

IBM WebSphere Transformation Extender



Execution Commands

Version 8.1

Note

Before using this information, be sure to read the general information in "Notices" on page 41.

October 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. Execution maps	1
Command Server overview	1
Coordinating execution commands with compiled settings	1
Compiled map files	2
Command files	3
Using Command Server commands.	3
General rules for execution commands.	4
From a command line	4
From the Platform API	5
From a RUN() function in a map rule	5
Chapter 2. Execution commands	7
List of execution commands	7
MapAudit (-A)	8
Close Window (-B)	11
CenturyCentury (-D)	11
Fail on Warnings (-F)	11
Ignore Validation Options (-G)	12
Input Source Override - Application (-IA)	13
Input Source Override - Database (-ID)	15
Input Source Override - Echo (-IE)	16
Input Source Override - File (-IF)	17
Input Source Override - Message (-IM)	19
List (-L)	20
Initialization File Override (-N)	21
Output Target Override - Application (-OA)	21
Output Target Override - Database (-OD)	22
Output Target Override - Echo (-OE)	23
Output Target Override - File (-OF)	24
Output Target Override - Message (-OM)	25
WorkSpace PageSize (-P)	27
Refresh Status (-R)	27
MapTrace (-T)	28
Stop Validation (-V)	28
WorkSpace (-W)	29
Retry (-Y)	30
Ignore Warnings (-Z)	30
Command availability	31
Chapter 3. Examples	33
Running a map located in a different directory	33
Creating work files in an alternate directory	33
Executing multiple maps from a single command	34
Overriding a database target.	34
Executing a command file	35
Using a run function and echoed outputs in memory	35
Audit Log in memory	36
Chapter 4. INI commands for debugging maps	37
Logging	37
Logging categories	37
Debug	39
Notices	41

Programming interface information	43
Trademarks and service marks	43
Index	45

Chapter 1. Execution maps

This documentation discusses the various ways to use execution commands and their options when executing maps. Execution commands are used to specify how to execute a map.

Command Server overview

Use execution commands to override the map settings or card settings in a compiled map when a map is run using one of the following methods:

- from a command line prompt
- from the Map Designer
- from the Command Server user interface
- using the RUN() function in a map rule
- using the API

Use the map settings and input and output card settings of the Map Designer or the execution settings in the Integration Flow Designer to define map execution. These map execution settings are included in the compiled map file. Many settings compiled into the map can be overridden (or partially overridden) using executions commands and options.

Some execution commands are limited because they can only be used with a Command Server and cannot be used from within a RUN() function or using the Platform API. Also, some commands are not available on all platforms. See "Command Availability" for more information.

Coordinating execution commands with compiled settings

The following table lists valid execution commands and the **MapSetting** or input/output card settings that they override.

Map settings

Execution Command	Map Setting
-------------------	-------------

MapAudit (-A)	MapAudit
----------------------	----------

Close Window (-B)	-
--------------------------	---

Century (-D)	Century
---------------------	---------

Fail on Warnings (-F)	Warnings
------------------------------	----------

Ignore Validation Options (-G)	Validation > RestrictionError, SizeError, PresentationError
---------------------------------------	---

List (-L)	-
------------------	---

Initialization File Override (-N)

-

WorkSpace Page Size (-P)

WorkSpace > PageSize

Refresh Status (-R)

-

MapTrace (-T)

MapTrace

Stop Validation (-V)

Validation > OnValidationError

WorkSpace (-W)

WorkSpace

Retry (-Y)

Retry

Ignore Warnings (-Z)

Warnings

Card settings

Execution Command

Card Setting

Input Overrides (-I)

WorkFiles (W)

SourceRule > FetchAs > WorkArea

Retry (R)

SourceRule > GET > Retry

Rollback (B)

SourceRule > GET > Transaction > OnFailure

Delete (X)

SourceRule > GET > Transaction > OnSuccess

Output Overrides (-O)

Retry (R)

TargetRule > PUT > Target > Retry

Append (+)

TargetRule > PUT > Transaction > Target > OnSuccess

Delete (X)

TargetRule > PUT > Transaction > OnSuccess

Rollback (B)

TargetRule > PUT > Transaction > OnFailure

Compiled map files

You can create compiled map (.mmc) files using the Map Designer, Integration Flow Designer, or Command Server commands.

Map Designer

Using the Map Designer, specify how you want each map to run. These specifications, which are compiled into the map file, include map settings and card settings, as well as execution command overrides for RUN() functions which

execute a map from a map rule. See the *Map Designer* documentation for more information about map and card settings and the *Functions and Expressions* documentation for information about the RUN() function.

Integration Flow Designer

Using the Integration Flow Designer, you can create systems of maps and define execution settings, map settings, and input/output settings. These settings are recorded into the compiled map file that results from building the map within the Integration Flow Designer.

If the system has an execution mode of Command, a command file can be generated that can be executed with a Command Server. Edit this command file using the appropriate execution commands and options.

The following is an example of a command file generated by the Integration Flow Designer for a system named **Inbound EDI**:

```
;
;
; Date:   Wed Jun 16 18:15:15 2001 (Version 6.0(126))
; File:   D:\MapsAndTrees\ANSIACK\ProcessEDI.msd
; System: Inbound EDI
;
; Component: ReceiveEDI
; Map File: D:\MapsAndTrees\ANSIACK\Audit997.mms
; Executable Map: ReceiveEDI
;
D:\MapsAndTrees\ANSIACK\ReceiveEDI.mmc
  -GR
;
; Component: ACK997
; Map File: D:\MapsAndTrees\ANSIACK\Audit997.mms
; Executable Map: ACK997
;
D:\MapsAndTrees\ANSIACK\ACK997.mmc
  -OF1BR10:2 'OutboundACK.EDI'
;
; End System: Inbound EDI
;
```

Notice that the execution settings defined in an Integration Flow Designer command file correspond with the execution commands and options. Refer to the *Integration Flow Designer* documentation for more information.

Command files

You can create command files that contain listings of maps to execute with execution commands that specify how you want each map to run. A command file is a text file that can contain multiple maps to run, as well as multiple execution commands and options for each map; however, a command file cannot run another command file.

Using Command Server commands

When you run a map, you can:

- Override sources and targets for the cards used in the map
- Create trace and audit log files
- Control a variety of characteristics of the run-time environment

The execution commands entered when you run a map do not permanently affect the compiled map file.

The command options that you specify override the settings compiled into the map file. If you do not use a specific execution command, the corresponding map and card settings compile into the map.

General rules for execution commands

The general rules that apply when specifying execution commands are:

- Each execution command must begin with a hyphen.
- At least one space is required between commands (for example, `-T -V`).
- There are no spaces within an option except for the commands that allow you to override sources and targets (for example, `-I` and `-O`), which require a space before the source or target name.
- If a source or target name contains a space, enclose the entire name in single quotes.
- Execution commands and options are not case-sensitive.
- If spaces or single quotes can occur in echoed data specified using the `-IE` execution command, the size option should be used (`-IExS`).
- Multiple options can be specified for a single map.
- Command files can contain multiple lines. Line breaks can be used to replace the required space that must exist between commands. The line breaks are converted to spaces before processing.

From a command line

You can execute a particular map or a command file containing a list of maps to execute from the command line of the system hosting your Command Server. The syntax is:

```
DTXCMSDV {map [exec_command...]|@command_file}...
```

Ensure your current directory is where the Command Server is located, or specify the relative path or full path, as applicable, when entering the command.

Option Description

DTXCMSDV

Specifies the name of the Command Server execution command for both UNIX and Windows platforms.

map Specifies the name of the compiled map (**.mmc**) file to run. If applicable, include the full path or a relative path to the directory that contains the map file.

exec_command

Specifies any execution command and its associated options that you want to use to override the settings compiled into the map file when the specified map is run.

command_file

Specifies the name of a command file that contains a list of maps to execute. If applicable, include the full path or a relative path to the directory that contains the command file.

Command line help

You can access Help from the command line by typing:

```
DTXCMSDV -?
```

From the Platform API

To run the server using a Platform API, as part of an argument to the **RunMap()** function, specify:

```
map [exec_command...]|@command_file
```

Option Description

map Specify the name of the compiled map (**.mmc**) file to run. If applicable, include the full path or a relative path to the directory that contains the map file.

exec_command

Specify any execution command and its associated options you want to override the settings compiled into the map file when the specified map is run.

command_file

Specify the name of a command file that contains a list of maps to execute. If applicable, include the full path or a relative path to the directory that contains the command file.

From a RUN() function in a map rule

To execute a map from a map rule, use the **RUN()** function as follows:

```
RUN ( "map", "exec_command..." )
```

Option Description

map Specify the name of the compiled map (**.mmc**) file to run. If applicable, include the full path or a relative path to the directory that contains the map file.

exec_command

Specify any execution command and its associated options you want to override the settings compiled into the map file when the specified map is run.

Chapter 2. Execution commands

This documentation discusses the execution commands and associated options, including the syntax of each command and its particular usage.

List of execution commands

The following is a list of execution commands and the command syntax.

