IBM Software Group

WebSphere Transformation Extender Utility Commands (Part I)

Tim Rabbitt (trabbitt@us.ibm.com)
Websphere Transformation Extender L2 Support
August 29, 2012

Agenda

The presentation covers the Utility Commands that are available with WebSphere Transformation Extender (WTX):

- Why use them
- What can they do for you
- Configuring your environment to use Utility Commands
- Types of Utility Commands
- Examples, Questions and Answers

Why use them

- Automate processing (compiling, analyzing, converting) of large volumes of maps/trees/systems via batch scripts
- Use WTX components without the need for X-Windows or Windows GUIs

What can they do for you

- Analyze type trees
- Import and export type trees, map source files and systems
- Convert type tree properties from bytes to characters
- Compile maps
- Generate a map report HTML file containing information about a map source file
- Deploy systems
- Create one map that can transform any input data into XML output, and a second map that can transform the XML that the first map produced into output data in a format described by the type tree that you specified
- Edit the Resource Registry
- Start, Stop, Administer the Launcher

Configuring your environment to use Utility Commands

- Windows
  - Set the environment variable PATH to reference your WTX installation
  - Set PATH=C:\IBM\WebSphere Transformation Extender 8.4;%PATH%
- UNIX
  - Run the WTX setup command or set environment variables
    - Setup Command
      - . <path to WTX installation directory>/setup
    - Set environment variables PATH and LD_LIBRARY_PATH/LIBPATH to reference the WTX directories.
      - PATH:
        - export PATH=/opt/ibm/wtx/bin:$PATH

Configuring your environment to use Utility Commands

- LIBPATH (AIX)
  - export LIBPATH=/opt/ibm/wtx/libs:$LIBPATH

- LD_LIBRARY_PATH (Solaris, Linux, zLinux, HP-UX)
  - export LD_LIBRARY_PATH=/opt/ibm/wtx/libs:$LD_LIBRARY_PATH

Configuring your environment to use Utility Commands

- Utility Commands return success/failure in an environment variable:
  - Windows: ERRORLEVEL
  - UNIX: status
- Utility commands return the following values in the ERRORLEVEL or status environment variable:
  - 0 - This value indicates that the utility command was successful.
  - 1 - This value indicates that the utility command was not successful.
- The ERRORLEVEL and status environment variable can also be used in batch and shell scripts to retrieve the value of the return code from the utilities.

Types of Utility Commands

- Utilities are grouped by functionality within WTX:
  - Type Designer
  - Integration Flow Designer
  - Resource Registry*
  - Map Tuning **
  - Map Designer
  - XML*
  - Launcher *

* A future presentation will address XML, Resource Registry and Launcher related Utility Commands on 2/6/2013

** Map Tuning was discussed in the WSTE presentation on 8/15/12, titled “Map Tuning Utilities and Debugging In Design Studio”.

Type Designer Utilities

- **tanalyze** - Analyzes type trees
- **timport** - Imports metadata definitions to create type trees
- **texport** - Exports type trees to a text file
- **tbcconv** - Converts type tree properties from bytes to characters
- Executed in Windows only environments
- Equivalent GUI functionality found with the Design Studio
- Documentation Link: http://pic.dhe.ibm.com/infocenter/wtxdoc/v8r4m0/index.jsp?topic=%2Fcom.ibm.websphere.dtx.utilcmd.doc%2Ftopics%2Fg_utility_commands_Overview_of_Type_Designer_Utility_Commands.htm

Type Designer Utilities - tanalyze

- Analyze one or more type trees
- Eliminates the need to open individual trees in the Design Studio
- Syntax: tanalyze <.mtt file> { -L | -S | -L –S } [-R [<.dbe file name/location >]] [-SAVE] [(-LOG [. | <log file name/location >] [-FAIL] [-VERBOSE] [-APPEND]) | -NOLOG]
  - .mtt file Type tree file name that needs to be analyzed (required)
  - -L Analyze type tree logically
  - -S Analyze type tree by structure
  - -R Specify the location and/or file name for the analyzing result file

