

IBM WebSphere Transformation Extender



Utility Commands

Version 8.1

Note

Before using this information, be sure to read the general information in "Notices" on page 77.

30 June 2006

This edition of this document applies to IBM WebSphere Transformation Extender Version 8.1; and to all subsequent releases and modifications until otherwise indicated in new editions.

To send us your comments about this document, email DTX_doc_feedback@us.ibm.com. We look forward to hearing from you.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 2006. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. Utility Commands introduction	1
Chapter 2. Troubleshooting	3
ERRORLEVEL environment variable	3
Chapter 3. Utility Commands for Type Designer	5
tanalyze utility command	5
Syntax summary for tanalyze.	5
Utility command options for tanalyze	5
Command line help for tanalyze.	7
Using the tanalyze command	8
timport utility command	8
Syntax summary for timport	8
Utility command options for timport	9
Command line help for timport	13
Using the timport command.	13
texport utility command	14
Syntax summary for texport.	14
Utility command options for texport	15
Command line help for texport.	17
Using the texport command	17
dsxmlconv Utility command.	18
Syntax summary for dsxmlconv	18
Utility command options for dsxmlconv.	18
Command line help for dsxmlconv	21
Using the dsxmlconv command	21
Chapter 4. Utility Commands for Map Designer	23
mcompile utility command	23
Syntax summary for mcompile	23
Utility command options for mcompile	24
Command line help for mcompile.	28
Using the mcompile command	28
mimport utility command	30
Syntax summary for mimport	30
Utility command options for mimport	30
Command line help for mimport	32
Using the mimport command	32
mexport utility command.	33
Syntax summary for mexport	33
Utility command options for mexport	33
Command line help for mexport	35
Using the mexport command	36
Chapter 5. Utility Commands for IFD	37
sdeploy utility command	37
Syntax summary for sdeploy	37
Utility command options for sdeploy.	38
Command line help for sdeploy	45
Using the sdeploy command	45
msdimport utility command.	48
Syntax summary for msdimport	48
Utility command options for msdimport.	48
Command line help for msdimport	51

Using the msdimport command	51
msdexport utility command	52
Syntax summary for msdexport	52
Utility command options for msdexport	52
Command line help for msdexport	55
Using the msdexport command.	55
Chapter 6. Utility Commands for map tuning	57
dtxpage utility command	57
Syntax summary for dtxpage	57
Command line help for dtxpage	58
Using the dtxpage command	58
dtxprof utility command	59
Syntax summary for dtxprof.	59
Utility command options for dtxprof	59
Command line help for dtxprof.	61
Using the dtxprof command.	61
Chapter 7. Utility Commands for map conversion	65
dsmapconv utility command	65
Syntax summary for dsmapconv	65
Utility command options for dsmapconv	66
Command line help for dsmapconv	68
Using the dsmapconv command	68
Chapter 8. dtxany2xml utility command.	69
Syntax summary for dtxany2xml	69
Command line help for dtxany2xml	69
Using the dtxany2xml command	69
dtxany2xml example	69
To create a map that can transform any input data into XML output	70
Chapter 9. Utility Commands for Resource Registry	71
Resource Registry utility command	71
Syntax summary for the resource registry	71
Utility command options for Resource Registry	71
Command line help for Resource Registry	73
Using the Resource Registry command	73
Notices	77
Programming interface information	79
Trademarks and service marks	79
Index	81

Chapter 1. Utility Commands introduction

You can use the utility commands to carry out the following functions:

- Analyzing type trees
- Importing to and exporting type trees
- Converting type trees
- Compiling maps
- Importing to and exporting map source files
- Deploying systems
- Importing to and exporting systems
- Calculating suggested memory page size and count for maps
- Analyzing map execution behavior
- Converting XML type definitions
- Creating a map that can transform any input data into XML output
- Modifying resource values in MRN and MRC files

The commands offer basic functions that are performed by the Design Studio, but allow them to be executed from the command line or within a command script without running the Design Studio applications.

There are utility commands for the Type Designer, Map Designer, and Integration Flow Designer (IFD) applications. The commands are briefly defined in the following table:

Application/utility command	Use
Type Designer	
tanalyze utility command	analyzes type trees
timport utility command	imports to type trees
texport utility command	exports type trees
dsxmlconv utility command	converts type trees that were generated in older versions (before 8.0) of WebSphere Transformation Extender
Map Designer	
mcompile utility command	compiles maps
mimport utility command	imports to maps
mexport utility command	exports maps
IFD	
sdeploy utility command	deploys systems
msdimport utility command	imports to system files
msdexport utility command	exports system files
Map Tuning	
dtxpage utility command	calculates suggested settings for memory page size and count for maps
dtxprof utility command	profiles and analyzes map execution behavior

Application/utility command	Use
Utility commands for XML	
dsmapconv utility command	automatically converts new XML type definitions in existing maps
dtxany2xml utility command	automatically produce a map that can transform any input data into XML output
Resource Registry utility command	
ResourceRegistryHelper.bat and ResourceRegistryHelper.sh	enables you to modify a resource value in the MRN (.mrn) and MRC (.mrc) files

Chapter 2. Troubleshooting

After executing a utility command, a value is returned into the *ERRORLEVEL* environment variable indicating the success or failure of the execution. The *ERRORLEVEL* variable is a Windows environment variable that contains the return code of the last DOS command you executed.

ERRORLEVEL environment variable

To retrieve the value of the *ERRORLEVEL* variable, run the following DOS command.

```
ECHO % ERRORLEVEL %
```

Utility commands return the following valid values in the *ERRORLEVEL* variable:

Return Code	Description
0	This value indicates that the utility command was successful.
1	This value indicates that the utility command was <i>not</i> successful.

The *ERRORLEVEL* variable can also be used in batch files to retrieve the value of the return code.

Chapter 3. Utility Commands for Type Designer

Type Designer utility commands permit analyzing type trees, importing to and exporting type trees and converting type trees.

The utility commands for this application are:

- "tanalyze Utility Command" - analyzes type trees
- "timport Utility Command" - imports metadata definitions to type trees
- "texport Utility Command" - exports type trees
- "dsxmlconv Utility Command" - converts type trees that were generated in older versions of WebSphere Transformation Extender.

tanalyze utility command

To analyze one or more type trees from the command line, use the `tanalyze` command. This utility command is used to analyze a type tree from the command line, outside the Type Designer GUI. The `tanalyze` utility command returns 0 if the analysis is successful, and 1 if it is not successful. It is a batch operation and can also be used for automation. See "Troubleshooting" section for details about capturing and evaluating the return code resulting from the execution of this utility command.

Syntax summary for tanalyze

`tanalyze` is the name of the utility command used to analyze type trees.

The `.mtt` file is a required field and is the name of the type tree file that needs to be analyzed. If the full path of the type tree file name is not specified, the `tanalyze` utility command will search for the type tree file specified in the current directory.

In the `tanalyze` utility command, all the options are case insensitive. The following is the syntax of the `tanalyze` utility command:

```
tanalyze <.mtt file> [-L] [-S]
          [-R [<.dbe file name/location>]
          [( -LOG [.
            | <log file name/location>]
              [-FAIL] [-VERBOSE] [-APPEND]
            | -NOLOG]
```

Utility command options for tanalyze

The following command options are available with the `tanalyze` utility command:

• -L tanalyze Option	• -FAIL tanalyze Option
• -S tanalyze Option	• -VERBOSE tanalyze Option
• -R tanalyze Option	• -APPEND tanalyze Option
• -LOG tanalyze Option	• -NOLOG tanalyze Option

If you type `tanalyze` with no options, the command line help that describes the command will display on your screen.

-L analyze option

The logic (-L) option of the `tanalyze` utility command is used to analyze a type tree for logic. If this option is specified, the type tree logic will be analyzed.

-S analyze option

The structure (-S) option of the `tanalyze` utility command is used to analyze a type tree by structure. If this option is specified, the type tree structure will be analyzed. If both (-L and -S) options of the `tanalyze` utility commands are specified, the type tree is analyzed for both logic and structure.

-R analyze option

The results file (-R) option of the `tanalyze` utility command has a parameter to specify the path and the file name for the analyze results file produced by the type tree analyzer. The specified parameter could be the name of the directory where the analyze results file will be written, the full file path specification in which a wildcard can be used to represent the analyze result file, or the full path in which the complete filename can be specified.

The default naming convention for the analyze results file is: **type_tree_name.dbe**.

If a wildcard is specified in the full path, for example, `tanalyze mytree.mtt -R C:\MyDev\AnalyzeResults\2002-02-28_*.err`, the analyze results file will be produced in the **C:\MyDev\AnalyzeResults** directory with the **2002-02-28_mytree.err** naming convention.

If the complete file name is specified, for example, `tanalyze mytree.mtt -R C:\MyDev\AnalyzeResults\MyAnalyzeResults.txt`, the **MyAnalyzeResults.txt** file will be produced in the **C:\MyDev\AnalyzeResults** directory.

If the (-R) option is not specified, the analyze results will be written to the console, and if (-R) is specified without a parameter, the analyze results will be written to the same directory as the type tree file using the default **type_tree_name.dbe** naming convention.

-LOG analyze option

The log (-LOG) option of the `tanalyze` utility command is used to enable logging. The log file produced shows the results of the type tree analysis. This option has an optional parameter to specify the file name and location for the log file. The log is produced when the `tanalyze` utility command is executed. The default file name for the `tanalyze` log will be **tanalyze_type_tree_name.log**. The default location for the `tanalyze` log will be the directory where the type tree file is located.

The file name or the location parameter, if specified with the (-LOG) option of the `tanalyze` utility command, could be the name of the directory where the `tanalyze` log file will be written, the full file path specification in which a wildcard can be used to represent the type tree file, or the full path in which the complete filename can be specified. If the `.` is specified with the (-LOG) option, the `tanalyze` log will be written to the default location (or the directory where the type tree file is located) using the default naming convention. If a wildcard is specified in the full path, for example, `tanalyze mytree.mtt -LOG C:\MyDev\AnalyzeResults\2002-02-`

28_*.results, the tanalyze log file will be produced in the C:\MyDev\AnalyzeResults directory with the 2002-02-28_mytree.results naming convention. If the complete file name is specified, for example, tanalyze mytree.mtt -LOG C:\MyDev\AnalyzeResults\mytree_results.txt, the mytree_results.txt file will be produced in the C:\MyDev\AnalyzeResults directory.

If the tanalyze utility command is unable to create the tanalyze log file, the type tree analyzing process will terminate, and the Fatal error : Could not start the process, unable to create the log file fatal error message will display on the console. If nothing is specified in the command line for -LOG (neither -LOG or -NOLOG) or if no parameter is provided for -LOG, the command line assumes the (-LOG) option is included in the command line, and writes the log messages to the console.

-FAIL tanalyze option

The fail (-FAIL) option of the tanalyze utility command is used to indicate that only the type trees that had analyze errors or warnings during the analysis should be included in the tanalyze log file. If the log failures only (-FAIL) option is selected, the tanalyze log will contain only the type trees that had errors or warnings. If the log failures only (-FAIL) option is not selected in the command line, all the analyze results will be written to the tanalyze log file.

-VERBOSE tanalyze option

The verbose (-VERBOSE) option of the tanalyze utility command is used to indicate whether or not the verbose tanalyze log should be produced. The verbose tanalyze log will contain an entry for the type tree being analyzed or attempted, the type tree name, the analyzing start date/time, the analyzing end date/time, the analyzing result, and the analyze results file name. If the Verbose (-VERBOSE) option is not selected, a concise version of the tanalyze log will be produced, which includes an entry for the type tree name and the result of the type tree analysis.

-APPEND tanalyze option

The append (-APPEND) option of the tanalyze utility command is used to indicate that the current tanalyze execution messages should be appended to the existing tanalyze log file if it already exists. If the tanalyze log -APPEND option is selected, the tanalyze log messages will be appended to the existing file if it exists. Otherwise, a new file will be created. If the -APPEND option is not selected, the tanalyze log file will be created if no log file exists, or it will be overwritten if the log file does exist.

-NOLOG tanalyze option

The disable log (-NOLOG) option of the tanalyze utility command is used to turn off logging. If the disable logging option is selected, the log file will be not produced. The optional (-FAIL, -APPEND, and -VERBOSE) options of the (-LOG) option cannot be used with the disable log (-NOLOG) option. If these options are selected with the (-NOLOG) option, the command line is invalid, and the Not a valid command line : (LOG) optional commands (-APPEND), (-FAIL), (-VERBOSE) cannot be used with disable log command (-NOLOG). error message will display on the console.

Command line help for tanalyze

There is help that describes the tanalyze utility command usage in the command line.

To view the command line help for the `tanalyze` utility command

Enter `tanalyze` at the command line prompt in the `install_dir`.

The syntax and descriptions of all the options appear.

Using the `tanalyze` command

Use the `tanalyze` utility command to analyze one or more type trees from the command line, outside the Type Tree Designer GUI. The following code shows an example of how the `tanalyze` utility command is used:

```
install_dir> tanalyze mytree.mtt
-L -S
-R C:\MyDev\AnalyzeResults\mytree.dbe
-LOG -APPEND
```

The following list displays three possible combinations of outcomes of this utility command example:

- If the **mytree.mtt** type tree file is analyzed successfully, the Type tree analysis successful. message will display on the console.
- The **mytree.dbe** analyze results file will be produced in the **C:\MyDev\AnalyzeResults** directory.
- If the **mytree.log** log file does not exist, it will be produced in the same directory as the type tree file. If the **mytree.log** log file already exists, the log messages will be appended to the existing file.

timport utility command

The `timport` utility command is used to import metadata definitions to a type tree from the command line, outside the Type Designer GUI. The `timport` utility command returns 0 if the tree import is successful and 1 if it is not successful. It is a batch operation and can also be used for automation.

See "Troubleshooting" for details about capturing and evaluating the return code resulting from the execution of this utility command.

Syntax summary for `timport`

`timport` is the name of the utility command used to import metadata definitions to type trees.

All the options in the `timport` utility command are case sensitive. The following is the syntax of the `timport` utility command:

```
timport [-IMP <IMPORTER OPTIONS>]
        [-NO]
        [-O <.mtt file name/location >]
        [( -LOG [. | <log file name/location>]
          [-FAIL] [-VERBOSE] [-APPEND]
          [-HELP] [-KEEPMTS <.mts filename>])
        | -NOLOG]
```

Utility command options for timport

The following command options are available with the timport utility command:

• -APPEND timport Option	• -HELP timport Option	• -NO timport Option
• -BYTEORDER timport Option	• -IMP timport Option	• -NOLOG timport Option
• -CHARSET timport Option	• -KEEPMTS timport Option	• -O timport Option
• -CICS timport Option	• -LANG timport Option	• -VALIDATION timport Option
• -FAIL timport Option	• -LOG timport Option	• -VERBOSE timport Option

If you type timport with no options, the command line help that describes the command will display on your screen.

-APPEND timport option

The append (- APPEND) option of the timport utility command is used to indicate that the current timport execution messages should be appended to the existing timport log file if it already exists. If the timport log -APPEND option is selected, the timport log messages will be appended to the existing file, if it already exists. If it does not exist, a new file will be created. If the -APPEND option is not selected, the timport log file will be created if no log file exists, or it will be overwritten if the log file does exist.

-BYTEORDER timport option

The byte order (-BYTEORDER) option of the timport utility command is available to use with the COPYBOOK importer option to import metadata definitions to a type tree with the COBOL Copybook importer. It is specified together with a byte order set argument. The byte order set is the convention a machine processor uses to position its lowest byte within a word to begin either from the leftmost or the rightmost position and describes the execution-time data. Valid values for the byte order set argument are NATIVE, BIGENDIAN and LITTLEENDIAN. If no byte order set argument is specified, the default value is NATIVE.

-CHARSET timport option

The character set (-CHARSET) option of the timport utility command is available to use with the COPYBOOK importer option to import metadata definitions to a type tree with the COBOL Copybook importer. It is specified together with a character set argument. The character set is the standard collection of letters, numbers and symbols that describes the execution-time data. Valid values for the character set argument are NATIVE, EBCDIC and ASCII. If no character set argument is specified, the default value is NATIVE.

-CICS timport option

The CICS (-CICS) option of the timport utility command is available to use with the COPYBOOK importer option to import metadata definitions to a type tree with the COBOL Copybook importer. It specifies to the importer that it should generate a type tree for a CICS adapter.

-FAIL timport option

The fail (-FAIL) option of the `timport` utility command is used to indicate that only the type trees that had errors during the importing should be included in the `timport` log file. If the log failures only when the (-FAIL) option is selected, the `timport` log will contain only the type trees that had failed importing. If the log failures only -FAIL option is not selected in the command line, all the imported results will be written to the `timport` log file.

-HELP timport option

The help (-HELP) option of the `timport` utility command is used to display information about the specified importer. It is used with the importer options presented in the "IMPORTER OPTIONS" .

The syntax is:

```
timport -IMP <importer option> -HELP
```

Example

An example of the syntax to use to display information about the COPYBOOK importer option is as follows:

```
timport -IMP COPYBOOK -HELP
```

-IMP timport option

The import (-IMP) option of the `timport` utility command is used to specify the importer type to use to import the metadata definitions to the type tree. The available importer types are presented in the "IMPORTER OPTIONS" .

-KEEPMTS timport option

The keep mts file import (-KEEPMTS) option of the `timport` utility command is specified together with the system file location and name. This system file is the intermediate mts file generated when the the `timport` utility command runs. The -KEEPMTS option ensures that the specified system file generated by the utility command, will remain after the command is run, instead of being deleted.

The keep mts file import (-KEEPMTS) option of the `timport` utility command is not available to be used with the MTS importer option.

-LANG timport option

The language (-LANG) option of the `timport` utility command is available to use with the XMLSCHEMA and XMLDTD importer options to import the metadata definitions to a type tree with the XML Schema or XML DTD importers. It is specified together with a language option argument. The language option argument is the language type that describes the execution-time data. Valid values for the language option argument are JAPANESE and WESTERN. If no language option argument is specified, the default value is WESTERN.

-LOG timport option

The log (-LOG) option of the `timport` utility command is used to enable logging. The log file will be produced showing the result of the imported type tree. The -LOG option has an optional parameter to specify the location or file name, or both, for the `timport` log file. The log will be produced when the `timport` utility

command is executed. The default file name for the `timport` log will be **mts_file_name.log**. The default location for the `timport` log will be the directory where the `mts` file is located.

The file name or the location parameter, if specified with the (`-LOG`) option of the `timport` utility command, could be the name of the directory where the `timport` log file will be written, the full file path specification in which a wildcard can be used to represent the `mts` file, or the full path in which the complete filename can be specified. If `.` is specified with the `-LOG` option, the `timport` log will be written to the default location (directory where the type tree file is located) using the default naming convention. If a wildcard is specified in the full path, for example, `timport mymts.mts -LOG C:\MyDev\ImportResults\2002-02-28_*.results`, the `timport` log file will be produced in the **C:\MyDev\ImportResults** directory with the **2002-02-28_mymts.results** naming convention. If the complete file name is specified, for example, `timport mymts.mts -LOG C:\MyDev\ImportResults\mymts_results.txt`, the **mymts_results.txt** will be produced in the **C:\MyDev\ImportResults** directory.

If the `timport` utility command is unable to create the `timport` log file, importing of the metadata definitions to the type tree will terminate, and the Fatal error - Could not start importing the type tree, unable to create the log file fatal error message will display on the console. If nothing is specified in the command line for the log, by default, the command line assumes the `-LOG` option will be included in the command line, and writes the log messages to the console.

-NO timport option

The no overwrite (`-NO`) option of the `timport` utility command is used to indicate that an existing imported `mts` file (**.mts**) is not to be overwritten. If the no overwrite (`-NO`) option is specified in the command line of the `timport` utility command, and the file with the name of the tree file resulting from the importing process exists, it will not be overwritten, the tree exporting will fail, and the Tree Import error - The imported `mts` file cannot be overwritten, file already exists message will be written to the `timport` log file, if enabled. If the (`-NO`) option is not specified, the file with the name of the type tree resulting from the importing process will be overwritten.

