OpendX

personnel data exchange

OpenDX User Manual

OPEN OPTIONS™

ACCESS TECHNOLOGY

OpenDX personnel data exchange

This manual has been written for OpenDX™ version 6.8.0.0 or higher

Print Date: March 14, 2018
Manual Number: DX-6.8

Warranty

All Open Options products are warranted against defect in materials and workmanship for one year from the date of shipment. Open Options will repair or replace products that prove defective and are returned to Open Options within the warranty period with shipping prepaid. The warranty of Open Options products shall not apply to defects resulting from misuse, accident, alteration, neglect, improper installation, unauthorized repair, or acts of God. Open Options shall have the right of final determination as to the existence and cause of the defect. No other warranty, written or oral is expressed or implied.

OPEN OPTIONS®
ACCESS TECHNOLOGY

16650 Westgrove Dr | Suite 150
Addison, TX 75001
Phone: (972) 818-7001
Fax (972) 818-7003
www.ooaccess.com

Open Options, L.P. Software License Agreement and Warranty

THE ENCLOSED SOFTWARE PACKAGE IS LICENSED BY Open Options, L.P. TO CUSTOMERS FOR THEIR NON-EXCLUSIVE USE ON A COMPUTER SYSTEM PER THE TERMS SET FORTH BELOW.

DEFINITIONS: Open Options shall mean Open Options, L.P., which has the legal right to license the computer application known as OpenDX™ herein known as the Software. Documentation shall mean all printed material included with the Software. Licensee shall mean the end user of this Open Options Software. This Software Package consists of copyrighted computer software and copyrighted user reference manual(s).

LICENSE: Open Options, L.P., grants the licensee a limited, non-exclusive license (i) to load a copy of the Software into the memory of a single (one) computer as necessary to use the Program, and (ii) to make one (1) backup or archival copy of the Software for use with the same computer. The archival copy and original copy of the Software are subject to the restrictions in this Agreement and both must be destroyed or returned to Open Options if your continued possession or use of the original copy ceases or this Agreement is terminated.

RESTRICTIONS: Licensee may not sub license, rent, lease, sell, pledge or otherwise transfer or distribute the original copy or archival copy of the Software or the Documentation. Licensee agrees not to translate, modify, disassemble, decompile, reverse engineer, or create derivative works based on the Software or any portion thereof. Licensee also may not copy the Documentation. The license automatically terminates without notice if Licensee breaches any provision of this Agreement.

TRANSFER RIGHTS: Reseller agrees to provide this license and warranty agreement to the end user customer. By installation and acceptance of the software package, the end user customer and reseller agree to be bound by the license agreement and warranty.

LIMITED WARRANTY: Open Options warrants that it has the sole right to license the Software to licensee. Open Options further warrants that the media on which the Software is furnished will be free from defects in materials and workmanship under normal use for a period of ninety (90) days following the delivery of the Software to the licensee. Open Options' entire liability and your exclusive remedy shall be the replacement of the Software if the media on which the Software is furnished proves to be defective. This warranty is void if the media defect has resulted from accident, abuse, or misapplication. Open Options does not warrant that the Software will meet the end user customer requirements or that operation of the Software will be uninterrupted or that the Software will be error-free.

THE ABOVE WARRANTIES ARE THE ONLY WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. NEITHER OPEN OPTIONS, NOR ITS VENDORS SHALL BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF USE, INTERRUPTION OF BUSINESS, NOR FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND WHETHER UNDER THIS AGREEMENT OR OTHERWISE.

IN NO CASE SHALL OPEN OPTIONS' LIABILITY EXCEED THE PURCHASE PRICE OF THE SOFTWARE. The disclaimers and limitations set forth above will apply regardless of whether you accept the Software.

TERMINATION: Open Options may terminate this license at any time if licensee is in breach of any of its terms or conditions. Upon termination, licensee will immediately destroy the Software or return all copies of the Software to Open Options, along with any copies licensee has made.

APPLICABLE LAWS: This Agreement is governed by the laws of the State of Texas, including patent and copyright laws. This Agreement will govern any upgrades, if any, to the program that the licensee receives and contains the entire understanding between the parties and supersedes any proposal or prior agreement regarding the subject matter hereof.

Table of Contents

Installation 1

In This Chapter

√ OpenDX Overview
√ Installation and Configuration

Overview

OpenDX™ is a personnel data management program that delivers a seamless interface between the DNA Fusion™ access control system and an ADO-compliant database, such as Microsoft's Active Directory. It automates the data entry process of personnel/cardholder records into DNA Fusion and provides a user-friendly interface to configure the imported data.

OpenDX can perform automated data transfers from any valid data source or scan a specific directory for CSV or text files generated by third-party systems. The program, which contains built-in scripting support for Pascal and Basic programming languages, can be tailored to fit a variety of data import tasks.

Installation

The OpenDX installation does not require any knowledge of the software. To begin, open the setup file (Open DX Setup.exe) provided in the license e-mail or contact Open Options Technical Support to obtain the file.

! *The Fusion COM objects must be installed on the same workstation.*

1. **Double-click** the Open DX Setup.exe file.

   The Select Destination Location dialog opens.

2. **Click** Next to accept the default location or **click** Browse to select a different folder.

   Default location:

   - 32-bit OS — C:\Program Files\Importer
   - 64-bit OS — C:\Program Files (x86)\Importer

   The Select Start Menu Folder screen appears.

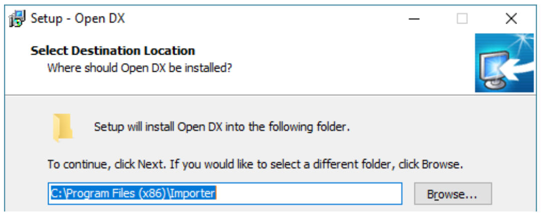
3. **Click** Next to accept the default Start Menu location or **click** Browse to select a different shortcut folder.

   The Ready to Install screen appears.

4. **Click** Install to start the installation process.

   If the OpenDX service is running, the installation will stop the service and a dialog will appear; **click** OK to continue.

5. When the installation is complete, **click** Finish.

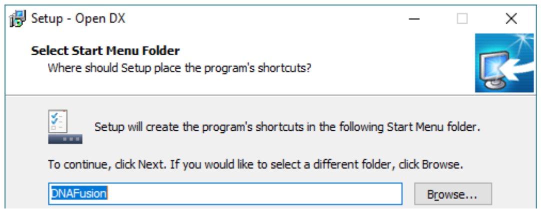


Select Destination Location
Where should Open DX be installed?

Setup will install Open DX into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files (x86)\Importer

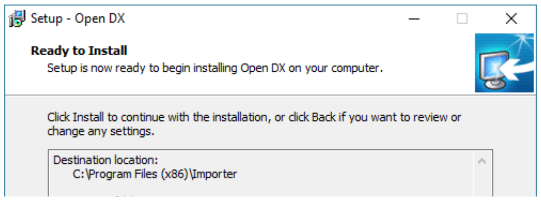


Select Start Menu Folder
Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

DNAFusion



Ready to Install
Setup is now ready to begin installing Open DX on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

Destination location:
C:\Program Files (x86)\Importer

Best Practices

Open Options recommends the following installation guidelines for OpenDX:

- Install OpenDX on a DNA server; a client workstation will slow the import process.
- Configure the OpenDX import service to run under a Windows user account.

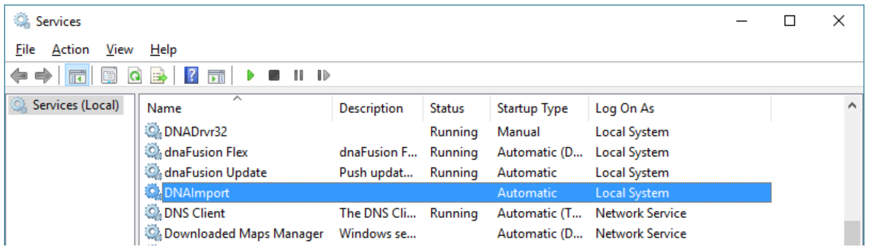
This Page Intentionally Left Blank

OpenDX Configuration

The DNAImport service must be configured to run under a specific user.

1. From the Control Panel, **open** the Administrative Tools dialog.
2. **Double-click** on Services.

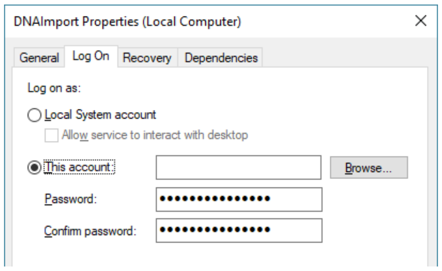
   The Services dialog opens.



Name	Description	Status	Startup Type	Log On As
DNADrvr32		Running	Manual	Local System
dnaFusion Flex	dnaFusion F...	Running	Automatic (D...	Local System
dnaFusion Update	Push updat...	Running	Automatic	Local System
DNAImport			Automatic	Local System
DNS Client	The DNS Cli...	Running	Automatic (T...	Network Service
Downloaded Maps Manager	Windows se...		Automatic (D...	Network Service

3. **Right-click** on the DNAImport service and **select** Properties.

   The DNAImport Properties (Local Computer) dialog opens.



Log on as:

( ) Local System account

[ ] Allow service to interact with desktop

(●) This account:

Password: •••••••••••••••

Confirm password: •••••••••••••••

4. **Select** the Log On tab.
5. **Select** This Account and **enter** the Username and Password.

   The account information must be obtained from the customer.
6. **Click** OK to save the settings.

