of records identified.

Example

```
-TRACKSTATUSINTERVAL=0
```

A.6.69 TRACKUID

Use the TRACKUID switch to provide the tracking database ODBC user ID. This switch requires you to use the TRACKDSN switch.The default value is the DSN ID.

Syntax

```
-TRACKUID=<user ID>
```

The only argument for this switch is the ODBC user ID. There is no default value.

Example

```
-TRACKUID=KBuser
```

Supported in

8.0.301 and later

A.6.70 TRIM_LINE_END_SPACES

If you design spaces in a text area at the end of a line, the spaces can grow outside the printable area. Use the TRIM_LINE_END_SPACES switch to force trim spaces present at the end of text lines and avoid overflowing text areas.

Syntax

```
-TRIM_LINE_END_SPACES
```

This switch does not have any arguments.

Supported in

6.1.015 and later

A.6.71 TRK_CAMP_AND_MSG_USED_ONLY

Use the TRK_CAMP_AND_MSG_USED_ONLY switch to produce output record types 2 and 3 for campaigns and 7 and 8 for campaign messages when the campaign or campaign message qualifies and is used. Only campaigns and campaign messages that qualify and are used are reported in the tracking file.

Syntax

```
-TRK_CAMP_AND_MSG_USED_ONLY
```

This switch does not have any arguments.

Supported in

5.0.056 and later

A.6.72 TWO_DIGIT_CENTURY_LIMIT

Use the TWO_DIGIT_CENTURY_LIMIT engine switch to change the cutoff year for interpreting how years in two-digit format (YY) should be transposed into four-digit format (YYYY). Specifically, this switch allows you to specify the correct century for a year in two-digit format.

If the value of a two-digit year is less than the value supplied by the switch, the date is considered to fall in the current century (20xx). Two-digit year values greater than or equal to the switch value fall in the previous century (19xx).

For example, if the switch value is set at 50:

- An incoming date of 2/28/07 is processed as 2007.
- An incoming date of 1/22/72 is processed as 1972.
- An incoming date of 12/18/50 is processed as 1950.

Note: In Exstream Design and Production version 16.3 and later, you can change the cutoff year using the **Century cutoff** check box and value field on the **Data Area Properties** dialog box. If you have selected the **Century cutoff** check box and specified a value, the TWO_DIGIT_CENTURY_LIMIT engine switch does *not* override that setting.

Syntax

```
-TWO_DIGIT_CENTURY_LIMIT
```

A single integer value, 00–99 (inclusive), is the only required argument.

The default cutoff value is 20.

Example

```
-TWO_DIGIT_CENTURY_LIMIT=50
```

Supported in

- 8.0.352 and later maintenance releases
- 8.6.121 and later maintenance releases
- 9.0.118 and later maintenance releases
- 9.5.304 and later maintenance releases
- 16.2.1 and later maintenance releases
- 16.3.0 and later versions

A.7 Switches S-Z

A.7.1 UPLOAD_PACKAGE_TO_CAS

Summary

Use the UPLOAD_PACKAGE_TO_CAS packaging switch to upload a package file to the common asset service (CAS).

When you use this switch, you must also use the EXSTREAMUSER and EXSTREAMPASSWORD switches to specify the Exstream user that you want to use to sign in to the design database and connect to the CAS repository. The specified user must be an OTDS user.

For more information about OTDS authentication, see *System Administration* in the Exstream Design and Production documentation.

You can use the following switches to specify additional upload settings:

- CASPACKAGENAME
- CASPACKAGEDESCRIPTION
- CASPACKAGEUPLOADTIMEOUT
- CASCREATENEWPACKAGE
- CASSTOPLEVEL

For more information about uploading a package file to the CAS from the command line, see *Preparing Applications for Production* in the Exstream Design and Production documentation.

Syntax

```
-UPLOAD_PACKAGE_TO_CAS
```

Arguments

This switch does not have any arguments.