Name	Syntax
MapAudit (-A)	-A [D[R W]][E[R W]][B[R W]] [C[R W]][P[R W]][M[S] [U][+ !+][={file dir}]]
Close Window (-B)	-B
CenturyCentury (-D)	-D [ccyy 0]
Fail on Warnings (-F)	-F ![warning_code[:warning_code...]]
Ignore Validation Options (-G)	-G [R][P][S][T]
Input (Source) Overrides	
Input Source Override - Application (-IA)	-IA [alias]card_num[W !W][Rc:i] [B][K['filename'] +'filename'] [U'filepath'] [U] [1'filename'] [1+'filename'] [U1'filepath'] [U1]]{src-adptr-cmd .}
Input Source Override - Database (-ID)	-ID card_num[W !W][Rc:i][B] [K['filename'] +'filename'] [U'filepath'] [U] [1'filename'] [1+'filename'] [U1'filepath'] [U1]]{src-adptr-cmd .}
Input Source Override - Echo (-IE)	-IE card_num[Ssize][W !W] source
Input Source Override - File (-IF)	-IF card_num[W !W][X X0][Rc:i] [B][K['filename'] +'filename'] [U'filepath'] [U] [1'filename'] [1+'filename'] [U1'filepath'] [U1]]{source .}
Input Source Override - Message (-IM)	-IM [alias]card_num[W !W][X X0] [Rc:i][B][K['filename'] +'filename'] [U'filepath'] [U] [1'filename'] [1+'filename'] [U1'filepath'] [U1]]{src-adptr-cmd .}
List (-L)	-L
Initialization File Override (-N)	-N =[<'INI_filename'>]
Output (Target) Overrides	
Output Target Override - Application (-OA)	-OA [alias]card_num[Rc:i][B] [K['filename'] +'filename'] [U'filepath'] [U] [1'filename'] [1+'filename'] [U1'filepath'] [U1]]{target-adptr-cmd .}

Name	Syntax
Output Target Override - Database (-OD)	-OD <i>card_num</i> [Rc:i][B][K ['filename'] [+ 'filename'] [U 'filepath'] [U] [1 'filename'] [1 + 'filename'] [U1 'filepath'] [U1]]{ <i>target-adptr-cmd</i> \.}
Output Target Override - Echo (-OE)	-OE <i>card_num</i> [S X]
Output Target Override - File (-OF)	-OF <i>card_num</i> [+ !+][X X0][Rc:i] [B][K ['filename'] [+ 'filename'] [U 'filepath'] [U] [1 'filename'] [1 + 'filename'] [U1 'filepath'] [U1]]{ <i>target</i> \.}
Output Target Override - Message (-OM)	-OM [<i>alias</i>] <i>card_num</i> [X X0][Rc:i] [B][K ['filename'] [+ 'filename'] [U 'filepath'] [U] [1 'filename'] [1 + 'filename'] [U1 'filepath'] [U1]]{ <i>target-adptr-cmd</i> \.}
WorkSpace PageSize (-P)	-P <i>size:count</i>
Refresh Status (-R)	-R <i>update_rate</i>
MapTrace (-T)	-T [I IC <i>card_num</i> I <i>from[:to]</i>] [O OC <i>card_num</i>][S][= <i>dir</i>]
Stop Validation (-V)	-V
WorkSpace (-W)	-W [M] {[D][U][= <i>dir</i>]}
Retry (-Y)	-Y <i>count:interval</i>
Ignore Warnings (-Z)	-Z [!][<i>warning_code[:warning_code...]</i>]

MapAudit (-A)

Use the MapAudit execution command (-A) to control the creation of audit log information. Using this command, audit information is recorded which can include data audit information and execution audit information.

When used *with* other specified options, the **MapAudit** settings specified with the MapAudit command (-A) override *all* **MapAudit** settings compiled into the map. When used *without* other specified options, no audit log is produced.

```
-A[D[R|W] {1|2}] [E[R|W]] [B[R|W]] [C[R|W]] [P[R|W]]
  [M[S] | [U] [+ | !+] [= {file | dir}]]
```

For information about the audit log files, refer to the *Map Designer* documentation.

Option Description

D Include data audit information in the audit log. The recorded information is based on the **MapAudit** settings configured in the Map Designer for a particular map.

The sub-options for D include: R, W, and 1 or 2

R Generate audit files each time the map is run only if the outcome of the map's execution is an error condition. Used with the burst audit data information (D), execution summary audit information (E), burst execution information (B), card (data) settings (C), and map settings (P) options.

W Generate audit files each time the map is run only if the outcome of the map's execution is a warning or an error condition. Used with the burst

audit data information (D), execution summary audit information (E), burst execution information (B), card (data) settings (C), and map settings (P) options.

1|2 1 is the audit code for fail size (**WrongSize**).

2 is the audit code for fail minimum size or fail maximum size (**TooLongTooShort**).

Note If using the 1 or 2 option as well as the R or W option with the D sub-option, the 1 or 2 must follow the R or W. For example:

-AEDW2

This example represents SummaryAudit Execution, SummaryAudit Data, OnWarning, and TooLongTooShort.

B The burst execution information includes:

- the burst return value
- the time it took for the burst to execute
- the number of bytes, the adapter return value, and the content return value for each card

Example:

```
<ExecutionLog burstreturn="0" ElapsedSec="0.6846">
  <inputstatus card="1" bytes="4904"
  adapterreturn="0"          contentreturn="0"/>
  <inputstatus card="2" bytes="270"
  adapterreturn="0"          contentreturn="0"/>
  <outputstatus card="1" bytes="18695
  adapterreturn="0"          contentreturn="0"/>
</ExecutionLog>
```

E The execution summary audit information includes:

- return codes resulting from the map execution
- execution time (in seconds)
- command line used to execute the map
- number of input objects found and output objects built (Command Server only)
- details about the map sources and destinations
- work file names and sizes

C The card (data) settings audit information includes:

- the data settings for sources and target cards

P The map settings audit information includes:

- the map settings

M The memory setting can be used only when the map is run by the RUN() function or an API. If memory is set, audit information is stored in memory and appended to the last set of data returned during map execution.

When the map is run by a Command Server or Launcher, this setting is ignored and the audit log is created in a file in the map directory, using the **<mapname>.log** naming convention.

If the audit in memory fails or there is insufficient available memory, a Disk write error or Not enough memory to execute map message can be returned.

- S** Only use with the memory option (**M**). This option prefixes the audit data with a text number indicating the size (in bytes) of the audited data followed by a space.

This option causes the resulting audit log entry to be returned as an ASCII number followed by a space, then followed by the specified number of bytes of data. This helps to distinguish it from multiple character strings in the returned data when at least one output and the audit log are echoed.

For example, if used in a **RUN()** function:

```
RUN ("mymap.mmc","-OE3S -ADEMS")
```

If the data resulting from output card #3 was This and That, the string returned by this **RUN()** function is:

```
13 This and That1031 BEGIN RUN Function for Windows....
```

- U** Generate audit files with unique names each time the map is run. The name of the audit log for top-level maps is **mer_<mapname>_<process_key>_<map_counter>.log**, where *map_counter* is a value representing each unique instance of a map within the same process. Run maps use **run_<mapname>_<process_key>_<map_counter>.log**. Unique names are globally unique; each process has a unique value.

Unique log file names are typically used in a multi-threaded environment so that more than one instance of the same map can run at the same time. In a command environment, unique log file names are typically used when more than one command specifies the same map to run and you do not want one log file to overwrite the other. Or, use this option when multiple servers run the same map. This option prevents one log file from overwriting another log file.

This option cannot be used with the append option (+).

- +** Each time the map is run, append specified information to the audit log. If no audit log exists at the start of map execution, an audit log is created. Turn off the + option. Discontinue appending specified information to the audit log. This option cannot be used with the **U** option for unique names.
- !+** Turn off the + option. Discontinue appending specified information to the audit log. This option cannot be used with the **U** option for unique names.

=file|dir

Specify a file name for appending audit information or specify a full directory path and file name in which you want the audit log file created. By default, the audit log file is created in the same directory as the compiled map (**.mmc**) file, using the **<mapname>.log** naming convention.

Examples

- AD** BurstAudit Data
- AD1** BurstAudit Data, WrongSize
- AD2** BurstAudit Data, TooLongTooShort
- AB** BurstAudit Execution
- AE** SummaryAudit Execution
- AC** SettingsAudit Data
- AP** SettingsAudit Map
- ABD** BurstAudit Execution and BurstAudit Data

- AEB SummaryAudit Execution and BurstAudit Execution
- ABDE BurstAudit Execution, BurstAudit Data, and SummaryAudit Execution
- AEDW2 SummaryAudit Execution, SummaryAudit Data, OnWarning, and TooLongTooShort
- AER SummaryAudit Execution and OnError
- ADW BurstAudit Data and OnWarningOrError
- AERCWPW SummaryAudit Execution and OnError and SettingsAudit Data and OnWarningOrError and SettingsAudit Map and OnWarningOrError

Close Window (-B)

When executing a map using a Command Server for Windows or AS/400 OS/400, use the Close Window execution command (-B) to exit the Command Server application after the map has completed.