Type Designer Utilities - tanalyze

- -SAVE Save the type tree after analysis.
- -LOG Enables logging (default). If not specified, displays log information to the screen
- -LOG options Log tanalyze options
  - . | <log file name/location > Specify the location and/or file name for the log file. If no name is specified it will take the basename of the tree being analyzed with an extension of .log
  - -FAIL Log/Report failures only to the log file
  - -APPEND Append to the existing log file
  - -VERBOSE Verbose log file
- -NOLOG Disable logging

Type Designer Utilities - tanalyze

- Example: tanalyze.exe mytree.mtt -L -S -SAVE
  –R mytree_failure.dbe -LOG tanalyze_summary.log
  -APPEND -FAIL -VERBOSE
- NOTE: -LOG, -NOLOG, -FAIL, -APPEND and –VERBOSE apply to most if not all of the Utility Commands and have the same meaning for the respective utility

Type Designer Utilities - timport

- Allows you to import various metadata files (mts, xsd, dtd) and COBOL Copybook (cpy) to create a WTX Type Tree
- Files that can be imported
  - mts – WTX Type Tree Maker document
  - xsd – xml schema file
  - dtd – Document Type Definition file
  - cpy – COBOL Copybook
- Syntax: timport [-IMP <IMPORTER OPTIONS> [-HELP]] [-NO] [-O <.mtt *file name/location* >] [-KEEPMTS <.mts *filename*>] [(-LOG [. | <*log file name/location*>] [-FAIL] [-VERBOSE] [-APPEND]) | -NOLOG]

Type Designer Utilities - timport

- IMPORTER OPTIONS:
  - MTS <mts file> | <mts file>
  - XMLSCHEMA [<.xsd file> {-LANG [WESTERN | JAPANESE]}

    {-VALIDATION [XERCES | CLASSIC]}]
  - XMLDTD [<.dtd file> {-LANG [WESTERN | JAPANESE]}

    {-VALIDATION [XERCES | CLASSIC]}]
  - COPYBOOK [<filename> { -BYTEORDER NATIVE|BIGENDIAN|LITTLEENDIAN} {-CHARSET NATIVE|<code page> -CICS]

Type Designer Utilities - timport

- IMPORTER OPTIONS (Continued):
  - For COBOL COPYBOOK imports, the COPYBOOK to be imported must be fully qualified (path/filename)
  - LANG - Language type that describes the execution-time data. Valid values are WESTERN (default) and JAPANESE
  - VALIDATION - Validation type that describes the execution-time data. Valid values are XERCES (default) and CLASSIC
  - BYTEORDER - The convention a machine processor uses to position its lowest byte within a word to begin either from the leftmost or the rightmost position and describes the execution-time data. Valid values are NATIVE (default), BIGENDIAN and LITTLEENDIAN. Applies to binary numbers.

Type Designer Utilities - timport

- IMPORTER OPTIONS (Continued):
  - CHARSET - The standard collection of letters, numbers and symbols that describes the execution-time data. The values for the character set argument are any one of the valid character set codes (code page). If no character set argument is specified, the default value is NATIVE
  - -CICS Specifies to the importer that it should generate a type tree for a CICS adapter.
  - Brackets [ ] are required to enclose the IMPORTER OPTIONS type parameters
- -HELP Display syntax help for the –IMP options

Type Designer Utilities - timport

- -NO Prohibit an existing type tree from being overwritten.
- -O Specifies the location and/or name of the generated type tree. If not specified when importing a tree, the resulting type tree will be saved to the WTX installation port subdirectory. When importing a MTS file and –O is not specified the tree will be saved to the location specified in the NEWTREE=Filename entry of the mts file. If the Filename does not specify a path, it will be saved to the port subdirectory of your WTX installation.
- -KEEPMTS <mts filename> Keep the intermediate file created during the import process and name it as specified by <mts filename>
- -LOG Enable logging (default). Same options as tanalyze
- -NOLOG Disable logging