Example

```
-UPLOAD_PACKAGE_TO_CAS
-CASPACKAGENAME=MyPackageName
-CASPACKAGEDESCRIPTION="This is a sample package file."
-CASPACKAGEUPLOADTIMEOUT=300
-CASSTOPLEVEL=ERROR
```

Supported versions

- 16.3.3 and later maintenance releases
- 16.4.0 Update 1 and later versions

A.7.2 USE_7BIT_EMAIL

In Exstream Design and Production versions earlier than 16.3.0, HTML (email) output was produced using the 7-bit ASCII character set. In version 16.3.0, the default output was changed to use the 8-bit UTF-8 character set. This character encoding change results in HTML structure and binary differences.

For backward compatibility after you upgrade to Exstream Design and Production 16.3.0, use the USE_7BIT_EMAIL switch to produce HTML (email) output using the ASCII character set.

This switch is automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 16.2.999 or earlier.

Syntax

```
-USE_7BIT_EMAIL
```

This switch does not have any arguments.

Supported in

16.3.0 and later

A.7.3 USE_DOCX_NUMPAGES_VAR

Use the USE_DOCX_NUMPAGES_VAR engine switch to force the engine to use the built-in Microsoft Word `NUMPAGES` variable so that the page numbers appear correctly in the DOCX output when using Microsoft Word print preview, or when printing using Microsoft Word. If you use the USE_DOCX_NUMPAGES_VAR engine switch, you must make sure that you select the appropriate printing options in Microsoft Word to update required fields for printing.

For more information about selecting printing options in Microsoft Word, go to the Microsoft Web site.

When you use the USE_DOCX_NUMPAGES_VAR engine switch, keep in mind the following changes in behavior:

- Page numbers do not include the expected values for the first page of your document.
- If the DOCX file includes multiple documents, then the page total value for the first document appears as the page total for all subsequent documents in the file.

Syntax

```
-USE_DOCX_NUMPAGES_VAR
```

This switch does not have any arguments.

Supported in

8.0.318 and later

A.7.4 USE_FONT_FONT_MAPPING_FOR_EMBEDDED_PDF_FONTS

Use the USE_FONT_FONT_MAPPING_FOR_EMBEDDED_PDF_FONTS engine switch to use font name mapping when you select **Embed all fonts** from the **Font Usage** list on the

Resource Management tab of the properties of the output object in Design Manager.

Syntax

```
USE_FONT_FONT_MAPPING_FOR_EMBEDDED_PDF_FONTS
```

This switch does not have any arguments.

Supported in

- 7.0.642 and later maintenance releases
- 8.0.335 and later maintenance releases
- 8.6.108 and later maintenance releases
- 9.0.102 and later maintenance releases
- 9.5.101 and later versions

A.7.5 USE_LEGACY_SUPERSCRIPT_SUBSCRIPT_IMPORT

When you upgrade from version 7.0 or earlier to version 8.0 and later, use the USE_LEGACY_SUPERSCRIPT_SUBSCRIPT_IMPORT engine switch to produce output from an imported RTF file that uses a 5.3 font size and baseline shifts. This switch will bypass a superscript and subscript enhancement created for versions 8.0 and later that causes a decrease from 5.3-point font size to 5.2-point font size.

Syntax

```
-USE_LEGACY_SUPERSCRIPT_SUBSCRIPT_IMPORT
```

This switch does not have any arguments.

Supported in

8.0 and later

A.7.6 USE_LOCAL_CONTEXT_BASED_MISSING_NODE_CHECK

When you produce output from an application that includes a schema model data file that contains two XML tags with the same name, the data associated with the second tag is missing from the output. Use the USE_LOCAL_CONTEXT_BASED_MISSING_NODE_CHECK engine switch to fix this issue.

Syntax

```
-USE_LOCAL_CONTEXT_BASED_MISSING_NODE_CHECK
```

This switch does not have any arguments.

Supported in

- 9.5.306 and later maintenance releases
- 16.2.2 and later maintenance releases
- 16.3.0 and later versions

A.7.7 USE_MCF_FOR_AFP_REF_FONTS_FOR_ANY_NAME

Use the USE_MCF_FOR_AFP_REF_FONTS_FOR_ANY_NAME switch to override IBM's naming conventions for MCF records used for reference fonts. With this switch the engine uses MCFs for fonts with any name in the name table instead of using only reference names that begin with "X".

Syntax

```
-USE_MCF_FOR_AFP_REF_FONTS_FOR_ANY_NAME
```

This switch does not have any arguments.