-B[I]

Option Description

- I When using a Command Server for Windows, specify to exit the Command Server application **only** when the map completes successfully.

CenturyCentury (-D)

Use the Century execution command (-D) to specify the start of a 100-year range used to supply a missing century portion for a Date & Time. This command is useful for the transition to the year 2000 so that dates with an YYMMDD format can be interpreted for the correct century.

-D[ccyy|0]

Option Description

- ccyy A missing century for a Date & Time item is derived based on the *ccyy* value.
- 0 Turn off the Century execution command.

For example, if you specify **-D1960**, a YYMMDD-formatted date between 60mmdd and 99mmdd is interpreted as having a century value of 19. Dates between 00mmdd and 59mmdd are interpreted as having a century value of 20. In this example, data with a date of 580101 is assigned a century of 20; however, data with a date of 600101 is assigned a century of 19.

Fail on Warnings (-F)

Use the Fail on Warnings (-F) execution command to specify that the map should fail when any of the selected warning codes are returned. When used *with* one or more warning codes, a map-failed condition is returned for those warning codes. When used *without* other options specified, all warning codes result in a map-failed condition. Using this command impacts **OnSuccess** settings, **OnFailure** settings, and so on.

-F[!][warning_code[:warning_code...]]

Option Description

! Use to specify warning codes that, when encountered, remain in their default states—they remain as warnings and do not issue a map failed message.

When used with **-F** only (for example, **-F!**), a map-failed condition is returned if any of the warning codes is encountered.

warning_code

Specify the warning codes to be acted upon when encountered. Valid warning code selections include: 14, 18, 21, 26, 27, 28, and 29.

Examples

-F A map-failed message is returned when any of the warning codes is encountered.

-F! A map-failed message is not returned when any of the warning codes is encountered.

-F18 A map-failed message is returned when a Page Usage Count Error (18) is encountered.

-F!18 Excluding the Page Usage Count Error (18), a map-failed message is returned for all of the other warning codes.

-F18:27:28

A map-failed message is returned when any of the specified warning codes is encountered (18, 27, or 28 in this case).

If the **-F** command is used and the map fails due to the cause associated with the specified warning, the execution audit includes an Error map status with the appropriate warning code preceded by a minus sign and the map-failed message.

For example:

```
<Summary MapStatus="Error" Return="-28">
  <ElapsedTimeInSec>0.0898</ElapsedTimeInSec>
  <Message>Input type contains errors.</Message>
  <CommandLine>'install_dir\example\ansiack\receiveEDI.mmc'
</CommandLine>
  <ObjectsFound>0</ObjectsFound>
  <ObjectsBuilt>0</ObjectsBuilt>
</Summary>
```

Note: When using this command, you cannot specify the same warning code using the **-Z** command. If you do, the setting of the latter command in the command is used.

Note: For example, if your command line included **-F18:26 -Z18:27:28**, a return code of 18 (Page Use Count error) would be ignored because the Ignore Warnings command (**-Z**) follows the Fail on Warnings command (**-F**).

Ignore Validation Options (-G)

Use the Ignore Validation Options execution command (**-G**) to specify the item properties, if any that should be ignored during validation. When used *with* other specified options, only those item properties are ignored. When used *without* other specified options, no item properties are ignored.

-G[R] [P] [S] [T]

Option Description

- R Ignore restrictions for all items appearing in the input data.
Restrictions of syntax items are not ignored because they are used in determining the value of a syntax object.
- P Ignore the presentation settings of items.
For example, if a number is defined as an integer but it appears in the data as a decimal number, it validates anyway.
If the Ignore command is used and an item has an invalid presentation, it is mapped as if it were NONE. However, if the input is converted to text using the TEXT function, it is mapped.
- S Ignore the minimum and maximum size of items that appear in delimited data.
For example, if an item is defined as having a minimum of three bytes, but the data contains only two bytes, it validates anyway.
- T Ignore restrictions for all items appearing in the input data with no warnings.

Input Source Override - Application (-IA)

Use the Input Source Override - Application execution command (-IA) to override specifications in the compiled map file for a specific input card for a single execution of a map. Only those adapter settings specified are used to override the settings compiled into the map.

For example, if the WorkArea option (W or !W) is not specified, the **WorkArea** setting compiled into the map is used.

```
-IA[alias]card_num[W|!W][Rcount:interval][B]
  [K['filename']|['+'filename']|[U'filepath']|[U]|
  [1'filename']|[1+'filename']|[U1'filepath']|[U1]]
  {src-adptr-cmd}.}
```

Option Description

alias This is the adapter alias for the specific application adapter. For example, FTP is the adapter alias for the FTP adapters. See the appropriate adapter documentation in the *Resource Adapters* documentation.

card_num

This is the card number of the input to override.

W After a map runs for the first time, the work area created for the input card is not deleted. For subsequent executions of the same map, the data for this input card is *not* validated and the work area information for the card is retrieved from the existing work area.

!W The work area for input is created when map execution begins.

Rcount:*interval*

Specify **Retry** settings. If the source is unavailable, the adapter will attempt to access the source as many times as specified by the *count* setting at the interval specified by the *interval* setting.

count number of attempts to access the application

interval number of seconds to wait between attempts

Note: If this option is not specified, the **Retry** settings compiled into the map are used. To eliminate adapter retry, specify **R0:0**.

B If the map, burst, or card does not successfully complete, roll back any changes that were made to this data source.

Note: If this option is not specified, the **OnFailure** setting compiled into the map is used.

K When the map, burst, or card runs, no backup file will be created.

K'*filename'*

When the map, burst, or card runs, a backup file will always be created to *filename*.

Note: If '*filename*' is not an absolute path, the map directory will be pre-pended.

K+*filename'*

When the map, burst, or card runs, a backup file will always be appended to *filename*.

Note: If '*filename*' is not an absolute path, the map directory will be pre-pended.

KU'*filepath'*

When the map, burst, or card runs, a backup file will always be created using a unique backup name in the directory *filepath*.

Note: If '*filepath*' is not an absolute path, the map directory will be pre-pended.

KU When the map, burst, or card runs, a backup file will always be created using a unique backup name in the map directory.

K1'*filename'*

If the map, burst, or card does not successfully complete, a backup file will be created to *filename*.

Note: If '*filename*' is not an absolute path, the map directory will be pre-pended.

K1+*filename'*

If the map, burst, or card does not successfully complete, a backup file will be appended to *filename*.

Note: If '*filename*' is not an absolute path, the map directory will be pre-pended.

KU1'*filepath'*

If the map, burst, or card does not successfully complete, a backup file will be created using a unique backup name in the directory *filepath*.

Note: If '*filepath*' is not an absolute path, the map directory will be pre-pended.

KU1 If the map, burst, or card does not successfully complete, a backup file will be created using a unique backup name in the map directory.

src-adptr-cmd

This option provides the adapter-specific commands to connect to the data source and retrieve the input data. See the adapter-specific information in the *Resource Adapters* documentation.

- . Use **Command** in the compiled map file.

Input Source Override - Database (-ID)

Use the Input Source Override - Database execution command (-ID) to override specifications in the compiled map file for a specific input card for a single execution of a map. Only those adapter settings specified are used to override the settings compiled into the map.

For example, if the WorkArea option (W or !W) is not specified, the **WorkArea** setting compiled into the map is used.

```
-IDcard_num[W|!W] [Rcount:interval] [B]
  [K['filename'] | ['+'filename'] | [U'filepath'] | [U] |
  [1'filename'] | [1+'filename'] | [U1'filepath'] | [U1]]
  {src-adptr-cmd|.}
```

Option Description

card_num

This is the card number of the input to override.

W After a map runs for the first time, the work area that is created for the input card is not deleted. Then, on subsequent executions of the same map, the data for this input card is *not* validated and the work area information for the card is retrieved from the existing work area.

!W Input's work area is created when map execution begins.

Rcount:interval

Specify **Retry** settings. If the source is unavailable, the adapter attempts to access the source as many times as specified by the *count* setting at the interval specified by the *interval* setting.

count number of attempts to access the database

interval number of seconds to wait between attempts.

If this option is not specified, the **Retry** settings compiled into the map are used. To eliminate adapter retry, specify **R0:0**.

B If the map, burst, or card does not complete successfully, roll back any changes that were made to this data source. If this option is not specified, the **OnFailure** setting compiled into the map is used.

K When the map, burst, or card runs, no backup file will be created.

K'filename'

When the map, burst, or card runs, a backup file will always be created to *filename*. If *filename'* is not an absolute path, the map directory will be pre-pended.

K+'filename'

When the map, burst, or card runs, a backup file will always be appended to *filename*. If *filename'* is not an absolute path, the map directory will be pre-pended.

KU'filepath'

When the map, burst, or card runs, a backup file will always be created

using a unique backup name in the directory *filepath*. If '*filepath*' is not an absolute path, the map directory will be pre-pended.