-NOLOG timport option

The disable log (`-NOLOG`) option of the `timport` utility command is used to turn off the logging function of the `timport` utility command. If the disable logging option is selected, the log file will not be produced. The optional (`-LOG`) options (`-APPEND`, and `-VERBOSE`) cannot be used with the disable log (`-NOLOG`) option. If these options are selected with the (`-NOLOG`) option, the command line will be invalid, and the Not a valid command line - (LOG) optional commands (`-APPEND`), (`-VERBOSE`) cannot be used with disable log command (`-NOLOG`). error message will display on the console.

-O timport option

The output (`-O`) option of the `timport` utility command will have a parameter to specify the location or file name, or both, for the output **.mtt** type tree file produced by importing the **.mts** file. The specified parameter could be the name of the directory where the **.mtt** file will be written, the full file path specification in which a wildcard can be used to represent the **.mtt** file, or the full path in which the complete filename can be specified. The default naming convention for the **.mtt** file is **mts_file_name.mtt**. If a wildcard is specified in the full path, for example,

`timport mymts.mts -0 C:\MyDev\ImportOutputs\2002-02-28_*.mtt`, the resulting `.mtt` file will be produced in the `C:\MyDev\ImportOutputs` directory with the `2002-02-28_mymts.mtt` naming convention. If the complete file name is specified, for example, `timport mymts.mts -0 C:\MyDev\ImportOutputs\MyMtt.mtt`, the `MyMtt.mtt` file will be produced in the `C:\MyDev\ImportOutputs` directory. If the `(-0)` is selected and if the file name is not specified, the command line is invalid, and the Not a valid command line - the file name parameter is required if command option `(-0)` is selected error message will display on the console.

-VALIDATION timport option

The validation (`-VALIDATION`) option of the `timport` utility command is available to use with the `XMLSCHEMA` and `XMLDTD` importer options to import the metadata definitions to a type tree with the XML Schema or XML DTD importers. It is specified together with a validation type option argument. The validation type option argument is the validation type that describes the execution-time data. Valid values for the validation option argument are `XERCES` and `CLASSIC`. If no validation option argument is specified, the default value is `XERCES`.

-VERBOSE timport option

The verbose (`-VERBOSE`) option of the `timport` utility command indicates if the verbose `timport` log should be produced. The verbose `timport` log will contain an entry for the `.mts` file being imported, the import start date/time, the import end date/time, and the result of the `timport` utility command. If the `-VERBOSE` option is not selected, a concise version of the `timport` log will be produced, which includes an entry for the `.mts` file name and the result of the import.

IMPORTER OPTIONS

The following available `IMPORTER OPTIONS` of the `timport` utility command can be supplied:

“`MTS timport importer option`” (Default)
| “`COPYBOOK timport importer option`” on page 13 | “`XMLSCHEMA timport importer option`” on page 13 | “`XMLDTD timport importer option`” on page 13]

These are descriptions and the command syntaxes for each of these importer options.

Each importer-specific command line must be contained within a set of brackets (`[]`). To see syntax examples, see the specific importer option documentation.

To see how to view the command line help information for additional syntax and usage details, see the “`-HELP timport Option`” .

MTS timport importer option

`MTS` is one of the `MTS` options that can be specified for `IMPORTER OPTIONS` on the `timport` utility command. It is used to import an `mts` file and is specified as follows:

```
[MTS <.mts file>
- or -
<.mts file>
```


Because MTS is the default, the MTS option is optional; only the mts file must be specified. If the MTS option is specified, the mts file has to be provided. If it is not provided, the command line is invalid, and the Not a valid command line - the file name parameter is required if command option (MTS) is selected error message will display on the console.

The **.mts** file is a required field and is the file name of the mts file that needs to be imported. If the full path of the mts file name is not specified, the timport utility command will search for the mts file in the current directory.

COPYBOOK timport importer option

COPYBOOK is one of the options that can be specified for IMPORTER OPTIONS on the timport utility command. It is used to designate the COBOL Copybook importer as the type tree importer type and is specified as follows:

```
COPYBOOK [<.cpy file> -BYTEORDER timport Option [byte order]
          -CHARSET timport Option [charset] -CICS timport Option]
```

XMLDTD timport importer option

XMLDTD is one of the options that can be specified for IMPORTER OPTIONS on the timport utility command. It is used to designate the XML DTD importer as the type tree importer type and is specified as follows:

```
XMLDTD [<.dtd file> -LANG timport Option [language]
        -VALIDATION timport Option [validation_type]]
```

The full path of the dtd file must be specified in the command line.

XMLSCHEMA timport importer option

XMLSCHEMA is one of the options that can be specified for IMPORTER OPTIONS on the timport utility command. It is used to designate the XML Schema importer as the type tree importer type and is specified as follows:

```
XMLSCHEMA [<.xsd file> -LANG timport Option [language]
           -VALIDATION timport Option [validation_type]]
```

The full path of the xsd file must be specified in the command line.

Command line help for timport

There is help that describes the timport utility command usage in the command line.

To view the command line help for the timport utility command

Enter timport at the command line prompt in the *install_dir*.

The syntax and descriptions of all the options appear.

Using the timport command

This section describes how to use the timport utility command to:

- import metadata definitions to a type tree file from the command line, outside the Type Tree Designer GUI.

- import Copybook metadata definitions to a type tree from the command line, outside the Type Tree Designer GUI.

Importing a type tree file example

In the following example, the `timport` utility command line option is used to import metadata definitions in the `mymts.mts` file to the `mymts.mtt` type tree file.

```
install_dir> timport mymts.mts
      -O C:\MyDev\ImportResults\mymts.mtt
      -LOG -APPEND
```

The following list displays three possible outcomes of this utility command example:

- If the `mymts.mts` file is imported successfully, the `timport - completed successfully.` message will display on the console.
- A `mymts.mtt` type tree will be created in the `C:\MyDev\ImportResults` directory.
- If the `mymts.log` log file does not exist, it will be produced in the same directory as the `.mts` file. If the `mymts.log` log file already exists, the log messages will be appended to the existing file.

Importing a copybook type tree example

In the following example, the `timport` utility command line option is used to import the COBOL Copybook `wrapped.cpy` metadata file to a type tree for a CICS adapter from the command line. The COBOL Copybook importer is specified to use the NATIVE byte order set, NATIVE character set, create the `wrapped.log` file, and to retain the generated intermediate `wrapped.mts` system file.

```
timport -IMP COPYBOOK
      [c:\install_dir\examples\general\copybook\wrapped.cpy
      -BYTEORDER NATIVE -CHARSET NATIVE -CICS]
      -O c:\install_dir\examples\general\copybook\wrapped.mtt
      -LOG c:\install_dir\examples\general\copybook\wrapped.log
      -KEEPMTS
      c:\install_dir\examples\general\copybook\wrapped.mts
```

texport utility command

The `texport` utility command is used to export a type tree from the command line, outside the Type Designer GUI. The `texport` utility command returns 0 if the tree export is successful and 1 if it is not successful. It is a batch operation and can also be used for automation. See "Troubleshooting" for details about capturing and evaluating the return code resulting from the execution of this utility command.

Syntax summary for texport

`texport` is the name of the utility command used to export type trees. The `.mtt` file is a required field and is the file name of the type tree file that needs to be exported. If the full path of the type tree file name is not specified, the `texport` utility command will search for the type tree file specified in the current directory.

All the options in the `texport` utility command are case insensitive. The following is the syntax of the `texport` utility command:

```
texport <.mtt file>
      [-T <type name>]
      [-NO]
```

```
[-O <.mts file name/location>]
[( -LOG [.
  | <log file name/location>]
  [-FAIL] [-VERBOSE] [-APPEND])
  | -NOLOG]
```

Utility command options for texport

The following command options are available with the texport utility command:

• -T texport Option	• -FAIL texport Option
• -O texport Option	• -APPEND texport Option
• -NO texport Option	• -VERBOSE texport Option
• -LOG texport Option	• -NOLOG texport Option

If you type texport with no options, the command line help that describes the command will display on your screen.

-T texport option

The type name (-T) option of the texport utility command will have a parameter to specify the type name that needs to be exported. The type name is case sensitive and it should be the full path of the type. If type name (-T) option is selected and no type name is specified in the command line, it is an invalid command line, and the Not a valid command line - the type name parameter is required if type name command option (-T) is selected error message will display on the console. If the specified type name is not found in the type tree specified in the command line, the texport execution will terminate and the Not a Valid type name given to export - Type name given message will be written to the log file if enabled. If the (-T) is not specified, the whole type tree from the root will be exported.

-O texport option

The output (-O) option of the texport utility command will have a parameter to specify the location or file name, or both, for the exported mts file produced by exporting the type tree. The parameter specified could be the name of the directory where the exported mts file will be written, the full file path specification in which a wildcard can be used to represent the exported mts file, or the full path in which the complete filename can be specified. The default naming convention for the exported mts file is **typetree_name.mts**. If a wildcard is specified in the full path, for example, `texport mytree.mtt -O C:\MyDev\ExportResults\2002-02-28_*.mts`, the analyze results file will be produced in the **C:\MyDev\ExportResults** directory with the **2002-02-28_mytree.mts** naming convention. If the complete file name is specified, for example, `texport mytree.mtt -O C:\MyDev\ExportResults\MyExportedMts.mts`, the **MyExportedMts.mts** mts file will be produced in the **C:\MyDev\ExportResults** directory. If the export output (-O) option is selected and if the file name is not specified, the command line is invalid, and the Not a valid command line - the file name parameter is required if export command option (-O) is selected error message will display on the console.

-NO texport option

The no overwrite (-NO) option of the texport utility command is used to indicate that an existing exported mts file (.mts) is not to be overwritten. If the no

overwrite (-NO) option is specified in the command line of the texport utility command, and the file with the name of the tree file resulting from the exporting process exists, it will not be overwritten, the tree exporting will fail, and the Tree Export error - The exported mts file cannot be overwritten, file already exists message will be written to the texport log file, if enabled. If the (-NO) option is not specified, the file with the name of the type tree resulting from the exporting process, will be overwritten.

-LOG texport option

The log (-LOG) option of the texport utility command is used to enable logging. The log file will be produced showing the result of the exported type tree. (-LOG) has an optional parameter to specify the location or file name, or both, for the texport log file. The log will be produced when the texport utility command is executed. The default file name for the texport log will be **type_tree_file_name.log**. The default location for the texport log will be the directory where the type tree file is located.

The file name or the location parameter, if specified with the (-LOG) option of the texport utility command, could be the name of the directory where the texport log file will be written, the full file path specification in which a wildcard can be used to represent the type tree file, or the full path in which the complete filename can be specified. If the . is specified with the (-LOG) option, the texport log will be written to the default location (directory where the type tree file is located) using the default naming convention. If a wildcard is specified in the full path, for example, `texport mytree.mtt -LOG C:\MyDev\ExportResults\2002-02-28_*.results`, the texport log file will be produced in the **C:\MyDev\ExportResults** directory with the **2002-02-28_mytree.results** naming convention. If the complete file name is specified, for example, `texport mytree.mtt -LOG C:\MyDev\ExportResults\mytree_results.txt`, the **mytree_results.txt** will be produced in the **C:\MyDev\ExportResults** directory.

If the texport utility command is unable to create the texport log file, the exporting type tree will terminate, and the Fatal error - Could not start exporting the type tree, unable to create the log file fatal error message will display on the console. If nothing is specified in the command line for the log, by default, the command line assumes the (-LOG) option is included in the command line, and writes the log messages to the console.

-FAIL texport option

The fail (-FAIL) option of the texport utility command is used to indicate that only the type trees that had errors during the exporting should be included in the texport log file. If log failures only (-FAIL) option is selected, the texport log will contain only the type trees that had failed exporting. If the log failures only (-FAIL) option is not selected in the command line, all the exported results will be written to the texport log file.

-APPEND texport option

The append (-APPEND) option of the texport utility command is used to indicate that the current texport execution messages should be appended to the existing texport log file if it already exists. If the texport log append (-APPEND) option is selected, the texport log messages will be appended to the existing file if it already exists. Otherwise, a new file will be created. If the (-APPEND) option is not selected, the texport log file will be created if no log file exists, or it will be overwritten if the log file does exist.

-VERBOSE texport option

The verbose (-VERBOSE) option of the texport utility command is used to indicate if the verbose texport log should be produced. The verbose texport log will contain an entry for the type tree being exported, the type tree name, the exporting start date/time, the exporting end date/time, the exported xml file name, and the result of the texport utility command. If the verbose (-VERBOSE) option is not selected, a concise version of the texport log will be produced, which includes an entry for the type tree, type tree name, and exporting result.

-NOLOG texport option

The disable log (-NOLOG) option of the texport utility command is used to turn off the texport logging capability. If the disable logging option is selected, log file will not be produced. The (-LOG) optional (-APPEND, and -VERBOSE) options cannot be used with the disable log (-NOLOG) option. If these options are selected with the (-NOLOG) option, the command line is invalid, and the Not a valid command line - (LOG) optional commands (-APPEND), (-VERBOSE) cannot be used with disable log command (-NOLOG). error message will display on the console.

Command line help for texport

There is help that describes the texport utility command usage in the command line.

To view the command line help for the texport utility command

Enter texport at the command line prompt in the *install_dir*.

The syntax and descriptions of all the options appear.

Using the texport command

The following examples show how you can use the texport command.

Example 1

Use the texport utility command to export a type tree from the command line, outside the Type Designer GUI. The following command shows an example of how it is used:

```
install_dir> texport mytree.mtt  
-O C:\MyDev\ExportResults\mytree.mts  
-LOG -APPEND
```

The following list displays three possible combinations of outcomes of this utility command example:

- If the **mytree.mtt** type tree file is exported successfully, the texport - completed successfully. message will display on the console.
- The exported **mytree.mts** xml file will be produced in the **C:\MyDev\ExportResults** directory. Because no type name is specified, the whole type tree from the root will be exported.
- If the **mytree.log** log file does not exist, it will be produced in the same directory as the type tree. If the **mytree.log** log file already exists, the log messages will be appended to the existing file.

Example 2

Use the `texport` utility command to export a type tree from the command line, outside the Type Designer GUI. The following command shows an example of how it is used:

```
install_dir> texport mytree.mtt
-T MyType
-O C:\MyDev\ExportResults\2002-02-28_mytree.mts
-NOLOG
```

The following list displays three possible combinations of outcomes of this utility command example:

- If the **MyType** type of the **mytree.mtt** type tree file is exported successfully, the `texport - completed successfully.` message will display on the console.
- The exported **2002-02-28_mytree.mts** xml file will be produced in the **C:\MyDev\ExportResults** directory. The **2002-02-28_mytree.mts** xml file will contain only the exported information of the **MyType** type.

No log file will be produced because the `(-NOLOG)` option has been selected.

dsxmlconv Utility command

The `dsxmlconv` utility command is used to convert type trees that were generated in older versions (before 8.0) of WebSphere Transformation Extender from the command line, outside the Type Designer GUI. (It is called the XML Type Tree Compatibility Utility in the GUI.) When run in batch mode, the `dsxmlconv` utility command returns 0 if the tree conversion is successful and 1 if it is not successful. It is a batch operation, which can also be used for batch automation.

The `.mtt` file is a required field and is the file name of the type tree file that needs to be converted to the current format. If the full path of the type tree file name is not specified, the `dsxmlconv` utility command will search for the type tree file specified in the current directory.

See "Troubleshooting" for details about capturing and evaluating the return code resulting from the execution of this utility command.

Syntax summary for dsxmlconv

`dsxmlconv` is the name of the utility command used to convert type trees.

The following is the syntax of the `dsxmlconv` utility command:

```
dsxmlconv <.mtt_file_name/location>
[-T DTD|SCHEMA] [-V 6.7|6.7.1|6.7.2|7.5] [-L]
[-S DTD_or_SCHEMA_file_name/location ]
[-B backup_file_name/location [-O]]
[( -LOG [.]< log_file_name/location >]
[-FAIL] [-VERBOSE] [-APPEND])
| -NOLOG]
```

Utility command options for dsxmlconv

The following command options are available with the `dsxmlconv` utility command:

- | | |
|-----------------------|-------------------------|
| • -T dsxmlconv Option | • -LOG dsxmlconv Option |
|-----------------------|-------------------------|

• -V dsxmlconv Option	• -FAIL dsxmlconv Option
• -L dsxmlconv Option	• -APPEND dsxmlconv Option
• -S dsxmlconv Option	• -VERBOSE dsxmlconv Option
• -B dsxmlconv Option	• -NOLOG dsxmlconv Option
• -O dsxmlconv Option	

If you type `dsxmlconv` with no options, the command line help that describes the command will display on your screen.

-T dsxmlconv option

The importer type (-T) option of the `dsxmlconv` utility command will have a parameter to specify the importer type used to generate the type tree (*.mtt_file_name*). The valid values are:

- DTD
- Schema (the default value)

-V dsxmlconv option

The version (-V) option of the `dsxmlconv` utility command has parameters to specify the version of the importer used to generate the type tree. The valid values are:

- 6.7
- 6.7.1
- 6.7.2
- 7.5 (default)

-L dsxmlconv option

The language (-L) option argument of the `dsxmlconv` utility command is the language type that describes the execution-time data. The valid values are:

- JAPANESE
- WESTERN

If no language option argument is specified, the default value is WESTERN.

-S dsxmlconv option

The source document file name or URL (-S) option of the `dsxmlconv` utility command will have a parameter to specify the source document used to generate the original type tree. This will either be a *.dtd* (DTD) or *.xsd* (schema) file and will represent the new XML grammar that will be used to generate the type tree.

If there is no value specified, the utility will automatically use the value that is specified in the type tree. If a value is specified, the utility will use that specified value instead of the one from the type tree.

-B dsxmlconv option

The backup file (-B) option of the dsxmlconv utility command will have a parameter to specify the name of the backup file to be created. If no option is specified, a default backup file with the same name and the .omt extension is created.

-O dsxmlconv option

The overwrite (-O) option of the dsxmlconv utility command is used if a file with the same name as that file specified in the -B option exists. If this option is not specified, and a file with the same name exists, the conversion process is aborted.

-LOG dsxmlconv option

The log (-LOG) option of the dsxmlconv utility command is used to enable logging. The log option has an optional parameter to specify the user-defined location or file name, or both, for the dsxmlconv log file.

If the user-defined location or file name, or both, is not specified, the results of the converted type tree will appear on the screen.

If the user-defined location or file name, or both, is specified, a log file will be created containing the results of the converted type tree.

The information will be produced when the dsxmlconv utility command is run.

-FAIL dsxmlconv option

The fail (-FAIL) option of the dsxmlconv utility command is used to indicate that only the type trees that had errors during the converting should be included in the dsxmlconv log file. If log failures only (-FAIL) option is selected, the dsxmlconv log will contain only the type trees that had failed converting. If the log failures only (-FAIL) option is not selected in the command line, all the converted results will be written to the dsxmlconv log file.

-APPEND dsxmlconv option

The append (-APPEND) option of the dsxmlconv utility command is used to indicate that the current dsxmlconv execution messages should be appended to the existing dsxmlconv log file if it already exists. If the dsxmlconv log append (-APPEND) option is selected, the dsxmlconv log messages will be appended to the existing file if it already exists. Otherwise, a new file will be created. If the (-APPEND) option is not selected, the dsxmlconv log file will be created if no log file exists, or it will be overwritten if the log file does exist.

