bannerr_rmx.gif

DIS Conversion Tool Installation Guide

[Overview 1](#_Toc332639736)

[Pre-Installation Steps 2](#_Toc332639737)

[Installation 3](#_Toc332639738)

[To Install DIS Conversion Tool 3](#_Toc332639739)

[Post Installation Steps 9](#_Toc332639740)

[To Create XML Format Definition File 9](#_Toc332639741)

[To Create DIS Access Database 13](#_Toc332639742)

# Overview

DIS Conversion Tool is an independent utility that helps clients to import arbitrary, third-party data files into RMX via the DA DIS template. The utility can transfer data from fixed width text data files into an Access database file, as defined by a format definition file, for processing by the DA DIS template.

Primary user functions consist of specifying files, initializing the import process and optionally clearing or saving log files. The tool will asynchronously keep the user apprised of the import process status and also provide clear, understandable error messages in case of abnormal execution or invalid input.

# Pre-Installation Steps

The following steps must be completed before continuing to the Installation phase.

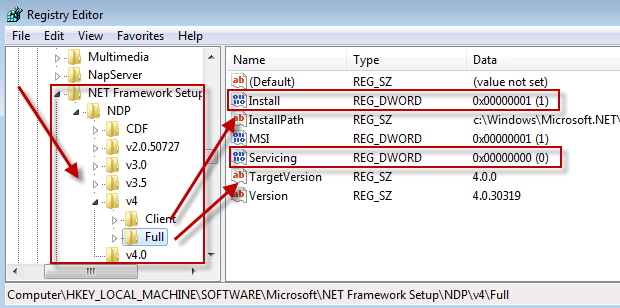
1. Request the DIS Conversion Tool installation files from [www.riskmaster.com](http://www.riskmaster.com) or by contacting RISKMASTER Support directly.
2. Download the DIS Conversion Tool installation files to a local machine or server where the Tool will be installed.
3. Confirm that the Microsoft .Net 4.0 Framework is installed on the machine/server where the DIS Conversion Tool will run.

Please refer to the following website for more details on determining which Microsoft .NET Framework versions are installed:

[**http://msdn.microsoft.com/en-us/kb/kbarticle.aspx?id=318785**](http://msdn.microsoft.com/en-us/kb/kbarticle.aspx?id=318785)

Use the registry information below to determine which version(s) of the Microsoft .NET Framework are installed.

1. Select Start, enter regedit in the Search programs and files box and then press Enter.
2. In the Registry Editor, locate the Registry Key Name in the table below and check the value for the related .NET Framework.



1. Confirm the 2007 or 2010 Microsoft Office System Driver is installed on the machine/server where the DIS Conversion tool will run.

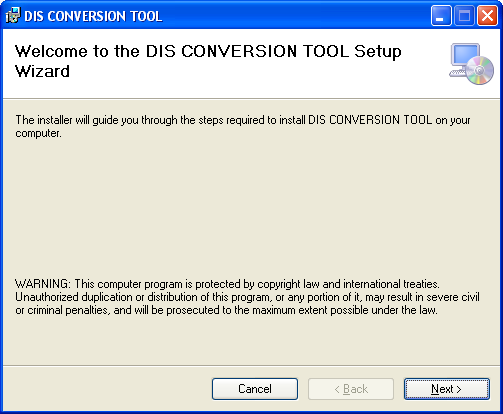
The Microsoft.ace.oledb.12.0 driver (AccessDatabaseEngine.exe) can be downloaded from the Microsoft Download Center on the following website: <http://www.microsoft.com/downloads/en/details.aspx?FamilyID=7554F536-8C28-4598-9B72-EF94E038C891&displaylang=en>. If the machine is a 64bit machine, then the 32bit driver must be installed.

# Installation

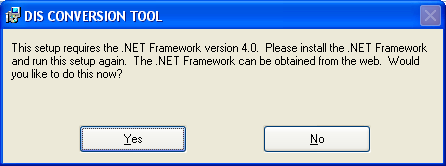
## To Install DIS Conversion Tool

1. Run DIS CONVERSION TOOL SETUP.msi. The install files need to be downloaded as part of the Pre-Installation steps.

The DIS Conversion Tool window displays.



1. If .Net Framework 4.0 is not installed in the machine, it will display the message as shown below.

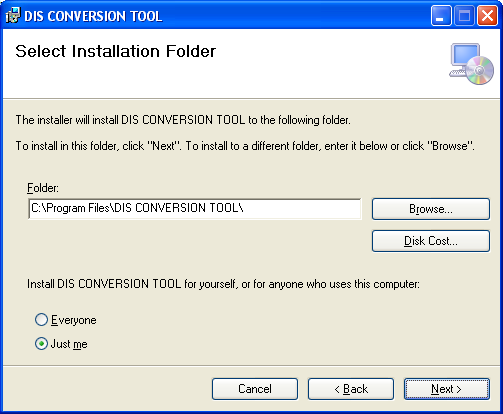


Select ‘Yes’ to open a link [***http://www.microsoft.com/enus/download/details.aspx?id=17113***](http://www.microsoft.com/enus/download/details.aspx?id=17113) for downloading Microsoft .NET Framework 4 Client Profile (Web Installer).