KU When the map, burst, or card runs, a backup file will always be created using a unique backup name in the map directory.

K1'*filename'*

If the map, burst, or card does not successfully complete, a backup file will be created to *filename*. If '*filename*' is not an absolute path, the map directory will be pre-pended.

K1+''*filename'*

If the map, burst, or card does not successfully complete, a backup file will be appended to *filename*. If '*filename*' is not an absolute path, the map directory will be pre-pended.

KU1'*filepath'*

If the map, burst, or card does not successfully complete, a backup file will be created using a unique backup name in the directory *filepath*. If '*filepath*' is not an absolute path, the map directory will be pre-pended.

KU1 If the map, burst, or card does not successfully complete, a backup file will be created using a unique backup name in the map directory.

src-adptr-cmd

Specify the settings for the source using the adapter-specific commands. See the appropriate database adapter documentation in the *Resource Adapters documentation*.

. Use **Command** in the compiled map file.

Input Source Override - Echo (-IE)

Use the Input Source Override - Echo execution command (-IE) to override specifications in the compiled map file for a specific input card for a single execution of a map.

`-IEcard_num[Ssize] [W|!W] source`

Option Description

card_num

This is the card number of the input to override.

Ssize

This is the size of the data specified by *source* in bytes.

W

After a map runs for the first time, the WorkArea created for the input card is not deleted. On subsequent executions of the same map, the data for this input card is *not* validated and the WorkArea information for the card is retrieved from the existing WorkArea.

!W

The WorkArea for input is created when map execution begins.

source

This is the data to be used for the input.

For example, to pass the value ABC123XYZ to a map as the data for input card #2, type the following command:

```
-IE2 ABC123XYZ
```

If the data to be echoed contains spaces or hyphens, it must be surrounded by single quotes as shown in the following example:

```
-IE2 `ABC 123 XYZ'
```

If the echoed data contains single quotes or the null character (hex 00), use the **S** option to specify the number of bytes of echoed data. Changing the example above to use the **S** option, the single quotes are omitted:

```
-IE1S11 ABC 123 XYZ
```

The examples above demonstrate using the Input Source Override - Echo execution command (-IE) in a map rule.

If you are using the Input Source Override - Echo execution command (-IE) in the Command Server command line and the data contains spaces or hyphens, in addition to surrounding the data to be echoed with single quotes, you must then add double-quotes outside the single-quotes as shown in the following example:

```
-IE2 "'ABC 123 XYZ'"
```

Also, if you are using the Input Source Override - Echo execution command (-IE) in the Command Server command line and the data contains single quotes or the null character (hex 00), in addition to using the **S** option to specify the number of bytes of echoed data, and omitting the single quotes, if the data contained them, you must also surround the data with double-quotes as shown in the following example:

```
-IE1S11 "ABC 123 XYZ"
```

One use for echoed input data is in conjunction with the **RUN** function to pass data from one map to another. For an example of using an echoed data source as an input argument using the **RUN** function, see the **RUN** function in the *Functions and Expressions documentation*. Another example of using an echoed data source is provided in the *Platform API documentation*.

Input Source Override - File (-IF)

Use the Input Source Override - File execution command (-IF) to override specifications in the compiled map file for a specific input card for a single execution of a map. Only the specified adapter settings are used to override the settings compiled into the map.

For example, if the **WorkArea** option (**W** or **!W**) is not specified, the **WorkArea** setting compiled into the map is used.

```
-IFcard_num[W|!W][X|X0][Rcount:interval][B]
  [K['filename']|['+filename']|['U'filepath']|['U']]
  [1['filename']|['1+filename']|['U1'filepath']|['U1']]
  {source|.}
```

Option Description

card_num

This is the card number of the input to override.

W After a map runs the first time, the **WorkArea** created for the input card is not deleted. For subsequent executions of the same map, the data for this input card is *not* validated and the **WorkArea** information for the card is retrieved from the existing **WorkArea**.

!W The **WorkArea** of the input is created when map execution begins.

X If the map, burst, or card completes successfully, the input data file is deleted.

X0 If the map, burst or card completes successfully, the input data file is

deleted only if it has no content it is zero bytes. If neither of the delete options (X or X0) is specified, the **OnSuccess** setting compiled into the map is used.

Rcount:interval

Specify **Retry** settings. If the source is unavailable, the adapter attempts to access the source as many times as specified by the *count* setting at the interval specified by the *interval* setting.

count number of attempts to access the data file

interval number of seconds to wait between attempts. If this option is not specified, the **Retry** settings compiled into the map are used. To eliminate adapter retry, specify **R0:0**.

B If the map, burst, or card does not complete successfully, roll back any changes made to this data source. If this option is not specified, the **OnFailure** setting compiled into the map is used.

K When the map, burst, or card runs, no backup file will be created.

K'filename'

When the map, burst, or card runs, a backup file will always be created to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.

K+'filename'

When the map, burst, or card runs, a backup file will always be appended to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.

KU'filepath'

When the map, burst, or card runs, a backup file will always be created using a unique backup name in the directory *filepath*. If *'filepath'* is not an absolute path, the map directory will be pre-pended.

KU When the map, burst, or card runs, a backup file will always be created using a unique backup name in the map directory.

K1'filename'

If the map, burst, or card does not successfully complete, a backup file will be created to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.

K1+'filename'

If the map, burst, or card does not successfully complete, a backup file will be appended to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.

KU1'filepath'

If the map, burst, or card does not successfully complete, a backup file will be created using a unique backup name in the directory *filepath*. If *'filepath'* is not an absolute path, the map directory will be pre-pended.

KU1 If the map, burst, or card does not successfully complete, a backup file will be created using a unique backup name in the map directory.

source Specify the name of the input file. If the input file is not located in the same directory as the map file, the full path must be specified using the platform-specific syntax.

. Use the **FilePath** in the compiled map file.

Input Source Override - Message (-IM)

Use the Input Source Override - Message execution command (-IM) to override specifications in the compiled map file for a specific input card for a single execution of a map. Only the specified adapter settings are used to override the settings compiled into the map.

For example, if the WorkArea option (W or !W) is not specified, the **WorkArea** setting compiled into the map is used.

```
-IM[alias]card_num[W|!W][X|X0][Rcount:interval][B]
  [K['filename']|[+'filename']|[U'filepath']|[U]]
  [1'filename']|[1+'filename']|[U1'filepath']|[U1]]
{src-adptr-cmd}.
```

Option Description

alias This is the adapter alias for the specific messaging adapter. For example, AQ is the adapter alias for the Oracle AQ adapters. See the adapter-specific information in the *Resource Adapters documentation*. Refer to the release notes (**readme.txt**) file for platform and version-specific information.

card_num

This is the card number of the input to override.

- W** After a map runs the first time, the WorkArea created for the input card is not deleted. For subsequent executions of the same map, the data for this input card is *not* validated and the WorkArea information for the card is retrieved from the existing WorkArea.
- !W** The WorkArea of input is created when map execution begins.
- X** If the map, burst, or card completes successfully, the message is deleted from its source.
- X0** If the map, burst, or card completes successfully, the message is deleted from its source, only if it has no content. If neither delete option (X or X0) is specified, the **OnSuccess** setting compiled into the map is used.

Rcount:interval

Specify **Retry** settings. If the source is unavailable, the adapter attempts to access the source as many times as specified by the *count* setting at the interval specified by the *interval* setting.

count number of attempts to access the message queue.

interval number of seconds to wait between attempts. If this option is not specified, the **Retry** settings compiled into the map are used. To eliminate adapter retry, specify **R0:0**.

B If the map, burst, or card does not complete successfully, roll back any changes made to this data source. If this option is not specified, the **OnFailure** setting compiled into the map is used.

K When the map, burst, or card runs, no backup file will be created.

K'filename'

When the map, burst, or card runs, a backup file will always be created to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.

K+'filename'

When the map, burst, or card runs, a backup file will always be appended to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.

KU'filepath'

When the map, burst, or card runs, a backup file will always be created using a unique backup name in the directory *filepath*. If *'filepath'* is not an absolute path, the map directory will be pre-pended.

KU When the map, burst, or card runs, a backup file will always be created using a unique backup name in the map directory.

K1'filename'

If the map, burst, or card does not successfully complete, a backup file will be created to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.

K1+'filename'

If the map, burst, or card does not successfully complete, a backup file will be appended to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.

KU1'filepath'

If the map, burst, or card does not successfully complete, a backup file will be created using a unique backup name in the directory *filepath*. If *'filepath'* is not an absolute path, the map directory will be pre-pended.

KU1 If the map, burst, or card does not successfully complete, a backup file will be created using a unique backup name in the map directory.

src-adptr-cmd

Specify settings for the source using the adapter-specific commands. See the adapter-specific information in the *Resource Adapters documentation*.

. Use **Command** in the compiled map file.

List (-L)

When using a Command Server on a platform other than Windows, use the List execution command (-L) to display a list of the data sources and targets that are compiled into a map file.

This command does not execute the map. It displays the list of sources and destinations.