-VERBOSE dsxmlconv option

The verbose (-VERBOSE) option of the dsxmlconv utility command is used to indicate if the verbose dsxmlconv log should be produced. The verbose dsxmlconv log will contain an entry for the type tree being converted, the type tree name, the conversion start date/time, the conversion end date/time, the converted xml file name, and the result of the dsxmlconv utility command. If the verbose (-VERBOSE) option is not selected, a concise version of the dsxmlconv log will be produced, which includes an entry for the type tree, type tree name, and conversion result.

-NOLOG dsxmlconv option

The disable log (-NOLOG) option of the dsxmlconv utility command is used to turn off the dsxmlconv logging capability. If the disable logging option is selected, log file will not be produced. The (-LOG) optional (-APPEND, and -VERBOSE) options cannot be used with the disable log (-NOLOG) option. If these options are selected with the (-NOLOG) option, the command line is invalid, and the Invalid argument in command line. error message will display on the console.

Command line help for dsxmlconv

There is help that describes the dsxmlconv utility command usage in the command line.

To view the command line help for the dsxmlconv utility command

Enter dsxmlconv at the command line prompt in the *install_dir*.

The syntax and descriptions of all the options appear.

Using the dsxmlconv command

The following example shows how you can use the dsxmlconv command.

dsxmlconv example

Use the dsxmlconv utility command to convert a type tree from the command line, outside the Type Designer GUI. The following command shows an example of how it is used:

```
install_dir> dsxmlconv mytree.mtt
-T DTD
-S C:\MyDev\ConversionResults\mydtd.dtd
-LOG -APPEND
```

The following list displays three possible combinations of outcomes of this utility command example:

- If the **mytree.mtt** type tree file is converted successfully, the Conversion succeeded. message will display on the console.
- The **mydtd.dtd** DTD file, specified as the input parameter and represents the new XML grammar, will be used to generate the type tree.
- If the **mytree.log** log file does not exist, it will be produced in the same directory as the type tree. If the **mytree.log** log file already exists, the log messages will be appended to the existing file.

Chapter 4. Utility Commands for Map Designer

Map Designer utility commands permit compiling maps, importing to maps, and exporting maps.

The utility commands for this application are:

- "mcompile Utility Command" - compiles maps
- "mimport Utility Command" - imports to maps
- "mexport Utility Command" - exports maps

mcompile utility command

The `mcompile` utility command permits compiling one or more maps from the command line, outside of the Map Designer GUI. The `mcompile` utility command returns 0 if the compilation is successful, and 1 if not. It is a batch operation that can also be used for automation.

See "Troubleshooting" for details about capturing and evaluating the return code resulting from running this utility command.

Syntax summary for mcompile

`mcompile` is the name of the utility command used to compile maps.

The `.mms` file is a required field and is the name of the map source file that needs to be compiled. The map source file can have any number of executable maps in it. The executable maps shall be defined as a map for which all sources and targets have the minimum amount of information specified for the actual source or target of data for that card's data. If the full path of the map source file name is not specified, the `mcompile` utility command will search for the map source file in the current directory. If the `.mms` file is not specified, the Not a valid command line - map source file name should be given for the command error message will display on the console.

In the `mcompile` utility command, all the options, except for the executable map name, are case insensitive. The following is the syntax of the `mcompile` utility command:

```
mcompile <.mms file> -L
- or -
mcompile <.mms file> -A [-K] [-E]
[-NO]
[-P <platform>]
[-O <.mmc file name/location>]
[-R <.mme file name/location>]
[( -LOG [.
| <log file name/location>]
[-FAIL] [-VERBOSE] [-APPEND])
| -NOLOG]
- or -
mcompile <.mms file>
<executable map>
[-E]
[-NO]
[-P <platform>]
```

```

[-O <.mmc file name/location>]
[-R <.mme file name/location>]
[( -LOG [.
  | <log file name/location>]
  [-FAIL] [-VERBOSE] [-APPEND])
  | -NOLOG]

```

Utility command options for mcompile

The following command options are available with the mcompile utility command:

• -L mcompile Option	• -R mcompile Option
• -A mcompile Option	• -LOG mcompile Option
• -K mcompile Option	• -FAIL mcompile Option
• -E mcompile Option	• -APPEND mcompile Option
• -P mcompile Option	• -VERBOSE mcompile Option
• -O mcompile Option	• -NOLOG mcompile Option
• -NO mcompile Option	

If you type mcompile with no options, the command line help that describes the command will display on your screen.

-L mcompile option

The list (-L) option of the mcompile utility command is used to list all executable maps that are present in the specified map source file. When this option is selected, the list of all executable maps will display in alphabetical order, each on a separate line. No other option can be selected with the (-L) option in the mcompile utility command. If any other option of the mcompile utility command is selected with the (-L) option, the command line is invalid, and the Not a valid command line - No other option is allowed to with '-L' option error message will display on the console.

-A mcompile option

The all (-A) option of the mcompile utility command is used to compile all executable maps that are present in a map source file. The executable map name should not be specified in the command line if (-A) option is selected. If the executable map name is also specified with the (-A) option, the command line is invalid, and the Not a valid command line - Executable map name should not be specified when '-A' all maps are selected error message will display on the console.

-K mcompile option

The continue (-K) option of the mcompile utility command is used to continue the compiling of the executable maps that are present in the map source file when any errors are returned by the map compiler or upon the first mcompile error. This option can be used only with the all maps (-A) option. If the continue (-K) option is specified without (-A), the command line is invalid, and the Not a valid command line - Continue compiling option cannot be specified without '-A' all maps command option. error message will display on the console. If the continue compilation (-K) option is not selected, the compilation of the map on which the

first error has occurred will be completed, and the `mcompile` will terminate. It will not continue with the rest of the executable maps that are present in the map source file. If the continue compilation (`-K`) option is selected, the map compilation will be performed for all the executable maps that are present in the map source file, regardless of whether the map compiler returns warnings or errors for any of the individual maps.

-E mcompile option

The errors (`-E`) option of the `mcompile` utility command is used to control the type of messages (warnings or errors, or both) written to the build results file for each executable map that has been compiled. If the (`-E`) option is selected, the messages written to the build results file will be limited to map compiler errors only. The warning messages will be ignored and will not be written to the build results file. If the (`-E`) option is not selected, all the error and warning messages will be written to the build results.

-P mcompile option

The parameter (`-P`) option of the `mcompile` utility command is used to indicate a parameter to specify the platform for which the executable map is to be compiled. The `mcompile` command line takes `WINDOWS` as the default platform if the (`-P`) option is not specified. The other platforms that are allowed are `AIX`, `HIMALAYA`, `HPITANIUM`, `HPNONSTOP`, `HPUX`, `LINUX`, `SOLARIS` and `ZOS`. If the (`-P`) option is specified and no platform parameter is provided, the command line is invalid, and the Not a valid command line -Platform parameter should be provided if option `-P` is selected. The valid platforms are `AIX`, `HIMALAYA`, `HPUX`, `SOLARIS`, `HPNONSTOP`, `ZOS`, `LINUX`, `HPITANIUM` and `WINDOWS` error message will display on the console.

-O mcompile option

The output (`-O`) option of the `mcompile` utility command is used to indicate a parameter to specify the location or file name, or both, for the resulting compiled map file. There are 2 different cases where (`-O`) can be used in the `mcompile` command line.

In the first case, (`-O`) is used with the compile all maps (`-A`) option. The parameter specified could be either the name of the directory where the resulting compiled map files will be written or the full file path specification in which a wildcard can be used to represent the compiled map file naming convention. The default naming convention for the compiled map file is **executable_map_name.mmc**. If a wildcard is specified in the full path as `mcompile mymaps.mms -A -O C:\MyDev\Compiledmaps\2002-02-28_*.win`, the compiled maps will be produced in the **C:\MyDev\Compiledmaps** directory with the **2002-02-28_<executable_map_name>.win** naming convention.

In the second case, (`-O`) is used without the compile all maps (`-A`) option. The parameter specified could be the name of the directory where the resulting compiled map files will be written, the full file path specification in which a wildcard can be used to represent the compiled map file, or the full path in which the complete filename can be specified. The default naming convention for the compiled map file is **executable_map_name.mmc**. If a wildcard is specified in the full path as `mcompile mymaps.mms MyExecMap -O C:\MyDev\Compiledmaps*.win`, the compiled maps will be produced in the **C:\MyDev\Compiledmaps** directory with the **<executable_map_name>.win** naming convention. If the complete file name is

specified, for example, `mcompile mymaps.mms MyExecMap -O C:\MyDev\Compiledmaps\MyExecForwin.win`, the **MyExecForWin.win** file will be produced in the **C:\MyDev\Compiledmaps** directory.

If the (-O) option is not specified, the compile map will be written to the same directory as of the map source file using the default naming convention `executable_map_name.mmc`. If the (-O) option is specified and the parameter is not provided, the command line is invalid, and the Not a valid command line - The file name for the resulting compiled map file should be provided when -O option is selected. error message will display on the console. If the file name or the path provided for the (-O) option is not valid or returns error while attempting to open or write to the file, the Compilation error - The file/path provided for the option (-O) is not valid, unable to write to the specified directory/file message will be written to the mcompile log file if it is enabled.

-NO mcompile option

The no overwrite (-NO) option of the mcompile utility command is used to indicate that an existing compiled map file (**.mmc**) is not to be overwritten. If the no overwrite (-NO) option is specified in the command line of the mcompile utility command, and a file with the same name as the resulting compiled map file exists, it will not be overwritten, the map compilation will fail, and the Compilation error - The compiled map file cannot be overwritten, file already exists message will be written to the mcompile log file, if enabled. If the (-NO) option is not specified, the file with the same name as the compiled map file resulting from the compilation process will be overwritten.

-R mcompile option

The results (-R) option of the mcompile utility command will have a parameter to specify the location or file name, or both, for the build results file produced by the map compiler for each executable map. There are 2 different cases where (-R) can be used in the mcompile utility command.

In the first case, the (-R) option is used with the compile all maps (-A) option. The parameter specified could be either the name of the directory where the build results files will be written or the full file path specification in which a wildcard can be used to represent the executable map naming convention. The default naming convention for the build results file is **executable_map_name.mme**. If a wildcard is specified in the full path as `mcompile mymaps.mms -A -R C:\MyDev\CompiledResults*.err`, the build results files will be produced in the **C:\MyDev\CompiledResults** directory with the **executable_map_name.err** naming convention.

In the second case, the (-R) option is used without the compile all maps (-A) option. The parameter specified could be the name of the directory where the build results files will be written, the full file path specification in which a wildcard can be used to represent the build result file, or the full path in which the complete filename can be specified. The default naming convention for the build results file is **executable_map_name.mme**. If a wildcard is specified in the full path as `mcompile mymaps.mms MyExecMap -R C:\MyDev\CompiledResults\2002-02-28_*.err`, the build results file will be produced in the **C:\MyDev\CompiledResults** directory with **2002-02-28_MyExecMap.err** naming convention. If the complete file name is specified as `mcompile mymaps.mms MyExecMap -R C:\MyDev\CompiledResults\MyExecForwin.txt`, the **MyExecForWin.txt** will be produced in the **C:\MyDev\CompiledResults** directory.

If the (-R) option is not specified, the build results will be written to the console. If the (-R) option is specified, but no parameter is provided, the build results will be written to the same directory as of the map source file using the default **executable_map_name.mme** naming convention. If the file name or the path provided for the (-R) option is not valid or returns error while attempting to open or write to the file, the Compilation error - The file/path provided for the option (-R) is not valid, unable to write to the specified directory/file message will be written to the mcompile log file, if it is enabled.

-LOG mcompile option

The log (-LOG) option of the mcompile utility command is used to enable logging. The log file will be produced showing the results of each executable map's compilation. (-LOG) has an optional parameter to specify the location or file name, or both, for the mcompile log file. The log will be produced when the mcompile utility command is executed. The mcompile log contains all the map compilations resulting from the current mcompile utility command on the command line. The default file name for the mcompile log will be **map_source_file_name.mcl**. The default location for the mcompile log will be the directory where the source map file is located.

The file name or the location parameter, if specified with the (-LOG) option of the mcompile utility command, could be the name of the directory where the mcompile log file will be written, the full file path specification in which a wildcard can be used to represent the map source file, or the full path in which the complete filename can be specified. The default naming convention for the mcompile log file is **map_source_file_name.mcl**. If the . is specified with the (-LOG) option, the mcompile log will be written to the default location (directory where the map source file is located) using the default naming convention. If a wildcard is specified in the full path as `mcompile mymaps.mms -A -LOG C:\MyDev\CompileResults\2002-02-28_*.build.results`, the mcompile log file will be produced in the **C:\MyDev\CompileResults** directory with the **2002-02-28_mymaps.build.results** naming convention. If the complete file name is specified as `mymaps.mms -A -LOG C:\MyDev\CompileResults\mymaps_buildresults.txt.txt`, the **mymaps_buildresults.txt** will be produced in the **C:\MyDev\CompileResults** directory.

If the mcompile utility command is unable to create the mcompile log file, the map compilation will terminate and the Fatal error - Could not start map compilation, unable to create the log file fatal error message will display on the console. If nothing is specified in the command line for the log, by default, the command line assumes the (-LOG) option is included in the command line, and writes the log messages to the console.

-FAIL mcompile option

The fail (-FAIL) option of the mcompile utility command is used to indicate that only the maps that had compilation errors or warnings during compilation should be included in the mcompile log file. If the log failures only (-FAIL) option is selected, the mcompile log will contain only the maps that had errors or warnings. If the log failures only (-FAIL) option is not selected in the command line, all the compilation results will be written to the mcompile log file.

-APPEND mcompile option

The append (-APPEND) option of the mcompile utility command is used to indicate that the current mcompile execution messages should be appended to the existing

mcompile log file if it already exists. If the mcompile log append (-APPEND) option is selected, the mcompile log messages will be appended to the existing file if it already exists. Otherwise, a new log file will be created. If the (-APPEND) option is not selected, the mcompile log file will be created if no log file exists, or it will be overwritten if the log file does exist.

-VERBOSE mcompile option

The verbose (-VERBOSE) option of the mcompile utility command is used to indicate that a verbose mcompile log should be produced. The verbose mcompile log will contain an entry for each map compilation attempted, the map name, the compile start date/time, the compile end date/time, the compilation result, and the name of the build results file. If the verbose (-VERBOSE) option is not selected, a concise version of the mcompile log will be produced, which includes an entry for each map compilation attempted, map name, and compilation result.

-NOLOG mcompile option

The disable log (-NOLOG) option of the mcompile utility command is used to turn off logging. If the disable logging option is selected, the log file will not be produced. The (-APPEND, -FAIL, and -VERBOSE) options of the (-LOG) option cannot be used with the disable log (-NOLOG) option. If these options are selected with the (-NOLOG) option, the command line is invalid, and the Not a valid command line - (LOG) optional commands (-APPEND), (-FAIL), (-VERBOSE) cannot be used with disable log command (-NOLOG). error message will display on the console.

Command line help for mcompile

There is help that describes the mcompile utility command usage in the command line.

To view the command line help for the mcompile utility command

Enter mcompile at the command line prompt in the *install_dir*.

The syntax and descriptions of all the options appear.

Using the mcompile command

The following example shows how you can use the mcompile utility command.

Using the mcompile to display all executable maps - example 1

Use the mcompile utility command to display all the executable maps on the console from the command line, outside the Map Designer GUI. The following command is an example of using the (-L) option on the mcompile utility command:

```
install_dir> mcompile mymaps.mms  
-L
```

Result:

```
Executable maps in mymaps.mms: FristMyExecMap MyExecMap NewExecMap  
Total executable maps: 3
```

The list of all executable maps for the specified map source file will display in alphabetical order on the console. The total number of executable maps will display as well.

Using mcompile to compile all executable maps - example 2

Use the mcompile utility command to compile all the executable maps on the console from the command line, outside the Map Designer GUI. The following command is an example of using the (-A) option on the mcompile utility command:

```
install_dir> mcompile mymaps.mms
-A -E -P WINDOWS
-O C:\MyDev\Compiledmaps\*.mmc
-R C:\MyDev\CompiledResults\mymaps_*.mme
-LOG C:\MyDev\CompiledResults\
    *.buildresults.txt
-FAIL -APPEND
```

The following list displays four possible combinations of outcomes of this utility command example:

- If all the maps are compiled successfully, the mcompile - completed successfully. message will display on the console.
- The **FristMyExecMap.mmc**, **MyExecMap.mmc**, **NewExecMap.mmc** compiled map files will be produced for WINDOWS in the **C:\MyDev\Compiledmaps** directory.
- The **mymaps_FristMyExecMap.mme**, **mymaps_MyExecMap.mme**, **mymaps_NewExecMap.mme** build results files will be produced in the **C:\MyDev\CompiledResults** directory.

If the **mymaps_buildresults.txt** log file does not exist, it will be produced in the **C:\MyDev\CompiledResults** directory. If the **mymaps_buildresults.txt** log file already exists, only the failed maps' log messages will be appended to the existing file.

Using mcompile to compile an executable map

Use the mcompile utility command to compile a specific executable map on the console from the command line, outside the Map Designer GUI. The following command shows an example of how it is used:

```
install_dir> mcompile MyExecMap
-E -P WINDOWS
-O C:\MyDev\Compiledmaps\MyCompiledMap.mmc
-R C:\MyDev\CompiledResults\
    MyExecMapErrorFile.err
-NOLOG
```

The following list displays four possible combinations of outcomes of this utility command example:

- If the **MyExecMap** map compiled successfully, the mcompile - completed successfully. message will display on the console.
- The **MyCompiledMap.mmc** compiled map file will be produced for WINDOWS in the **C:\MyDev\Compiledmaps** directory.
- The **MyExecMapErrorFile.err** build results file will be produced in the **C:\MyDev\CompiledResults** directory.

No log file will be produced because (-NOLOG) option has been selected.

mimport utility command

The mimport utility command is used to import an xml file to a map source file from the command line, outside the Map Designer GUI. The mimport utility command returns 0 if the map import is successful and 1 if it is not successful. It is a batch operation and can also be used for automation.

Refer to "Troubleshooting" for details about capturing and evaluating the return code resulting from the execution of this utility command.

Syntax summary for mimport

mimport is the name of the utility command used to import xml files to maps.

The **.xml** file is a required field and the file name of the xml source file. If the full path of the xml source file name is not specified, the mimport utility command will search for the xml source file in the current directory.

The following is the syntax of the mimport utility command:

```
mimport <.xml file>
  [-NO]
  [-O <.mms file name/location>]
  [(-LOG [. | <log file name/location>]
    [-FAIL] [-APPEND] [-VERBOSE])
  |-NOLOG]
```

In the mimport utility command, all the options are case insensitive. Typing mimport with no options will display the command summary on the console.

Utility command options for mimport

The following command options are available with the mimport utility command:

• --O mimport Option	• -APPEND mimport Option
• -NO mimport Option	• -VERBOSE mimport Option
• -LOG mimport Option	• -NOLOG mimport Option
• -FAIL mimport Option	

--O mimport option

The output (-O) option of the mimport utility command will have a parameter to specify the location or file name, or both, for the imported map source file produced by importing the xml file. The parameter specified could be the name of the directory where the imported map source file will be written, the full file path specification in which a wildcard can be used to represent the imported map source file, or the full path in which the complete filename can be specified. The default naming convention for the imported map source file is **xml_file_name.mms**. If a wildcard is specified in the full path, for example, `mimport myxmlfile.xml -O C:\MyDev\ImportResults\2002-02-28_*.mms`, the resulting map source file will be produced in the **C:\MyDev\ImportResults** directory with the **2002-02-28_myxmlfile.mms** naming convention. If the complete file name is specified, as in `mimport myxmlfile.xml -O C:\MyDev\ImportResults\Myxmlmaps.mms`, the **Myxmlmaps.mms** map source file will be produced in the

C:\MyDev\ImportResults directory. If the import map (-M) option is selected and if the file name is not specified, the command line is invalid, and the Not a valid command line - the file name parameter is required if import command option (-O) is selected error message will display on the console.