This Page Intentionally Left Blank

Getting Started 2

In This Chapter

√ Starting OpenDX
√ OpenDX Environment
√ OpenDX Components

Starting OpenDX

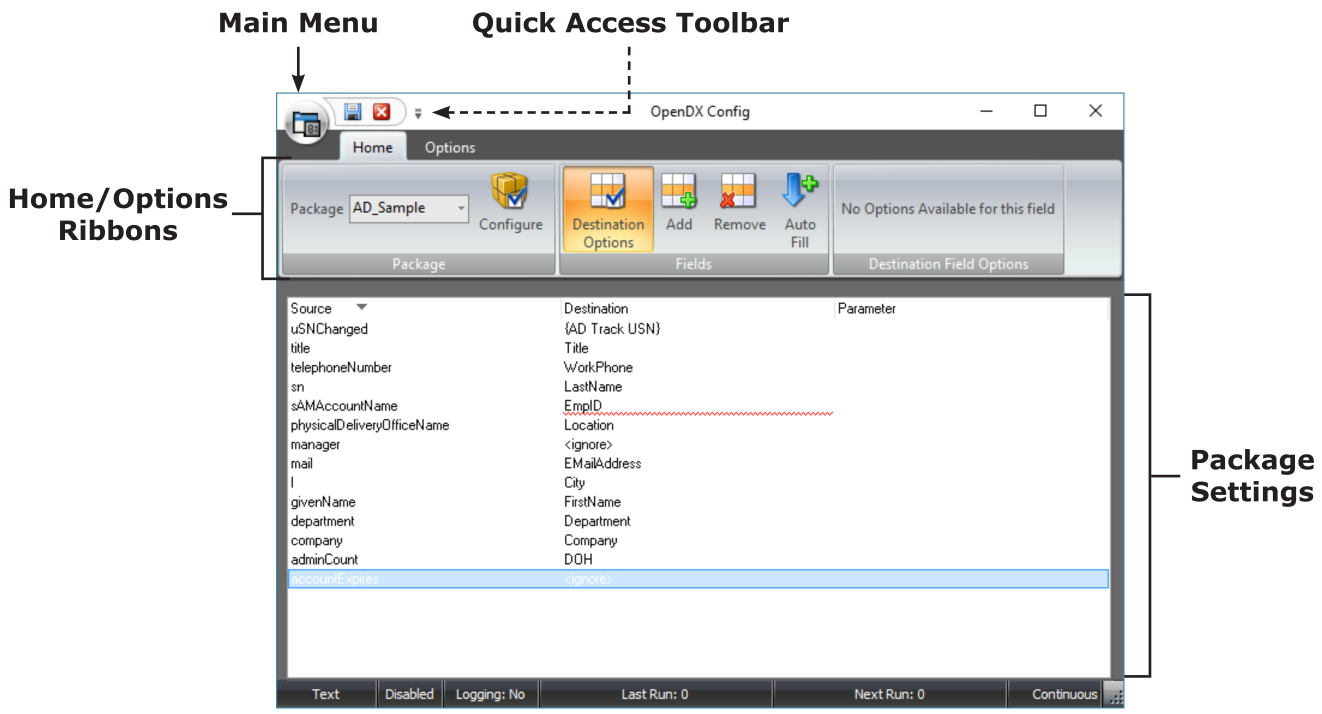
Once OpenDX is installed on the workstation, select ImportConfig from the Start Menu to launch the configuration program.

> *OpenDX requires additional licensing; contact your dealer for questions about licensing.*

The OpenDX Environment

The OpenDX interface is designed for user operability and navigation. The main screen, displayed in the image below, consists of five (5) primary elements:

- Main Menu
- Quick Access Toolbar
- Home Ribbon
- Options Ribbon
- Package Settings



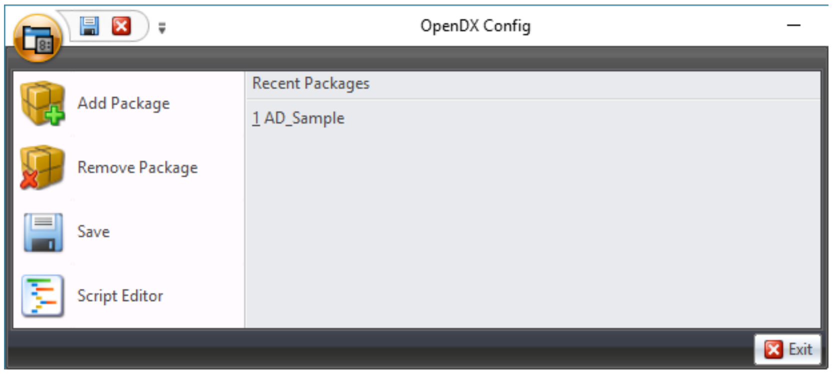
Source	Destination	Parameter
uSNChanged	{AD Track USN}	
title	Title	
telephoneNumber	WorkPhone	
sn	LastName	
sAMAccountName	EmpID	
physicalDeliveryOfficeName	Location	
manager	<ignore>	
mail	EMailAddress	
l	City	
givenName	FirstName	
department	Department	
company	Company	
adminCount	DOH	
accountExpires	<ignore>	

Text | Disabled | Logging: No | Last Run: 0 | Next Run: 0 | Continuous

This Page Intentionally Left Blank

Main Menu

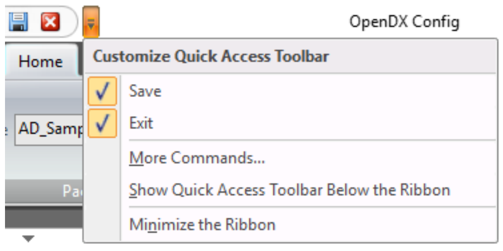
The Main Menu drop-down provides access to the Add Package, Remove Package, Save, and Script Editor commands. It also includes a quick-access list to open Recent Packages. Each item is discussed in detail in a later section of the manual.



Quick Access Toolbar

The Quick Access Toolbar is located to the right of the Main Menu. By default, the Save and Exit commands are available in the toolbar.

To customize the toolbar, click the drop-down arrow to the right of the toolbar and select one of the following options:

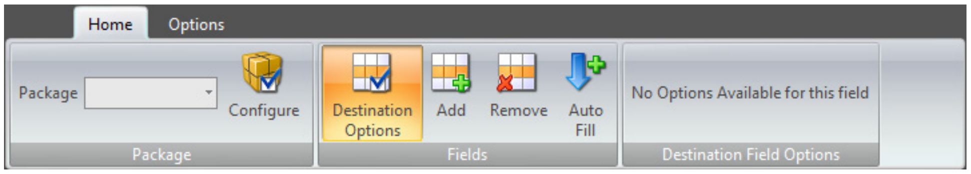

OpenDX Config
Home
AD_Samp
Customize Quick Access Toolbar
Save
Exit
More Commands...
Show Quick Access Toolbar Below the Ribbon
Minimize the Ribbon

- More Commands - Opens the Customize dialog to add and remove toolbars and toolbar commands.
- Show Quick Access Toolbar Below/Above the Ribbon - Toggles the location of the Quick Access Toolbar above or below the Home/Options Ribbon.
- Minimize the Ribbon - Toggles the Home/Options Ribbon in the OpenDX Config dialog.

With the Customize dialog open, **drag** *and* **drop** *Commands to and away from the Quick Access Toolbar or Home/Options Ribbons to add and remove the commands.*

Home Ribbon

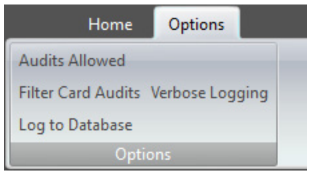
The Home Ribbon displays the Package, Fields, and Destination Field Options toolbars to open dialogs and perform specific tasks. It also populates the Package Settings in the data window below the ribbon, where the user can edit and configure settings for the active package.



- Package - Selects a configured package and displays its settings in the Package Settings window.
- Configure - Opens the Configuration dialog for the selected package. See Chapter 3 for more information.
- Destination Options - Toggles the Destination Field Options toolbar.
- Add - Adds a field row to the bottom of the active package's table.
- Remove - Deletes the selected row from the active package's table.
- Auto Fill - Populates the Package Settings window with the user-designated fields for the selected package.
- Destination Field Options - Displays the Destination Field Options for the selected field. See page 3-21 for more information.

Options Ribbon

The Options Ribbon displays the Options toolbar to configure OpenDX settings; when selected, the setting is applied to all configured packages. Additional settings appear in the data window below the toolbar.



The Options toolbar contains four (4) selections:

- Audits Allowed - If checked, OpenDX will log all data entries to the DNA_Audits table. If unchecked, OpenDX will not log audits. Personnel information audits are logged per field if a change is detected.
- Filter Card Audits - If selected, changes made to cards by OpenDX will not be logged in the System Audit Trail report in DNA Fusion.
- Log to Database - If checked, OpenDX will log record changes to the DNAImport_Change table in the DNA database. By default, this table is not included in the database and requires the user to execute a table creation script. Contact Open Options Technical Support for more information on creating the database table.

Open Options does not recommend the Log to Database *setting for DNA Fusion installations that use MSDE due to the size limitation of the database.*

- Verbose Logging - Do not check this option unless actively troubleshooting with an Open Options dealer or technical support representative. Verbose Logging is separate from the debug logging for each package.

In addition to the Options toolbar, the following fields populate in the data window:

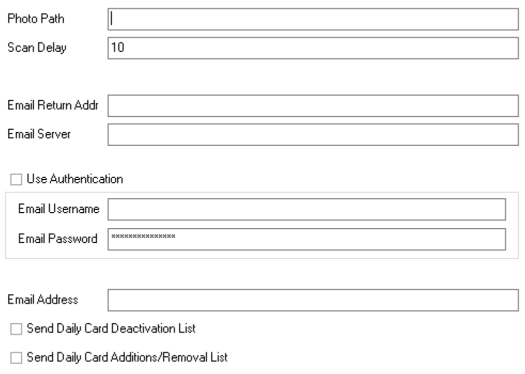


Photo Path	
Scan Delay	10
Email Return Addr	
Email Server	
☐ Use Authentication	
Email Username	
Email Password	***************
Email Address	
☐ Send Daily Card Deactivation List	
☐ Send Daily Card Additions/Removal List	

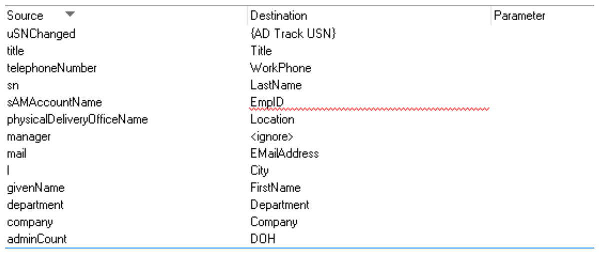
- Photo Path - Sets the import path for photos; this setting is ignored if a Photo path is configured for DNA Station Number 1.
- Scan Delay - Sets the number of seconds between complete package loops. (Default = 10 seconds)
- Email Return Addr - Designates a return e-mail address for recipients who click the Reply button.
- Email Server - Determines which e-mail server is used.
- Use Authentication - If checked, uses NT Authentication. Enter an Email Username and Password in the fields below.
- Email Address - Designates which e-mail address will receive the activity list e-mails.
- Send Daily Card Deactivation List - If checked, sends a daily list of deactivated cards to the specified e-mail address.
- Send Daily Card Additions/Removal List - If selected, sends a daily list of added and removed cards to the specified e-mail address.