For example, if you are using the Command Server on a UNIX platform, you can display the sources and destinations for the **sinkmap** example map by using the following command:

```
dtxcmdsv sinkmap.mmc -L
```

The result would be similar to the following:

```
Map file: /examples/general/map/sinkmap/sinkmap.mmc -L
```

```
Card information for map /examples/general/map/sinkmap/
sinkmap.mmc:
```

```
T column is data type: F=File, A=Application, D=Database
```

	Name	T Data
	----	- ----
Input # 1	all_employees	F allemps.txt
Input # 2	employee_to_lookup	F employee.txt

Output #	1	week	F
Output #	2	tthours	F
Output #	3	emp_hours	F emphours.txt

This information can be useful when determining the current source or target names, as well as to confirm the exact source, target types, or locations.

Initialization File Override (-N)

To override the default initialization (**dtx.ini**) file for the Command Server, use the **-N** command line option as the first command to the Command Server, as indicated in the following syntax:

```
-N=< INI_filename >
```

where *< INI_filename >* is the name of the renamed initialization file. If the path is not absolute, it will be resolved to the location of the default **INI** file, which is **dtx.ini**. Also, if spaces exist in this path/name, enclose the *< INI_filename >* value in single quotes. When running a map using this override, the entire syntax should be as follows:

```
[< INI_filename >] [< mapname > < commands > ...]
```

The following example uses the **myinit.ini** initialization file to run the **sdq.mmc** map with an execution audit on a Windows platform:

```
dtxcmsv -n=c:\dtx_n.n\myinit.ini sdq.mmc -ae
```

Output Target Override - Application (-OA)

Use the Output Target Override - Application execution command (**-OA**) to override specifications in the compiled map file for a specific output card for a single execution of a map.

```
-OA[alias]card_num[Rcount:interval] [B]
    [K['filename']|['+filename']|['U'filepath']|['U']|
    [1'filename']|['1+filename']|['U1'filepath']|['U1']]
    {target-adptr-cmd|.}
```

Option Description

alias This is the adapter alias for the specific application adapter. For example, FTP is the adapter alias for the FTP adapters.

card_num
This is the card number of the output to override.

Rcount:interval
Specify **Retry** settings. If the target is unavailable, the adapter attempts to access the target as many times as specified by the *count* setting at the interval specified by the *interval* setting.

count number of attempts to access the application

interval number of seconds to wait between attempts. If this option is not specified, **Retry** settings compiled into the map are used. To eliminate adapter retry, specify **R0:0**.

B If the map, burst, or card does not complete successfully, roll back any changes made to this data target. If this option is not specified, the **OnFailure** setting compiled into the map is used.

- K** When the map, burst, or card runs, no backup file will be created.
- K'filename'**
When the map, burst, or card runs, a backup file will always be created to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.
- K+'filename'**
When the map, burst, or card runs, a backup file will always be appended to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.
- KU'filepath'**
When the map, burst, or card runs, a backup file will always be created using a unique backup name in the directory *filepath*. If *'filepath'* is not an absolute path, the map directory will be pre-pended.
- KU** When the map, burst, or card runs, a backup file will always be created using a unique backup name in the map directory.
- K1'filename'**
If the map, burst, or card does not successfully complete, a backup file will be created to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.
- K1+'filename'**
If the map, burst, or card does not successfully complete, a backup file will be appended to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.
- KU1'filepath'**
If the map, burst, or card does not successfully complete, a backup file will be created using a unique backup name in the directory *filepath*. If *'filepath'* is not an absolute path, the map directory will be pre-pended.
- KU1** If the map, burst, or card does not successfully complete, a backup file will be created using a unique backup name in the map directory.
- target-adptr-cmd**
This option provides the adapter-specific commands that connect to the data target and process the output data. See the adapter-specific documentation in the *Resource Adapters documentation*.
- . Use **Command** in the compiled map file.

Output Target Override - Database (-OD)

Use the Output Target Override - Database execution command (-OD) to override specifications in the compiled map file for a specific output card for a single execution of a map.

```
-ODcard_num[Rcount:interval][B]
  [K['filename']|['+'filename']|['U'filepath']|['U']|
  [1'filename']|['1+'filename']|['U1'filepath']|['U1']]
  {target-adptr-cmd|.}
```

Option Description

card_num

This is the card number of the output to override.

Rcount:interval

Specify **Retry** settings. If the target is unavailable, the adapter attempts to

access the target as many times as specified by the *count* setting at the interval specified by the *interval* setting.

count number of attempts to access the database

interval number of seconds to wait between attempts. If this option is not specified, the **Retry** settings compiled into the map are used. To eliminate adapter retry, specify **R0:0**.

B If the map, burst, or card does not complete successfully, roll back any changes made to this data target. If this option is not specified, the **OnFailure** setting compiled into the map is used.

K When the map, burst, or card runs, no backup file will be created.

K'*filename'*

When the map, burst, or card runs, a backup file will always be created to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.

K+*filename'*

When the map, burst, or card runs, a backup file will always be appended to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.

KU'*filepath'*

When the map, burst, or card runs, a backup file will always be created using a unique backup name in the directory *filepath*. If *'filepath'* is not an absolute path, the map directory will be pre-pended.

KU When the map, burst, or card runs, a backup file will always be created using a unique backup name in the map directory.

K1'*filename'*

If the map, burst, or card does not successfully complete, a backup file will be created to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.

K1+*filename'*

If the map, burst, or card does not successfully complete, a backup file will be appended to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.

KU1'*filepath'*

If the map, burst, or card does not successfully complete, a backup file will be created using a unique backup name in the directory *filepath*. If *'filepath'* is not an absolute path, the map directory will be pre-pended.

KU1 If the map, burst, or card does not successfully complete, a backup file will be created using a unique backup name in the map directory.

target-adapter-cmd

Specify the settings for the target using the adapter-specific commands. See the appropriate adapter information in the *Resource Adapters* documentation.

. Use **Command** in the compiled map file.

Output Target Override - Echo (-OE)

Use the Output Target Override - Echo execution command (-OE) to override specifications in the compiled map file for a specific output card for a single execution of a map.

Use this command when you want the data resulting from one or more output cards passed back to a calling application or a map, rather than being directed to a file, database, application, or message. If this command is used to override more than one output, the results of the outputs are concatenated without separators.

For example, use this command in conjunction with the **RUN()** function to pass data from one map to another:

```
-OEcard_num[S|X]
```

Option Description

card_num

This is the card number of the output to override.

- S** Return the size (in bytes) of the output data. This option causes the resulting output data to be returned as an ASCII number representing the size of the data, followed by a space, followed by the data itself.

For example, if used in a **RUN()** function:

```
RUN ("mymap.mmc","-AE -OE2S -OE4S")
```

If the data resulting from output card #2 is "This and That" and the output from output card #4 is "19439.27", the string returned by this example **RUN()** function is "13 This and That 8 19439.27".

If the size option is not used, the data returned by this example **RUN()** function is "This and That 19439.27".

- X** Exclude this card from echo. The output is stored in memory and not echoed back to the specified map or application.

You might use this option to optimize performance when a map has an output used strictly as a *scratch* card, that is, it is built only to be used by a later output.

Output Target Override - File (-OF)

Use the Output Target Override - File execution command (-OF) to override data destination specifications for a specific output card in a compiled map file for a single execution of a map.

```
-OFcard_num[+|!+] [X|X0] [Rcount:interval] [B]
  [K['filename'] | ['filename'] | [U'filepath'] | [U] |
  [1'filename'] | [1+'filename'] | [U1'filepath'] | [U1]]
  {target|.}
```

Option Description

card_num

This is the card number of the output to override.

- +** If the map, burst, or card completes successfully, the output data is appended to an existing data file. If the data file does not exist, it is created.
- !+** The data file is created at the start of map execution. If the file exists, it is overwritten. If more than one burst occurs, data is appended to the same data file.

Rcount:interval

Specify **Retry** settings. If the target is unavailable, the adapter attempts to access the target as many times as specified by the *count* setting at the interval specified by the *interval* setting.

count number of attempts to access the file

interval number of seconds to wait between attempt. If this option is not specified, the **Retry** settings compiled into the map are used. To eliminate adapter retry, specify **R0:0**.

B If the map, burst, or card does not complete successfully, roll back any changes made to this data target. If this option is not specified, the **OnFailure** setting compiled into the map is used.

K When the map, burst, or card runs, no backup file will be created.

K'*filename'*

When the map, burst, or card runs, a backup file will always be created to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.

K+*filename'*

When the map, burst, or card runs, a backup file will always be appended to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.

KU'*filepath'*

When the map, burst, or card runs, a backup file will always be created using a unique backup name in the directory *filepath*. If *'filepath'* is not an absolute path, the map directory will be pre-pended.

KU When the map, burst, or card runs, a backup file will always be created using a unique backup name in the map directory.

K1'*filename'*

If the map, burst, or card does not successfully complete, a backup file will be created to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.

K1+*filename'*

If the map, burst, or card does not successfully complete, a backup file will be appended to *filename*. If *'filename'* is not an absolute path, the map directory will be pre-pended.