Type Designer Utilities - timport

- EXAMPLES:
  - timport.exe -IMP XMLDTD ["C:\IBM\WebSphere Transformation Extender 8.4\examples\xml\podata.dtd" -LANG WESTERN -VALIDATION XERCES] -LOG "timport_summary.log" –FAIL -APPEND –VERBOSE –O podata.mtt
  - timport.exe -IMP XMLSCHEMA ["C:\IBM\WebSphere Transformation Extender 8.4\examples\xml\ipo.xsd” -LANG WESTERN -VALIDATION XERCES] -LOG "timport_summary.log“   -FAIL -APPEND –VERBOSE –O ipo.mtt
  - timport.exe -IMP COPYBOOK ["C:\IBM\WebSphere Transformation Extender 8.4\examples\general\copybook\wrapped.cpy"] –LOG mylog.txt -VERBOSE -FAIL –O mycb.mtt

Type Designer Utilities - texport

- Export a particular type within a tree or the entire type tree to a mts file.
- Useful to make global changes with a text editor. Import and analyze once changes are completed.
- Syntax: texport <.mtt *file*> [-T <*type name*>] [-NO] [-O <.mts *file name/location*>] [(-LOG [. | <*log file name/location*>] [-FAIL] [-VERBOSE] [-APPEND]) | -NOLOG]
  - .mtt file Type tree file name to be exported (required)
  - -T <type name> Specify the type to be exported. Without this option, the entire tree is exported.

Type Designer Utilities - texport

- -NO Do not overwrite an existing mts file
- -O Path and/or mts file name
- -LOG Enable logging (default). -LOG has the same parameters as listed with the tanalyze utility
- -NOLOG Disable logging.

Type Designer Utilities - texport

- EXAMPLES:
  - texport mytree.mtt -O C:\MyDev\ExportResults\mytree.mts -LOG -FAIL –VERBOSE –APPEND
  - texport mytree.mtt -T MyType -O C:\MyDev\ExportResults\2012-02-28_mytree.mts –LOG –FAIL –VERBOSE -APPEND

Type Designer Utilities - tbcconv

- Used to automatically convert certain objects in a type tree sized in bytes to characters from the command line.
- Max/Min sizes of Bytes and Characters introduced with version 8.2
- Version 8.1 and earlier did not have Max/Min size representation of characters.
- Max/Min sizes of characters introduced to accommodate a multitude of character sets
- The behavior of the command depends on the object types in the type tree being converted (Text, Syntax, Number, and Date & Time) and the specific command options you use. The object types are represented by the Item Subclass property values in the type tree.

Type Designer Utilities - tbconv

- Syntax: tbcconv <.mtt *file*> [-P] [-R [<.dbe *file name/location* >]] [(-LOG [. | <*log file name/location* >] [-FAIL][-VERBOSE] [-APPEND]) | -NOLOG]
  - <.mtt *file*> - Type tree name to convert
  - -P Pad sizing option. Used to indicate for those items that have the PAD characteristic turned on, the PAD sizing value for the SizedAs property will be changed from Bytes to Characters. It also determines how the PAD value is sized when the data is being validated and written as output.
  - Items that have the Pad characteristic are Text, Date & Time and Number. Syntax items have no Pad property.

Type Designer Utilities - tbconv

- For those objects that meet this criteria (the Pad property is set to Yes, and Byte is specified for Text, Date & Time and Number items for the SizedAs property in the type tree), when the tbcconv utility command is run with the -P option, the value for the SizedAs property set as Byte will be changed to Character.
- -R [<.dbe *file name/location* >] Results file. Display results of the conversion to the screen or to the file specified by [<.dbe *file name/location* >]
- -LOG Enable logging (default). -LOG has the same parameters as listed with the tanalyze utility.
- -NOLOG Disable logging.