-NO mimport option

The no overwrite (-NO) option of the mimport utility command is used to indicate that an existing imported mms file (.mms) is not to be overwritten. If the no overwrite (-NO) option is specified in the command line of mimport , and if the file with the name of the map resulting from the imported process file exists, it will not be overwritten, the map importing will fail, and the Map Import error - The imported map source file cannot be overwritten, file already exists message will be written to the mimport log file if enabled. If the (-NO) option is not specified, the file with the name of the map resulting from the importing process will be overwritten.

-LOG mimport option

The log -LOG option of the mimport utility command is used to enable logging. The log file will be produced showing the result of the imported map source file. (-LOG) has an optional parameter to specify the location or file name, or both, for the mimport log file. The log will be produced when the mimport utility command is executed. The default file name for the mimport log will be **xml_source_file_name.log**. The default location for the mimport log will be the directory where the xml file is located.

The file name or the location parameter, if specified with the (-LOG) option of the mimport utility command, could be the name of the directory where the mimport log file will be written, the full file path specification in which a wildcard can be used to represent the xml source file, or the full path in which the complete filename can be specified. If the . is specified with the (-LOG) option, the mimport log will be written to the default location (directory where the xml file is located) using the default naming convention. If a wildcard is specified in the full path, for example, mimport myxmlfile.xml -LOG C:\MyDev\ImportResults\2002-02-28_*.results, the mimport log file will be produced in the C:\MyDev\ImportResults directory with the **2002-02-28_myxmlfile.results** naming convention. If the complete file name is specified, for example, mimport myxmlfile.xml -LOG C:\MyDev\ImportResults\myxmlimport_results.txt, the **myxmlimport_results.txt** will be produced in the C:\MyDev\ImportResults directory.

If the mimport utility command is unable to create the mimport log file, the importing of the map source file will terminate and the Fatal error - Could not start importing the map source file, unable to create the log file fatal error message will display on the console. If nothing is specified in the command line for the log, by default, the command line assumes the (-LOG) option is included in the command line, and writes the log messages to the console.

-FAIL mimport option

The fail (-FAIL) option of the mimport utility command is used to indicate that only the maps that had errors during the importing should be included in the mimport log file. If the log failures only (-FAIL) option is selected, the mimport log will contain only the maps that had failed exporting. If the log failures only (-FAIL) option is not selected in the command line, all the exported results will be written to the mimport log file.

-APPEND mimport option

The append (-APPEND) option of the mimport utility command is used to indicate that the current mimport execution messages should be appended to the existing mimport log file if it already exists. If the mimport log append (-APPEND) option is selected, the mimport log messages will be appended to the existing file if it already exists. Otherwise, a new file will be created. If the (-APPEND) option is not selected, the mimport log file will be created if no log file exists, or it will be overwritten if the log file does exist.

-VERBOSE mimport option

The verbose (-VERBOSE) option of the mimport utility command is used to indicate if the verbose mimport log should be produced. The verbose mimport log will contain an entry for the xml file being imported, the importing start date/time, the importing end date/time, the imported map source file name, and the result of the mimport utility command. If the verbose (-VERBOSE) option is not selected, a concise version of the mimport log will be produced, which includes an entry for the map source file name, and importing result.

-NOLOG mimport option

The disable log (-NOLOG) option of the mimport utility command is used to turn off the mimport logging capability. If the disable logging option is selected, log file will not be produced. The (-APPEND, and -VERBOSE) optional commands of the (-LOG) option cannot be used with the disable log (-NOLOG) option. If these options are selected with the (-NOLOG) option, the command line is invalid, and the Not a valid command line - (LOG) optional commands (-APPEND), (-VERBOSE) cannot be used with disable log command (-NOLOG). message will display on the console.

Command line help for mimport

There is help that describes the mimport utility command usage in the command line.

To view the command line help for the mimport utility command

Enter mimport at the command line prompt in the *install_dir*.

The syntax and descriptions of all the options appear.

Using the mimport command

Use the mimport utility command to import the xml file to the map source file from the command line, outside the Map Designer GUI. The following command shows an example of how it is used:

```
install_dir> mimport myxmlfile.xml  
-O C:\MyDev\ImportResults\myimportedmaps.mms  
-LOG -APPEND
```

The following list displays three possible combinations of outcomes of this utility command example:

- If the **myxmlfile.xml** xml file is imported successfully, the mimport - completed successfully. message will display on the console.

- The imported **myimportedmaps.mms** map source file will be produced in the **C:\MyDev\ImportResults** directory.
- If the **myxmlfile.log** log file does not exist, it will be produced in the same directory as the xml file. If the **myxmlfile.log** log file already exists, the log messages will be appended to the existing file.

mexport utility command

The **mexport** utility command is used to export a map source file from the command line, outside the Map Designer GUI. The **mexport** utility command returns 0 if the map export is successful and 1 if it is not successful. It is a batch operation and can also be used for automation.

See "Troubleshooting" for details about capturing and evaluating the return code resulting from running this utility command.

Syntax summary for mexport

mexport is the name of the utility command used to export map source files.

The **.mms** file is a required field and is the file name of the map source file that needs to be exported. The map source file can have any number of maps in it. All maps will be exported to the same xml file if provided. If the full path of the map source file name is not specified, the **mexport** utility command will search for the map source file in the current directory.

The following is the syntax of the **mexport** utility command:

```
mexport <.mms file> [-NO]
      [-O <.xml file name/location>]
      [( -LOG [.] <log file name/location>]
      [-FAIL] [-APPEND] [-VERBOSE])
      [-NOLOG]
```

In the **mexport** utility command, all the options are case insensitive. Typing **mexport** with no options will display the command summary on the console.

Utility command options for mexport

The following command options are available with the **mexport** utility command:

• -O mexport Option	• -APPEND mexport Option
• -NO mexport Option	• -VERBOSE mexport Option
• -LOG mexport Option	• -NOLOG mexport Option
• -FAIL mexport Option	

If you type **mexport** with no options, the command line help that describes the command will display on your screen.

-O mexport option

The output (-O) option of the **mexport** utility command will have a parameter to specify the location or file name, or both, for the exported xml file produced by

exporting the map source file. The parameter specified could be the name of the directory where the exported xml file will be written, the full file path specification in which a wildcard can be used to represent the exported xml file, or the full path in which the complete filename can be specified. The default naming convention for the exported xml file is **map_source_file_name.xml**. If a wildcard is specified in the full path, for example, `mexport mymaps.mms -O C:\MyDev\ExportResults\2002-02-28_*.xml`, the resulting xml file will be produced in the **C:\MyDev\ExportResults** directory with the **2002-02-28_mymaps.xml** naming convention. If the complete file name is specified, for example, `mexport mymaps.mms -O C:\MyDev\ExportResults\MymapsExportedfile.xml`, an xml file **MymapsExportedfile.xml** will be produced in the **C:\MyDev\ExportResults** directory. If the export output (-O) option is selected, and if the file name is not specified, the command line is invalid, and the Not a valid command line - the file name parameter is required if export command option (-O) is selected error message will display on the console.

-NO mexport option

The no output (-NO) option of the mexport utility command is used to indicate that an existing exported xml file (.xml) is not to be overwritten. If the no overwrite (-NO) option is specified in the command line of mexport, if the file with the name of the map source file resulting from the exporting process exists, it will not be overwritten, the map exporting process will fail, and the Map Export error - The exported xml file cannot be overwritten, file already exists message will be written to the mexport log file, if enabled. If the (-NO) option is not specified, the log file with the name of the map resulting from the exporting process will be overwritten.

-LOG mexport option

The log (-LOG) option of the mexport utility command is used to enable the mexport logging capability. The log file will be produced showing the result of the exported map source file. (-LOG) has an optional parameter to specify the location or file name, or both, for the mexport log file. The log will be produced when mexport utility command is executed. The default file name for the mexport log will be **map_source_file_name.log**. The default location for the mexport log will be the directory where the map source file is located.

The file name, or the location parameter, if specified with the (-LOG) option of the mexport utility command, could be the name of the directory where the mexport log file will be written, the full file path specification in which a wildcard can be used to represent the map source file, or the full path in which the complete filename can be specified. If the . is specified with the (-LOG) option, the mexport log will be written to the default location (directory where the map source file is located) using the default naming convention. If a wildcard is specified in the full path, for example, `mexport mymaps.mms -LOG C:\MyDev\ExportResults\2002-02-28_*.results` the mexport log file will be produced in the **C:\MyDev\ExportResults** directory with the **2002-02-28_mymaps.results** naming convention. If the complete file name is specified, for example, `mexport mymaps.mms -LOG C:\MyDev\ExportResults\mymaps_results.txt`, the **mymaps_results.txt** file will be produced in the **C:\MyDev\ExportResults** directory.

If the mexport utility command is unable to create the mexport log file, the exporting of the map source file will terminate and the Fatal error - Could not start exporting the map source file, unable to create the log file fatal error message will display on the console. If nothing is specified in the command

line for the log, by default, the command line assumes the (-LOG) option is included in the command line, and writes the log messages to the console.

-FAIL mexport option

The fail (-FAIL) option of the mexport utility command is used to indicate that only the maps that had errors during the exporting should be included in the mexport log file. If the log failures only (-FAIL) option is selected, the mexport log will contain only the maps that had failed exporting. If the log failures only (-FAIL) option is not selected in the command line, all the exported results will be written to the mexport log file.

-APPEND mexport option

The append (-APPEND) option of the mexport utility command is used to indicate that the current mexport execution messages should be appended to the existing mexport log file if it already exists. If the mexport log append (-APPEND) option is selected, the mexport log messages will be appended to the existing file if it already exists. Otherwise, a new file will be created. If the (-APPEND) option is not selected, the mexport log file will be created if no log file exists, or it will be overwritten if the log file does exist.

-VERBOSE mexport option

The verbose (-VERBOSE) option of the mexport utility command is used to indicate if the verbose mexport log should be produced. The verbose mexport log will contain an entry for the map source file being exported, the map source name, the exporting start date/time, the exporting end date/time, the exported xml file name, and the result of the mexport utility command. If the verbose (-VERBOSE) option is not selected, a concise version of the mexport log will be produced, which includes entry for the map source file name and exporting result.

-NOLOG mexport option

The disable log (-NOLOG) option of the mexport utility command is used to turn off the mexport logging capability. If the disable logging option is selected, the log file will not be produced. The optional (-APPEND, and -VERBOSE) options of the (-LOG) option cannot be used with the disable log (-NOLOG) option. If these options are selected with the (-NOLOG) option, the command line is invalid, and the Not a valid command line - (LOG) optional commands (-APPEND), (-VERBOSE) cannot be used with disable log command (-NOLOG). error message will display on the console.

Command line help for mexport

There is help that describes the mexport utility command usage in the command line.

To view the command line help for the mexport utility command

Enter mexport at the command line prompt in the *install_dir*.

The syntax and descriptions of all the options appear.

Using the mexport command

Use the `mexport` utility command to export the map source file from the command line, outside the Map Designer GUI. The following command shows an example of how it is used:

```
install_dir> mexport mymaps.mms  
-O C:\MyDev\ExportResults\mymaps.xml  
-LOG -APPEND
```

The following list displays three possible combinations of outcomes of this utility command example:

- If the **mymaps.mms** map source file is exported successfully, the `mexport` - completed successfully. message will display on the console.
- The exported **mymaps.xml** xml file will be produced in the **C:\MyDev\ExportResults** directory.
- If the **mymaps.log** log file does not exist, it will be produced in the same directory as the map source file. If it already exists, the log messages will be appended to the existing file.

Chapter 5. Utility Commands for IFD

The IFD utility commands permits deploying systems and importing to and exporting systems.

The utility commands for this application are:

- "sdeploy Utility Command" - deploys systems
- "msdimport Utility Command" - imports xml files into systems
- "msdexport Utility Command" - exports xml files from systems

sdeploy utility command

The sdeploy utility command permits analyzing systems, building maps in the system, generating and deploying the system to the specified server from the command line, outside of the IFD GUI. The sdeploy utility command returns 0 if the compilation is successful, and 1 if not. It is a batch operation that can also be used for automation. See "Troubleshooting" for details about capturing and evaluating the return code resulting from running this utility command.

Syntax summary for sdeploy

sdeploy is the name of the utility command used to analyze systems, build maps in the system, and generate and deploy systems.

The **.msd** file is a required field and is the name of the system definition file, located on disk, that needs to be analyzed, compiled, generated, or deployed to the specified server. The system definition file can have any number of systems in it.

If you make modifications, such as changing the number of input or output cards in your maps, you must open the **.msd** file in the Integration Flow Designer and save the changes in the file there to synchronize the changes before you can deploy the system using the sdeploy utility command.

If the full path of the **.msd** file name is not specified, the sdeploy command will search for the **.msd** file in the current directory. Because the sdeploy command expects the map source definition (**.msd**) file to be specified after it, if the **.msd** file is not specified, the following error message will display on the console:

```
"Invalid or empty map source file. Cannot open file path\command_parameters"
```

The following is the syntax of the sdeploy utility command:

```
sdeploy <.msd file>
  -L
  - or -
sdeploy <.msd file>
  -A [-K] [-NO]
    <ANALYZE OPTIONS | BUILDMAPS OPTIONS
    | GENERATE OPTIONS>
  [(-LOG [. | <log file name/location>]
    [-FAIL] [-VERBOSE] [-APPEND])
  | -NOLOG]
  - or -
sdeploy <.msd file> <system> [-NO]
  <ANALYZE OPTIONS | BUILDMAPS OPTIONS
```

```

|GENERATE OPTIONS| DEPLOY OPTIONS>
[(-LOG [. | <log file name/location>]
  [-FAIL] [-VERBOSE] [-APPEND])
| -NOLOG]

```

ANALYZE options

```

-ANALYZE [-EMODE ES | CS] [-R
  <MSEAnlz.txt file name/location>]

```

BUILDMAPS OPTIONS

```

-BUILDMAPS [-P <platform>]
  [-O <file location>]
  | -A [-R
  <MSEBlds.txt
  file name/location>]

```

GENERATE OPTIONS

```

-GENERATE [-EMODE ES | CS]
  [-P <platform>]
  [-O <.msl file name/location>]

```

DEPLOY OPTIONS

```

-DEPLOY <script name> [-EMODE ES | CS]
  [<-SERVER <server name>
  | LOCAL >
  | <-USER <userid> -PW <password> -IP <IP address> >]
  [-R <MSEDeploy.txt file name/location>]

```

In the sdeploy utility command, the system name, script name, server name, if provided, userid and password options, are case sensitive.

Typing sdeploy with no options will display the command summary on the console.

Utility command options for sdeploy

The following command options are available with the sdeploy utility command:

• -L sdeploy Option	• -P sdeploy Option
• -A sdeploy Option	• -O sdeploy Option
• -K sdeploy Option	• -NO sdeploy Option
• -EMODE sdeploy Option	• -R sdeploy Option
• -ANALYZE sdeploy Option	• -FAIL sdeploy Option
• -BUILDMAPS sdeploy Option	• -APPEND sdeploy Option
• -GENERATE sdeploy Option	• -VERBOSE sdeploy Option
• -DEPLOY sdeploy Option	• -LOG sdeploy Option
• -SERVER sdeploy Option	• -NOLOG sdeploy Option
• -USER -PW -IP -sdeploy Option	

If you type sdeploy with no options, the command line help that describes the command will display on your screen.

-L sdeploy option

The list (-L) option of the sdeploy utility command is used to list all systems that are present in the specified msd file. When this option is selected, the list of all systems will display in alphabetical order, each on a separate line. No other option can be selected with the (-L) option in the sdeploy utility command. If any other option of the sdeploy utility command is selected with the (-L) option, the command line is invalid, and the Not a valid command line - No other option is allowed to with '-L' option error message will display on the console.

-A sdeploy option

The all (-A) option of the sdeploy utility command is used to analyze all systems, compile all maps that are present in all systems and to generate msl files for all systems present in the msd file. The system name should not be specified in the command line if (-A) option is selected. If the system name is also specified with the (-A) option, the command line is invalid, and the Not a valid command line - System name should not be specified when '-A' process all systems are selected error message will display on the console.

-O command line option cannot be used with the -A command line option.

-K sdeploy option

The continue (-K) option of the sdeploy utility command is used to indicate whether to continue the specified process for all systems even though an error is found for a system. This option can be used only with the all maps (-A) option. If the continue (-K) option is specified without the (-A) option, the command line is invalid, and the Not a valid command line : Continue compilation option (-K) can only be used along with compile all option (-A) error message will display on the console. If the continue compilation (-K) option is not selected, the processing of the systems will not be completed and the sdeploy will terminate. It will not continue with the rest of the systems that are present in the msd file. If the continue compilation (-K) option is selected, the processing will be performed for all the systems that are present in the msd file, regardless of any errors found during the processing for any of the individual systems.

-EMODE sdeploy option

The specify mode (-EMODE) option of the sdeploy utility command is used to specify the mode in which the operations need to be performed. The available options for (-EMODE) are ES (Eventserver mode) and CS (Commandserver mode). If the (-EMODE) option is not specified, the ES (Eventserver) mode will be the default. If the (-EMODE) option is specified and the (ES | CS) options are not specified, or if both are specified, the command line is invalid, and the Not a valid command line - ES(Eventserver) or CS(Commandserver) modes has to be specified when (-EMODE) option is selected. error message will display on the console. If the (-EMODE) option is specified with the (-BUILDMAPS) option, the command line is invalid, and the Not a valid command line - (-EMODE) option cannot be specified along with (-BUILDMAPS) command option. error message will display on the console.

-ANALYZE sdeploy option

The analyze (-ANALYZE) option of the sdeploy utility command is used to analyze the systems present in the msd file. If the (-ANALYZE) option is specified with the (-A) option, it will process all systems' options. It is specifying that all the systems

need to be analyzed, and the results files will be generated, one for each system that has been analyzed. If the (-ANALYZE) option is specified with a system name only, that system will be analyzed.

For ANALYZE OPTIONS, the following information can be supplied:

```
-ANALYZE [-EMODE ES | CS]
          [-R <MSEAnlz.txt file name/location>]
```

-BUILDMAPS sdeploy option

The build maps (-BUILDMAPS) option of the sdeploy utility command is used to build all the maps that are in the systems present in the msd file. If the (-BUILDMAPS) option is specified with the (-A) option, process all the systems' options. It is specifying that all the maps in all systems need to be compiled with the platform specified in the command line, the output .mmc files will be produced in the specified directory, and the results files will be generated one for each system that has been called. If the (-BUILDMAPS) option is specified with a system name only, all the maps in that system will be compiled.

-O command line option cannot be used with the -A command line option.

BUILDMAPS OPTIONS information

For BUILDMAPS OPTIONS, the following information can be supplied:

```
-BUILDMAPS [-P <platform>]
            [-O <file location>]
            | -A [-R
                <MSEBlds.txt file name/location>]
```

-GENERATE sdeploy option

The generate (-GENERATE) option of the sdeploy utility command is used to generate the systems present in the msd file. If the (-GENERATE) option is specified with the (-A) option, process all the systems' options. It is specifying that all the systems need to be generated with the platform specified in the command line and with the mode specified. The output .msl files will be produced one for each system in the specified directory and the results files will be generated one for each system that has been called. If the (-GENERATE) option is specified with a system name only, that system will be generated.

GENERATE OPTIONS information

For GENERATE OPTIONS, the following information can be supplied:

```
-GENERATE [-EMODE ES | CS]
          [-P <platform>]
          [-O <.msl file name/location>]
```

-DEPLOY sdeploy option

The deploy (-DEPLOY) option of the sdeploy utility command is used to deploy the deploy scripts to the specified server in the msd file. The (-DEPLOY) option cannot be specified with the (-A) option. It is specifying that all systems' options be processed as the script name is specific to each system, and can only be specified with a system name. The script defined in the msd file for a system can be deployed to the specified server with the (ES | CS) mode specified in the command

line. The server can be specified with the (-SERVER) option. The `MSEDploy_system_name.txt` results file will be generated for the system that has been called.