KU1'*filepath'*

If the map, burst, or card does not successfully complete, a backup file will be created using a unique backup name in the directory *filepath*. If *'filepath'* is not an absolute path, the map directory will be pre-pended.

KU1 If the map, burst, or card does not successfully complete, a backup file will be created using a unique backup name in the map directory.

target Specify the name of the output file. If the output file is not located in the same directory as the map file, the full path must be specified using the platform-specific syntax.

. Use the **FilePath** in the compiled map file.

Output Target Override - Message (-OM)

Use the Output Target Override - Message execution command (-OM) to override data destination specifications for a specific output card in a compiled map file for a single execution of a map.

```
-OM[alias]card_num[X|X0][Rcount:interval][B]
[K['filename']|['+'filename']|['U'filepath']|['U']|
[1'filename']|[1+'filename']|[U1'filepath']|[U1]]
{target-adptr-cmd|.}
```

Option Description

alias This is the adapter alias for the specific messaging adapter. For example, MSMQ is the adapter alias for the Microsoft MSMQ adapters. See the adapter-specific information in the Resource Adapters documentation.

card_num This is the card number of the output to override.

X If the map, burst, or card completes successfully, the message is deleted from its target.

X0 If the map, burst, or card completes successfully, the message is deleted from its target only if it has no content. If neither delete option (**X** or **X0**) is specified, the **OnSuccess** setting compiled into the map is used.

Rcount:interval Specify **Retry** settings. If the target is unavailable, the adapter attempts to access the target as many times as specified by the *count* setting at the interval specified by the *interval* setting.

count number of attempts to access the message queue.

interval number of seconds to wait between attempts. If this option is not specified, the **Retry** settings compiled into the map are used. To eliminate adapter retry, specify **R0:0**.

B If the map, burst, or card does not complete successfully, roll back any changes made to this data target. If this option is not specified, the **OnFailure** setting compiled into the map is used.

K When the map, burst, or card runs, no backup file will be created.

K'filename' When the map, burst, or card runs, a backup file will always be created to *filename*. If *'filename'* is not an absolute path, the map directory will be prepended.

K+'filename' When the map, burst, or card runs, a backup file will always be appended to *filename*. If *'filename'* is not an absolute path, the map directory will be prepended.

KU'filepath' When the map, burst, or card runs, a backup file will always be created using a unique backup name in the directory *filepath*. If *'filepath'* is not an absolute path, the map directory will be prepended.

KU When the map, burst, or card runs, a backup file will always be created using a unique backup name in the map directory.

K1'filename' If the map, burst, or card does not successfully complete, a backup file will be created to *filename*. If *'filename'* is not an absolute path, the map directory will be prepended.

K1+'filename' If the map, burst, or card does not successfully complete, a backup file will be appended to *filename*. If *'filename'* is not an absolute path, the map directory will be prepended.

KU1*'filepath'*

If the map, burst, or card does not successfully complete, a backup file will be created using a unique backup name in the directory *filepath*. If *'filepath'* is not an absolute path, the map directory will be prepended.

KU1 If the map, burst, or card does not successfully complete, a backup file will be created using a unique backup name in the map directory.

trgt-adptr-cmd

This option provides the adapter-specific commands that connect to the data source and retrieve the input data. See the adapter-specific information in the Resource Adapters documentation.

. Use **Command** in the compiled map file.

WorkSpace PageSize (-P)

When a file-based WorkSpace is used, the amount of memory accessed at a given time depends on the number of pages and the size of pages you configure for a map to run. Use the WorkSpace PageSize execution command (-P) to specify the size of and number of pages to be used.

If this execution command is used without a corresponding WorkSpace execution command (-W), the WorkSpace settings in the compiled map are used.

Generally, for large data files, you might want to increase the number of pages by specifying a large count. For maps that include a large number of type references, you might want to increase the page size.

-Psize:count

Option Description

size Size in kilobytes of a single page of memory used during map execution. Page size is measured in kilobytes and is specified as a multiple of 2 between 2 and 1024.

count Number of pages used during map execution. The count is specified as an integer greater than 0.

Refresh Status (-R)

The Refresh Status execution command (-R) is used to specify the frequency of map status update when displayed in the Command Server window using the Windows operating system or in the console when using all other operating systems. To reflect execution time, as well as the number of input objects found and output objects built, specify the interval at which the map status is refreshed.

-Rupdate_rate

Option Description

update_rate

This is the number of times per second the map status is refreshed. The default refresh interval is approximately twice per second. If you specify 0 as the refresh interval, the map status is not displayed.

MapTrace (-T)

The MapTrace execution command (-T) creates a trace file that can be used for diagnosing invalid data or incorrect type definitions. For a detailed discussion of the trace file see the *Map Designer* documentation.

Use this command to specify a particular card, or a range of input and output data objects to trace. A trace summary can also be produced. When the -T command is used without other options specified, no trace log is produced.

If you specify the MapTrace execution command (-T), you must specify all necessary **MapTrace** settings because the settings in the MapTrace execution command override *all* **MapTrace** settings in the compiled map.

```
-T[I|Iccard_num|Ifrom[:to]] [O|Ocard_num] [S] [=dir]
```

Option Description

I This produces detailed trace information for all input cards.

Iccard_num

This produces detailed trace information for a single input card. For example, the command **-TIC3** produces detailed trace information for only input card #3.

Ifrom[:to]

This produces detailed trace information for a range of input objects. For example, to produce trace information for objects 1000 to 1320, the option could be specified as **-TI1000:1320**. The upper limit of the object range is optional, such that to trace from input object 1000 to the end of the file, the option could be specified as **-TI1000**.

O This produces detailed trace information for all output cards.

Ocard_num

This produces detailed trace information for a single output card. For example, the command **-TOC2** produces detailed trace information for only output card #2.

S This produces summary level trace information for all input and output cards providing a message for each card. For example, the message for a particular card might be either Input 1 was valid, but 34 bytes of unknown data found or Output 2 built successfully.

=dir This is a directory path in which you want the trace file created. By default, the trace file is created in the same directory as the compiled map (**.mmc**) file.

Stop Validation (-V)

The Stop Validation execution command (-V) specifies to stop validation at the card where the first error is detected. When this command is used, validation occurs only through the input card where the first error is encountered. Map execution is stopped after validating that input card, even if there are multiple input cards.

Under certain circumstances, multiple errors are detected during validation. You can stop by using **-V** at the first error detected or continue (with no **-V** validating data until all errors are found).

If you use the `-V` command, validation only occurs through the input card where the first error occurred. It stops executing the map after validating that input card, even if there are multiple input cards. If the `-V` command is not used, the **Validation OnValidationError** setting is used.

In all situations, if an error is found in the input, no output objects are built.

Workspace (-W)

When you run a map, the Command Server uses its own paging scheme to access information about the data as efficiently as possible. Portions of both workspace data and actual data are paged as needed. The workspace can be either a combination of files and memory or memory-based. When file-based workspace is used, the amount of memory accessed at a given time depends on the number of pages and the count of pages configured for map execution.

When the `WorkSpace` execution command (`-W`) is used without other specified options, a file-based `WorkSpace` is used and the work files are named using the map name; they are placed in the same directory as the compiled map file. If the `-W` command is used without a corresponding `WorkSpace PageSize` execution command (`-P`), the **WorkSpace** settings in the compiled map are used.

You can optimize the speed of map processing using the workspace configuration setting. When processing a large amount of input data in a map, use the file-based workspace with default paging. When small amounts of data are processed, smaller page sizes and memory-based workspaces are recommended. When using burst processing, adjust the workspace configuration for the size of burst data.

`-W[M] | { [D] [U] [=dir] }`

Option Description

- D** A file-based `WorkSpace` is used. The work files are deleted after a map is executed. All work files are deleted, except those being saved for reuse as specified on the input cards for each map.
- U** A file-based `WorkSpace` is used. A unique name is generated for each work file created and the files are automatically deleted after the map is executed. Unique work file names are useful if you are running more than one instance of the same map at the same time. Each time a map is run, a work file is created with a unique name, which eliminates any conflict between work files.
- M** A memory-based `WorkSpace` is used. This might enhance performance for specific maps depending on work file requirements and the amount of available memory by keeping the work file information in memory and eliminating the I/O necessary with file-based work files. If there is insufficient memory available to build the work file, the map returns a *Disk write error* message.

When this option is used, none of the other work file options can be used. For example, deleting work files, unique work file names, or specifying the directory.

- `=dir` Specify the directory path in which you want the work files to be created. By default, work files are created in the same directory as the compiled map (**.mmc**) file.

Retry (-Y)

The Retry execution command (-Y) is used to specify the retry rate for non-data files accessed during map execution. This includes the compiled map file, as well as, work files, the trace file, and the audit log. If one of these files cannot be accessed at execution time, an attempt is made to access the file as specified.

-Ycount:interval

Option Description

count This is the number attempts to be made to access the file.

interval

This is the number of seconds to wait between each attempt.