Supported in

6.0.015 and later

A.7.8 USE_NATIVE_ELEMENTS_FOR_ACCESSIBLE_HTML_LISTS

When you create accessible HTML5 or HTML (email) output that includes lists, the resulting HTML for those lists is structured using `<div>` tags and the ARIA roles `list` and `listitem`, by default. For example, in HTML5 output, an unordered list would be structured as:

```
<div role='list'>
       <div role='listitem' aria-level='1'><p>List item 1</p></div>
       <div role='listitem' aria-level='1'><p>List item 2</p></div>
</div>
```

You can use the USE_NATIVE_ELEMENTS_FOR_ACCESSIBLE_HTML_LISTS switch to create HTML list structures that use the native <ol> or <ul> and <li> tags. For example, in HTML5 output, an unordered list would be structured as:

```
<ul>
       <li><p>List item 1</p></li>
       <li><p>List item 2</p></li>
</ul>
```

Note that some email clients do not render the native list structure in an ideal way.

Syntax

```
-USE_NATIVE_ELEMENTS_FOR_ACCESSIBLE_HTML_LISTS
```

This switch does not have any arguments.

Supported in

16.2.0 and later

A.7.9 USE_OLD_AXIS_SCALING_FOR_SAME_MAX_MIN

For backward compatibility with versions 7.0 and earlier, use the USE_OLD_AXIS_SCALING_FOR_SAME_MAX_MIN engine switch when you produce output from an application that contains a chart and if you select **Automatic** or **Auto**, with 0 minimum from the **Range method** list on the **Y-axis** tab of the **Chart Format** dialog box in the chart properties in Designer. This switch will scale the chart axis values the same way that it did in Exstream Design and Production versions 7.0 and earlier.

Syntax

```
-USE_OLD_AXIS_SCALING_FOR_SAME_MAX_MIN
```

This switch does not have any arguments.

Supported in

- 8.0.335 and later maintenance releases
- 8.6.108 and later maintenance releases
- 9.0.103 and later maintenance releases
- 9.5.101 and later versions

A.7.10 USE_OLD_EMAIL_LISTS

In Exstream Design and Production versions prior to 16.2.0, bullets and numbers in lists were indented too far from the left margin in HTML (email) output, and the space between the number or bullet and the following text was inconsistent with the design. Additionally, only the paragraph styles of the first list items were honored. This led to visual differences between the formatting in Designer and the view in some mail clients or web browsers.

These issues were corrected in Exstream Design and Production version 16.2.0.

For backward compatibility after you upgrade to 16.2.0, use the USE_OLD_EMAIL_LISTS switch. This switch will be automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 9.5.299 or earlier.

Syntax

```
-USE_OLD_EMAIL_LISTS
```

This switch does not have any arguments.

Supported in

16.2.0 and later

A.7.11 USE_OLD_FONT_LABEL_CHART_SYMBOLS

For backward compatibility after you upgrade to version 7.0 or later, use the USE_OLD_FONT_LABEL_CHART_SYMBOLS switch to keep the original fonts for the data points in a traditional label chart.

Syntax

```
-USE_OLD_FONT_LABEL_CHART_SYMBOLS
```

This switch does not have any arguments.

Supported in

7.0.410 and later

A.7.12 USE_OLD_HTML_EMAIL_TABLE_WIDTHS

In Exstream Design and Production versions earlier than 9.5.301, table width calculations included borders and cell padding. This produced visual differences, such as text wrapping, in HTML (email) output when compared to the design.

This issue was corrected in Exstream Design and Production version 9.5.301.

For backward compatibility after you upgrade to 9.5.301, use the USE_OLD_HTML_EMAIL_TABLE_WIDTHS switch. This switch will be automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 9.5.299 or earlier.

Syntax

```
-USE_OLD_HTML_EMAIL_TABLE_WIDTHS
```

This switch does not have any arguments.

Supported in

9.5.301 and later

A.7.13 USE_OLD_HTML_NUMBERING_FONT_AFTER_BULLETS

When creating HTML output in Exstream Design and Production versions earlier than 16.2.0, the font for bullets was incorrectly applied to numbering in lists that had a mix of bullets and numbers.

This issue was corrected in Exstream Design and Production version 16.2.0.