Package Settings

The Package Settings appear in a data window below the Home Ribbon. After a Package is created, the user must configure each field.

Note that the Package Settings do not populate in the data window when the Options Ribbon is active.

For more information on these settings, see page 3-15.



Source	Destination	Parameter
uSNChanged	{AD Track USN}	
title	Title	
telephoneNumber	WorkPhone	
sn	LastName	
sAMAccountName	EmplD	
physicalDeliveryOfficeName	Location	
manager	<ignore>	
mail	EMailAddress	
l	City	
givenName	FirstName	
department	Department	
company	Company	
adminCount	DOH	

OpenDX Components

The OpenDX program is comprised of three (3) separate components: a configuration tool, a service application, and the DNA COM objects.

- ImportConfig.exe — The configuration tool used to create and edit packages.
- DNAImportSvc.exe — The service application that imports the data. The service loads and processes each package in order of the ID number.
- DNA COM Objects — OpenDX requires the DNAFusion COM objects to be installed on the same DNA server or client workstation. Open Options recommends installing OpenDX on the server.

This Page Intentionally Left Blank

Package Setup 3

In This Chapter

Data Sources

OpenDX updates personnel/cardholder information by communicating with a valid data source or by scanning a specific directory for text (.txt or .csv) or XML (.xml) files generated by third-party systems. OpenDX imports data from multiple sources and can scan the same source (database or Active Directory) multiple times with different query statements.

OpenDX can scan the following data sources:

- Windows® Active Directory
- Database
- Text File
- XML File

Packages

Packages, which are created and edited with the ImportConfig.exe utility, store the information used to connect to each data source.

A package identifies a single data source and contains both the connection and field mapping information. OpenDX will first scan the data source identified in each package for records to import, then move to the next package. After scanning the last package, OpenDX will pause for a predetermined amount of time (Scan Delay setting on page 2-4) before restarting with the first package.

Creating Packages

1. **Double-click** the ImportConfig.exe file.

   The OpenDX Config dialog opens.

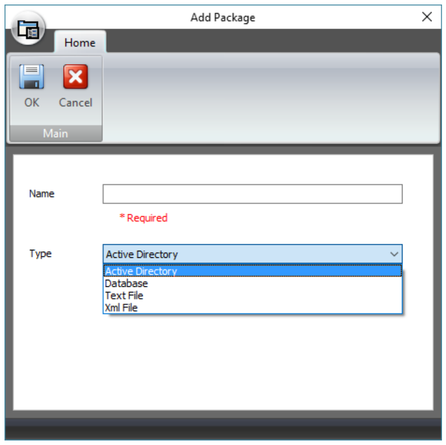
2. From the Main Menu, **select** Add Package. Add Package

   The Add Package dialog opens.

3. **Enter** a Name (required) and **select** the Type of data source from the drop-down.

   The configuration options change based on the selected data source.

   - Active Directory - See page 3-5 for configuration information.
   - Database - See page 3-9 for configuration information.
   - Text File - See page 3-11 for configuration information.
   - XML File - See page 3-13 for configuration information.



Name

* Required

Type: Active Directory

Active Directory
Database
Text File
Xml File

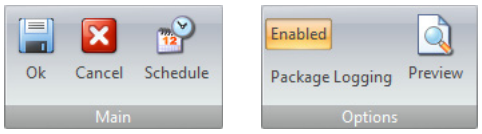
Text and XML file packages require the user to specify a unique target folder; packages can not scan the same target folder.

4. **Click** OK to save the package.

   The Configuration dialog appears.

   The Main and Options toolbars appear in the Home tab for each type of data source.

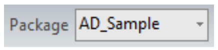
> The Configuration dialog also contains toolbar options specific to the data source. See the manual section that corresponds with the data source for more information.



- OK - Saves the Package Configuration Options.
- Cancel - Cancels the new package or disregards changes to existing packages.
- Schedule - Opens the Package Scheduling dialog to schedule when a package will run. See page 3-3 for more information.
- Enabled - If selected, allows OpenDX to scan the package (default). Deselect to use package for periodic imports.
- Package Logging - If selected, permits logging that is specific to the selected package. See page 2-4 for more information on logging.
- Preview Button - Previews the data source for the selected package. The package configuration determines the preview results.

Editing a Package

To edit an existing package:

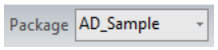
1. **Select** the Package from the drop-down. 

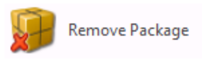
2. **Click** the Configure button to the right of the Package Name.

   The Configuration dialog opens.
3. **Edit** the package as needed.
4. **Click** OK to save.

Deleting a Package

To delete a package:

1. **Select** the Package from drop-down. 

2. From the Main Menu, **click** the Remove Package button. 

   The package is deleted and the package ID numbers are re-ordered; however, the package names remain the same.

> If a package error occurs, OpenDX will temporarily disable the package.

Scheduling and Email Notifications

OpenDX allows the user to schedule package runs and configure e-mail notifications for executed packages.

Scheduling

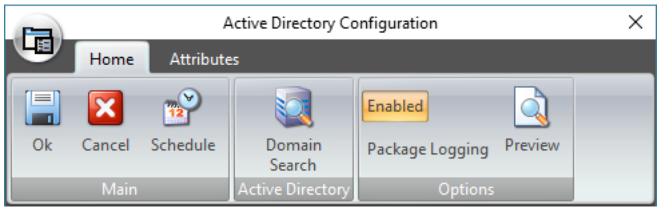
Each OpenDX package can be scheduled to run at a specific date and time or set to follow a schedule that runs on a daily, weekly, or monthly basis.

When scheduling file packages (such as .csv or .xml), it is important to set the scheduled time after the file is expected to arrive. If the package is executed according to a schedule and the file is not found, the package run is considered complete and the next run time for the package is auto-calculated based on the schedule.

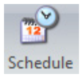
To schedule a package:

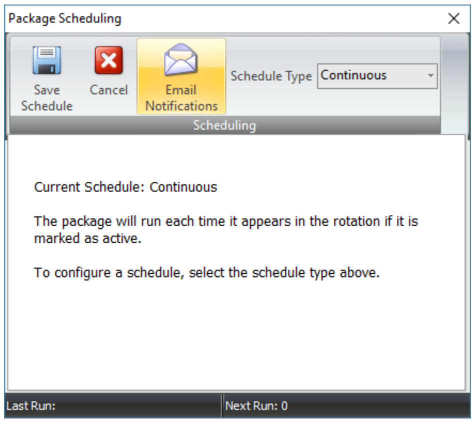
1. **Select** the Package from the drop-down list and **click** Configure.

   The Configuration dialog opens.



2. **Click** Schedule to open the Package Scheduling dialog.





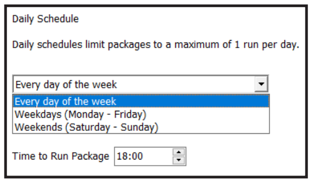
Current Schedule: Continuous

The package will run each time it appears in the rotation if it is marked as active.

To configure a schedule, select the schedule type above.

Last Run:
Next Run: 0

3. **Select** the Schedule Type from the drop-down:
   - Continuous - Default setting; if the package is active, it will run each time it appears in the rotation.
   - Daily - Limits packages to a maximum of one (1) run per day. **Select** the run cycle: Every Day of the Week, Weekdays (Monday-Friday), or Weekends (Saturday-Sunday); **enter** a package run time.
   - Weekly - Limits packages to a maximum of one (1) run per week. **Select** the week number and weekday from the drop-down lists and **enter** a package run time.
   - Monthly - Limits packages to a maximum of one (1) run per month. **Select** the run cycle: Day of the Month or Day of the Week; **select** the day and **enter** a package run time.
   - One Time - Limits packages to a maximum of one (1) run. **Select** the Scheduled Date and **enter** a package run time.

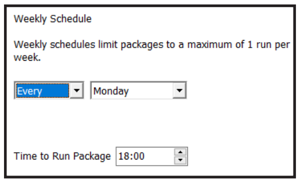


Daily Schedule

Daily schedules limit packages to a maximum of 1 run per day.

Every day of the week
Every day of the week
Weekdays (Monday - Friday)
Weekends (Saturday - Sunday)

Time to Run Package 18:00

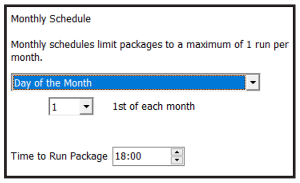


Weekly Schedule

Weekly schedules limit packages to a maximum of 1 run per week.

Every
Monday

Time to Run Package 18:00

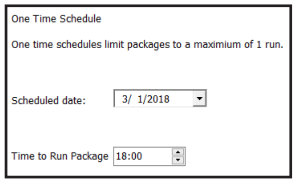


Monthly Schedule

Monthly schedules limit packages to a maximum of 1 run per month.

Day of the Month
1
1st of each month

Time to Run Package 18:00



One Time Schedule

One time schedules limit packages to a maximium of 1 run.

Scheduled date: 3/ 1/2018

Time to Run Package 18:00

4. **Click** Save Schedule to save the schedule and exit the dialog.