Ignore Warnings (-Z)

The execution Ignore Warnings execution command (-Z) specifies for a map that selected warning codes should be ignored and a 0 return code should be issued. When -Z is used *with* one or more warning codes, a 0 return code is issued for those warning codes. When used *without* other options specified, all warning codes result in a 0 return code Using this command impacts **OnSuccess** settings, **OnFailure** settings, and so on.

-Z[!][warning_code[:warning_code...]]

Option Description

! Use to specify those warning codes that, when encountered, remain in their default state; they will remain as warnings and a 0 return code is not issued, followed by the warning code message.

When used with -Z only (-Z!), a 0 return code followed by the warning code message is returned if any of the warning codes is encountered.

warning_code

Specify the warning codes to be acted upon when encountered.
Valid warning code selections include: 14, 18, 21, 26, 27, 28, and 29.

Examples using Ignore Warnings (-Z)

Z This causes a 0 return code with the warning message to be returned when any of the warning codes is encountered.

Z18 This causes a 0 return code with the warning message to be returned when a Page Usage Count Error (18) is encountered.

Z!18 Excluding the Page Usage Count Error (18), a 0 return code with the warning message is returned for all of the other warning codes.

Z18:27:28

This causes a 0 return code with the warning message to be returned when any of the specified warning codes is encountered (18, 27, or 28 in this case).

If the Ignore Warnings command (-Z) is enabled and a map encounters the condition for any warning code as specified, the execution section of the audit log includes a Valid map status with a 0 return code and the associated warning message. For example:

```

<Summary MapStatus="Valid" Return="0">
  <ElapsedTimeInSec>0.1026</ElapsedTimeInSec>
  <Message>Input type contains errors.</Message>
  <CommandLine>'install_dir\examples\ansiack\receiveEDI.mmc'
</CommandLine>
  <ObjectsFound>0</ObjectsFound>
  <ObjectsBuilt>0</ObjectsBuilt>
</Summary>

```

When using the `-Z` command, you cannot specify the same warning code using the Fail on Warnings command (`-F`).

Command availability

Some commands are limited because they can only be used with a Command Server and cannot be used from within a `RUN()` function or using the Platform API. Also, there are some commands that are not available on all platforms. The Integration Flow Designer, the Map Designer, and the Command Server have a user interface for these commands where they apply.

The following table lists the execution commands and where they can be executed (3), as well as those commands that have limited platform availability.

Execution Command	Command Server	RUN Function	Platform API	Platform Availability
MapAudit (-A)	Y	Y	Y	
Close Window (-B)	Y			
Century (-D)	Y	Y	Y	
Fail on Warnings (-F)	Y	Y	Y	
Ignore Validation (-G)	Y	Y	Y	
Source Overrides				
Application (-IA)	Y	Y	Y	
Database (-ID)	Y	Y	Y	
Echo (-IE)	Y	Y	Y	
File (-IF)	Y	Y	Y	
Message (-IM)	Y	Y	Y	
List Sources/Targets (-L)	Y			Non-Windows only
Initialization File Override (-N)	Y			
Target Overrides				
Application (-OA)	Y	Y	Y	
Database (-OD)	Y	Y	Y	
Echo (-OE)	Y*	Y	Y	
File (-OF)	Y	Y	Y	

Execution Command	Command Server	RUN Function	Platform API	Platform Availability
Message (-OM)	Y	Y	Y	
WorkSpace PageSize (-P)	Y	Y	Y	
Refresh Status (-R)	Y			
MapTrace (-T)	Y	Y	Y	
Stop Validation (-V)	Y	Y	Y	
WorkSpace (-W)	Y	Y	Y	
Retry (-Y)	Y	Y	Y	
Ignore Warnings (-Z)	Y	Y	Y	

Chapter 3. Examples

This documentation presents a variety of examples involving the execution of maps using various execution commands in a **RUN()** function and from the command line in UNIX and Windows environments.

Running a map located in a different directory

From a UNIX command line, enter:

```
/install_dir/dtxcmds /x12/edi.mmc -TIO -IF2!WXR5:1 dtin -OF1B out.txt
```

Option Description

/X12/edi.mmc

This specifies the full path and name of the map to be executed.

-TIO A trace file is created or all input and output objects that are placed in the same directory as the map are named using the *<mapname>.mtr* file naming convention.

-IF2!WXR5:1 dtin

This changes the source of input card #2 to be a file named **dtin**. The other adapter settings for this card include:

Do not re-use workfiles.

If the map successfully executes, delete the **dtin** file.

If the file is not available, attempt to access it five more times at one-second intervals.

-OF1B out.txt

This changes the destination for the data from output card #1 to be a file named **out.txt**. The B option causes any changes to **out.txt** to be backed out (rolled back), if the map does not successfully execute. All other adapter settings are as compiled into the map file.

The file locations in this example are relative to the location of the compiled map file.

Creating work files in an alternate directory

From a Windows command line, enter:

```
install_dir\dtxcmds k:\maps\invoices.mmc -AE -T -WD=c:\tmp
```

Option Description

k:\maps\invoices.mmc

This specifies the name of the map to be executed.

-AE This produces an execution audit log using the **AuditLocation** settings for **MapAudit** compiled into the map file.

-T This specifies that *no* trace information is to be produced.

-WD=c:\tmp

This creates the work files in the `c:\tmp` directory and deletes the work files after the map file is executed.

Executing multiple maps from a single command

From a Windows command line, enter:

```
dtxcmdsv mymap.mmc -R2 -D1949 flatfile.mmc -F -Y10:2
```

Option Description

mymap.mmc

This specifies the name of the first map to be executed, **mymap.mmc**.

-R2 This causes the server display to be updated two times per second.

-D1949

This option specifies that dates without a century value should be converted. Years between 00 and 49 are interpreted as 20xx and years between 50 and 99 are interpreted as 19xx.

flatfile.mmc

This option specifies the name of another map to be executed, **flatfile.mmc**. When **flatfile.mmc** is executed, it uses the default options, not the options that precede it on the command line.

-F Fail on the return of any warning code.

-Y10:2 If any non-data file resource is unavailable, such as the map file, audit log file, work file, and so on, up to ten attempts should be made at two-second intervals to access the unavailable resource.

Overriding a database target

Suppose you want to run the map **multitran.hp**, overriding some of the settings for the fourth output card, which is a database target.

Use the following command:

```
/install_dir/dtxcmdsv multitran.hp -G -OD4R5:3B "`-DELETE -TRACE ERROR'"
```

Option Description

multitran.hp

This specifies the name of the map to be executed, **multitran.hp**.

-G All item properties, including size, presentation, and restrictions, are fully validated.

-OD4R5:3B "`...'"

This changes settings for the database target for the data of output card #4 as follows:

R5:3

- If the database resource is unavailable, attempt to make the connection five more times at three-second intervals

B

- If the map does not complete successfully, roll back any changes made to the database table

"^-DELETE -TRACE ERROR"

These database-specific adapter commands described in the *Resources Adapters* documentation indicate that all rows in the target database table should be deleted before processing the output data and a trace file should be generated that only contains error information.

Executing a command file

As an alternative to typing execution commands on the command line, you can run maps using maps and execution commands in a command file. To use a command file named **cmd_file.txt**, from a UNIX command line, enter:

```
dtxcmsv @cmd_file.txt
```

A command file can contain spaces and new lines to separate arguments and maps to be run. The following is an example of a command file to be executed on a UNIX platform:

```
/x12/edi.mmc -TIO -IF2!WXR5:1 dtin -OF1B out.txt  
/maps/invoices.mmc -AE -T -WD=/tmp  
mymap.mmc -R2 -D1949  
flatfile.mmc -F -Y10:2  
/maps/daily/multitrans.hp -G -OD4R5:3B `^-DELETE -TRACE ERROR'
```

Using this command file, the maps are executed sequentially. These command files can be manually created or can be generated by the Integration Flow Designer for systems with an execution mode of Command.

Using a run function and echoed outputs in memory

When using the Output Target Override - Echo execution command (-OE) with the X option, the output is stored in memory and is not echoed back to the calling map or application. A typical scenario in which this is useful and can serve to optimize performance is when a map has an output that is only used as a *scratch* card. That is, it is built to be used only by a subsequent output.

For example, if used in a **RUN()** function:

```
RUN ("some_map.mmc", "-WM -OE1 -OE3X -OE4")
```

In this example, a memory-based Workspace is used and the data for output cards 1 and 4 are echoed back to the rule in which the **RUN** function is located. All other map settings and input and output card settings for executing this map, such as **MapAudit**, **Century**, and so on, would use the settings compiled into **some_map.mmc**.

This results in the execution of **some_map.mmc**. The data produced for output card 1 and 4 is concatenated and *echoed* back to the rule containing the **RUN** function. The data for output card 3 is not *echoed* back to the rule because of the use of the X option. Given that output card 1 builds 103910^ABC, output card 3 builds justsome data, and output card 4 builds 01/23/98, the **RUN** function returns 103910^ABC01/23/98.

The most common use for this feature of echoed outputs is when the data for an output card is needed only during the execution of the map, perhaps for use on a later output card, and can be discarded after the map has completed execution.