DEPLOY OPTIONS information

For DEPLOY OPTIONS, the following information can be supplied:

```
-DEPLOY <script name>
  [-EMODE ES | CS]
  [<-SERVER <server name> | LOCAL >
  [<-USER <userid> -PW <password> -IP <IP address> >]
  [-R <MSEDploy.txt file name/location>]
```

-SERVER sdeploy option

The server (-SERVER) option of the sdeploy utility command is used to specify the server name to which the deploy script needs to be deployed. The (-SERVER) option can be specified only with the (-DEPLOY) option. If specified with the (-ANALYZE, -BUILDMAPS, or -GENERATE) or the (-USER, -PW, or -IP) options, the command line is invalid, and the Not a valid command line - -SERVER option cannot be specified with -ANALYZE, -BUILDMAPS, -GENERATE, -USER, -PW, -IP. error message will display on the console. The server name can be LOCAL or any defined server name in the msd file. A new server can be defined by specifying the IP address of the server with the (-IP) option, the username with the (-USER) option, and the password with the (-PW) option.

-USER -PW -IP -sdeploy option

The server (-USER, -PW, and -IP) options of the sdeploy utility command are used to specify a new server for deploying the deploy script. All three options are mandatory for specifying a new server. If any one of the three options are not specified, the command line is invalid, and the Not a valid command line - all three (-USER), (-PW), (-IP) need to be specified to for a new server. error message will display on the console. If you are able to create a new server, the deploy script will be deployed to this new server, or the Unable to create a server with the user options and IP given error message will be written to the log file, if enabled, and the operation will be terminated.

-P sdeploy option

The parameter (-P) option of the sdeploy utility command is used to indicate a parameter to specify the platform for which the maps are to be compiled, the platform for which the systems need to be generated and to which the systems need to be deployed. The sdeploy utility command line uses WINDOWS as the default platform if the (-P) option is not specified. The other platforms that are allowed are AIX, HIMALAYA, HPITANIUM, HPNONSTOP, HPUX, LINUX, SOLARIS and ZOS. If the (-P) option is specified and no platform parameter is provided, the command line is invalid, and the Not a valid command line -Platform parameter should be provided if option -P is selected. The valid platforms are AIX, HIMALAYA, HPUX, SOLARIS, HPNONSTOP, ZOS, LINUX, HPITANUIUM and WINDOWS error message will display on the console.

-O sdeploy option

The output (-O) option of the sdeploy utility command is used to indicate a parameter to specify the location or file name, or both, for the resulting compiled map files when (-BUILDMAPS) option is chosen, or resulting generated msl file when

(-GENERATE) option is chosen. This is a description of how the output (-O) option can be used in the sdeploy utility command.

The specified parameter for the output (-O) option could be the name of the directory where the resulting compiled map files or the resulting generated msl file will be written, the full file path specification in which a wildcard can be used to represent the compiled map file or the generated msl file, or the full path in which the complete filename can be specified only for the msl file. The default naming convention for the compiled map file is **executable_map_name.mmc**, and for the generated msl file is **system_name.msl**. If a wildcard is specified in the sdeploy mysystems.msd MySystem -BUILDMAPS -O C:\MyDev\Compiledmaps*.win full path, the compiled maps will be produced in the **C:\MyDev\Compiledmaps** directory with the **<executable_map_name>.win** naming convention. If a wildcard is specified in the sdeploy mysystems.msd MySystem -GENERATE -O C:\MyDev\Systemfiles*.win full path, the generated msl will be produced in the **C:\MyDev\Systemfiles** directory with the **<system_name>.win** naming convention. If the complete file name is specified, for example, sdeploy mysystems.msd MySystem -GENERATE -O C:\MyDev\Systemfiles\MyExecForwin.win, the **MyExecForWin.win** file will be produced in the **C:\MyDev\Systemfiles** directory.

If the (-O) option is not specified, the compile map will be written to the same directory as the map source file using the default **executable_map_name.mmc** naming convention and the msl file will be written to the source directory of the msd file with the **system_name.msl** default naming convention. If the (-O) option is specified and the parameter is not provided, the command line is invalid, and the Not a valid command line : The file name for the resulting output file should be provided when (-O) option is selected error message will display on the console. If the file name or the path provided for the (-O) option is not valid or returns error while attempting to open or write to the file, the Compilation error - The file/path provided for the option (-O) is not valid, unable to write to the specified directory/file error message will be written to the sdeploy log file, if it is enabled. If the (-O) option is specified without the (-BUILDMAPS or -GENERATE) command options, the command line is invalid, and the Not a valid command line : (-GENERATE) or (-BUILDMAPS) or (-DEPLOY) or (-L) or (-ANALYZE) is missing. error message will display on the console.

Note: -O command line option cannot be used with the -A command line option.

-NO sdeploy option

The no overwrite (-NO) option of the sdeploy utility command is used to indicate that an existing (**.mmc** or **.msl**) output file is not to be overwritten. If the no overwrite (-NO) option is specified in the command line of sdeploy, and a file with the same name as the resulting output file exists, it will not be overwritten, the sdeploy operation will fail, and the Compilation error - The output file cannot be overwritten, file already exists message will be written to the sdeploy log file, if enabled. If the (-NO) option is not specified, the file with the same name as the resulting output file will be overwritten.

-R sdeploy option

The results (-R) option of the sdeploy utility command will have a parameter to specify the location or file name, or both, for the results file produced by the sdeploy operation for each system. There are 2 different cases where (-R) can be used in the sdeploy utility command.

In the first case, the (-R) option is used with the process all systems (-A) option. The parameter specified could either be the name of the directory where the results files will be written, or the full file path specification in which a wildcard can be used to represent the system or executable map naming convention. The default naming convention for the results file is **MSEBlds_system_name.txt** for the (-BUILDMAPS) option and **MSEAnlz_system_name.txt** for the (-ANALYZE) option. If a wildcard is specified in the `sdeploy mysystems.msd -ANALYZE -A -R C:\MyDev\AnalyzeResults*.err` full path, the results files will be produced in the **C:\MyDev\AnalyzeResults** directory with the **MSEAnlz_system_name.err** naming convention. If a wildcard is specified in the full path as `sdeploy mysystems.msd -BUILDMAPS -A -R C:\MyDev\BuildResults*.err`, the results files will be produced in the **C:\MyDev\BuildResults** directory with the **MSEBlds_system_name.err** naming convention.

In the second case, the (-R) option is used without the process all systems (-A) option. The parameter specified could be the name of the directory where the results files will be written, the full file path specification in which a wildcard can be used to represent the result file, or the full path in which the complete filename can be specified. The default naming convention for the results file is **MSEBlds_system_name.txt** for the (-BUILDMAPS) option, **MSEAnlz_system_name.txt** for the (-ANALYZE), and **MSEDeploy_system_name.txt** for the (-DEPLOY) option. If a wildcard is specified in the full path as `sdeploy mysystems.msd mysystem -ANALYZE -R C:\MyDev\AnalyzeResults*.err`, the results files will be produced in the **C:\MyDev\AnalyzeResults** directory with the **MSEAnlz_system_name.err** naming convention. If a wildcard is specified in the full path as `sdeploy mysystems.msd mysystem -BUILDMAPS -R C:\MyDev\BuildResults*.err`, the results files will be produced in the **C:\MyDev\BuildResults** directory with the **MSEBlds_system_name.err** naming convention. If a wildcard is specified in the full path as `sdeploy mysystems.msd mysystem -DEPLOY -R C:\MyDev\DeployResults*.err`, the results files will be produced in the **C:\MyDev\DeployResults** directory with the **MSEDeploy_system_name.err** naming convention.

If the (-R) option is not specified, the results will be written to the console. If the (-R) option is specified, but no parameter is provided, the command line is invalid, and the Not a valid command line - The file name for the resulting result file should be provided when -R option is selected. error message will display on the console. If the file name or the path provided for the (-R) option is not valid or returns error while attempting to open or write to the file, the Compilation error - The file/path provided for the option (-R) is not valid, unable to write to the specified directory/file message will be written to the sdeploy log file, if it is enabled. If the (-R) option is specified without the (-ANALYZE, -BUILDMAPS, or -DEPLOY) options, the command line is invalid, and the Not a valid command line - (-R) option can be specified only with (-ANALYZE) or (-BUILDMAPS) or (-DEPLOY) command options. error message will display on the console.

-LOG sdeploy option

The log (-LOG) option of the sdeploy utility command is used to enable the sdeploy logging capability. The log file will be produced with the results of each operation performed. The (-LOG) option has an optional parameter to specify the location or file name, or both, for the sdeploy log file. The log will be produced when the sdeploy utility command is executed. The scope of the sdeploy log depends on which (-ANALYZE, -BUILDMAPS, -GENERATE, or -DEPLOY) option has been chosen with

the sdeploy utility command. The default file name for the sdeploy log will be **msd_file_name.scl**. The default location for the sdeploy log will be the directory where the msd file is located.

The file name or the location parameter, if specified with the (-LOG) option of the sdeploy utility command, could be the name of the directory where the sdeploy log file will be written, the full file path specification in which a wildcard can be used to represent the msd file, or the full path in which the complete filename can be specified. The default naming convention for the sdeploy log file is **msd_file_name.scl**. If the . is specified with the (-LOG) option, the sdeploy log will be written to the default location (directory where the msd file is located) using the default naming convention. If a wildcard is specified in the full path as sdeploy mysystems.mscl -A -LOG C:\MyDev\Systemfiles\2002-02-28_*.results, the sdeploy log file will be produced in the **C:\MyDev\Systemfiles** directory with the **2002-02-28_mysystems.results** naming convention. If the complete file name is specified as sdeploy mysystems.mscl -A -LOG C:\MyDev\Systemfiles\mysystems_results.txt, the **mysystems_results.txt** will be produced in the **C:\MyDev\Systemfiles** directory.

If the sdeploy utility command is unable to create the sdeploy log file, the any operation of sdeploy will terminate and the Fatal error - Could not start processing, unable to create the log file fatal error message will display on the console. If nothing is specified in the command line for the log, by default, the command line assumes the (-LOG) option is included in the command line, and writes the log messages to the console.

-FAIL sdeploy option

The fail (-FAIL) option of the sdeploy utility command is used to indicate that only the process that had errors during operation should be included in the sdeploy log file. If the log failures only (-FAIL) option is selected, the sdeploy log will contain only the logs of failed operations. If the log failures only (-FAIL) option is not selected in the command line, all the results will be written to the sdeploy log file.

-APPEND sdeploy option

The append (-APPEND) option of the sdeploy utility command is used to indicate that the current sdeploy execution messages should be appended to the existing sdeploy log file if it already exists. If the sdeploy log append (-APPEND) option is selected, the sdeploy log messages will be appended to the existing file if it already exists. Otherwise, a new file will be created. If the (-APPEND) option is not selected, the sdeploy log file will be created if no log file exists, or it will be overwritten if the log file does exist.

-VERBOSE sdeploy option

The verbose (-VERBOSE) option of the sdeploy utility command is used to indicate that a verbose sdeploy log should be produced. The verbose sdeploy log will contain an entry for each (-ANALYZE, -BUILDMAPS, -GENERATE, and -DEPLOY) operation attempted, the **.mscl** file name, operation start date/time, operation end date/time, operation result, and the name of the results file, if it exists. If the verbose (-VERBOSE) option is not selected, a concise version of the sdeploy log will be produced, which includes an entry for each mscl file, attempted system name, and operation result.

-NOLOG sdeploy option

The disable log (-NOLOG) option of the sdeploy utility command is used to turn off the sdeploy logging capability. If the disable logging option is selected, the log file will not be produced. The (-APPEND, -FAIL, and -VERBOSE) options of the (-LOG) option cannot be used with the disable log (-NOLOG) option. If these options are selected with (-NOLOG), the command line is invalid, and the Not a valid command line - (LOG) optional commands (-APPEND), (-FAIL), (-VERBOSE) cannot be used with disable log command (-NOLOG). error message will display on the console.

Command line help for sdeploy

There is help that describes the sdeploy utility command usage in the command line.

To view the command line help for the sdeploy utility command

Enter sdeploy at the command line prompt in the *install_dir*.

The syntax and descriptions of all the options appear.

Using the sdeploy command

The following example shows how you can use the sdeploy utility command to:

- display all systems on the console from the command line
- analyze all systems on the console from the command line
- build maps in all systems from the command line (outside of the IFD GUI)

Using sdeploy to display all systems on the Console

Use the sdeploy utility command to display all systems on the console from the command line, outside the Integration Flow Designer GUI. The following command is an example of using the (-L) option on the sdeploy utility command to display all the systems on the console:

```
install_dir> sdeploy mysystems.msdl -L
```

Result:

```
Systems in mysystems.msdl:  
  FristSystem  
  MySystem  
  NewSystem  
Total systems: 3
```

The list of all systems will display in alphabetical order and the total number of systems will display on the console.

Using sdeploy to analyze all systems

Use the sdeploy utility command to analyze all systems from the command line, outside the Integration Flow Designer GUI. The following command is an example of using the (-A) option on the sdeploy utility command to analyze all the systems on the console:

```
install_dir> sdeploy mysystems.msdl
-ANALYZE -A -EMODE ES
-R C:\MyDev\AnalyzeResults\*.txt
-LOG C:\MyDev\AnalyzeResults\*.results.txt
-FAIL -APPEND
```

The following list displays three possible combinations of outcomes of this utility command example:

- If all the systems are analyzed successfully, the sdeploy - completed successfully. message will display on the console.
- The **MSEAnlz_FristSystem.txt**, **MSEAnlz_MySystem.txt**, **MSEAnlz_NewSystem.txt** analyzed results files will be produced for launcher mode in the **C:\MyDev\AnalyzeResults** directory.
- If the **mysystems_results.txt** analyzed results file does not exist, the sdeploy's log file will be produced in the **C:\MyDev\AnalyzeResults** directory. If the **mysystems_results.txt** analyzed results file already exists, only the failed system's log messages will be appended to the existing file.

Using sdeploy to build maps from the command line - example 1

Use the sdeploy utility command to build maps in all systems from the command line, outside the Integration Flow Designer GUI. The following command is an example of using the (-A) option on the sdeploy utility command to analyze all the systems on the console:

```
install_dir> sdeploy mysystems.msdl
-BUILDMAPS -A -P WINDOWS
-O C:\MyDev\BuildResults
-R C:\MyDev\BuildResults\*.txt
-LOG C:\MyDev\BuildResults\*.results.txt
-FAIL -APPEND
```

The following list displays three possible combinations of outcomes of this utility command example:

- If all the maps in the systems are compiled successfully, the sdeploy - completed successfully. message will display on the console.
- The **MSEBlds_FristSystem.txt**, **MSEBlds_MySystem.txt**, **MSEBlds_NewSystem.txt** build results files and the compiled **executable_map_name.mmc** compiled map will be produced in the **C:\MyDev\BuildResults** directory.

If the **mysystems_results.txt** file does not exist, the sdeploy's log file will be produced in the **C:\MyDev\BuildResults** directory. If the **mysystems_results.txt** already exists, only the failed systems log messages will be appended to the existing file.

Using sdeploy to build maps from the command line - example 2

Use the sdeploy utility command to build maps in all systems from the command line, outside the Integration Flow Designer GUI. The following command is an example of using the (-A) option on the sdeploy utility command to analyze all the systems on the console:

```
install_dir> sdeploy mysystems.msdl
-GENERATE -A -EMODE ES -P WINDOWS
-O C:\MyDev\GenerateResults\*.msl
-LOG C:\MyDev\GenerateResults\*.results.txt
-FAIL -APPEND
```

The following list displays three possible combinations of outcomes of this utility command example:

- If all the systems are generated successfully, the `sdeploy - completed successfully.` message will display on the console.
- The `FristSystem.msl`, `MySystem.msl`, `NewSystem.msl` generated output files will be produced in the `C:\MyDev\GenerateResults` directory.
- If the `mysystems_results.txt` file does not exist, the `sdeploy`'s log file will be produced in the `C:\MyDev\GenerateResults` directory. If the `mysystems_results.txt` file already exists, only the failed systems log messages will be appended to the existing file.

Using sdeploy to build maps from the command line - example 3

Use the `sdeploy` utility command to build maps in all systems from the command line, outside the Integration Flow Designer GUI. The following command is an example of using the `(-A)` option on the `sdeploy` utility command to analyze all the systems on the console:

```
install_dir> sdeploy mysystems.msd NewSystem
-DEPLOY MyScript -EMODE ES -SERVER MyServer
-R C:\MyDev\DeployResults\*.txt
-LOG C:\MyDev\DeployResults\*.results.txt
-FAIL -APPEND
```

- If the deploy script is deployed successfully, the `sdeploy - completed successfully.` message will display on the console.
- The `MyScript` map will be deployed to the `MyServer` and the results files will be produced in the `C:\MyDev\DeployResults` directory.
- If the `mysystems_results.txt` file does not exist, the `sdeploy`'s log file will be produced in the `C:\MyDev\DeployResults` directory. If `mysystems_results.txt` file already exists, only the failed systems log messages will be appended to the existing file.

Using sdeploy to build maps from the command line - example 4

Use the `sdeploy` utility command to build maps in all systems from the command line, outside the Integration Flow Designer GUI. The following command is an example of using the `(-A)` option on the `sdeploy` utility command to analyze all the systems on the console:

```
install_dir> sdeploy mysystems.msd NewSystem
-DEPLOY MyScript -EMODE ES
-USER userid -PW password -IP 192.82.65.46
-R C:\MyDev\DeployResults\*.txt
-LOG C:\MyDev\DeployResults\*.results.txt
-FAIL -APPEND
```

- If the deploy script is deployed successfully, the `sdeploy - completed successfully.` message will display on the console.
- The `MyScript` map will be deployed to the new server created with the IP address provided, the `userid`, and the `password`, and the results files will be produced in the `C:\MyDev\DeployResults` directory.
- If the `mysystems_results.txt` file does not exist, it will be created in the `C:\MyDev\DeployResults` directory. If the file exists, append only the failed systems log messages.

msdimport utility command

The msdimport utility command is used to import an xml file to a system definition (.msd) source file in a system from the command line, outside of the IFD GUI. The msdimport utility command returns 0 if the import is successful, and 1 if not. It is a batch operation that can also be used for automation.

Refer to the "Troubleshooting" section for details about capturing and evaluating the return code resulting from running this command.

Syntax summary for msdimport

msdimport is the name of the utility command used to import xml files to system source files.

The .xml file is a required field and the file name of the xml source file. If the full path of the xml source file name is not specified, the msdimport utility command will search for the xml source file in the current directory.

The following is the syntax of the msdimport utility command:

```
msdimport <.xml file>
  [-NO]
  [-O <.msd file name/location>]
  [-REFSYS] [-REFSVR] [-OSYS] [-OSVR]
  [-ASVR [<server name>] | [-ASYS [<system names>]]]
  [(-LOG [. | <log file name/location>]
    [-FAIL] [-APPEND] [-VERBOSE])
  | -NOLOG]
```

In the msdimport utility command, all the options are case insensitive.

Typing msdimport with no options will display the command summary on the console.

Utility command options for msdimport

The following command options are available with the msdimport utility command:

• -APPEND msdimport Option	• -O msdimport Option
• -ASVR msdimport Option	• -OSVR msdimport Option
• -ASYS msdimport Option	• -OSYS msdimport Option
• -FAIL msdimport Option	• -REFSVR msdimport Option
• -LOG msdimport Option	• -REFSYS msdimport Option
• -NO msdimport Option	• -VERBOSE msdimport Option
• -NOLOG msdimport Option	

-APPEND msdimport option

The append (-APPEND) option of the msdimport utility command is used to indicate that the current msdimport execution messages should be appended to the existing msdimport log file if it already exists. If the msdimport log append (-APPEND) option

is selected, the msdimport log messages will be appended to the existing file if it already exists. Otherwise, a new file will be created. If the (-APPEND) option is not selected, the msdimport log file will be created if no log file exists, or it will be overwritten if the log file does exist.