Email Notifications

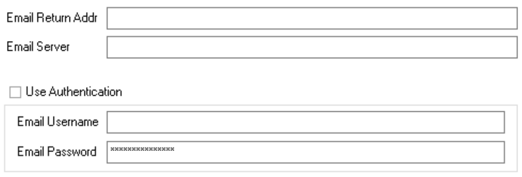
E-mail notifications are activated if the e-mail address is configured in the package schedule. If configured, the e-mail will be sent when a package is executed and the schedule for that package is not set to Continuous.

If a package is temporarily disabled, an e-mail notification is sent to the same address as the one configured for schedule notifications. This feature applies to all packages, including those running under a Continuous schedule.

Parameter Setup

The Email Return Address and Email Server fields must be configured in the Options tab in order for e-mail notifications to function successfully. See page 2-4 for more information.

If required by the e-mail server, the user may also need to check the Use Authentication box and enter a valid Username and Password.



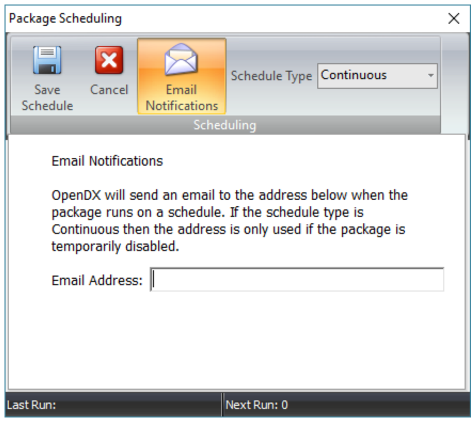
Email Return Addr
Email Server
☐ Use Authentication
Email Username
Email Password ***************

The Email Address box located under the authentication fields is used by the daily e-mail report for changes; it is not required for package e-mail notifications.

Notification Setup

1. From the Package Scheduling dialog, **click** the Email Notifications button.

   The Email Notifications screen appears in the dialog.



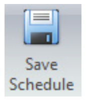
Schedule Type: Continuous

Email Notifications

OpenDX will send an email to the address below when the package runs on a schedule. If the schedule type is Continuous then the address is only used if the package is temporarily disabled.

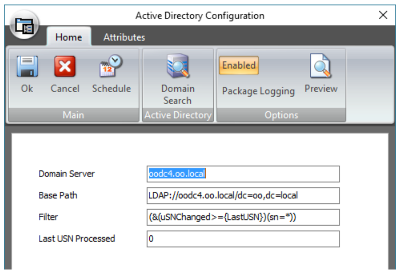
Email Address:

Last Run:
Next Run: 0

2. **Enter** an e-mail address and **click** the Save Schedule button. 

Windows Active Directory Package

Multiple packages can be used to connect to a single Active Directory (AD) forest to process different organizational units (OUs) with field mappings or scripts specific to that unit.



Domain Server: oodc4.oo.local
Base Path: LDAP://oodc4.oo.local/dc=oo,dc=local
Filter: (&(uSNChanged>={LastUSN})(sn=*))
Last USN Processed: 0

1. From the Main Menu, **select** the Add Package button. Add Package

   The Add Package dialog opens.

2. **Enter** a Name and **select** Active Directory from the Type drop-down list.

3. **Click** OK to save the package and continue the setup.

   The Active Directory Configuration dialog appears.

4. **Click** the Domain Search button.

   The name and path of the domain controller populate in the Domain Server and Base Path fields.

> The domain controller returned by the Domain Search button may change after the user logs in or out of Windows.

5. If desired, **edit** the Base Path.

   The Base Path is used to identify the name of the domain controller that will be used to import data. Because the USN values are unique to each server, it is important to identify a specific server.

   The Base Path is also used when filtering by organizational units. Enter the OU name(s) after the controller path to add the filter.

   - Example: LDAP://<Server.Domain>/ou=sales,dc=<Domain>,dc=local

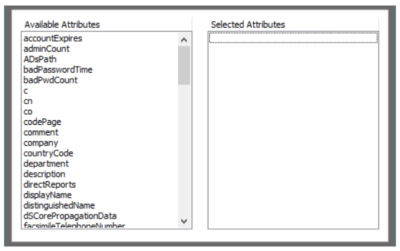
6. If desired, **edit** the Filter.

   Keep in mind the following notes about filters:

   - The filter will be invalid if it contains spaces.
   - Valid comparison signs are =, >=, <=, and =*.
   - Substring filtering, such as (sn=j*), is accepted; however, its performance may be affected if the wildcard character is not placed at the end. For example, (sn=*son) may cause slow performance since indexes can not be used.
   - When searching for boolean values, TRUE and FALSE must be capitalized.

7. **Enter** 0 in the Last USN Processed field.

> The USN is only applicable if the AD Filter contains the following statement: uSNChanged>={LastUSN}. After OpenDX processes an AD package, it saves the highest USN encountered. If the the Last USN Processed is set to 0, the package will reevaluate all the records on the next pass.



Available Attributes
accountExpires
adminCount
ADsPath
badPasswordTime
badPwdCount
c
cn
co
codePage
comment
company
countryCode
department
description
directReports
displayName
distinguishedName
dSCorePropagationData
facsimileTelephoneNumber

Selected Attributes

8. **Select** the Attributes tab.

   A list of available query fields populates the Available Attributes section.

9. **Select** the desired field(s) from the Available Attributes section and **click** the Add button.

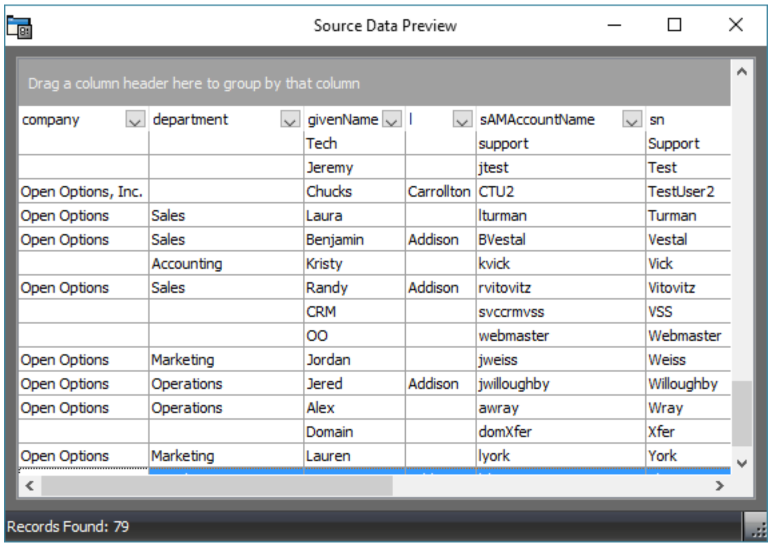
   Verify the field use with the domain administrator; the fields are added to the Selected Attributes section.

   Useful Attributes:

- sAMAccount - Unique identifier
- userAccountControl - Deactivates the user's card(s) when their AD account is disabled.
- uSNChanged - Tracks changes to the Update Serial Number (USN). See Scan Active Directory for Changes Only instructions on page 3-7.
- l - City field
- sn - Last Name field
- givenName - First Name field

10. **Select** the Home tab.
11. If desired, **click** the Preview button to preview the data source.

    The Source Data Preview dialog opens. OpenDX applies the filter(s) and displays the records for those that meet the criteria.



Source Data Preview

Drag a column header here to group by that column

company	department	givenName	l	sAMAccountName	sn
		Tech		support	Support
		Jeremy		jtest	Test
Open Options, Inc.		Chucks	Carrollton	CTU2	TestUser2
Open Options	Sales	Laura		lturman	Turman
Open Options	Sales	Benjamin	Addison	BVestal	Vestal
	Accounting	Kristy		kvick	Vick
Open Options	Sales	Randy	Addison	rvitovitz	Vitovitz
		CRM		svccrmvss	VSS
		OO		webmaster	Webmaster
Open Options	Marketing	Jordan		jweiss	Weiss
Open Options	Operations	Jered	Addison	jwilloughby	Willoughby
Open Options	Operations	Alex		awray	Wray
		Domain		domXfer	Xfer
Open Options	Marketing	Lauren		lyork	York

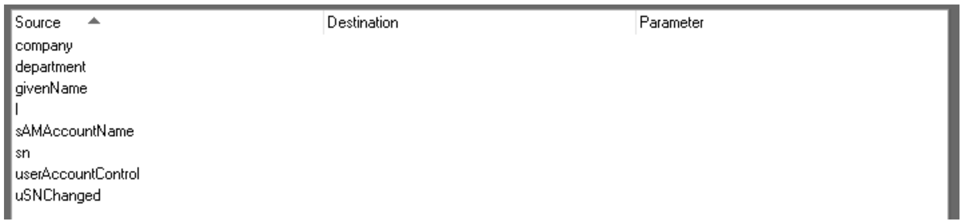
Records Found: 79

Drag and **drop** a column header to the Group By box at the top of the dialog to group the field results by the selected column.

12. **Click** OK to add the package.
13. **Click** the Auto Fill button to populate the Package Settings window with the fields.

    OR

    Click the Add button to add individual field rows and select the Field(s) from the drop-down.



Source	Destination	Parameter
company		
department		
givenName		
l		
sAMAccountName		
sn		
userAccountControl		
uSNChanged		

14. **Configure** the Package Fields.

    See page 3-15 for more information.
15. **Click** the Save button to save the package.

Permissions within Active Directory may prevent certain fields from importing into DNA Fusion. If some fields remain empty after an import even though data is in the field, contact the system administrator.

Scan Active Directory for Changes Only

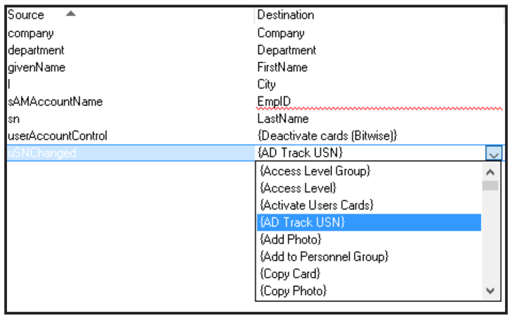
This feature adds support for Update Serial Numbers (USNs) in Active Directory by allowing OpenDX to scan for updates once it is fully synchronized with Active Directory. The maximum USN is stored in the .ini file in the event that the service is restarted; a full scan is not required.