Map Designer Utilities

- **mcompile** – Build/compile maps
- **mimport** - Import an xml file to a map source
- **mexport** - Export a map source file to a XML file
- **mreport** - Generate a map report HTML file from a map source file
- Executed in Windows environments only
- Equivalent GUI functionality found with the Design Studio
- Documentation Link: http://pic.dhe.ibm.com/infocenter/wtxdoc/v8r4m0/index.jsp?topic=%2Fcom.ibm.websphere.dtx.utilcmd.doc%2Ftopics%2Fg_utility_commands_Overview_of_Map_Designer_Utility_Commands.htm

Map Designer Utilities - mcompile

- Build/Compile one or more map source files
- Syntax format 1 (list maps if for DataPower or WTX):
  - mcompile <.mms *file*> -L [-DP | -TX]

- Syntax format 2 (compile all maps):
  - mcompile <.mms *file*> -A [-K] [-E] [-NO] [-DP] [-TX]

    [-P <*platform*>] [-O <.mmc *file name/location*>]

    [-R <.mme *file name/location*>]

    [(-LOG [. | <*log file name/location*>] [-FAIL] [-VERBOSE]

    [-APPEND]) | -NOLOG]

Map Designer Utilities - mcompile

- Syntax format 3 (compile specific map):

  mcompile <.mms *file*> <*executable map*> [-E] [-NO] [-P <*platform*>]

  [-O <.mmc *file name/location*>] [-R <.mme *file name/location*>]

  [(-LOG [. | <*log file name/location*>] [-FAIL] [-VERBOSE] [-APPEND])

  | -NOLOG]

Map Designer Utilities - mcompile

- Syntax 1 options:
  - -L List individual map names within the map source file
  - -DP List only those maps to be compiled for DataPower
  - -TX List only those maps to be compiled for WTX
  - What maps are compiled for can be set in the Design Studio in the Map Settings – MapRuntime option.
  - -L, -DP and –TX return the map name and the total number of executable maps

Map Designer Utilities - mcompile

- Syntax 2 options:
  - -A Builds all the maps within the map source file (mms file)
  - -DP Builds all the DataPower maps in the map source file
  - -TX Builds all the WTX maps in the map source file
  - -K Continue compiling all maps even if an error occurs. Without this option map compilation will stop with the first error encountered and not compile remaining maps. -K can only be used in conjunction with the –A option.

Map Designer Utilities - mcompile

- -E Report map compiler errors only. Map compiler warnings are ignored
- -NO Do not overwrite an existing compiled map
- -P <platform> Platform to compile the map for. Valid values are:

  - AIX
  - HPITANIUM
  - LINUX
  - SOLARIS
  - WINDOWS
  - ZLINUX
  - ZOS

Map Designer Utilities - mcompile

- -O Specify a path and/or file name location for the compiled maps
- -R Specify a path and/or file name location for the results of the compilation messages (mme file)
- -LOG Enable logging (default). -LOG has the same parameters as listed with the tanalyze utility
- -NOLOG Disable logging.

Map Designer Utilities - mcompile

- Syntax 3 options:
  - <.mms *file*> Map source file name
  - <*executable map*> Particular map within the mms file to compile
  - -E Report map compiler errors only. Map compiler warnings are ignored
  - -NO Do not overwrite an existing compiled map

Map Designer Utilities - mcompile

- Syntax 3 options:
  - -P <*platform*> Platform to compile the map for. Valid values are:
    - AIX
    - HIPITANIUM
    - LINUX
    - SOLARIS
    - WINDOWS
    - ZLINUX
    - ZOS
  - -O Specify a path and/or file name location for the compiled maps
  - -R Specify a path and/or file name location for the results of the compilation messages (mme file)

Map Designer Utilities - mcompile

- Syntax 3 options:
  - -LOG Enable logging (default). -LOG has the same parameters as listed with the tanalyze utility
  - -NOLOG Disable logging.