For backward compatibility after you upgrade to Exstream Design and Production 16.2.0, use the USE_OLD_HTML_NUMBERING_FONT_AFTER_BULLETS switch. This switch will be automatically included when the ENABLE_BACKWARD_COMPAT switch is used and the version specified is 9.5.999 or earlier.

Syntax

```
-USE_OLD_HTML_NUMBERING_FONT_AFTER_BULLETS
```

This switch does not have any arguments.

Supported in

16.2.0 and later

A.7.14 USE_OLD_MERGED_CELLS_HEIGHT_CALC

For backward compatibility after you upgrade to version 8.0.306, use the USE_OLD_MERGED_CELLS_HEIGHT_CALC engine switch. When you produce output from an application that contains vertically merged cells, this engine switch can be used to automatically calculate the height of the merged cells based on the first split cell.This engine switch will be automatically included when the ENABLE_BACKWARD_COMPAT engine switch is used and the version specified is 8.03.05 or earlier.

Syntax

```
-USE_OLD_MERGED_CELLS_HEIGHT_CALC
```

This switch does not have any arguments.

Supported in

- 8.0.339 and later maintenance releases
- 8.6.110 and later maintenance releases
- 9.0.105 and later maintenance releases
- 9.5.101 and later versions

A.7.15 USE_OLD_MULTI_XML_IMAGE_SIZE

For backward compatibility with versions 7.0 and earlier, use the USE_OLD_MULTI_XML_IMAGE_SIZE engine switch when you produce Multi-Channel XML output from an application that contains images that were imported at design time. This switch causes the engine to base the size of the image on the image resolution, so that the images appear in the output design size and aspect ratio that was expected in versions 7.0 and earlier.

In Exstream Design and Production 8.0 and later, if you produce Multi-Channel XML output from an application with imported images, the images in the output appear at the size and the aspect ratio that are specified in Designer.

Syntax

```
-USE_OLD_MULTI_XML_IMAGE_SIZE
```

This switch does not have any arguments.

Supported in

- 8.0.317 and later maintenance releases
- 8.5.103 and later versions

A.7.16 USE_OLD_SORTADD_FOR_VAR_IMPORTS

For backward compatibility after you upgrade to version 8.6.106, use the USE_OLD_SORTADD_FOR_VAR_IMPORTS engine switch. After you upgrade from an earlier version of Exstream Design and Production, you may see degraded engine performance when the engine processes applications that dynamically import TIFF files. This engine switch can be used to revert to the sorting algorithm used in earlier versions of Exstream Design and Production, and will restore engine performance.

This engine switch will be automatically included when the ENABLE_BACKWARD_COMPAT engine switch is used and the version specified is 8.01.05 or earlier.

Syntax

```
-USE_OLD_SORTADD_FOR_VAR_IMPORTS
```

This switch does not have any arguments.

Supported in

- 8.0.337 and later maintenance releases
- 8.6.110 and later maintenance releases
- 9.0.104 and later maintenance releases
- 9.5.101 and later versions

A.7.17 USE_PRINT_ORDER_FOR_DLF_INITIAL_ENGINE_TABLES

In Live documents, the engine organizes section-driven tables that are set to resolve rules at initial engine timing, based on how those tables are ordered in the original design, not based on how they will be ordered in the final output. The USE_PRINT_ORDER_FOR_DLF_INITIAL_ENGINE_TABLES switch lets you order section-driven tables in Live documents based on how the tables will be ordered in the final output.

Syntax

```
-USE_PRINT_ORDER_FOR_DLF_INITIAL_ENGINE_TABLES
```

This switch does not have any arguments.

Supported in

- 8.0.336 and later maintenance releases
- 8.6.109 and later maintenance releases
- 9.0.104 and later maintenance releases
- 9.5.101 and later versions

A.7.18 USE_REPEATING_DOC_TO_INDEX_BOOKMARK

Use the USE_REPEATING_DOC_TO_INDEX_BOOKMARK engine switch when you produce output from an application that contains document-level bookmarks that are controlled by an array variable to base the bookmark array index on repeating documents instead of the qualified section index. When you produce output from an application that includes the same document more than once for the same customer, and you use this engine switch, the document-level bookmarks will appear as expected in the output.

Syntax

```
-USE_REPEATING_DOC_TO_INDEX_BOOKMARK
```

This switch does not have any arguments.