For another example of using an echoed data destination, see the *Platform API* documentation.

Audit Log in memory

Similar to how the data built for an output card can be echoed back to a calling map or application, the audit log, including Data and Execution information, can be placed in memory using the MapAudit execution command (**-A**) with the **M** option and echoed back to its calling map or application with the **E** option.

If the audit in memory fails, either of these messages might be returned from the Command Server: Disk write error or Not enough memory to execute map.

The **-AEM** audit option can be useful when using a **RUN** function or the Platform API to place the audit log in memory during map execution and then to return it, along with any outputs being echoed back. Placing the audit log in memory avoids the file I/O necessary if the audit log was written to disk.

For example, if used in a **RUN()** function, enter:

```
RUN ("some_map.mmc", "-AEM -WM -OE1")
```

This results in the execution of **some_map.mmc** and returns the concatenation data created for output card 1 and the audit log data.

When the audit log data is returned in this function implementation, it still adheres to the format described by the **audit.mtt** type tree included in the examples directory.

When the audit log is written to memory and is echoed back to the calling map or application, it is appended to any echoed outputs. Therefore, if output cards 2 and 4 were echoed, along with the audit log, the data for output card 2 is followed by the data for output card 4, which is followed by the contents of the audit log.

Chapter 4. INI commands for debugging maps

This documentation presents the **dtx.ini** (INI) file commands you can use for the debugging of adapter connections for maps run using non event-driven execution methods not run through events using the Launcher. This includes using a command-driven execution method such as the Command Server, a command parameter file, batch file, shell script, graphical user interface (GUI), scheduling program, and JCL on the mainframe, or an application-embedded execution method using components of the Development Kit in which maps can be embedded in specific third-party applications for web-based integration scenarios.

For information about debugging maps run using event-driven execution methods, refer to the *Launcher* documentation, *Map Designer* documentation and individual adapter documentation.

The logging feature is used to help you debug adapter connection issues. To use this feature, configure the debugging settings in the **Connections Manager** section of the **dtx.ini** (INI) file.

Logging

The logging feature (non-Launcher usage) enables you to specify the type of information you want to capture about adapter connections in log files. You do this by configuring the following severity classes and debug components listed from lowest impact to highest logging categories along with their equivalent entry in the **Connections Manager** logging entry. The logging categories in the **INI** file will be configured to 0 to turn that category off or 1 to turn that logging category on.

Logging Category

Logging Category in the Connections Manager

Trace	LogTrace
Debug	LogDebug
Info	LogInfo
Warn	LogWarning
Error	LogError
Fatal	LogFatal

Logging categories

The logging categories, non-Launcher usage, provide a variety of debugging information. They are used to determine which severity levels and components to debug.

LogX=*n*

Provide the specific category, such as Trace, Debug, Info, Warning or Fatal, for X and 0 to turn that logging category off, or 1 (to turn that category on) for *n*.

Example: LogTrace=1

Logging Category	Value	Description
LogTrace	0	Disabled (default)
	1	Enables the logging of sequence of events Example Results: Before Resource Pending <map> After Resource Pending <map>
LogDebug	0	Disabled (default)
	1	Enables the logging of information to diagnose adapter connection behavior Example Results: New audit file name <name> Adapter wildcard.<wildcard>
LogInfo	0	Disabled (default)
	1	Enables the logging of general information about the state of adapter connections Example Results: :Map started InitPendingHigh reached
LogWarning	0	Disabled (default)
	1	Enables the logging of issues that are not severe, but might be important to know Example Results: Resource Manager table too small - using default Map warning <map>
LogError	0	Disabled (default)
	1	Enables the logging of critical adapter connection errors Example Results: Map failure <map> Socket failure <socket_type>
LogFatal	0	Disabled (default)

Logging Category	Value	Description
	1	Enables the logging of a fatal error Example Results: Corrupted MMC file during initialization <mmc_name> Adapter listener Initialization failure <adapter>

Debug

Use the **DebugName** and **DebugAppend** commands configured in the **dtx.ini** (INI) file to specify how you want to create the output log file.

DebugName

Use this setting to define the complete file name of the debug file. Remember that this logging category is only for non-Launcher usage and cannot be configured dynamically in the Management Console. When the map is started, it will use the settings that were configured in the INI file.

The default file name is **dtxconn.log**.

The syntax is the following:

DebugName=[*file_name*]

Option Description

file_name

The file name of the debug file for the Connections Manager

DebugAppend

Use this setting to append the debug information to an already existing debug log file. Remember that this logging category is only for non-Launcher usage and cannot be configured dynamically in the Management Console. When the map is started, it will use the settings that were configured in the INI file.

The default file name is **dtxconn.log**.

The syntax is the following:

DebugAppend={0|1}

Option Description

- 0 Disables the append specification so that the debug information is written to a new debug log file.
- 1 Enables the append specification so that the debug information overwrites the contents of an existing debug log file.

This is the default.

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 execution command 8, 36
- B execution command 11
- D execution command 11
- F execution command 11
- G execution command 12
- IA execution command 13
- ID execution command 15
- IE execution command 16
- IF execution command 17
- IM execution command 19
- L execution command 20
- N execution command 21
- OA execution command 21
- OD execution command 22
- OE execution command 23, 35
- OF execution command 24
- OM execution command 25
- P execution command 27
- R execution command 27
- T execution command 28
- V execution command 28
- W execution command 29
- Y execution command 30
- Z execution command 30

A

- Always SummaryAudit setting 8
- applications
 - example of using as an input 34
- audit logs
 - auditing the execution 9
 - full audit log support 8
 - full audit log support examples 10
 - stored in memory 36

B

- backup audit log option 25
- backUp audit log option 13, 15, 17, 19, 21, 22, 24

C

- card settings 2
- Century (-D) 11
- Close Window (-B) 11
- command availability 31
- command files 3
 - using to run a map 35
- command line
 - executing from 4
 - help 5
- command options
 - examples of using
 - creating work files in a different directory 33
 - mapping from an application 34
 - running a map from a different directory 33
 - running multiple maps from a single command 34

- command options (*continued*)
 - examples of using (*continued*)
 - using a command file 35
- Command Server 1
 - commands,using 3
- compiled map files 2

D

- data sources
 - listing 20
- data targets
 - listing 20
- database target
 - overriding 34
- debug 39
- DebugAppend 39
- DebugName 39
- dtx.ini 21
 - debugging configurations 37
 - overriding 21

E

- echoed data
 - explanation 16
 - syntax 4
- execution commands
 - Century (-D) 11
 - Close Window (-B) 11
 - Fail on Warnings (-F) 11
 - Ignore on Warnings (-Z) 30
 - Ignore Validation Options (-G) 12
 - Initialization File Override (-N) 21
 - Input Source Override Application (-IA) 13
 - Input Source Override Database (-ID) 15
 - Input Source Override Echo (-IE) 16
 - Input Source Override File (-IF) 17
 - Input Source Override Message (-IM) 19
 - Input Target Override Application (-OA) 21
 - Input Target Override Database (-OD) 22
 - Input Target Override Echo (-OE) 23
 - Input Target Override File (-OF) 24
 - Input Target Override Message (-OM) 25
 - list 7
 - List (-L) 20
 - MapAudit (-A) 8
 - MapTrace (-T) 28
 - options 4
 - Refresh Status (-R) 27
 - Retry (-Y) 30
 - Stop Validation (-V) 28
 - when running from a map rule 5
 - when running from a platform API 5
 - WorkSpace (-W) 29
 - WorkSpace PageSize (-P) 27

F

Fail on Warnings (-F) 11
full audit log support 8

G

general rules for execution commands 4

I

Ignore on Warnings (-Z) 30
Ignore Validation Options (-G) 12
Initialization File Override (-N) 21
Input Source Override Application (-IA) 13
Input Source Override Database (-ID) 15
Input Source Override Echo (-IE) 16
Input Source Override File (-IF) 17
Input Source Override Message (-IM) 19
Integration Flow Designer 3

L

List (-L) 20
logging 37
 DebugAppend 39
 DebugName 39

M

Map Designer 2
map settings 1
MapAudit (-A) 8, 36
maps
 running
 from a command file 35
 from a different directory 33
 from a map rule 5
 from a platform API 5
 multiple from a single command 34
 UNIX example 33
MapTrace (-T) 28
memory page size and count
 explanation 27

O

OnError SummaryAudit setting 8
OnWarningsOnError SummaryAudit setting 8
output stored in memory 35
Output Target Override Application (-OA) 21
Output Target Override Database (-OD) 22
Output Target Override Echo (-OE) 23, 35
Output Target Override File (-OF) 24
Output Target Override Message (-OM) 25

P

page size and count
 explanation 27

R

Refresh Status (-R) 27
Retry (-Y) 30

S

sources and targets
 listing 20
Stop Validation (-V) 28
store in memory
 audit log 36
 output 35

U

Unique map audit log file name option 8
UNIX example 33

W

WorkSpace (-W) 29
WorkSpace PageSize (-P) 27



Printed in USA