Map Designer Utilities - mcompile

- mcompile runmap.mms –L

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

- mcompile runmap.mms gethours -E -P AIX -O C:\MyDev\Compiledmaps\*.mmc -R C:\MyDev\CompiledResults\mymaps_*.mme -LOG C:\MyDev\CompiledResults\ *.buildresults.txt -FAIL –APPEND

Map Designer Utilities - mimport

- Used to import an xml file and create a map source file from the command line
- Syntax: mimport <.xml *file*> [-NO] -O <.mms *file name/location*>]
    - [(-LOG [. | <*log file name/location*>] [-FAIL][-APPEND][-VERBOSE]) |-NOLOG]
  - -NO   Do not overwrite existing map source file
  - -O   Map source name
  - -LOG   Enable logging (default). -LOG has the same parameters as listed with the tanalyze utility
  - -NOLOG Disable logging.

Map Designer Utilities - mimport

- When to use
  - Import exported maps that were generated by the mexport utility.
    - Large scale changes made to exported maps with a text editor.
    - Maps that were previously exported, should also have their trees exported (texport) and then imported (timport) before the map is imported.
  - Import after the Any-2-XML utility (dtxany2xml) generates a xml file

Map Designer Utilities - mimport

- Import after WSDL import feature a creates xml file. For additional about the WSDL import process, refer to: http://pic.dhe.ibm.com/infocenter/wtxdoc/v8r4m0/index.jsp?topic=%2Fcom.ibm.websphere.dtx.ttimporters.doc%2Fconcepts%2Fc_type_import_WSDL_Importer.htm&resultof%3D%2522%2577%2573%2564%256c%2522%2520%2522%2569%256d%2570%256f%2572%2574%2522%2520%2522%2578%256d%256c%2522%2520%2522%256d%2561%2570%2522%2520

- Example: mimport myxmlfile.xml -O C:\MyDev\ImportResults\myimportedmaps.mms -LOG -APPEND

Map Designer Utilities - mexport

- Used to export a map source file to an xml file from the command line
- Useful to create text version of map(s) to make large scale changes
- Syntax: mexport <.mms *file*> [-NO] [-O <.xml *file name/location*>] [(-LOG [.| <*log file name/location*>] [-FAIL][-APPEND] [-VERBOSE]) | -NOLOG]
  - -NO Do not overwrite existing xml file
  - -O Exported map file (xml) name
  - -LOG Enable logging (default). -LOG has the same parameters as listed with the tanalyze utility
  - -NOLOG Disable logging.

Map Designer Utilities - mexport

- Example:

  mexport mymaps.mms -O C:\MyDev\ExportResults\mymaps.xml -LOG –APPEND

Map Designer Utilities - mreport

- mreport - used to generate a map report in HTML format containing information about a map source
- Syntax: mreport <.mms *file*> [-NO] [-O <.html *file name/location*>] [(-LOG [. | <*log file name/location*>] [-FAIL][-VERBOSE] [-APPEND]) |-NOLOG]
  - -NO Do not overwrite existing xml file
  - -O Exported map file (xml) name
  - -LOG Enable logging (default). -LOG has the same parameters as listed with the tanalyze utility
  - -NOLOG Disable logging.

Map Designer Utilities - mreport

- mreport mymmsfile.mms -O C:\MyDev\MapReportResults\myhtmlfile.html -LOG –APPEND

- Displays Audit, Trace, Page Size/Count, Input/Output cards and rules in HTML format viewable as a web page.