Supported in

8.0 and later

A.7.19 USE_RTF_PAGE_MARGINS

Use the USE_RTF_PAGE_MARGINS switch to allow an RTF placeholder to work like a placeholder document. RTF documents that are imported by way of placeholder variables retain the margins from the original document just as a placeholder document does.

Syntax

```
-USE_RTF_PAGE_MARGINS
```

This switch does not have any arguments.

A.7.20 USE_RTF_PARA_PROPS

Use the USE_RTF_PARA_PROPS switch to force the final RTF import paragraph to be created using the previous paragraph properties instead of the default initialization paragraph properties.

Syntax

```
-USE_RTF_PARA_PROPS
```

This switch does not have any arguments.

A.7.21 USE_SHORT_TABLE_HEADER_ ACCUMULATORS

After you upgrade from Exstream Design and Production version 8.0.312 to a later version, tables sometimes flow across pages differently in output. For example, two tables that fit on one page in output produced in version 8.0.312 and earlier are placed on two separate pages in output that is produced in later versions.

For backward compatibility after you upgrade to the Exstream Design and Production versions listed below, use the USE_SHORT_TABLE_HEADER_ACCUMULATORS engine switch to maintain the table flow behavior seen in output produced in version 8.0.312 and earlier.

Syntax

```
-USE_SHORT_TABLE_HEADER_ACCUMULATORS
```

This switch does not have any arguments.

Supported in

- 8.6.123 and later maintenance releases
- 9.0.120 and later maintenance releases
- 9.5.308 and later maintenance releases
- 16.2.4 and later maintenance releases
- 16.3.3 and later maintenance releases
- 16.4 Update 1 and later versions

A.7.22 USE_UNPREFIXED_INTERNAL_HTML_IDS

In Exstream Design and Production 16.3.0, the default format for engine-generated HTML `id` attributes was changed, so that each `id` value includes the prefix `ex-`. For example, what would be formatted as `id="textbox1"` in earlier versions would be formatted as `id="ex-textbox1"` in versions 16.3.0 and later .

For backward compatibility after you upgrade to 16.3.0, use the USE_UNPREFIXED_INTERNAL_HTML_IDS switch to exclude the `ex-` prefix on `id` attribute values. This switch is automatically included when theENABLE_BACKWARD_COMPAT switch is used and the version specified is 162999 or earlier.

Note: This switch has no impact on custom HTML id attribute values in the HTML output, only the engine-generated id values.

Syntax

```
-USE_UNPREFIXED_INTERNAL_HTML_IDS
```

This switch does not have any arguments.

Supported in

16.3.0 and later

A.7.23 USEQUEUE

Use the USEQUEUE switch to specify whether to use the queue information in the current engine run.

Syntax

```
-USEQUEUE=<option>
```

Arguments	Required	Supported values
`<line_end_command>`	Yes	• `YES`—Use the queue information. This is the default. • `NO`—Do not use the queue information.

A.7.24 VARDELLIST

Use the VARDELLIST switch to provide a list of all unused variables deleted from the data mapping. This switch is valid only if the NOVARDEL switch is not used.

Syntax

```
-VARDELLIST
```

This switch does not have any arguments.

A.7.25 VARSCOPE

Use the VARSCOPE switch to control the initialization of a variable when its run time is not known at the time the application is packaged. The VARSCOPE switch is often required when you are using data aggregation or processing Live document applications. For example, suppose a variable is being used to capture a button value that is used in a data section-driven table in a Live document. You can use the VARSCOPE switch to initialize the variable so that it is used in the section data processing.

Syntax

```
-VARSCOPE=<variable>,<initialization time>
```

Arguments	Required	Supported values
`<variable>`	Yes	The name of the variable as it appears in Design Manager.
`<initialization time>`	Yes	• `AUTOMATIC`—The variable is used as it would be normally. This is the default. • `CUSTOMER`—The variable is treated as if it were outside of a data section. • `SECTION`—The variable is initialized and used in section data processing. To initialize the variable for a specific data section, follow the Section argument with the name of the data section. Otherwise, the variable is initialized for all data sections.