The AD filter includes the uSN statement; follow the steps below to set up the feature:

1. In the Attributes tab of the Active Directory Configuration dialog, **add** the uSNChanged field to the Selected Attributes section.
2. **Select** the Home tab and **click** OK to save the dialog.
3. **Select** the Package from the drop-down.

   The attributes selected in Step 1 populate in the data window.
4. In the Destination field for the uSNChanged source, select AD Track USN from the drop-down list.
5. **Configure** the remaining Package Fields.

   See page 3-15 for more information.
6. **Click** the Save button.



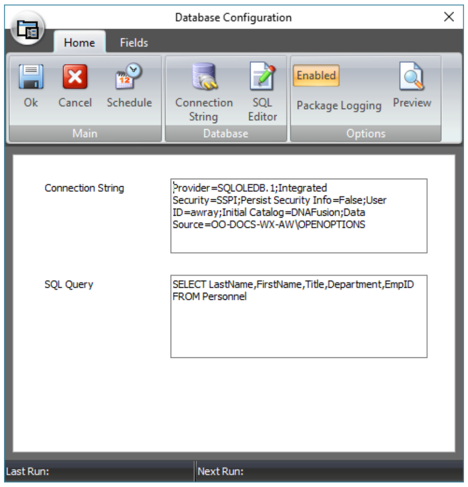
Source	Destination
company	Company
department	Department
givenName	FirstName
l	City
sAMAccountName	EmpID
sn	LastName
userAccountControl	{Deactivate cards (Bitwise)}
uSNChanged	{AD Track USN}

{Access Level Group}
{Access Level}
{Activate Users Cards}
{AD Track USN}
{Add Photo}
{Add to Personnel Group}
{Copy Card}
{Copy Photo}

NOTES:

Database Package

Database packages use ActiveX Data Objects (ADOs) to connect to databases. Use a Connection String and SQL Query to make a connection with the ADO(s).



Connection String: Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;User ID=awray;Initial Catalog=DNAFusion;Data Source=OO-DOCS-WX-AW\OPENOPTIONS

SQL Query: SELECT LastName,FirstName,Title,Department,EmpID FROM Personnel

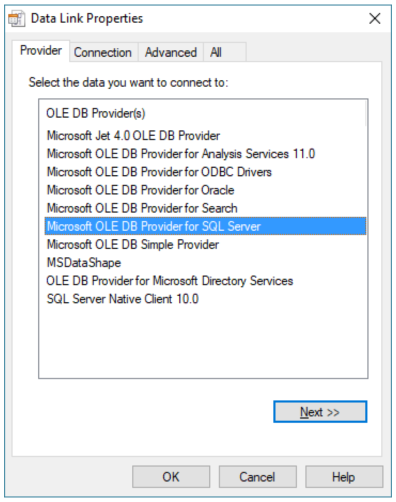
1. From the Main Menu, **select** the Add Package button. Add Package

   The Add Package dialog opens.
2. **Enter** a Name and **select** Database from the Type drop-down list.
3. **Click** OK to save the package and continue the setup.

   The Database Configuration dialog appears.
4. **Click** the Connection String button.

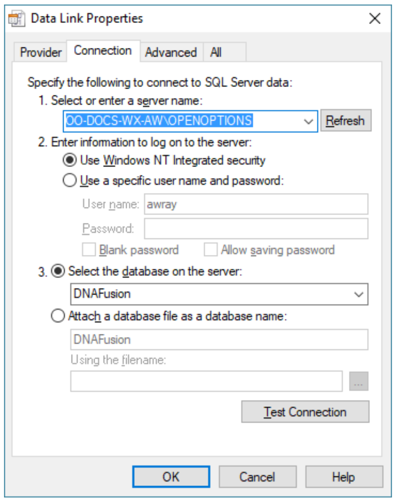
   The Data Link Properties dialog opens. This dialog helps build the connection string required to connect to the database. The Connection String typically identifies the server, database, and possibly a username and password for authentication.
5. **Select** the Database Provider from the list and **click** Next.

   The list includes all the ADO providers installed on the computer. The Connection dialog will vary depending on the selection, and the information will be used to build the Connection String.



Select the data you want to connect to:

OLE DB Provider(s)
Microsoft Jet 4.0 OLE DB Provider
Microsoft OLE DB Provider for Analysis Services 11.0
Microsoft OLE DB Provider for ODBC Drivers
Microsoft OLE DB Provider for Oracle
Microsoft OLE DB Provider for Search
Microsoft OLE DB Provider for SQL Server
Microsoft OLE DB Simple Provider
MSDataShape
OLE DB Provider for Microsoft Directory Services
SQL Server Native Client 10.0



Specify the following to connect to SQL Server data:
1. Select or enter a server name: OO-DOCS-WX-AW\OPENOPTIONS
2. Enter information to log on to the server:
   ● Use Windows NT Integrated security
   ○ Use a specific user name and password:
   User name: awray
   Password:
   ☐ Blank password ☐ Allow saving password
3. ● Select the database on the server: DNAFusion
   ○ Attach a database file as a database name: DNAFusion
   Using the filename:

6. **Enter** or **select** a Server Name.
7. If desired, **select** the Refresh button to renew the drop-down list. Depending on the selected Provider, the user may need to enter a Data Source instead of a server.

> If the server is using SQL Server Express or a named instance of SQL, the server field may need to include the name after the server, e.g. OO-TRAINER-XP\sqlexpress.

8. **Select** the desired Log On method.

   If Use a Specific User Name and Password is selected, **enter** a User Name and Password in the correct fields or **check** the Blank Password box.

   These fields are not used for databases that support Windows Authentication; login information is not required.
9. **Select** the Database on the server from the drop-down list.

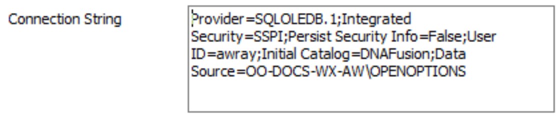
   OR

   Select the Attach a Database File as a Database Name radio button, **enter** a Name, and **click** the Browse button to locate the .mdf file.
10. **Click** the Test Connection button to verify database communication.

    If successful, **click** OK to close the dialog; if unsuccessful, **exit** the dialog and **verify** the settings.

11. **Click** OK to close the Data Link Properties dialog.

    The Connection String populates in the Database Configuration dialog.



Connection String

```
Provider=SQLOLEDB.1;Integrated
Security=SSPI;Persist Security Info=False;User
ID=awray;Initial Catalog=DNAFusion;Data
Source=OO-DOCS-WX-AW\OPENOPTIONS
```

12. **Enter** the query into the SQL Query field.

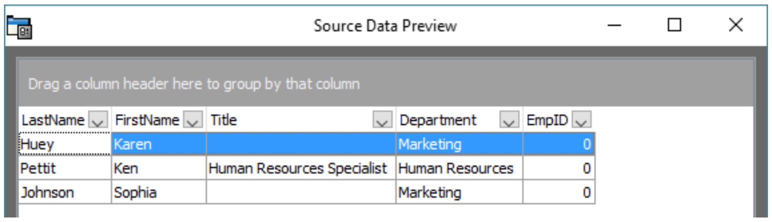
    If desired, **click** the SQL Editor button to open a SQL Query window.

    The SQL Query identifies which table(s), fields, and records are returned.

    Example: SELECT LastName,FirstName,Title,Department,EmpID FROM Personnel

13. If desired, **click** the Preview button to preview the data source.

    The Source Data Preview dialog opens; only the records and fields indentified in the query are displayed.



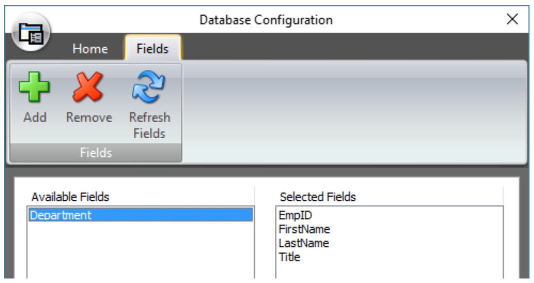
LastName	FirstName	Title	Department	EmpID
Huey	Karen		Marketing	0
Pettit	Ken	Human Resources Specialist	Human Resources	0
Johnson	Sophia		Marketing	0

14. **Select** the Fields tab and **click** the Refresh Fields button.

    The queried fields populate the Available Fields list.

15. **Select** the desired field(s) from the Available Fields list and **click** the Add button.

    The designated fields are added to the Selected Fields list.



Available Fields: Department

Selected Fields: EmpID, FirstName, LastName, Title

16. **Select** the Home tab and **click** OK to add the package.

17. **Click** the AutoFill button to populate the fields in the Package Settings window.

    OR

    Click the Add button to add individual field rows and **select** the Field(s) from the drop-down.

18. **Configure** the Package Fields; see page 3-15 for more information.

19. **Click** the Save button.

Sample Oracle Connection Strings

Microsoft OLEDB

ADOConn=Provider=SQLOLEDB.1;IntegratedSecurity=SSPI;PersistSecurityInfo=False;InitialCatalog=<database>;DataSource=<server\instance>

Oracle OLEDB

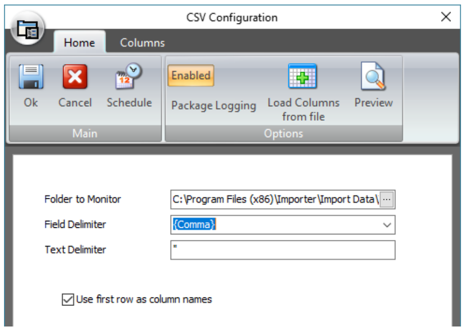
ADOConn=Provider=OraOLEDB.Oracle;Server=<server>;DataSource=<dbname>;UserID=<uid>;Password=<pwd>

Text File Package

Each text package must reference a target folder that has limited access to avoid processing any invalid data. If there will be multiple text packages, Open Options recommends creating a main folder with several subfolders that contain the information for each package. Text files require a .txt or .csv extension.

1. From the Main Menu, **select** the Add Package button. Add Package

   The Add Package dialog opens.