Integration Flow Designer Utilities

- **sdeploy** – Analyze systems, build maps in the system, generate and deploy the system to the specified server from the command line
- **msdimport** - Import an xml file to a system definition (.msd) source file in a system from the command line,
- **msdexport** - Export a system source definition (.msd) file from the command line
- Executed in Windows only environments
- Equivalent GUI functionality found with the Integration Flow Designer
- Documentation Link: http://pic.dhe.ibm.com/infocenter/wtxdoc/v8r4m0/index.jsp?topic=%2Fcom.ibm.websphere.dtx.utilcmd.doc%2Ftopics%2Fg_utility_commands_Overview_of_the_IFD_Utility_Commands.htm

Integration Flow Designer Utilities - sdeploy

- Syntax format 1 (List/Display):
  - sdeploy <.msd file> -L

- Syntax format 2 (Analyze/Build/Generate):
  - sdeploy <.msd file> -A [-K] [-NO]

    <ANALYZE OPTIONS | BUILDMAPS OPTIONS | GENERATE OPTIONS>

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

    -NOLOG]

Integration Flow Designer Utilities - sdeploy

- Syntax format 3 (Deploy):
  - sdeploy <.msd file> <system> [-NO]

    <ANALYZE OPTIONS | BUILDMAPS OPTIONS |GENERATE OPTIONS|

    DEPLOY OPTIONS>

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

    | -NOLOG]

Integration Flow Designer Utilities - sdeploy

- Syntax format 1 options:
  - <.msd file>　The system to be listed
  - -L　Lists/Displays all the systems within the msd

- Example: sdeploy launcher.msd -L

Integration Flow Designer Utilities - sdeploy

- Syntax format 2 options:
  - -A Analyze and compile all maps that are present in all systems and generate msl files for all systems present in the msd file (required)
  - -K Continue compilation even if an error is encountered. Must be used with the –A option
  - -NO Do not overwrite an existing msl

Integration Flow Designer Utilities - sdeploy

- Syntax format 2 options:
  - ANALYZE OPTIONS: -ANALYZE [-EMODE ES | CS] [-R <*MSEAnlz.txt file name/location*>]
    - -ANALYZE Analyze the msd (do not deploy maps or msd)
    - -EMODE Analyze the msd for Launcher (ES) or Command Server (CS). If –EMODE is not specified, the default is –EMODE ES
    - -R < MSEAnlz.txt file name/location> file name to save deploy analyze results to. Without a file name specified, the default is MSEAnlz_system_name.txt

Integration Flow Designer Utilities - sdeploy

- Syntax format 2 options:
  - BUILDMAPS OPTIONS: -BUILDMAPS [-P <platform>] [-O <file location>] [-R <MSEBlds.txt file name/location>]

-BUILDMAPS	Build/Compile the maps within the msd
-P	Platform to compile the map for. Valid values are:

- AIX
- LINUX
- WINDOWS
- ZOS
- HPITANIUM
- SOLARIS
- ZLINUX

Integration Flow Designer Utilities - sdeploy

- Syntax format 2 options:
  - BUILDMAPS OPTIONS:
    - -O <file location> Specify a path and/or file name for where compiled maps are to be saved.
    - -R < *MSEBlds.txt file name/location*> File name to save deploy build results to. Without a file name specified, the default is MSEBlds_system_name.txt

Integration Flow Designer Utilities - sdeploy

- Syntax format 2 options:
  - GENERATE OPTIONS:
    - -GENERATE [-EMODE ES | CS] [-P <*platform*>] [-O <.msl *file name/location*> or <command server file>]
    - -GENERATE | -GENERATE –EMODE ES Create the msl from the specified msd
    - -GENERATE –EMODE CS Create a text file used to execute a map with the Command Server.

Integration Flow Designer Utilities - sdeploy

- Syntax format 2 options:
  - GENERATE OPTIONS
    - -P Platform to compile the map for. Valid values are:

• AIX	• HPITANIUM	• LINUX
• SOLARIS	• WINDOWS	• ZOS
• ZLINUX		

    - -O <.msl *file name/location*> or <command server file>
    - If EMODE = ES, generate a msl (binary file)
    - If EMODE = CS, generate a text file to be used with the dtxcmdsv command. Without the –O option, the generated text file is named by the basename of the msd file (no extension). To execute: dtxcmdsv @<command server file>

Integration Flow Designer Utilities - sdeploy

- Syntax format 2 options:
  - -LOG Enable logging (default). -LOG has the same parameters as listed with the tanalyze utility
  - -NOLOG Disable logging.