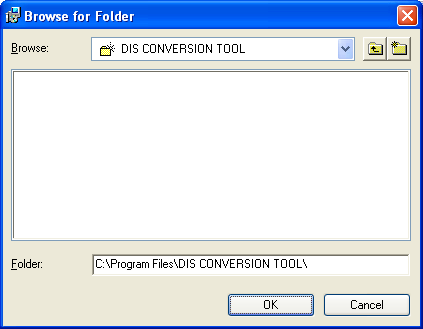
Select ‘No’ to close the application.

1. Select the Next button.

The Select Installation Folder window displays.

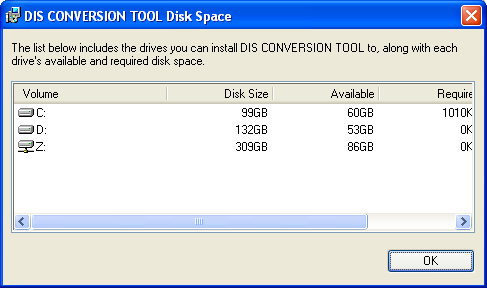


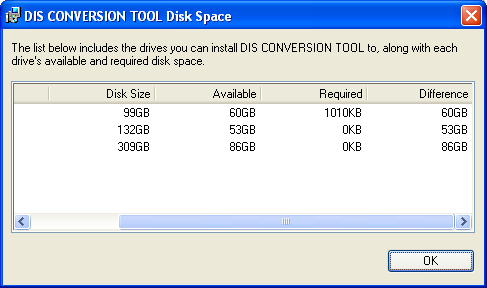
1. In the Installation Folder field, enter the installation folder or browse to the installation directory where the DIS Conversion Tool will be installed.
2. Select “Everyone” or “Just me” based on your requirement.
3. Select ‘Browse’ button to install the tool in different folder. It gives the option as below.



The Disk Cost button specifies the space available in each drive and space required for this tool.

For Example-

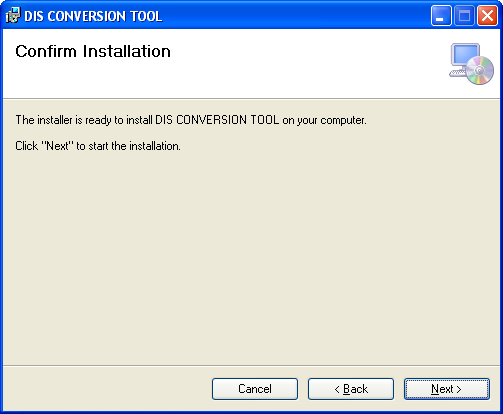




In this way, you can view the space availability/requirement in each drive and by selecting ‘Browse’ button, can change the folder path.

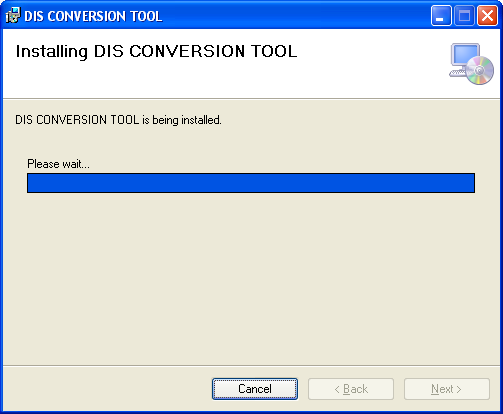
1. Select the Next button.

The Confirm Installation window displays.



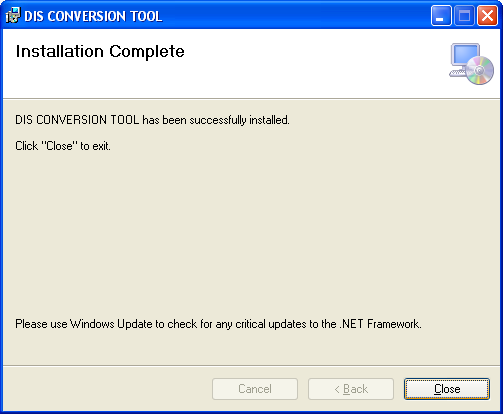
1. Select the Next button to start the installation.

The Installing DIS Conversion Tool window opens and displays the installation progress.



The Installation complete window displays.

The window will display a message confirming the installation process completed.



1. Select the Close button to exit the installation.
2. Verify that the following files were placed in the folder selected in Step 3.
3. DA\_DIS\_Conversion\_Tool.exe
4. FrmtDef.xsd

# Post Installation Steps

The following steps must be completed prior to running the DIS Conversion Tool.