Folder to Monitor: C:\Program Files (x86)\Importer\Import Data\
Field Delimiter: {Comma}
Text Delimiter: "
☑ Use first row as column names

2. **Enter** a Name and **select** Text File from the Type drop-down list.
3. **Click** OK to save the package and continue the setup.

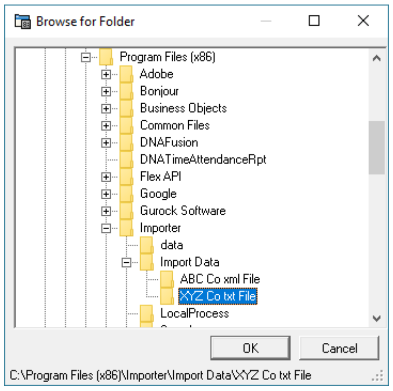
   The CSV Configuration dialog opens.

4. **Click** the Browse button in the Folder to Monitor field to locate the folder that will be scanned for the files.

   The Browse for Folder dialog opens.

5. **Select** the desired folder and **click** OK to save the path.

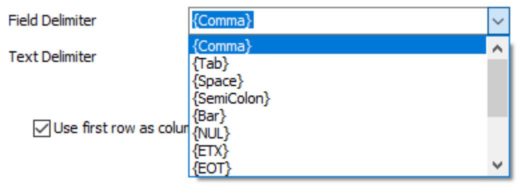
   The scan path populates in the field; verify that the path contains the folder name.



C:\Program Files (x86)\Importer\Import Data\XYZ Co txt File

6. If needed, **select** a character from the Field Delimiter drop-down.

   The Field Delimiter identifies the character that separates each field in the text file.



{Comma}
{Tab}
{Space}
{SemiColon}
{Bar}
{NUL}
{ETX}
{EOT}

7. If needed, **enter** the Text Delimiter.

   The text delimiter identifies the character that signals the start and end of quoted text. The scan will ignore field delimiters located inside quoted text.

8. If needed, **select** the Use First Row as Column Names checkbox.

   If checked, OpenDX assumes that the first row of the text file contains the column names. This row is only used to identify the columns and will not be imported as data.

   If unchecked, OpenDX will assign the column names as F1 through Fx (where x = max. column number).

9. **Select** the Columns tab and **click** the Load Columns from File button.

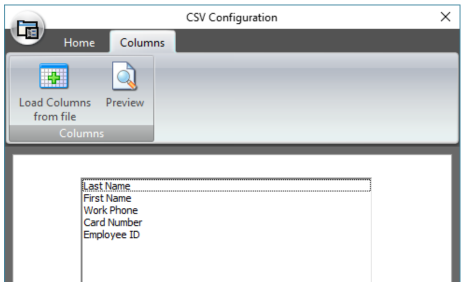
   The Sample File Needed dialog appears.

10. **Click** Yes to locate the sample file or **No** to exit the dialog.

    The sample file is used to verify the columns.

11. **Browse** to the desired text file (.txt or .csv) and **click** Open.

    The field names populate in the dialog.



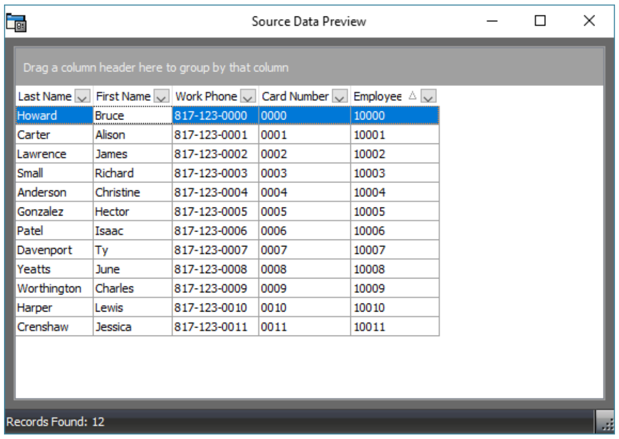
Last Name
First Name
Work Phone
Card Number
Employee ID

12. If desired, **click** the Preview button.

    The Sample File Needed dialog appears; **click** Yes to locate the file.

13. **Browse** to the desired text flie (.txt or .csv) and **click** Open.

    The Source Data Preview dialog opens with the returned records.



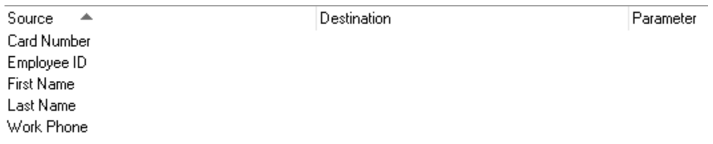
Last Name	First Name	Work Phone	Card Number	Employee
Howard	Bruce	817-123-0000	0000	10000
Carter	Alison	817-123-0001	0001	10001
Lawrence	James	817-123-0002	0002	10002
Small	Richard	817-123-0003	0003	10003
Anderson	Christine	817-123-0004	0004	10004
Gonzalez	Hector	817-123-0005	0005	10005
Patel	Isaac	817-123-0006	0006	10006
Davenport	Ty	817-123-0007	0007	10007
Yeatts	June	817-123-0008	0008	10008
Worthington	Charles	817-123-0009	0009	10009
Harper	Lewis	817-123-0010	0010	10010
Crenshaw	Jessica	817-123-0011	0011	10011

Records Found: 12

14. **Verify** the data and **close** the Source Data Preview dialog to continue.
15. **Select** the Home tab and **click** OK to add the package.
16. **Click** the Auto Fill button to populate the Package Settings window with the fields.

    OR

    Click the Add button to add individual field rows and **select** the field(s) from the drop-down.



Source	Destination	Parameter
Card Number		
Employee ID		
First Name		
Last Name		
Work Phone		

17. **Configure** the Package Fields.

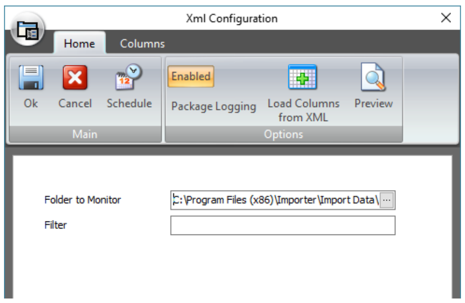
    See page 3-15 for more information.
18. **Click** the Save button.

XML File Package

Each XML package must reference a target folder that has limited access to avoid processing any invalid data. If there will be multiple XML packages, Open Options recommends creating a main folder with several subfolders that contain the information for each package. XML files require a .xml extension.

1. From the Main Menu, **select** the Add Package button. Add Package

   The Add Package dialog opens.



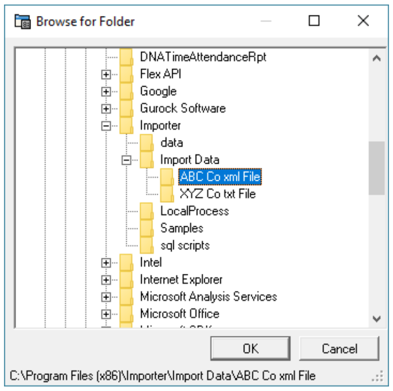
Folder to Monitor: C:\Program Files (x86)\Importer\Import Data\

Filter:

2. **Enter** a Name and **select** XML File from the Type drop-down list.
3. **Click** OK to save the package.

   The XML Configuration dialog opens.
4. **Click** the Browse button in the Folder to Monitor field to locate the folder that will be scanned for the files.

   The Browse for Folder dialog opens.



C:\Program Files (x86)\Importer\Import Data\ABC Co xml File

5. **Select** the desired folder and **click** OK to save the path.

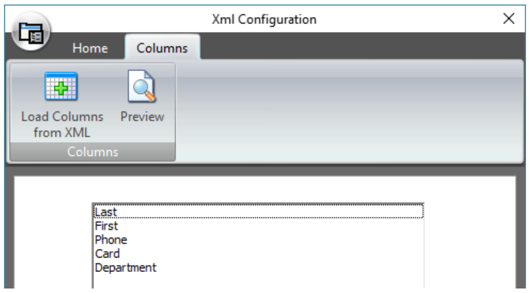
   The scan path populates in the field; verify that the path contains the folder name.
6. If desired, **enter** information in the Filter field.

   If the XML document contains additional first child objects, enter the name of the objects in the Filter field to filter them out of the import.
7. **Select** the Columns tab and **click** the Load Columns from XML button.

   The Sample File Needed dialog appears.
8. **Click** Yes to locate the sample file or **No** to exit the dialog.

   The sample file is used to verify the fields.
9. **Browse** to the desired XML file (.xml) and **click** Open.

   The field names populate in the dialog.

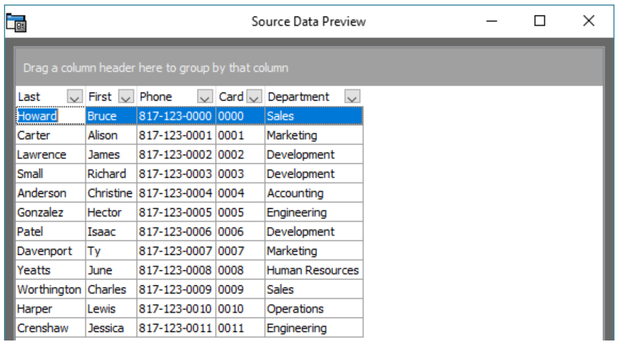


Last
First
Phone
Card
Department

10. If desired, **click** the Preview button.

    The Sample File Needed dialog appears; **click** Yes to locate the file.
11. **Browse** to the desired XML (.xml) file and **click** Open.

    The Source Data Preview dialog opens with the returned records.