-ASVR msdimport option

The server (-ASVR) option of the msdimport utility command is used to import only servers from the xml file. If the option is used alone, the msdimport utility command will import all the servers from the xml file. If the option is followed by a string of server names separated by commas, for example, -ASVR server1, server2, server3, the msdimport utility command will import those specific servers from the xml file.

-ASYS msdimport option

The system (-ASYS) option of the msdimport utility command is used to import only systems from the xml file. If the option is used alone, the msdimport utility command will import all the systems from the xml file. If the option is followed by a string of system names separated by commas, for example, -ASYS system1, system2, system3, the msdimport utility command will import those specific systems from the xml file.

Any combination of the -ASYS and -ASVR options used with the msdimport utility command will be executed as separate individual options.

-FAIL msdimport option

The fail (-FAIL) option of the msdimport utility command is used to indicate that only the systems that had errors during the importing should be included in the msdimport log file. If the log failures only (-FAIL) option is selected, the msdimport log will contain only the systems that failed importing. If the log failures only (-FAIL) option is not selected in the command line, all the imported results will be written to the msdimport log file.

-LOG msdimport option

The log (-LOG) option of the msdimport utility command is used to enable logging. The log file will be produced showing the result of the imported system source file. (-LOG) has an optional parameter to specify the location or file name, or both, for the msdimport log file. The log will be produced when the msdimport utility command is executed. The default file name for the msdimport log will be **system_source_file_name_import.log**. The default location for the msdimport log will be the directory where the system definition (**.msd**) source file is located.

The file name or the location parameter, if specified with the (-LOG) option of the msdimport utility command, could be the name of the directory where the msdimport log file will be written, the full file path specification in which a wildcard can be used to represent the xml source file, or the full path in which the complete filename can be specified. If the . is specified with the (-LOG) option, the msdimport log will be written to the default location (directory where the xml file is located) using the default naming convention. If a wildcard is specified in the full path, for example, msdimport myxmlfile.xml -LOG C:\MyDev\ImportResults\2003-10-03_*.results, the msdimport log file will be produced in the **C:\MyDev\ImportResults** directory with the **2003-10-03_myxmlfile.results** naming convention. If the complete file name is specified, for example, msdimport

myxmlfile.xml -LOG C:\MyDev\ImportResults\myxmlimport_results.txt, the **myxmlimport_results.txt** will be produced in the **C:\MyDev\ImportResults** directory.

If the msdimport utility command is unable to create the msdimport log file, the importing of the system source file will terminate and the Fatal error - Could not start importing the system source file, unable to create the log file fatal error message will display on the console. If nothing is specified in the command line for the log, by default, the command line assumes the (-LOG) option is included in the command line, and writes the log messages to the console.

-NO msdimport option

The no overwrite (-NO) option of the msdimport utility command is used to indicate that an existing imported system definition (**.msd**) source file is not to be overwritten. If the no overwrite (-NO) option is specified in the command line of msdimport, and a file with the same name as the resulting **.msd** file exists, it will not be overwritten, the msdimport operation will fail, and the IFD import error - The imported system source file cannot be overwritten, file already exists message will be written to the msdimport log file, if enabled. If the (-NO) option is not specified, the file with the same name as the resulting **.msd** file, will be overwritten.

-NOLOG msdimport option

The disable log (-NOLOG) option of the msdimport utility command is used to turn off the msdimport logging capability. If the disable logging option is selected, the log file will not be produced. The (-APPEND, and -VERBOSE) optional commands of the (-LOG) option cannot be used with the disable log (-NOLOG) option. If these options are selected with the (-NOLOG) option, the command line is invalid, and the Not a valid command line - (LOG) optional commands (-APPEND), (-VERBOSE) cannot be used with disable log command (-NOLOG). message will display on the console.

-O msdimport option

The output (-O) option of the msdimport utility command is used to specify, through a parameter, the location or file name, or both, of the system definition (**.msd**) file produced by importing the xml file. The specified parameter could be the name of the directory to where the imported **.msd** file will be written, or the full file path specification in which a wildcard can be used to represent the imported system source file, or the full path in which the complete file name can be specified. The default naming convention for the imported **.msd** file is ``xml_file_name.msd'`. If a wildcard is specified in the full path, for example, `MSDImport myxmlfile.xml -O C:\MyDev\ImportResults\2003-10-03_*.msd`, the resulting **.msd** file will be produced in the **C:\MyDev\ImportResults** directory with the **2003-10-03_myxmlfile.msd** naming convention. If the complete file name is specified, for example, `msdimport myxmlfile.xml -O C:\MyDev\ImportResults\Myxmlsystems.msd`, a **Myxmlsystems.msd** system definition (**.msd**) file will be produced in the **C:\MyDev\ImportResults** directory.

-OSVR msdimport option

The output (-OSVR) option of the msdimport utility command is used to override servers if the servers are already present in the system definition (**.msd**) file.

-OSYS msdimport option

The output (-OSYS) option of the msdimport utility command is used to override systems if the systems are already present in the system definition (.msd) file.

-REFSVR msdimport option

The reference server (-REFSVR) option of the msdimport utility command is used to allow reference servers to be imported.

-REFSYS msdimport option

The reference system (-REFSYS) option of the msdimport utility command is used to allow reference systems to be imported, also. If the -REFSYS command is specified, the msdimport utility command will also import the dependent systems.

-VERBOSE msdimport option

The verbose (-VERBOSE) option of the msdimport utility command is used to indicate if the verbose msdimport log should be produced. The verbose msdimport log will contain an entry for the xml file being imported, the importing start date/time, the importing end date/time, the imported system source file name, and the result of the msdimport utility command. If the verbose (-VERBOSE) option is not selected, a concise version of the msdimport log will be produced, which includes an entry for the system source file name, and importing result.

Command line help for msdimport

There is help that describes the msdimport utility command usage in the command line.

To view the command line help for the msdimport utility command

Enter msdimport at the command line prompt in the *install_dir*.

The syntax and descriptions of all the options appear.

Using the msdimport command

Use the msdimport utility command to import the xml file to a system definition (.msd) file from the command line, outside the Integration Flow Designer GUI. The following command shows an example of how it is used:

```
install_dir> msdimport myxmlfile.xml  
-O C:\MyDev\ImportResults\myimportedsystem.msd  
-LOG -APPEND
```

The following list displays three possible combinations of outcomes of this utility command example:

- If the **myxmlfile.xml** xml file is imported successfully, the msdimport - completed successfully. message will display on the console.
- The imported **myimportedsystem.msd** system definition file will be produced in the **C:\MyDev\ImportResults** directory.

If the **myxmlfile.log** log file does not exist, it will be produced in the same directory as the system file. If the **myxmlfile.log** log file already exists, the log messages will be appended to the existing file.

msdexport utility command

The msdexport utility command is used to export a system source definition (.msd) file from the command line, outside of the IFD GUI. The msdexport utility command returns 0 if the export is successful, and 1 if not. It is a batch operation that can also be used for automation.

Syntax summary for msdexport

msdexport is the name of the utility command used to export system source files.

The .msd file is a required field and is the file name of the system source file that needs to be exported. If the full path of the system source file name is not specified, the msdexport utility command will search for the system source file in the current directory.

In the msdexport utility command, all the options are case insensitive. The following is the syntax of the msdexport utility command:

```
msdexport <.msd file>
  [-NO]
  [-O <.xml file name/location>]
  [-EPWD] [-REFSYS] [-REFSVR]
  [(-LOG [. | <log file name/location>]
    [-ASVR [<server name>] | [-ASYS [<system names>]
      [-FAIL] [-APPEND] [-VERBOSE])
  [-NOLOG]
```

Typing msdexport with no options will display the command summary on the console.

Utility command options for msdexport

The following command options are available with the msdexport utility command:

• -APPEND msdexport Option	• -NO msdexport Option
• -ASVR msdexport Option	• -NOLOG msdexport Option
• -ASYS msdexport Option	• -O msdexport Option
• -EPWD msdexport Option	• -REFSVR msdexport Option
• -FAIL msdexport Option	• -REFSYS msdexport Option
• -LOG msdexport Option	• -VERBOSE msdexport Option

-APPEND msdexport option

The append (-APPEND) option of the msdexport utility command is used to indicate that the current msdexport execution messages should be appended to the existing msdexport log file if it already exists. If the msdexport log append (-APPEND) option is selected, the msdexport log messages will be appended to the existing file if it already exists. Otherwise, a new file will be created. If the (-APPEND) option is not selected, the msdexport log file will be created if no log file exists, or it will be overwritten if the log file does exist.

-ASVR msdexport option

The server (-ASVR) option of the msdexport utility command is used to export only servers from the system definition (.msd) file. If the option is used alone, the msdexport utility command will export all the servers from the .msd file. If the option is followed by a string of server names separated by commas, for example, -ASVR server1, server2, server3, the msdexport utility command will export those specific servers from the .msd file.

-ASYS msdexport option

The system (-ASYS) option of the msdexport utility command is used to export only systems from system definition (.msd) file. If the option is used alone, the msdexport utility command will export all the systems from the .msd file. If the option is followed by a string of system names separated by commas, for example, -ASYS system1, system 2, system 3, the msdexport utility command will export those specific systems from the .msd file.

Any combination of the -ASYS and -ASVR options used with the msdexport utility command will be executed as separate individual options.

-EPWD msdexport option

The encrypt password (-EPWD) option of the msdexport utility command is used to encrypt the password while writing to a file.

-FAIL msdexport option

The fail (-FAIL) option of the msdexport utility command is used to indicate that only the systems that had errors during the exporting should be included in the msdexport log file. If the log failures only (-FAIL) option is selected, the msdexport log will contain only the systems that failed exporting. If the log failures only (-FAIL) option is not selected in the command line, all the exported results will be written to the msdexport log file.

-LOG msdexport option

The log (-LOG) option of the msdexport utility command is used to enable logging. The log file will be produced showing the result of the exported system source file. (-LOG) has an optional parameter to specify the location or file name, or both, for the msdexport log file. The log will be produced when the msdexport utility command is executed. The default file name for the msdexport log will be **system_source_file_name_export.log**. The default location for the msdexport log will be the directory where the system definition (.msd) file is located.

The file name or the location parameter, if specified with the (-LOG) option of the msdexport utility command, could be the name of the directory where the msdexport log file will be written, the full file path specification in which a wildcard can be used to represent the xml source file, or the full path in which the complete filename can be specified. If the . is specified with the (-LOG) option, the msdexport log will be written to the default location (directory where the (.msd) file is located) using the default naming convention. If a wildcard is specified in the full path, for example, msdexport mysystems.msd -LOG C:\MyDev\ExportResults\2003-10-03_*.results, the msdexport log file will be produced in the C:\MyDev\ExportResults directory with the **2003-10-03_mysystems.results** naming convention. If the complete file name is specified, for example, msdexport

`mysystems.mms -LOG C:\MyDev\ExportResults\mysystemsexport_results.txt`, the `mysystemsexport_results.txt` will be produced in the `C:\MyDev\ExportResults` directory.

If the `msdexport` utility command is unable to create the `msdexport` log file, the exporting of the system source file will terminate and the Fatal error - Could not start exporting the system source file, unable to create the log file fatal error message will display on the console. If nothing is specified in the command line for the log, by default, the command line assumes the `(-LOG)` option is included in the command line, and writes the log messages to the console.

-NO msdexport option

The no overwrite `(-NO)` option of the `msdexport` utility command is used to indicate that an existing exported xml `(.xml)` file is not to be overwritten. If the no overwrite `(-NO)` option is specified in the command line of `msdexport`, and a file with the same name as the resulting `.msd` file exists, it will not be overwritten, the `msdexport` operation will fail, and the IFD export error - The exported xml file cannot be overwritten, file already exists message will be written to the `msdexport` log file, if enabled. If the `(-NO)` option is not specified, the file with the same name as the resulting `.msd` file, will be overwritten.

-NOLOG msdexport option

The disable log `(-NOLOG)` option of the `msdexport` utility command is used to turn off the `msdexport` logging capability. If the disable logging option is selected, the log file will not be produced. The optional `(-APPEND)`, and `(-VERBOSE)` options of the `(-LOG)` option cannot be used with the disable log `(-NOLOG)` option. If these options are selected with the `(-NOLOG)` option, the command line is invalid, and the Not a valid command line - (LOG) optional commands `(-APPEND)`, `(-VERBOSE)` cannot be used with disable log command `(-NOLOG)`. error message will display on the console.

-O msdexport option

The output `(-O)` option of the `msdexport` utility command will have a parameter to specify the location or file name, or both, for the exported xml file produced by exporting the system source file. The parameter specified could be the name of the directory where the exported xml file will be written, the full file path specification in which a wildcard can be used to represent the exported xml file, or the full path in which the complete filename can be specified. The default naming convention for the exported xml file is `system_source_file_name.xml`. If a wildcard is specified in the full path, for example, `msdexport mysystem.msd -O C:\MyDev\ExportResults\2003-10-03_*.xml`, the resulting xml file will be produced in the `C:\MyDev\ExportResults` directory with the `2003-10-03_mysystem.xml` naming convention. If the complete file name is specified, for example, `mexport mysystem.mms -O C:\MyDev\ExportResults\MysystemExportedfile.xml`, an xml file `MysystemExportedfile.xml` will be produced in the `C:\MyDev\ExportResults` directory. If the export output `(-O)` option is selected, and if the file name is not specified, the command line is invalid, and the Not a valid command line - the file name parameter is required if export command option `(-O)` is selected error message will display on the console.

-REFSVR msdexport option

The reference server `(-REFSVR)` option of the `msdexport` utility command is used to allow reference servers to be exported.

-REFSYS msdexport option

The reference system (-REFSYS) option of the msdexport utility command is used to allow reference systems to be exported, also. If the -REFSYS command is specified, the msdexport utility command will also export the dependent systems.

-VERBOSE msdexport option

The verbose (-VERBOSE) option of the msdexport utility command is used to indicate if the verbose msdexport log should be produced. The verbose msdexport log will contain an entry for the system source file being exported, the exporting start date/time, the exporting end date/time, the exported system source file name, and the result of the msdexport utility command. If the verbose (-VERBOSE) option is not selected, a concise version of the msdexport log will be produced, which includes an entry for the system source file name, and exporting result.

Command line help for msdexport

There is help that describes the msdexport utility command usage in the command line.

To view the command line help for the msdexport utility command

Enter msdexport at the command line prompt in the *install_dir*.

The syntax and descriptions of all the options appear.

Using the msdexport command

Use the msdexport utility command to export the system source file from the command line, outside the Integration Flow Designer GUI. The following command shows an example of how it is used:

```
install_dir> msdexport mysystem.mms  
-O C:\MyDev\ExportResults\mysystem.xml  
-LOG -APPEND
```

The following list displays three possible combinations of outcomes of this utility command example:

- If the **mysystems.mms** system source file is exported successfully, the msdexport - completed successfully. message will display on the console.
- The exported **mysystems.xml** xml file will be produced in the **C:\MyDev\ExportResults** directory.
- If the **mysystems.log** log file does not exist, it will be produced in the same directory as the system source file. If it already exists, the log messages will be appended to the existing file.

Chapter 6. Utility Commands for map tuning

Map tuning utility commands are tools for calculating map memory page size and count and running your maps to create output consisting of statistical data regarding the execution of your maps. This data can assist you in tuning your maps to attain improved performance.

The following list briefly describes the functions of each of the map tuning utility commands:

- "dtxpage Utility Command" - calculates suggested settings for memory page size and count for maps
- "dtxprof Utility Command" - profiles maps and analyzes map execution behavior

dtxpage utility command

The dtxpage utility command invokes the **Page Setting Assistant for Maps** application used to calculate suggested settings for memory page size and count for work files used in maps. The dtxpage utility command returns 0 if the compilation is successful, and 1 if not successful.

The **.mmc** file is a required parameter and is the file name of the map for which the page size and count is being calculated. If the full path of the map file name is not specified, the dtxpage utility command will search for the map in the current directory.

The **Page Setting Assistant for Maps** application will run iterations of the specified map to calculate suggested memory page size and count settings. Because of the multiple iterations of map runs, the overall run time will be longer than a typical run. Because of the additional run time, it is recommended that you:

- use a smaller representation of the input data when you are using a map that normally takes a long time to run
- use this utility command only when there is a limited number of other processes running so that CPU utilization is not adversely affected

See "Troubleshooting" for details about capturing and evaluating the return code resulting from the execution of this utility command.

You can use the dtxpage utility command to calculate suggested settings for memory page size and count for work files used in maps that you are running with the z/OS Batch Command Server. See "z/OS Batch Command Server dtxpage Usage".

Syntax summary for dtxpage

dtxpage is the name of the utility command used to calculate suggested memory page size and count settings for work files.

The following is the syntax of the dtxpage utility command:

```
dtxpage <.mmc file name/location>
```

There are no other options besides the compiled map (**.mmc**) file name and location.

Typing `dtxpage` with no options will display the command summary on the console.

Command line help for dtxpage

There is help that describes the `dtxpage` utility command usage in the command line.

To view the command line help for the `dtxpage` utility command

1. Enter `dtxpage` at the command line prompt in the *install_dir*.

The syntax and descriptions of all the options appear.

Using the dtxpage command

The following example shows how you can use the `dtxpage` utility command.

Example

Use the `dtxpage` utility command to calculate suggested memory page size and count settings for work files from the command line. The following command shows an example of how it is used:

```
install_dir> dtxpage mymap.mmc
```

The following list displays the outcomes of this utility command example:

- If the suggested settings for memory page size and count are calculated for the map file successfully, first a list of the sizes and counts for each map run by the `dtxpage` utility command will display on the console. The list will also include the time stamp for each map run.
- Following the list of the sizes and counts for each map, will be the suggested settings for memory page size and count that the `dtxpage` utility command calculated.

z/OS Batch Command Server dtxpage usage

The `dtxpage` utility command is available to use with the z/OS Batch Command Server. It will calculate suggested settings for memory page size and count for the work files used in a specified map.

To use the `dtxpage` utility command

1. Make a copy of the JCL file that executes the map on which the `dtxpage` utility command will perform its calculations.
2. Change the program name from **MERCATOR** or **DTXCMDSV** to **DTXPAGE** in one of the copies of the JCL file.

The **Page Analysis Usage** report will be written to an output specified on the **SYSPRINT** Data Definition (**DD**) statement.

dtxprof utility command

The `dtxprof` utility command is used to profile maps and analyze map execution behavior from the command line, outside of the Map Designer GUI. The `dtxprof` utility command returns 0 if the Map Profiler is successful and 1 if it is not successful.

The Map Profiler is a user-configurable utility that captures and reports map execution statistics. The profiler focuses on component and mapping rules and the functions and types within those rules. The resulting information enables you to see where performance is lagging and in turn make improvements in your maps.

For example, in the profile report you see that the processing time for a LOOKUP function is significantly more than other functions. You can modify the rule to use a better choice, such as the SEARCHUP function (because in this specific case the data in the LOOKUP is ascending), and as a result the processing time is greatly reduced. See *Data Search Usage* in the WebSphere Transformation Extender *Performance* documentation for more information about recommended usages of the different search functions to achieve your performance objectives.

See "Troubleshooting" for details about capturing and evaluating the return code resulting from the execution of this utility command.

You can use the `dtxprof` utility command to calculate suggested settings for memory page size and count for work files used in maps that you are running with the z/OS Batch Command Server. See "z/OS Batch Command Server `dtxprof` Usage".

Syntax summary for `dtxprof`

`dtxprof` is the name of the utility command used to analyze map execution behavior.