## To Create XML Format Definition File

An XML Format Definition File must be created prior to converting data using the DIS Conversion Tool. The Format Definition File will contain the data required by the application to determine how fields present in the input data file are to be mapped into the output DIS Access file. The Conversion Tool does not include any kind of automation or machine assistance with generating a FmtDef file, this file will need to be generated manually and must adhere to a certain format. The XML Schema for the FmtDef file is as follows. All Format Definitions will be validated against this XSD; formats that do not conform will prevent the execution of a process.

XML Format:

<?xml version="1.0" encoding="utf-8"?>

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="FmtDef">

<xs:complexType>

<xs:sequence>

<xs:element name="SkipLines" type="xs:unsignedByte" />

<xs:element name="CommentString" type="xs:string" />

<xs:element name="Tables">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="Table">

<xs:complexType>

<xs:simpleContent>

<xs:extension base="xs:string">

<xs:attribute name="PK" type="xs:string" use="optional" />

</xs:extension>

</xs:simpleContent>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="Fixed">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="Field">

<xs:complexType>

<xs:sequence>

<xs:element name="Start" type="xs: nonNegativeInteger" />

<xs:element name="Length" type="xs: nonNegativeInteger" />

<xs:element name="Table" type="xs:string" />

<xs:element name="Column" type="xs:string" />

<xs:element minOccurs="0" name="Mapping">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="Map">

<xs:complexType>

<xs:attribute name="From" type="xs:string" use="required" />

<xs:attribute name="To" type="xs:string" use="required" />

</xs:complexType>

</xs:element>

</xs:sequence>

<xs:attribute name="Default" type="xs:string" use="required" />

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

Example of a FmtDef File:

<?xml version="1.0" encoding="utf-8"?>

<FmtDef>

<SkipLines>0</SkipLines>

<CommentString>--</CommentString>

<Tables>

<Table PK="ENTITY\_ID">ENTITY</Table>

<Table PK="EMPLOYEE\_EID">EMPLOYEE</Table>

</Tables>

<Fixed>

<Field>

<Start>1</Start>

<Length>8</Length>

<Table>ENTITY</Table>

<Column>BIRTH\_DATE</Column>

</Field>

<Field>

<Start>9</Start>

<Length>1</Length>

<Table>ENTITY</Table>

<Column>SEX\_CODE</Column>

</Field>

<Field>

<Start>9</Start>

<Length>1</Length>

<Table>ENTITY</Table>

<Column>TITLE</Column>

<Mapping Default="">

<Map From="M" To="Mr." />

<Map From="F" To="Mrs." />

</Mapping>

</Field>

<Field>

<Start>65</Start>

<Length>5</Length>

<Table>EMPLOYEE</Table>

<Column>FULL\_TIME\_FLAG</Column>

<Mapping Default="0">

<Map From="FR" To="-1" />

</Mapping>

</Field>

<Field>

<Start>10</Start>

<Length>50</Length>

<Table>ENTITY</Table>

<Column>LAST\_NAME</Column>

</Field>

<Field>

<Start>110</Start>

<Length>12</Length>

<Table>EMPLOYEE</Table>

<Column>SALARY</Column>

</Field>

</Fixed>

</FmtDef>

FmtDef serves as the root node.

SkipLines allows the application to skip X number of lines at the beginning of the file. This format definition is optional.

CommentString allows the application to skip processing any rows in the file which begin with the specified string. This format definition is optional.

Tables serves as a meta-element containing a list of the tables required by the mapping, giving the application a heads up about which tables it will need to support in-memory without having to parse through the individual Target attributes specified in the following section.

Table defines a table implicated in the format. It uses a single optional attribute, PK to define the table’s primary key, if present. The table name is contained within the body of the element. When a PK is present, the same value will be used for all tables across a row.

Fixed serves as a meta-element. This indicates that the source file contains fixed-width data. Fixed is declared directly so that a CSV layout could be more easily added in the future. Note that support for a CSV layout is currently outside the scope of this document and project. Within the Fixed element are the individual field mappings.

Field element defines a single piece of data from the source file. Each Field requires four child elements:

* Start defines the row character offset where the field begins.
* Length defines the number of characters contained in the field.
* Table specifies the table the target field resides in.
* Target specifies which field in the table in the DIS Access file the data should be loaded into.

Field elements only have to map the fields the user actually wants from the file, the entire contents of a line (consuming the entire length) is not required. Field elements do not need to specify a data type; the data type for a field will be retrieved from the DIS Access file.

To aid in supporting additional types of data files, the specification for a field can also include a simple mapping of known values to other values. This is achieved by including a Mapping element in a Field. The Mapping must always include a Default value attribute and at least a single Map value. Each Map element must include a before-and-after pair of attributes specifying what each value from the file is mapped to in the output. Using this it is possible to have a single field on the input apply to multiple fields in the output.

## To Create DIS Access Database

Refer to the DA DIS Template Field mapping document for specifications for creating the DIS Access file. If the DA DIS Access file already exists, skip this step.