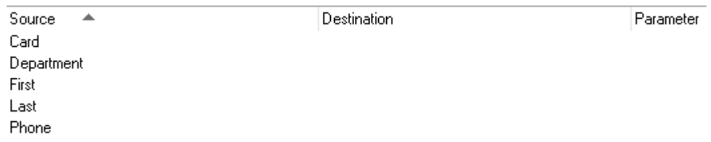
Last	First	Phone	Card	Department
Howard	Bruce	817-123-0000	0000	Sales
Carter	Alison	817-123-0001	0001	Marketing
Lawrence	James	817-123-0002	0002	Development
Small	Richard	817-123-0003	0003	Development
Anderson	Christine	817-123-0004	0004	Accounting
Gonzalez	Hector	817-123-0005	0005	Engineering
Patel	Isaac	817-123-0006	0006	Development
Davenport	Ty	817-123-0007	0007	Marketing
Yeatts	June	817-123-0008	0008	Human Resources
Worthington	Charles	817-123-0009	0009	Sales
Harper	Lewis	817-123-0010	0010	Operations
Crenshaw	Jessica	817-123-0011	0011	Engineering

12. **Verify** the data and **close** the Source Data Preview dialog to continue.

13. **Select** the Home tab and **click** OK to add the package.
14. **Click** the Auto Fill button to populate fields in the Package Settings window.

    OR

    Click the Add button to add field rows and select the field(s) from the drop-down.



Source	Destination	Parameter
Card		
Department		
First		
Last		
Phone		

15. **Configure** the Package Fields.

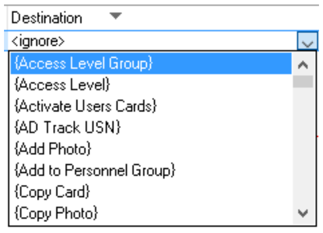
    See page 3-15 for more information.
16. **Click** the Save button.

XML Structure

It is important to remain consistent throughout the selected XML schema, which describes the structure of an XML document. The first line is the XML declaration; it defines the XML version (1.0) and the encoding used. The next line describes the root element of the document. An XML schema can only contain one (1) root element. The next lines contain both child and subchild elements for the root. The last line defines the end of the root element. All elements must contain a closing tag (/).

Package Configuration

After the package has been created, the fields must be configured in the Package Settings window. The user is required to select a Unique Identifier field for each package.



Destination

<ignore>

{Access Level Group}
{Access Level}
{Activate Users Cards}
{AD Track USN}
{Add Photo}
{Add to Personnel Group}
{Copy Card}
{Copy Photo}

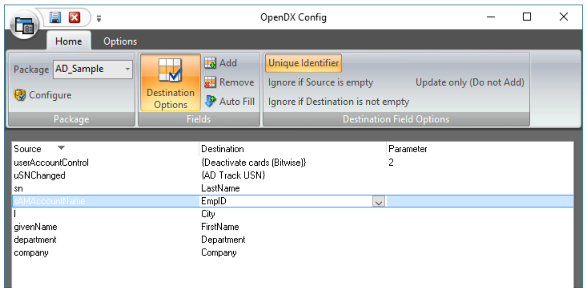
1. **Select** the Destination field and **click** the arrow to view the drop-down list.

   The list contains DNA Fusion fields that will populate in the Personnel Record as well as commands that execute a specific task. Braces ({}) denote commands.
2. **Select** a command ({}) or a DNA Fusion field.

   The table below describes the available selections and what Parameter, if any, is required.
3. If required, **enter** a value in the Parameter column.
4. **Identify** a field to serve as the unique identifier and **click** the Unique Identifier option in the Destination Field Options for the selected row.
5. If desired, **configure** the Destination Field Options for the remaining Destination fields.

   See page 3-21 for more information.
6. **Click** the Save button to save the current package.

   The package settings are stored in the ImportSettings.ini file. After the data has been imported and processed, the import file will be deleted (unless the Old Data folder was created; see page 4-1).



Source	Destination	Parameter
userAccountControl	{Deactivate cards (Bitwise)}	2
uSNChanged	{AD Track USN}	
sn	LastName	
sAMAccountName	EmpID	
l	City	
givenName	FirstName	
department	Department	
company	Company	

Destination Fields and Commands

Field/Command	Description	Parameter?	Length
{Access Level Group}	Adds an access level group (identified in the Source field by name or group number) to the card identified by the {Find Card} command. Can be used when a cardholder has multiple cards but the access level group only needs to be assigned to one card. If the {Find Card} command is not used, all of the user's cards will be affected. **Note:** the access level group will not be added if (a) the source data does not match an existing access level group or (b) the number in the {Find Card} command does not match a card for the cardholder.	No	
{Access Level}	Assigns the indicated access level (by name) to all of a cardholder's cards unless the {Find Card} command is used. "%" is a wildcard character.	No	
{Activate Users Cards}	Activates the user's card identified by the {Find Card} command if the data value from the Source field matches the value in the Parameters column. Can be used when only one of the cardholder's multiple cards needs to be activated. If the {Find Card} command is not used, all of the user's cards will be activated. If an invalid card number is passed, none of the cards will be activated.	Yes	
{AD Track USN}	Allows OpenDX to query Active Directory for changes instead of repeatedly processing all Active Directory records. The package filter should contain '(uSNChanged>={LastUSN})'.	No	
{Add Photo}	Adds a JPEG (.jpg) photo to the cardholder and links the photo by the file path. In the photo properties on the personnel record, the photo will be marked as Set Default and Displayed.	No	

Field/Command	Description	Parameter?	Length
{Add to Personnel Group}	Adds the person to a personnel group (identified in the Source field by name or group number). If no match is found, a new personnel group is created.	No	
{Copy Card}	Copies the card information to a new card. The original card will be identified by the source data associated with the {Copy Card} command. The target card (the one receiving the changes) is identified in the source data field associated with the {Find Card} command. The command will copy access levels and activation/deactivation dates; however, existing access levels associated with the target card will not be removed.	No	
{Copy Photo}	Copies a photo to the photo path and assigns it to the cardholder. The original photo path will be replaced with the path specified in the Options setting. In the photo properties on the personnel record, the photo will be marked as Set Default and Displayed. The Set Default flag will be cleared.	Possibly	
{Corp Card}	Used to add or update a card when using Corporate Mode. Requires {Corp FC} destination. Not compatible with {Find Card} or Keycard destinations.	No	
{Corp FC}	Identifies the facility code for a card when using Corporate Mode. Ignored unless {Corp Card} is used as a destination.	No	
{Deactivate Cards (Bitwise)}	Deactivates all the user's cards if the value from the Source field compared to the value in the Parameter column is greater than zero. Use with userAccountControl in Active Directory and set the Parameter to 2. This will deactivate a card when the user's account is disabled in Active Directory.	Yes	
{Activate Newest Card (Bitwise)}	Activates the most recent card for the user (based on the Start Date) if the value from the Source field compared to the value in the Parameter column is greater than zero. All other cards will be disabled.	Yes	
{Deactivate Users Cards}	Deactivates all the user's cards if the data value from the Source field matches the value in the Parameter column.	Yes	
{Delete Cards}	Removes all cards assigned to the cardholder if the data from the Source field matches the value in the Parameter column. If {Find Card} is used in the same package, only the identified card will be deleted.	Yes	
{Delete Specified Card}	Removes the card specified by the Source data.	No	
{Delete Person}	Deletes the cardholder and removes any cards assigned to the cardholder if the data from the Source field matches the value in the Parameter column.	Yes	
{Dismiss Record}	Does not process the current record if the data from the Source field matches the value in the Parameter column.	Yes	
{Exclusive Activate}	Deactivates all cards for the designated cardholder except for the card identified in the {Find Card} field. If {Find Card} is not included, the {Exclusive Activate} command will be ignored. If the cardholder does not have the card associated with the {Find Card} command, Open DX will create a new card unless the Update Only field option is selected. If the Update Only field is selected and the designated card is not found, the existing cards will not be modified.	Yes (Unless XML package)	

Field/Command	Description	Parameter?	Length
{Find Card}	Stores the ID of the card in the Source data, enabling card-specific commands. Negative numbers will be ignored. If the card number is not found, the card will be added to the cardholder unless prevented by field options. **Note**: Only allowed once per package.	No	
{Lookup Access Level Group}	Looks up the access level group ID in a custom table stored in the "ImportSettings.ini" file and identified by the Parameter value. If found, the corresponding AL group will be added to the cardholder. The AL group lookup table must be manually added to the .ini file.	Yes	
{Lookup Tenant ID}	Looks up the tenant ID in a table stored in the "ImportSettings.ini" file and identified by the Parameter.	Yes	
{Remove Access Level}	Removes access levels (by name) if the Source field matches the value in the Parameter field. Removes the access level from all cards assigned to a cardholder unless the {Find Card} or Keycard is used. "%" is a wildcard character.	Yes	
{Remove Access Level Group}	Removes the access level(s) (identified in the Source field by name or group number) from the card identified by the {Find Card} command. Can be used when a cardholder has multiple cards but only one needs the access level group removed. If the {Find Card} command is not used, all of the user's cards will be affected. The access level group will not be removed if the source data does not match an existing access level group or if the number in the {Find Card} command does not match a card for the cardholder. **Note**: If an access level is common to more than one access level group, the access level will be removed even if it was assigned via one of the other groups.	Yes	
{Remove All Access Levels}	Removes all of the cardholder's access levels if the Source field matches the Parameter value.	Yes	
{Remove from Personnel Group}	Removes the cardholder from a Personnel Group (identified in the Source field by name or group number). If the personnel group does not exist or the cardholder is not a member of the group, this command will not take effect.	No	
{Remove Photo}	Removes a specified photo from a cardholder. The file name (excluding the path) identifies the photo. **Note:** The file is not removed from the disk.	No	
{Set Activation Date for Users Cards}	Sets the activation date for the card identified by the {Find Card} command. Can be used when only one of the cardholder's multiple cards needs to be activated. If the {Find Card} command is not used, an activation date will bet set for all cards assigned to the user. Uses the short date/time format (MM/DD/YY HH:MM:SS). If AM or PM is not specified, uses 24-hr time. If an invalid card number is passed, none of the cards will be updated.	No	
{Set Deactivation Date for Users Cards}	Deactivates the card identified by the {Find Card} command on the date indicated in the Source field. Can be used when only one of the cardholder's multiple cards needs to be deactivated. If the {Find Card} command is not used, all cards assigned to the user are deactivated. Uses short date/time format (MM/DD/YY HH:MM:SS). If AM or PM is not specified, uses 24-hr time. If an invalid card number is passed, none of the cards will be updated.	No	