The output file name and compiled map name (**.mmc**) are required fields.

The following is the syntax of the `dtxprof` utility command:

```
dtxprof
[-f[x]]
[-t[x]]
[-fs]
[-ts]
[-d]
-o output_file
-dtx "map_name [map_options]"
```

In the `dtxprof` utility command, all the options are case insensitive. Typing `dtxprof` with no options will display the command summary on the console.

Utility command options for `dtxprof`

The following command options are available with the `dtxprof` utility command:

• -f <code>dtxprof</code> Option	• -ts <code>dtxprof</code> Option
• -fx <code>dtxprof</code> Option	• -d <code>dtxprof</code> Option

• -t dtxprof Option	• -o dtxprof Option
• -tx dtxprof Option	• -dtx dtxprof Option
• -fs dtxprof Option	

If you type dtxprof with no options, the command line help that describes the command will display on your screen.

-f dtxprof option

The function times (-f) option of the dtxprof utility command is used to specify that all the function processing times are to be reported in the output.

-fx dtxprof option

The function times exceeding *x* milliseconds (-fx) option of the dtxprof utility command is used to specify that all the function processing times that exceed a specified value (*x*) are to be reported in the output. The value must be between 0 and 250 (1/10 milliseconds).

-t dtxprof option

The type times (-t) option of the dtxprof utility command is used to specify that all the type processing times are to be reported in the output.

-tx dtxprof option

The type times exceeding *x* milliseconds (-tx) option of the dtxprof utility command is used to specify that all the type processing times that exceed a specified value (*x*) are to be reported in the output. The value must be between 0 and 250 (1/10 milliseconds).

-fs dtxprof option

The function times summary output (-fs) option of the dtxprof utility command is used to specify that all the function processing times are to be reported in summary form in the output.

-ts dtxprof option

The type times summary output (-ts) option of the dtxprof utility command is used to specify that all the type processing times are to be reported in summary form in the output.

-d dtxprof option

The comma-delimited output (-d) option of the dtxprof utility command is used to specify that the output format will be comma-delimited.

The default value is fixed-width so if the -d option is omitted, the output will be created in fixed-width format.

-o dtxprof option

The output file name (-o) option of the dtxprof utility command is used to specify the output file name (*filename*).

-dtx dtxprof option

The (-dtx) option of the dtxprof utility command is used to specify the compiled map name (*map_name*) with optional map options (*map_options*). Quotation marks around the map name are required.

Command line help for dtxprof

There is help that describes the dtxprof utility command usage in the command line.

To view the command line help for the dtxprof utility command

Enter dtxprof at the command line prompt in the *install_dir*.

The syntax and descriptions of all the options appear.

Using the dtxprof command

The profiler can capture the following statistics:

- Amount of time spent processing component and mapping rules and subordinate functions.
- Number of times rules and functions are executed (from validation through output).
- Number of times type objects are accessed.
- Depth (nesting level) of the object being profiled.

Type names

Similar to the map trace file, the type names referred to in the profile output represent the full type name in the original type tree, not the name of the type object that is visible in the Map Designer.

Function Times

The processing time of any subfunctions rolls up into the processing time of the top-level function. For example, if an EITHER function contains an IF function, the recorded time for the EITHER function would include the processing time of the IF function.

RUN maps

When a parent map contains a RUN map, the RUN map is not profiled. To profile a RUN map, it must be done outside of the RUN rule. You can do this by running it as a top-level executable map.

Output examples

Full report

The following command produces a full report containing both function and type information:

```
dtxprof -f -t -o sinkpro
-dtxcmdsv "C:\Program Files\IBM\WebSphere TX n.n\examples\
general\map\sinkmap\sinkmap.mmc"
```

Individual Function Breakdown

function	depth	iterations	time	map	type
-----	-----	-----	----	---	----
validation					
VALIDATE_CARD	0	1	298	sinkmap	
VALIDATE_CARD	0	1	59	sinkmap	
mapping					
BUILD_CARD	0	1	71	sinkmap	
LOOKUP	1	1	42	sinkmap	week example
TYPE_NAME	2	8	2	sinkmap	week example
BUILD_CARD	0	1	65	sinkmap	
NUMBERTOTEXT	1	1	44	sinkmap	total_hours element example
ADD_OP	2	1	43	sinkmap	total_hours element example
ADD_OP	3	1	37	sinkmap	total_hours element example
ADD_OP	4	1	33	sinkmap	total_hours element example
ADD_OP	5	1	28	sinkmap	total_hours element example
DIV_OP	6	1	24	sinkmap	total_hours element example
SUB_OP	7	1	22	sinkmap	total_hours element example
TEXTTONUMBER	8	1	20	sinkmap	total_hours element example
TIMETOTEXT	9	1	18	sinkmap	total_hours element example
TYPE_NAME	10	1	17	sinkmap	total_hours element example
TEXTTONUMBER	8	1	1	sinkmap	total_hours element example
TIMETOTEXT	9	1	1	sinkmap	total_hours element example
TYPE_NAME	10	1	0	sinkmap	total_hours element example
FLOAT_CONSTANT	7	1	1	sinkmap	total_hours element example
DIV_OP	6	1	4	sinkmap	total_hours element example
SUB_OP	7	1	3	sinkmap	total_hours element example
TEXTTONUMBER	8	1	1	sinkmap	total_hours element example
TIMETOTEXT	9	1	0	sinkmap	total_hours element example
TYPE_NAME	10	1	0	sinkmap	total_hours element example
TEXTTONUMBER	8	1	1	sinkmap	total_hours element example
TIMETOTEXT	9	1	0	sinkmap	total_hours element example
TYPE_NAME	10	1	0	sinkmap	total_hours element example
FLOAT_CONSTANT	7	1	0	sinkmap	total_hours element example
DIV_OP	5	1	4	sinkmap	total_hours element example
SUB_OP	6	1	3	sinkmap	total_hours element example
TEXTTONUMBER	7	1	0	sinkmap	total_hours element example
TIMETOTEXT	8	1	0	sinkmap	total_hours element example
TYPE_NAME	9	1	0	sinkmap	total_hours element example
TEXTTONUMBER	7	1	0	sinkmap	total_hours element example
TIMETOTEXT	8	1	0	sinkmap	total_hours element example
TYPE_NAME	9	1	0	sinkmap	total_hours element example
FLOAT_CONSTANT	6	1	0	sinkmap	total_hours element example
SUB_OP	5	1	3	sinkmap	total_hours element example
TEXTTONUMBER	6	1	1	sinkmap	total_hours element example
TIMETOTEXT	7	1	0	sinkmap	total_hours element example
TYPE_NAME	8	1	0	sinkmap	total_hours element example
TEXTTONUMBER	6	1	1	sinkmap	total_hours element example
TIMETOTEXT	7	1	0	sinkmap	total_hours element example
TYPE_NAME	8	1	0	sinkmap	total_hours element example
FLOAT_CONSTANT	5	1	0	sinkmap	total_hours element example
DIV_OP	3	1	3	sinkmap	total_hours element example
SUB_OP	4	1	2	sinkmap	total_hours element example
TEXTTONUMBER	5	1	1	sinkmap	total_hours element example
TIMETOTEXT	6	1	1	sinkmap	total_hours element example
TYPE_NAME	7	1	0	sinkmap	total_hours element example
TEXTTONUMBER	5	1	1	sinkmap	total_hours element example
TIMETOTEXT	6	1	0	sinkmap	total_hours element example
TYPE_NAME	7	1	0	sinkmap	total_hours element example
FLOAT_CONSTANT	4	1	0	sinkmap	total_hours element example
BUILD_CARD	0	1	50	sinkmap	
TYPE_NAME	1	1	1	sinkmap	employee example
TYPE_NAME	1	1	31	sinkmap	total_hours element example

Type per Rule Breakdown

time	map	type
----	---	----
validation		
mapping		
428	sinkmap	
44	sinkmap	week example
65	sinkmap	
322	sinkmap	total_hours element example
50	sinkmap	
1	sinkmap	employee example
31	sinkmap	total_hours element example

Summary report

The following command produces a summary report containing both function and type information:

```
dtxprof -fs -ts -o sinkmappro
-dtxcmds "examples\general\map\sinkmap\sinkmap.mmc"
```

Type per Rule Summary

iterations	time	map	type
-----	----	---	----
validation			
mapping			
1	228	sinkmap	week example
2	431	sinkmap	total_hours element example
1	1	sinkmap	employee example

Function Summary

function	iterations	time
-----	-----	----
VALIDATE_CARD	2	3153
BUILD_CARD	3	2415
TYPE_NAME	20	65
FLOAT_CONSTANT	5	1
ADD_OP	4	167
SUB_OP	5	37
DIV_OP	5	44
LOOKUP	1	226
NUMBERTOTEXT	1	51
TIMETOTEXT	10	34
TEXTTONUMBER	10	35

For best results

- If you have a large amount of data, the profiler output could be very large or unmanageable. For this reason, it is best to profile only a subset of the original data.
- You can reduce profiler output by specifying a time limit so that nothing is reported until it exceeds the time limit that you specify.
- When running the profiler, always limit or terminate other processes running on your machine to avoid adverse results.

z/OS Batch Command Server dtxprof usage

The dtxprof utility command is available to use with the z/OS Batch Command Server. It will profile maps and analyze map execution behavior.

To use the dtxprof utility command

1. Make a copy of the JCL file that executes the map on which the dtxprof utility command will profile.

2. Change the program name from **MERCATOR** or **DTXCMDSV** to **DTXPROF** in one of the copies of the JCL file.
3. Specify the command line switches you want to use with the `dtxprof` utility command in the **PARM** field on the **EXEC** statement of the JCL.

If the **PARM** field length exceeds the 100-character limit, place the **DTXCMDSV** map command line in a file and specify `-@ddname` in the **PARM** field along with the `dtxprof` command line you want to use. For more information about the `-@ddname` command line option, see the *Command Server* documentation.

dtxprof JCL example:

The following example JCL code demonstrates a use of the `dtxprof` utility command with the z/OS Batch Command Server.

```
//STEP1 EXEC PGM=DTXPROF,REGION=0M,  
// PARM='-f -t -fs -ts -o DD:PROFOUT -dtxcmdsv "-@CMD" '  
...  
...  
//CMD DD *  
REVERSE /VX0D,X0A REVERSEI /VX0D,X0A REVERSEO  
/*  
//PROFOUT DD SYSOUT=*  
...  
...
```

`-@ddname` identifies the Data Definition (**DD**) name that points to the file containing the **DTXCMDSV** map command line.

Chapter 7. Utility Commands for map conversion

The `dsmapconv` utility command (map migration utility) is used to automatically convert new XML type definitions in maps that were generated in older versions (before 8.0) of WebSphere Transformation Extender from the command line.

The utility will update the following two components:

- a specified map source file
- all the XML type trees the map references

The conversion process will not update any XML type trees that have already been converted by another process.

`dsmapconv` utility command

The utility command for this application is `dsmapconv`.

Although the `dsmapconv` utility command converts any XML type trees for the specified map, it is recommended that you:

- convert the XML type trees using either the `dsmapconv` utility command or the XML Type Tree Compatibility Utility (**Tree Convert...** option) in the Type Designer
- make modifications to the converted XML type trees as required
- launch the map migration utility (**`dsmapconv`**)

The `.mms` file is a required field and is the file name of the map source file that needs to be converted to the current format. If the full path of the map source file name is not specified, the `dsmapconv` utility command will search for the map source file specified in the current directory.

Before the map migration utility converts the XML type trees, it creates a backup file for each of them with the same name but with the `.omt` extension.

Refer to "Troubleshooting" for details about capturing and evaluating the return code resulting from the execution of this utility command.

Syntax summary for `dsmapconv`

`dsmapconv` is the name of the utility command used to convert maps.

In the `dsmapconv` utility command, all the options are case insensitive. The following is the syntax of the `dsmapconv` utility command:

```
dsmapconv <.mms file name/location>
[-V 6.7|6.7.1|6.7.2|7.5] [-L language]
[-B <backup file name/location> [-0]]
[( -LOG [.
| <log file name/location>]
[ -FAIL] [-VERBOSE] [-APPEND])
| -NOLOG]
```

The map migration utility expects to find a map source (`.mms`) file present after the `dsmapconv` utility command when other options are also specified. Therefore, if

you do not specify an .mms file but you do specify options, the map migration utility will try to interpret the first option it encounters as an .mms file. It will fail and display a message on the console that it could not find an .mms file.

Utility command options for dsmapconv

The following command options are available with the dsmapconv utility command:

• -APPEND dsmapconv Option	• -NOLOG dsmapconv Option
• -B dsmapconv Option	• -O dsmapconv Option
• -FAIL dsmapconv Option	• -V dsmapconv Option
• -L dsmapconv Option	• -VERBOSE dsmapconv Option
• -LOG dsmapconv Option	

If you type dsmapconv with no options, the command line help that describes the command will display on your screen.

-APPEND dsmapconv option

The append (-APPEND) option of the dsmapconv utility command is used to indicate that the current dsmapconv execution messages should be appended to the dsmapconv log file if it already exists. If the dsmapconv log append (-APPEND) option is selected, the dsmapconv log messages will be appended to the file if it already exists. If it does not exist, a new log file will be created. If the (-APPEND) option is not selected, the dsmapconv log file will be created if no log file exists, or it will be overwritten if the log file does exist.

-B dsmapconv option

The backup file (-B) option of the dsmapconv utility command will have a parameter to specify the name of the backup file to be created. If no options is specified, a default backup file with the same name and the .omm extension is created.

-FAIL dsmapconv option

The fail (-FAIL) option of the dsmapconv utility command is used to indicate that only the map and related type trees that had errors during the converting should be included in the dsmapconv log file. If the log failures only (-FAIL) option is selected, the dsmapconv log will contain only the map and type trees that had failed converting. If the log failures only (-FAIL) option is not selected in the command line, all the converted results will be written to the dsmapconv log file.

-L dsmapconv option

The language (-L) option of the dsmapconv utility command is used to specify the national language setting that was used when the type tree was originally generated using the importer.

The valid values for the language argument are:

- Japanese
- Western

and are specified with the language (-L) option as:

```
[-L western | japanese]
```

If you do not specify the language (-L) option, the default is **western**.

If you specify the language (-L) option without providing the value for the *language* argument, or if you specify the language (-L) option, but provide an invalid value for the argument, the command line is invalid and the `Invalid argument in command line: language should be WESTERN or JAPANESE. error` message will display on the console.

If you specify the language (-L) option with **japanese** as the value for the *language* argument, the version of WebSphere Transformation Extender that you have installed must be the international version.

-LOG dsmapconv option

The log (-LOG) option of the dsmapconv utility command is used to enable logging. The log option has an optional parameter to specify the user-defined location or file name, or both, for the dsmapconv log file.

If the user-defined location or file name, or both, is not specified, the results of the converted type tree will appear on the screen.

If the user-defined location or file name, or both, is specified, a log file will be created containing the results of the converted type tree.

The information will be produced when the dsmapconv utility command is run.

-NOLOG dsmapconv option

The disable log (-NOLOG) option of the dsmapconv utility command is used to turn off the dsmapconv logging capability. If the disable logging option is selected, the log file will not be produced. The (-LOG) optional (-APPEND and -VERBOSE) options cannot be used with the disable log (-NOLOG) option. If these options are selected with the (-NOLOG) option, the command line is invalid, and the `Invalid argument in command line. error` message will display on the console.

-O dsmapconv option

The overwrite (-O) option of the dsmapconv utility command is used to indicate that if a file with the same name as the file specified with the backup file (-B) option exists, then that file will be overwritten. If this option is not specified and a file with the same name as the file specified with the backup file (-B) option does exist, then the conversion process is stopped.

-V dsmapconv option

The version (-V) option of the dsmapconv utility command is used to specify the version of WebSphere Transformation Extender importer that was used when the type tree was originally generated.

The valid values are:

- 6.7
- 6.7.1
- 6.7.2

- 7.5

-VERBOSE dsmapconv option

The verbose (-VERBOSE) option of the dsmapconv utility command is used to indicate if a verbose dsmapconv log should be produced. The verbose dsmapconv log will contain an entry for the map and type trees being converted, the names of the map source file and type trees, the conversion start date/time, the conversion end date/time, the converted xml file names, and the result of the dsmapconv utility command. If the verbose (-VERBOSE) option is not selected, a concise version of the dsmapconv log will be produced, which includes an entry for each map source file, type trees, type tree names, and conversion result.

Command line help for dsmapconv

There is help that describes the dsmapconv utility command usage in the command line.

To view the command line help for the dsmapconv utility command

Enter dsmapconv at the command line prompt in the *install_dir*.

The syntax and descriptions of all the options appear.

Using the dsmapconv command

The following example shows how you can use the dsmapconv utility command.

dsmapconv example

Use the dsmapconv utility command to convert new XML type definitions in maps and the type trees they reference from the command line. The following command shows an example of how it is used:

```
install_dir> dsmapconv C:\MyDev\mymapsource.mms  
-L WESTERN  
-B C:\MyDev\originalmapsource\*.mms -O  
-LOG C:\MyDev\ConversionResults\  
*.conversionresults.txt  
-FAIL -APPEND
```

The following list displays the outcomes of this utility command example:

- If the map source file and the type trees it references are converted successfully, the conversion succeeded. message will display on the console.
- The **mymapsource.mms** original map source file will be copied to the **C:\MyDev\originalmapsource** directory as a backup.
- The original **mymapsource.mms** map source file in the **C:\MyDev** directory will be overlaid by the converted version.
- If the **mymaps_conversionresults.txt** log file does not exist, it will be produced in the **C:\MyDev\ConversionResults** directory. If the **conversionresults.txt** log file already exists, only the failed maps' log messages will be appended to the existing file.

Chapter 8. dtxany2xml utility command

Use the dtxany2xml utility to produce a map that can transform any input data into XML output.

The dtxany2xml utility performs the following tasks during execution:

- Exports the input type tree as an XML Schema
- Creates an XML type tree from the generated XML Schema
- Creates a map that uses the original type tree in the input card and the new XML type tree in the output card

Syntax summary for dtxany2xml

dtxany2xml is the name of the utility command used to create a map that can transform any input data into XML output.

Use the following the syntax for the dtxany2xml utility command:

```
dtxany2xml <.mtt file name/location>  
          <exported root type path>  
          <input file name/location>  
          <output file name/location>  
          <output map name>  
          [<log file name/location>]
```

Command line help for dtxany2xml

There is help that describes the dtxany2xml utility command usage in the command line.

To view the command line help for the dtxany2xml utility command:

1. Enter dtxany2xml at the command line prompt of the installation directory. The syntax and command descriptions are displayed.

Using the dtxany2xml command

This procedure assumes that you know the type tree and type name that models the XML output you want to produce.

1. Open a command prompt and change to the product installation directory.
2. Use the following command format. All command elements are required except for the log file name and path.

```
dtxany2xml <type tree (.mtt) name and path> <root type path to export>  
<input file name and path> <new output file name (.xml) and path> <new  
map name> [<log file name and path >] When the utility runs successfully,  
you can find the expected files in the directory that you selected. No "success"  
message is displayed. A log file is generated only when the process does not  
complete successfully.
```

dtxany2xml example

The following example shows how you can use the dtxany2xml utility command.

This exercise uses sample files that are shipped with the product.

To create a map that can transform any input data into XML output

1. From a command prompt, change your directory location:
`cd c:/Program Files/IBM/WebSphere Transformation Extender 8.1/examples/general/states`
2. Enter the following command:
`dtxany2xml states.mtt Data:Input:UnitedStates sts.txt xml_output.xml XML_map XML_log.log`
This command produces the following results:
 - A new XML Schema – named using the type tree name plus the selected type name (states_UnitedStates.xsd).
 - A new map source file (XML_map.mms) that contains the new map (XML_map).
 - The input card references the original type tree (states.mtt) and input file (sts.txt).
 - The output card references the newly created XML type tree (states_UnitedStates.mtt).
 - The type is Doc XSD.
 - The output file is xml_output.xml.
 - The log file (XML_log.log) is created only when an error occurs.
3. Using Map Designer, open *install_dir*\examples\general\states\XML_map.mms.
4. Build and run XML_map.