Example

```
-VARSCOPE=selectedZipCode,AUTOMATIC
```

```
-VARSCOPE=customerFirstName,CUSTOMER
```

```
-VARSCOPE=selectedPlaceholderArray,SECTION,ImageSection
```

A.7.26 VARSET

Use the VARSET switch to set customer or system variables to a specific value, in order to override the parameters that are in the initialization file. By default, variable values are set after the initialization or postsort initialization files are read, however, you can include the optional `variableTiming` argument to change when the value of the variable is set. You can use this switch as many times as needed to set the value of multiple variables.

Tip: Using this switch with a `variableTiming` value of `PREINIT` is the same as using the INITVARSET switch.

Syntax

```
-VARSET=<variableName>,<variableValue>,<variableTiming>
```

Argument name	Required	Supported values
`variableName`	Yes	The name of the variable that you want to set
`variableValue`	Yes	The value of the variable
`variableTiming`	No	• `PREINIT`—Sets the value of the variable before reading the initialization files • `POSTINIT`—Sets the value of the variable after reading the initialization files. This is the default. • `CUSTOMER`—Sets the value of the variable after reading each customer's data

Examples

```
-VARSET=FirstName,George
```

```
-VARSET=SYS_CustomerQueues,PDF_Queue_Main,PREINIT
```

A.7.27 VERBOSE

Use the VERBOSE switch to specify whether to write customer numbers to the display screen (when using the design engine in a testing scenario), or to write customer numbers to the engine message file (when using the production engine in a production scenario).

The VERBOSE switch is used to monitor the progress of the engine. If you use a z/OS control file, do not place VERBOSE first or before the MESSAGELEVEL switch. If you do, the engine writes all customers to SYSOUT regardless of the argument specified in the switch.

Syntax

```
-VERBOSE=<write option>
```

Arguments	Required	Supported values

<write option>	Yes	• ON— Write customer numbers to the display in the following manner:
		  • Customers 1 to 100 by 1
		  • Customers 101 to 1000 by 10
		  • Customers 1001 to end of customers by 1000
		  This is the default.
		• AUTO or AUTOMATIC—Same as ON.
		• OFF or NO—Do not write customer numbers to the display.
		• <*number*>—Write the record every "nth" customer.

A.7.28 VERSION_SIX_LINE_BREAKS

In versions earlier than 8.0.324, if a text line filled the width of a text box, and that line of text did not contain any breaking characters (such as spaces), then the engine placed an extra line break before the line. This issue was fixed in Exstream Design and Production 8.0.324.

If you upgrade to Exstream Design and Production 8.0.324 or later, you can use the VERSION_SIX_LINE_BREAKS switch for backward compatibility to maintain the extra line breaks.

Syntax

```
-VERSION_SIX_LINE_BREAKS
```

This switch does not have any arguments.

Supported in

8.0.324 and later

A.7.29 VMCHECK

Use the VMCHECK switch to check the virtual memory status.

Syntax

```
-VMCHECK=[CUSTOMER|SECTION|DRAW]
```

This switch has no default value. This switch requires an argument. Select from the following options:

- <CUSTOMER> — Checks the virtual memory at the start of each customer
- <SECTION> — Checks the virtual memory before processing each section

- <DRAW> — Checks the virtual memory before and after freeing TableDLMemory

Example

```
-VMCHECK=CUSTOMER
```

A.7.30 WEBSERVICE_ALLOW_INSECURE

To allow users unsecured access to HTTP connections over SSL, use the WEBSERVICE_ALLOW_INSECURE engine switch.

Syntax

```
-WEBSERVICE_ALLOW_INSECURE=<access>
```

Arguments	Required	Supported values
<access>	Yes	• YES—Ignore invalid or unsigned certificates when a web service accesses a URL over an SSL connection. • NO—Disable HTTP connections over SSL.

Supported in

8.5.106 and later

A.7.31 WORD_MINIMUM_MARGINS

Use the WORD_MINIMUM_MARGINS engine switch to specify the page margins that you want to use when the following is true about your application:

- Produces DOCX output
- Contains a text object that was set to **Above** from the **Move relative to object** list on the **Dynamic Size and Placement** tab
- Has page margins that are set to less than 0.17 inches for the top and bottom of the page
- Has page margins that are set to less than 0.24 inches from the left and right of the page

If you run your application and you do not include the WORD_MINIMUM_MARGINS engine switch, then you receive a message.