Integration Flow Designer Utilities - sdeploy

- Analyze example:

  -
    ```
    sdeploy launcher.msd -ANALYZE -A -EMODE ES -R
    C:\MyWTXSystems\AnalyzeResults\*.txt -LOG
    C:\MyWTXSystems\AnalyzeResults\*.results.txt -FAIL –APPEND
    ```
- Build all maps example:

  -
    ```
    sdeploy launcher.msd -BUILDMAPS -A -P WINDOWS -R
    C:\MyWTXSystems\BuildResults\*.txt -LOG
    C:\MyWTXSystems\BuildResults\*.results.txt -FAIL -APPEND
    ```

Integration Flow Designer Utilities - sdeploy

- Generate msl example:
  - sdeploy launcher.msd -A -GENERATE -P AIX -EMODE ES -LOG C:\MyWTXSystems\GenerateResults\*.results.txt -FAIL -APPEND

Integration Flow Designer Utilities - sdeploy

- Syntax format 3 (Deploy):
  - sdeploy <.msd file> <system> [-NO]

    <ANALYZE OPTIONS | BUILDMAPS OPTIONS |

    GENERATE OPTIONS| DEPLOY OPTIONS>

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

Integration Flow Designer Utilities - sdeploy

- Syntax format 3 (Deploy):
  - Additional option of DEPLOY OPTIONS compared to syntax 2

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

- -DEPLOY <script name> Deploy the deploy script using the script name to the specified server in the msd file.
- -EMODE ES Generate a msl (binary file)
- -EMODE CS Generate a text file to be used with the dtxcmdsv command

Integration Flow Designer Utilities - sdeploy

- Syntax format 3 (Deploy):
  - DEPLOY OPTIONS (continued)

- -SERVER <server name>   Server to deploy to
- -LOCAL   Deploy to the local host
- -USER <userid> -PW <password> -IP <IP address>   Deploy to the server specified by the IP address, logging in with the userid and password.
- -R MSEDploy_*system_name.err*

Integration Flow Designer Utilities - sdeploy

- Syntax format 3 (Deploy):
  - -LOG Enable logging (default). -LOG has the same parameters as listed with the tanalyze utility
  - -NOLOG Disable logging.

Integration Flow Designer Utilities - sdeploy

- Generate msl for particular system and deploy to a specified server example:
  - sdeploy launcher.msd NewSystem -DEPLOY MyScript -EMODE ES -SERVER MyServer -R C:\MyWTXSystems\DeployResults\*.txt -LOG C:\MyWTXSystems\DeployResults\*.results.txt -FAIL -APPEND
- Generate msl for particular system and deploy to a specified server using a particular userid/password example:
  - sdeploy launcher.msd NewSystem -DEPLOY MyScript -EMODE ES -USER userid -PW password -IP 192.82.65.46 -R C:\MyWTXSystems\DeployResults\*.txt -LOG C:\MyWTXSystems\DeployResults\*.results.txt -FAIL -APPEND

Integration Flow Designer Utilities - msdimport

- Import an xml file to a system definition (.msd) source file
- Used to import after changes made via text editor or reimport due to corruption
- Syntax: msdimport <.xml *file*> [-NO] [-O <.msd *file name/location*>]

  [-REFSYS] [-REFSVR] [-OSYS] [-OSVR]
  [-ASVR [<*server name*>]] [-ASYS [<*system names*>]]

  [(-LOG [. | <*log file name/location*>] [-FAIL] [-APPEND]

  [-VERBOSE]) |-NOLOG]

  - -NO Do not overwrite an existing msl
  - -O <.msd *file name/location*>
  - -REFSYS Import the dependent systems of the system being imported

Integration Flow Designer Utilities - msdimport

- -REFSVR — Import the dependent systems of the system being imported
- -OSYS — Override any system being imported if it already exists in the msd of the same name
- -OSVR — Override any server being imported if it already exists in the msd of the same name
- -ASVR [<*server name*>] — Import only servers from the xml file. With no server name specified, all servers are imported. If a server name is specified, only import that specific server(s). The actual system is not imported, just the server name.