The input (sts.txt) is converted to XML output (xml_output.xml).

Chapter 9. Utility Commands for Resource Registry

The **Resource Registry** utility command enables you to modify a resource value in the **MRN (.mrn)** and **MRC (.mrc)** files using various command options on the command line.

The utility command for this application is:

- **ResourceRegistryHelper.bat** on Windows and
- **ResourceRegistryHelper.sh** on UNIX

Resource Registry utility command

To modify a resource value in the **MRN (.mrn)** and **MRC (.mrc)** files from the command line, use the **ResourceRegistryHelper.bat** (Windows) or **ResourceRegistryHelper.sh** (UNIX) command. The **Resource Registry** utility command returns 0 if the analysis is successful, and 1 if it is not successful. It is a batch operation and can also be used for automation.

See "Troubleshooting" for details about capturing and evaluating the return code resulting from the execution of this utility command.

Syntax summary for the resource registry

ResourceRegistryHelper.bat on Windows and **ResourceRegistryHelper.sh** on UNIX are the names of the utility command used to modify a resource value in the **MRN (.mrn)** and **MRC (.mrc)** files.

All the options in the **Resource Registry** utility command are case insensitive. For UNIX, use **ResourceRegistryHelper.sh** instead of **ResourceRegistryHelper.bat**. The following is the syntax of the **Resource Registry** utility command:

```
resourceregistryhelper.bat <.mrn or .mrc location/file name>
  [-A [MODIFY [-R location/resource name]
        [-S location/server name]
        [-V value]
        [-E]]
  |[ADD [-R location/resource name]
        [-S location/server name]]]
  |[REMOVE [-R location/resource name]
        [-S location/server name]]]
```

Utility command options for Resource Registry

The following command options are available with the **Resource Registry** utility command:

• -A Resource Registry Option	• -S Resource Registry Option
• -E Resource Registry Option	• -V Resource Registry Option
• -R Resource Registry Option	

If you type `ResourceRegistryHelper.bat` with no options, the command line help that describes the command will display the command summary on the console.

-A Resource Registry option

The action (-A) option of the **Resource Registry** utility command indicates the type of update action (MODIFY, ADD or REMOVE) you want to perform on a specified resource value in the **MRN (.mrn)** and **MRC (.mrc)** files.

Valid update action options:

Option	Description
MODIFY	<ul style="list-style-type: none">• Modify a specified Resource, Server, or both Resource and Server• Valid options are -R or -S, or both, -V, and -E
ADD	<ul style="list-style-type: none">• Add a specified Resource or Server• Valid options are -R or -S
REMOVE	<ul style="list-style-type: none">• Remove a specified Resource or Server• Valid options are -R or -S

-E Resource Registry option

The encryption (-E) option of the **Resource Registry** utility command allows you to turn encryption on for a specified resource value in the **MRN (.mrn)** and **MRC (.mrc)** files. The presence of the -E option specifies that encryption is to be turned on. If you do not specify the -E option, there will be no encryption for the specified resource value; this is the default. This command option is only used when the MODIFY command option is specified.

-R Resource Registry option

The resource name (-R) option of the **Resource Registry** utility command allows you to specify the resource you want to modify, add or remove in the **MRN (.mrn)** and **MRC (.mrc)** files. Specify the location or the resource name, or both, for the *resource_name* argument.

-S Resource Registry option

The server name (-S) option of the **Resource Registry** utility command allows you to specify the server you want to modify, add or remove in the **MRN (.mrn)** and **MRC (.mrc)** files. Specify the location or the server name, or both, for the *server_name* argument.

-V Resource Registry option

The resource value (-V) option of the **Resource Registry** utility command allows you to specify the resource value you want to modify in the **MRN (.mrn)** and **MRC (.mrc)** files. This command option is only used when the MODIFY command option is specified. Specify the location or the resource value, or both, for the *value* argument.

Command line help for Resource Registry

There is help that describes the **Resource Registry** utility command usage in the command line.

To view the command line help for the Resource Registry utility command

Enter `ResourceRegistryHelper.bat` at the command line prompt in the *install_dir*.

The syntax and descriptions of all the options appear.

Using the Resource Registry command

Use the **Resource Registry** utility command to modify a resource value in the **MRN (.mrn)** and **MRC (.mrc)** files using various command options on the command line.

To use the Resource Registry utility command

1. Navigate to the folder in which the installation resides.

The *install_dir*\ path will appear.

2. On the command line next to the *install_dir*\ path, enter the utility command and command options according to the syntax described in "Syntax Summary for the Resource Registry" and Syntax Summary for the Resource Registry beginning with `resourceregistryhelper.bat`.

```
install_dir\resourceregistryhelper.bat ... < your command options >
```

3. Click **Enter**.

The **Resource Registry Helper** functionality is started and action is performed based on the command options you entered.

Resource Registry utility command examples

This information describes examples of different ways to use the **Resource Registry** utility command:

- "Modify examples"
- "Add examples" on page 74
- "Remove examples" on page 74

In Error Messages, the error messages that you might encounter if you use the **Resource Registry** utility command incorrectly, are listed with references to the correct command syntax.

Modify examples

These examples show how to use the **Resource Registry** utility command to modify a resource value in the **MRN (.mrn)** file.

These examples are for Windows platforms. For UNIX, use `ResourceRegistryHelper.sh` instead of `ResourceRegistryHelper.bat`.

- Modifying a specific Resource (test) for the resource file and turning encryption on (**modifyingresource.mrn**)

```
install_dir\resourceregistryhelper.bat modifyingresource.mrn  
-A modify  
-R test -E
```

- Modifying a specific Server (aix) for the resource file and turning encryption on (**modifyingresource.mrn**)

```
install_dir\resourceregistryhelper.bat modifyingresource.mrn
-A modify
-S aix -E
```

- Modifying a specific Server (windows) for the resource file (**modifyingresource.mrn**) without encryption

```
install_dir\resourceregistryhelper.bat modifyingresource.mrn
-A modify
-S windows
```

- Modifying a specific resource value (test) for the for all the servers (c:\server\location) in a specific resource (**modifyingresource.mrn**) without encryption

```
install_dir\resourceregistryhelper.bat modifyingresource.mrn
-A modify
-R test
-V c:\server\location
```

Add examples

These examples show how to use the **Resource Registry** utility command to add a resource value in the **MRN (.mrn)** file.

These examples are for Windows platforms. For UNIX, use ResourceRegistryHelper.sh instead of ResourceRegistryHelper.bat.

- Adding a specific Resource (test) to the resource file (**addingintoresource.mrn**)

```
install_dir\resourceregistryhelper.bat addingintoresource.mrn
-A add
-R test
```

- Adding a specific Server (aix) to the resource file (**addingresource.mrn**)

```
install_dir\resourceregistryhelper.bat addingintoresource.mrn
-A add
-S aix
```

- Adding a specific Server (windows) to the resource file (**addingresource.mrn**)

```
install_dir\resourceregistryhelper.bat addingintoresource.mrn
-A add
-S windows
```

Remove examples

These examples show how to use the **Resource Registry** utility command to remove a resource value in the **MRN (.mrn)** file.

These examples are for Windows platforms. For UNIX, use ResourceRegistryHelper.sh instead of ResourceRegistryHelper.bat.

- Removing a specific Resource (test) in the resource file (**removinginresource.mrn**)

```
install_dir\resourceregistryhelper.bat removinginresource.mrn
-A remove
-R test
```

- Removing a specific Server (aix) in the resource file (**removinginresource.mrn**)

```
install_dir\resourceregistryhelper.bat removinginresource.mrn
-A remove
-S aix
```

- Removing a specific Server (windows) in the resource file (**removinginresource.mrn**)

```
install_dir\resourceregistryhelper.bat removinginresource.mrn
-A remove
-S windows
```

Error messages

If you encounter the following error messages, see the related topics for the correct syntax for the **Resource Registry** utility command.

Error Messages

Syntax

The command line is missing argument for -S option

"-S Resource Registry Option"

The command line is missing argument for -R option

"-R Resource Registry Option"

The command line is missing argument for -V option

"-V Resource Registry Option"

There are no arguments present in the command line

"Syntax Summary for the Resource Registry"

The specified argument for teh ACTION command is invalid

"-A Resource Registry Option"

The command line is missing argument for -A option

"-A Resource Registry Option"

ADD or REMOVE must not have -V or -E in the command line

"-A Resource Registry Option"

MODIFY must have a value and a [Resource or server] in the command line

"-A Resource Registry Option"

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
577 Airport Blvd., Suite 800
Burlingame, CA 94010
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries, or both:

AIX
AIX 5L
AS/400
Ascential
Ascential DataStage
Ascential Enterprise Integration Suite
Ascential QualityStage
Ascential RTI
Ascential Software
Ascential
CICS
DataStage
DB2
DB2 Universal Database
developerWorks
Footprint
Hiperspace
IBM
the IBM logo
ibm.com
IMS
Informix
Lotus
Lotus Notes
MQSeries
MVS
OS/390
OS/400
Passport Advantage
Redbooks
RISC System/6000
Roma
S/390
System z
Trading Partner
Tivoli

WebSphere
z/Architecture
z/OS
zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

This product includes software developed by the Eclipse Project (<http://www.eclipse.org/>).



IBM WebSphere Transformation Extender, Version 8.1

Index

Special characters

- A option
 - mcompile 24
 - resourceregistryhelper 72
 - sdeploy 39
- ANALYZE option
 - sdeploy 39
- APPEND option
 - dsmapconv 66
 - dsxmlconv 20
 - mcompile 27, 66
 - mexport 35
 - mimport 32
 - msdexport 52
 - msdimport 48
 - sdeploy 44
 - tanalyze 7
 - texport 16
 - timport 9
- ASVR option
 - msdexport 53
 - msdimport 49
- ASYS option
 - msdexport 53
 - msdimport 49
- B option
 - dsmapconv 66
 - dsxmlconv 20
- BUILDMAPS option
 - sdeploy 40
- BYTEORDER option
 - timport 9
- CHARSET option
 - timport 9
- CICS option
 - timport 9
- DEPLOY option
 - sdeploy 40
- E option
 - mcompile 25
 - resourceregistryhelper 72
- EMODE option
 - sdeploy 39
- EPWD option
 - msdexport 53
- FAIL option
 - dsmapconv 66
 - dsxmlconv 20
 - mcompile 27, 66
 - mexport 35, 60, 72
 - mimport 31
 - msdexport 53
 - msdimport 49
 - sdeploy 44
 - tanalyze 7
 - texport 16
 - timport 10
- GENERATE option
 - sdeploy 40
- HELP option
 - timport 10
- IMP option
 - timport 10
- K option
 - mcompile 24
 - sdeploy 39
- KEEPMTS option
 - timport 10
- L option
 - dsmapconv 66
 - dsxmlconv 19
 - mcompile 24, 66
 - sdeploy 39
 - tanalyze 6
- LANG option
 - timport 10, 12
- LOG option
 - dsmapconv 67
 - dsxmlconv 20
 - mcompile 27, 67
 - mexport 34
 - mimport 31
 - msdexport 53
 - msdimport 49
 - sdeploy 43
 - tanalyze 6
 - texport 16
 - timport 10
- NO option
 - mcompile 26
 - mexport 34
 - mimport 31
 - msdexport 54
 - msdimport 50
 - sdeploy 42
 - texport 15
 - timport 11
- NOLOG option
 - dsmapconv 67
 - dsxmlconv 21
 - mcompile 28, 67
 - mexport 35
 - mimport 32
 - msdexport 54
 - msdimport 50
 - sdeploy 45
 - tanalyze 7
 - texport 17
 - timport 11
- O option
 - dsxmlconv 20
 - mcompile 25, 67
 - mexport 33
 - mimport 30
 - msdexport 54
 - msdimport 50
 - sdeploy 41
 - texport 15
 - timport 11
- OSVR option
 - msdimport 50

- OSYS option
 - msdimport 51
- P option
 - mcompile 25
 - sdeploy 41
- R option
 - mcompile 26
 - resourceregistryhelper 72
 - sdeploy 42
 - tanalyze 6
- REFSVR option
 - msdexport 54
 - msdimport 51
- REFSYS option
 - msdexport 55
 - msdimport 51
- S option
 - dsxmlconv 19
 - resourceregistryhelper 72
 - tanalyze 6
- SERVER option
 - sdeploy 41
- T option
 - dsxmlconv 19
 - texport 15
- USER PW -IP option
 - sdeploy 41
- V option
 - resourceregistryhelper 72
- VERBOSE option
 - dsmapconv 68
 - dsxmlconv 20
 - mcompile 28, 68
 - mexport 35
 - mimport 32
 - msdexport 55
 - msdimport 51
 - sdeploy 44
 - tanalyze 7
 - texport 17
 - timport 12

C

- command options
 - dsmapconv 66
 - dsxmlconv 18
 - dtxprof 59
 - mcompile 24
 - mexport 33
 - mimport 30
 - msdexport 52
 - msdimport 48
 - Resource Registry 71
 - sdeploy 38
 - tanalyze 5
 - texport 15
 - timport 9
- COPYBOOK importer option
 - timport 13

D

- dsmapconv
 - APPEND option 66
 - B option 66

- dsmapconv (*continued*)
 - FAIL option 66
 - L option 66
 - LOG option 67
 - NOLOG option 67
 - O option 67
 - V option 67
 - VERBOSE option 68
 - description 65
 - example 68
 - help 68
 - syntax summary 65
 - using 68

- dsxmlconv
 - APPEND option 20
 - B option 20
 - FAIL option 20
 - L option 19
 - LOG option 20
 - NOLOG option 21
 - O option 20
 - S option 19
 - T option 19
 - V option 19
 - VERBOSE option 20
 - description 18
 - help 21
 - syntax summary 18
 - using 21

- dtxany2xml 69
 - description 69
 - example 69
 - help 69
 - syntax summary 69

- dtxpage
 - description 57
 - help 58
 - syntax summary 57
 - using 58
 - with z/OS Batch Command Server 58

- dtxprof
 - comma-delimited output option 60
 - compiled map name option 61
 - description 59
 - function times exceeding x milliseconds option 60
 - function times option 60
 - function times summary output option 60
 - help 61
 - output file name option 60
 - syntax summary 59
 - type times exceeding x milliseconds option 60
 - type times option 60
 - type times summary output option 60
 - with z/OS Batch Command Server 63

E

- ERRORLEVEL environment variable 3

H

- help
 - dsxmlconv 21
 - dtxany2xml 69
 - dtxpage 58, 68, 73
 - dtxprof 61

- help (*continued*)
 - mcompile 28
 - mexport 35
 - mimport 32
 - msdexport 55
 - msdimport 51
 - Resource Registry 73
 - sdeploy 45
 - tanalyze 7
 - texport 17
 - timport 13

M

- Map Profiler utility 59
 - creating full output report for function and type information 61
 - creating summary output report for function and type information 63
 - defining Function Times in the output 61
 - defining Type Names in the output 61
 - dtxprof syntax summary 59
 - profiling RUN Maps 61

- mcompile
 - A option 24
 - APPEND option 27
 - E option 25
 - FAIL option 27
 - K option 24
 - L option 24
 - LOG option 27
 - NO option 26
 - NOLOG option 28
 - O option 25
 - P option 25
 - R option 26
 - VERBOSE option 28
 - command options 24, 66
 - description 23
 - help 28
 - syntax summary 23
 - using 28

- mexport
 - APPEND option 35
 - FAIL option 35
 - LOG option 34
 - NO option 34
 - NOLOG option 35
 - O option 33
 - VERBOSE option 35
 - command options 33, 59
 - description 33
 - help 35
 - syntax summary 33
 - using 36

- mimport
 - APPEND option 32
 - FAIL option 31
 - LOG option 31
 - NO option 31
 - NOLOG option 32
 - O option 30
 - VERBOSE option 32
 - command options 30
 - description 30
 - help 32
 - syntax summary 30

- mimport (*continued*)
 - using 32
- msdexport
 - APPEND option 52
 - ASVR option 53
 - ASYS option 53
 - EPWD option 53
 - FAIL option 53
 - LOG option 53
 - NO option 54
 - NOLOG option 54
 - O option 54
 - REFSVR option 54
 - REFSYS option 55
 - VERBOSE option 55
 - command options 52
 - description 52
 - help 55
 - syntax summary 52
 - using 55

- msdimport
 - APPEND option 48
 - ASVR option 49
 - ASYS option 49
 - FAIL option 49
 - LOG option 49
 - NO option 50
 - NOLOG option 50
 - O option 50
 - OSVR option 50
 - OSYS option 51
 - REFSVR option 51
 - REFSYS option 51
 - VERBOSE option 51
 - command options 48
 - description 48
 - help 51
 - syntax summary 48
 - using 51

- MTS importer option
 - timport 12

O

- overview
 - utility commands 5, 23, 37, 57, 65, 71

R

- Resource Registry
 - A option 72
 - E option 72
 - R option 72
 - S option 72
 - V option 72
 - action option 72
 - add examples 74
 - description 71
 - encryption option 72
 - error messages 75
 - help 73
 - modify examples 73
 - remove examples 74
 - resource name option 72
 - resource value option 72
 - server name option 72

Resource Registry (*continued*)
 syntax summary 71
 using 73
 ResourceRegistryHelper
 see also Resource Registry 71
 ResourceRegistryHelper.bat
 description 71
 ResourceRegistryHelper.sh
 description 71

S

sdeploy
 -A option 39
 -ANALYZE option 39
 -APPEND option 44
 -BUILDMAPS option 40
 -DEPLOY option 40
 -EMODE option 39
 -FAIL option 44
 -GENERATE option 40
 -K option 39
 -L option 39
 -LOG option 43
 -NO option 42
 -NOLOG option 45
 -O option 41
 -P option 41
 -R option 42
 -SERVER option 41
 -USER PW -IP option 41
 -VERBOSE option 44
 command options 38
 description 37
 help 45
 syntax summary 37
 using 45
 syntax summary
 dsxmlconv 18
 dtxpage 57, 65, 71
 dtxprof 59
 mcompile 23
 mexport 33
 mimport 30
 msdexport 52
 msdimport 48
 sdeploy 37
 tanalyze 5
 texport 14
 timport 8

T

tanalyze
 -APPEND option 7
 -FAIL option 7
 -L option 6
 -LOG option 6
 -NOLOG option 7
 -R option 6
 -S option 6
 -VERBOSE option 7
 command options 5, 71
 description 5
 help 7
 syntax summary 5

tanalyze (*continued*)
 using 8
 texport
 -APPEND option 16
 -FAIL option 16
 -LOG option 16
 -NO option 15
 -NOLOG option 17
 -O option 15
 -T option 15
 -VERBOSE option 17
 command options 15, 18
 description 14
 help 17
 syntax summary 14
 using 17
 timport
 -APPEND option 9
 -BYTEORDER option 9
 -CHARSET option 9
 -CICS option 9
 -FAIL option 10
 -HELP option 10
 -IMP option 10
 -KEEPMTS option 10
 -LANG option 10
 -LOG option 10
 -NO option 11
 -NOLOG option 11
 -O option 11
 -VALIDATION option 12
 -VERBOSE option 12
 command options 9
 COPYBOOK importer option 13
 description 8
 help 13
 MTS importer option 12
 syntax summary 8
 using 13
 XMLDTD importer option 13
 XMLSCHEMA importer option 13
 troubleshooting
 utility commands 3

U

using 69

X

XML Type Tree Compatibility Utility 18
 XMLDTD importer option
 timport 13
 XMLSCHEMA importer option
 timport 13



Printed in USA