Syntax

```
-WORD_MINIMUM_MARGINS=<top margin>,<bottom margin>,<left margin>,<right
margin>
```

The arguments for this switch are the numerical values of each of the minimum margins to use on the page.

Example

```
-WORD_MINIMUM_MARGINS=.17, .17, .24, .24
```

Supported in

8.0.315 and later

A.7.32 WRITEOUTPUTHEADER

When you are using EWS and SOAP Connector, you must use the WRITEOUTPUTHEADER engine switch when EWS requires information about the generated output file. When you use the WRITEOUTPUTHEADER engine switch, you must specify an output extension so the engine can write the DDA header to a separate location. The extension you specify determines the location from which EWS receives information about the generated output file.

Syntax

```
-WRITEOUTPUTHEADER=<OutputExtension>
```

The argument for this switch is the output extension. The default for this switch is `hdr`.

A.7.33 WRITE_JSON

If you are exporting sections or paragraphs to the common asset service (CAS), you can use the WRITE_JSON packaging switch during export to return the JSON data for each exported section and paragraph. The JSON data for each section and paragraph is saved separately as a .txt file in the same location that you specify for the CAS export log file. The name of each text file corresponds to the section or paragraph that it represents (`<section_name>.json.txt` or `<paragraph_name>.json.txt`).

Syntax

```
-WRITE_JSON
```

This switch does not have any arguments.

Supported in

16.4.10 and later

A.7.34 XMLMESSAGEFILE

Use the XMLMESSAGEFILE switch to specify an XML-based format for the message reporting file.

Syntax

```
-XMLMESSAGEFILE=<filename>
```

The only required argument is the file name of the XML file containing the message reporting file. There is no default value.

Example

```
-XMLMESSAGEFILE=MyXMLMessageFile.xml
```

A.7.35 XOBALLVERSIONS

During the unloading or loading process, the XOBALLVERSIONS switch unloads all of the versions of a versioned object. If you do not include this switch, Design Manager loads the latest version of the object.

If you use this switch, you cannot use theXOBKEEPAPPROVEDSTATUS switch.

Syntax

```
-XOBALLVERSIONS
```

This switch does not have any arguments.

A.7.36 XOBFOLDER

The XOBFOLDER switch specifies the folder where the target object resides. Folders are specified by a forward slash "/" . If this switch is not specified, it defaults to the root folder.

Syntax

```
-XOBFOLDER=<folder name>
```

The argument for this switch is the folder name of the folder where the target object resides.

Example

```
-XOBFOLDER=/XOB_FOLDER_2017
```

A.7.37 XOBKEEPAPPROVEDSTATUS

The XOBKEEPAPPROVEDSTATUS switch specifies that any approved objects retain their approved status when loaded.

Syntax

```
-XOBKEEPAPPROVEDSTATUS
```

This switch does not have any arguments.

A.7.38 XOBLOAD

The XOBLOAD switch specifies the name of the XOB file to load into the database.

Syntax

```
-XOBLOAD=<XOB file name>
```

The argument for this switch is the XOB file name.

Example

```
-XOBLOAD=JanEndorsementApp.xob
```

A.7.39 XOBLOADENVIRONMENTAL

The XOBLOADENVIRONMENTAL switch loads environmental objects (objects that cannot be placed into folders). If this switch is not specified, environmental objects are excluded from the load unless they are required as a dependency for another object that is being loaded.

Syntax

```
-XOBLOADENVIRONMENTAL
```

This switch does not have any arguments.

A.7.40 XOBLOADSYSTEMSETTINGS

The XOBLOADSYSTEMSETTINGS switch loads system settings if they are present in the XOB file. If the system settings are not present in the XOB file, then this switch is ignored. This switch is optional.

Syntax

```
-XOBLOADSYSTEMSETTINGS
```

This switch does not have any arguments.

A.7.41 XOBLOG

The XOBLOG switch specifies the file that is generated to report either unload activity and errors or load activity and errors.

Syntax

```
-XOBLOG=<logfilename>
```

The argument for this switch is the log file name.

Example

```
-XOBLOG=June2Log.txt
```

A.7.42 XOBPASSWORD

The XOBPASSWORD switch lets you specify a password to protect the XOB file that is created. You must also use this switch to specify the password for loading a password-protected XOB file. This switch is optional.