Integration Flow Designer Utilities - msdimport

- -ASYS [<*system name*>] Import only systems from the xml file. With no system name specified, all systems are imported. If a system name is specified, only import that specific systems(s)
- -LOG Enable logging (default). -LOG has the same parameters as listed with the tanalyze utility
- -NOLOG Disable logging

- Examples:
  - msdimport myxmlfile.xml –O C:\MyDev\ImportResults\myimportedsystem.msd -LOG –APPEND
  - msdimport myxmlfile.xml –O C:\MyDev\ImportResults\myimportedsystem.msd –ASYS myWTXsystem -LOG –APPEND

Integration Flow Designer Utilities - msdexport

- Export a msd file to xml file
- Used to make changes via a text editor.
- Syntax: msdexport <.msd *file*> [-NO] [-O <.xml *file name/location*>] [-EPWD] [-REFSYS] [-REFSVR] [-ASVR [<*server name*>]] [-ASYS [<*system names*>]] [(-LOG [. | <*log file name/location*>] [-FAIL][-APPEND] [-VERBOSE]) |-NOLOG]

<.msd file>	System name to export
-NO	Do not overwrite an existing exported msd (xml file)
-O	Location/Name of exported msd
-EPWD	Encrypt the password associated with servers

Integration Flow Designer Utilities - msdexport

- -REFSYS Export the dependent systems of the system being exported
- -REFSVR Export the dependent systems of the system being exported
- -ASVR [<*server name*>] Export only server information from the msd. With no server name specified, all servers are exported. If a server name is specified, only export that specific server(s). The actual system is not exported, just the server name.
- -ASYS [<*system name*>] Export only systems from the msd file. With no system name specified, all systems are exported. If a system name is specified, only export that specific systems(s)

Integration Flow Designer Utilities - msdexport

  - -LOG Enable logging (default). -LOG has the same parameters as listed with the tanalyze utility
  - -NOLOG Disable logging
- Example: msdexport mysystem.msd -O C:\MyDev\ExportResults\mysystem.xml –LOG -APPEND

Search APARs, Technotes and Info Center

- WTX Support Site: http://www-947.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_Transformation_Extender
- WTX Documentation (Information Center, Release Notes) http://www-01.ibm.com/software/integration/wdatastagetx/library/

Encountering an issue - Open a PMR

- On-Line problem submission:
  - http://www-306.ibm.com/software/support/probsub.html
- WTX MustGather: http://www-01.ibm.com/support/docview.wss?rs=2320&uid=swg21244201

Summary

- Review on-line Web based documentation for Utility Command usage and examples.

- Part II of WTX Utility Commands including Resource Registry, XML and Launcher commands to be offered on February 6, 2013

- Visit the WTX support page for additional information: http://www-947.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_Transformation_Extender

Examples

Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at: http://www.ibm.com/software/websphere/support/supp_tech.html
- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at: http://www.ibm.com/developerworks/websphere/community/
- Join the Global WebSphere Community: http://www.websphereusergroup.org
- Access key product show-me demos and tutorials by visiting IBM® Education Assistant: http://www.ibm.com/software/info/education/assistant
- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically: http://www.ibm.com/software/websphere/support/d2w.html
- Sign up to receive weekly technical My Notifications emails: http://www.ibm.com/software/support/einfo.html

Connect with us!

1. **Get notified on upcoming webcasts**
Send an e-mail to wsehelp@us.ibm.com with subject line "wste subscribe" to get a list of mailing lists and to subscribe

2. **Tell us what you want to learn**
Send us suggestions for future topics or improvements about our webcasts to wsehelp@us.ibm.com

3. **Be connected!**
Connect with us on Facebook
Connect with us on Twitter

Questions and Answers