Syntax

```
-XOBPASSWORD=<XOB password>
```

The argument for this switch is the database password.

Example

```
-XOBPASSWORD=xxx
```

A.7.43 XOBUNLOAD

The XOBUNLOAD switch specifies the name of the XOB file to which the object will be loaded. You can unload any object type that is available in the Design Manager Library.

Syntax

```
-XOBUNLOAD=<XOB file name>
```

The argument for this switch is the XOB file name.

Example

```
-XOBUNLOAD=JanEndorsementApp.xob
```

A.7.44 XOBUNLOADAPPEND

The XOBUNLOADAPPEND switch specifies that unloaded files are appended to an existing XOB file.

Syntax

```
-XOBUNLOADAPPEND
```

This switch does not have any arguments.

A.7.45 XOBUNLOADAPPROVEDONLY

The XOBUNLOADAPPROVEDONLY switch specifies that only approved versions that were approved on or before the specified date are unloaded.

Syntax

```
-XOBUNLOADAPPROVEDONLY=<approval date>
```

The argument for this switch is the approval date in the format of YYYYMMDD.

Example

```
-XOBUNLOADAPPROVEDONLY=20110927
```

A.7.46 XOBUNLOADTARGET

The XOBUNLOADTARGET switch specifies the target object to unload from the database. If the object name is omitted, then all of the objects of that type are unloaded from the specified folder.

Syntax

```
-XOBUNLOADTARGET=<target object>
```

Arguments	Required	Supported values
<target object>	Yes	• APPLICATION • APPROVALPROCESS • APPROVALSTATE • BANNER • BARCODE • BIN • CAMPAIGN • COLORFAMILY • COLORTABLE • CONNECTOR • DATAFILE • DESIGNLAYER • DICTIONARY • {{DOCUMENT} • DLFKEY • ENCODING • FOLDER • FONT • FLOWTARGET • FORMFIELD • FUNCTION • GROUP • INSERTER • JURISDICTION • LANGUAGE • LIBRARYCOMPONENT • LIBRARYRULE • LIBRARYSEARCHKEY • LIST • LIVEACTION • LIVESETTINGS • LIVETHEME • LIVETOOLBAR • LIVEVIEW • LOCALE • MESSAGE • MESSAGETYPE • METADATA

Arguments	Required	Supported values
		• MULTIPLEUP • NAMEDCOLOR • OUTPUT • OUTPUTQUEUE • PAGE • PAGETEMPLATE • PAPERTYPE • PARAGRAPH • RECIPIENTPROFILE • SECTION • STYLE • STYLESHEET • SYSTEMSETTINGS • TAGSET • TEMPLATE • USER • VARIABLE

A.7.47 ZERO_DATE_IS_ERROR

Use the ZERO_DATE_IS_ERROR engine switch when you produce output from an application that uses a date variable that contains an invalid zero value of "00000000" or "00000001", and you want these date values to generate an error message. A date value of all zeros can sometimes be used intentionally in an application to indicate that there is no date, or an unknown date. Date values of "00000002" or greater will generate an error as expected.

Syntax

```
-ZERO_DATE_IS_ERROR
```

This switch does not have any arguments.

Supported in

- 8.0.346 and later maintenance releases
- 8.6.115 and later maintenance releases
- 9.0.110 and later maintenance releases

- 9.5.103 and later maintenance releases
- 9.5.201 and later versions

A.7.48 ZOS_USE_X15_NEWLINE_CHAR

In versions 8.0336 and later, some z/OS applications read new line characters incorrectly in the data, which can cause empty lines to be removed from the data and can cause the data to be read as a single paragraph. This issue was introduced by a software change that changed the new line character in z/OS from x15 to x25. Use the ZOS_USE_X15_NEWLINE_CHAR switch to force the z/OS engine to use x15 to represent a new line character instead of x25 for backward compatibility.

Syntax

```
-ZOS_USE_X15_NEWLINE_CHAR
```

This switch does not have any arguments.

Supported in

- 8.0.336 and later maintenance releases
- 8.6.109 and later maintenance releases
- 9.0.103 and later maintenance releases
- 9.5.101 and later versions