Field/Command	Description	Parameter?	Length
{Set Card Type}	Populates the Card Type field. Requires the Source field to pass valid Card Type information (Normal, Visitor, Temp, Disabled, Contractor, Custom, etc.). If using the Custom type, valid values include: Custom1, Custom2, etc. Updates all cards for a given cardholder or uses {Find Card} to identify an individual card. **Note**: OpenDX logs a note if it does not find a match; no change is made.	No	
{Set Hot Stamp}	Sets the Hot Stamp field. Requires the {Find Card} command to be used in the package. If the {Find Card} command is not used or if a card number is not specified in the Source data, the {Set Hot Stamp} command will be ignored.	No	
{Set Issue Code}	Sets the Issue Code field. Requires the {Find Card} command to identify the card.	No	
{Set PIN}	Sets the PIN code value for the card identified by the {Find Card} command. The {Set PIN} command is ignored if the {Find Card} command is not specified.	No	
{Set Use Limit}	Sets the Use Limit for the cardholder's active cards. Requires the {Find Card} command to identify the card.	No	
<ignore>	Ignores the Source field.	No	
Address1	Populates the first Address field in the Employee Info (Page 2) tab of the Personnel Record.	No	100
Address2	Populates the second Address field in the Employee Info (Page 2) tab of the Personnel Record.	No	100
AssaFacCode	Populates the AssaFacCode setting for the selected keycard. Requires the {Find Card}, Keycard, or {Corp Card} command/field.	No	
AssaCredFormat	Populates the AssaCredentialFormat setting for the selected keycard. Requires the {Find Card}, Keycard, or {Corp Card} command/field.	No	
City	Populates the City field in the Employee Info (Page 2) tab of the Personnel Record.	No	100
Company	Populates the Company field in the Employee Info tab of the Personnel Record.	No	
Country	Populates the Country field in the Employee Info (Page 2) tab of the Personnel Record.	No	100
Department	Populates the Department field in the Employee Info tab of the Personnel Record.	No	100
DLNumber	Populates the Drivers License # field in the Employee Info (Page 2) tab of the Personnel Record.	No	25
DNAText1	Populates the Custom 1 field in the Employee Info (Page 2) tab of the Personnel Record.	No	255
DNAText2	Populates the Custom 2 field in the Employee Info (Page 2) tab of the Personnel Record.	No	255
DNAText3	Populates the Custom 3 field in the Employee Info (Page 2) tab of the Personnel Record.	No	255
DNAText4	Populates the Custom 4 field in the Employee Info (Page 2) tab of the Personnel Record.	No	
DNAText5	Populates the Custom 5 field in the Employee Info (Page 2) tab of the Personnel Record.	No	

Field/Command	Description	Parameter?	Length
DNAText6	Populates the Custom 6 field in the Employee Info (Page 2) tab of the Personnel Record.	No	
DNAText7	Populates the Custom 7 field in the Employee Info (Page 2) tab of the Personnel Record.	No	
DNAText8	Populates the Custom 8 field in the Employee Info (Page 2) tab of the Personnel Record.	No	
DNAText9	Populates the Custom 9 field in the Employee Info (Page 2) tab of the Personnel Record.	No	
DNAText10	Populates the Custom 10 field in the Employee Info (Page 2) tab of the Personnel Record.	No	
DNAText11	Populates the Custom 11 field in the Employee Info (Page 2) tab of the Personnel Record.	No	
DNAText12	Populates the Custom 12 field in the Employee Info (Page 2) tab of the Personnel Record.	No	
DNAText13	Populates the Custom 13 field in the Employee Info (Page 2) tab of the Personnel Record.	No	
DNAText14	Populates the Custom 14 field in the Employee Info (Page 2) tab of the Personnel Record.	No	
DNAText15	Populates the Custom 15 field in the Employee Info (Page 2) tab of the Personnel Record.	No	
DNAText16	Populates the Custom 16 field in the Employee Info (Page 2) tab of the Personnel Record.	No	
DNAVal1	Populates the Custom Value 1 field in the Employee Info (Page 2) tab of the Personnel Record. Numerical characters only. No letters or punctuation.	No	
DNAVal2	Populates the Custom Value 2 field in the Employee Info (Page 2) tab of the Personnel Record. Numerical characters only. No letters or punctuation.	No	
DNAVal3	Populates the Custom Value 3 field in the Employee Info (Page 2) tab of the Personnel Record. Numerical characters only. No letters or punctuation.	No	
DOH	Populates the Hire Date field in the Employee Info tab of the Personnel Record.	No	
EMailAddress	Populates the E-Mail field in the Employee Info tab of the Personnel Record.	No	100
EmpID	Populates the Employee ID field in the Employee Info (Page 2) tab of the Personnel Record. Alphanumeric characters allowed. Can be used as a Unique Identifier to synchronize data between the Source data and DNA Fusion database.	No	15
EmpNumber	Populates the Employee # field in the Employee Info (Page 2) tab of the Personnel Record. Numerical characters only. No letters or punctuation. Can be used as a Unique Identifier to synchronize data between the Source data and DNA Fusion database.	No	
FirstName	Populates the First Name field in the Employee Info tab of the Personnel Record.	No	100
Flags	Populates the appropriate record field.	No	
Home Phone	Populates the Home Phone field in the Employee Info (Page 2) tab of the Personnel Record.	No	20

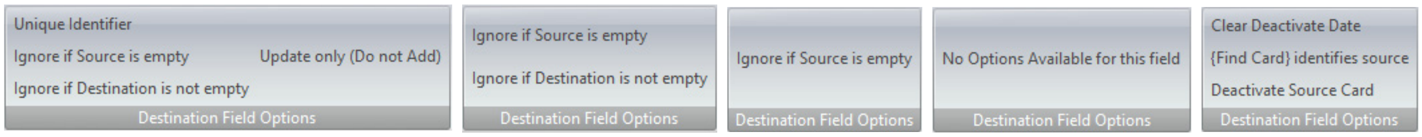
Field/Command	Description	Parameter?	Length
Keycard	Allows multiple card additions from a single record. Keycard can be used several times in the same package (unlike the {Find Card} command). If the card number is not found, the card will be added to the cardholder unless prevent by field options. Can also be used as a Unique Identifier. **Note**: Only one Keycard field can be specified as the Unique Identifier per package.	No	
LastName	Populates the Last Name field in the Employee Info tab of the Personnel Record.	No	100
Location	Populates the Location field in the Employee Info tab of the Personnel Record.	No	100
MiddleName	Populates the Middle Name field in the Employee Info tab of the Personnel Record.	No	100
Other	Populates the Other Personnel Information field on the Employee Info (Page 2) tab of the Personnel Record.	No	
PersonnelType	Populates the Type field in the Employe Info tab of the Personnel Record. Requires the Source field to pass valid Personnel Type information (Normal, Visitor, Temp, Disabled, Contractor, Custom, etc.) If using the Custom type, valid values include: Custom1, Custom2, etc. **Note**: Scripts can be used to reformat the input value to a valid input.	No	
Site	Populates the Site field in the Employee Info tab of the Personnel Record.	No	100
SSN	Populates the Employee ID field in the Employee Info (Page 2) tab of the Personnel Record.	No	15
State	Populates the State field in the Employee Info (Page 2) tab of the Personnel Record.	No	100
TenantID	Populates the Tenant field in the Employee Info tab of the Personnel Record. Only use this field if Tenants is enabled in DNA Fusion.	No	
Title	Populates the Title field in the Employee Info tab of the Personnel Record.	No	100
WorkPhone	Populates the Work Phone field in the Employee Info tab of the Personnel Record.	No	20
ZIP	Populates the Zip Code field on the Employee Info (Page 2) tab of the Personnel Record.	No	20

When OpenDX adds a card, the Deactivation Date defaults to one (1) year from the card's creation date unless the deactivation date is imported as part of the package.

Destination Field Options

The table below describes the Destination Field Options that populate in the Home Ribbon based on the Destination field selection. It is possible to select multiple Destination Field Options for the same Destination field.

A Unique Identifier must be specified for each package.


Unique Identifier
Ignore if Source is empty
Update only (Do not Add)
Ignore if Destination is not empty
Destination Field Options
Ignore if Source is empty
Ignore if Destination is not empty
Destination Field Options
Ignore if Source is empty
Destination Field Options
No Options Available for this field
Destination Field Options
Clear Deactivate Date
{Find Card} identifies source
Deactivate Source Card
Destination Field Options

Destination Field Option	Definition
Unique Identifier	Identifies the field that will be used to synchronize the records in the Source database to the DNA Fusion database. Each package requires one unique field.
Ignore if Source is Empty	Ignores the field if the Source field is empty. Prevents the data in DNA Fusion from being clear when the Source field is empty.
Ignore if Destination is Not Empty	Ignores the Source field information if data already exists in the DNA Fusion record.
Update Only (Do Not Add)	Only available when the Destination field is set to {Find Card}, Keycard, EmpID, or EmpNumber. Only updates the field information; does not add a card or personnel record if the search criteria is not met.
Clear Deactivation Date	Only available when the Destination field is set to {Copy Card}. Updates the Deactivation Date on the source card by subtracting one day from the executed date.
{Find Date} Identifies Source	Only available when the Destination field is set to {Copy Card}. If selected, the {Find Card} field identifies the source card.
Deactivate Source Card	Only available when the Destination field is set to {Copy Card}. Deactivates the designated source card when the copy process is complete.

This Page Intentionally Left Blank

Additional Features 4

In This Chapter

Old Data Folder

Processed, imported data can be saved to a folder for later review.

1. **Create** a subfolder named OldData in the same location as the DNAImportSvc.exe application file.

   If this folder is created, the OpenDX service application will copy the processed information to the OldData folder prior to deleting it from the scan folder.

Change Log

Changes made to a Personnel Record are saved in a Change Log.

1. **Open** the log file.

   The log files are stored in the same folder as the ImportConfig.exe file.

   Default location:

   - 32-bit — C:/Program Files/Importer
   - 64-bit — C:/Program Files (x86)/Importer

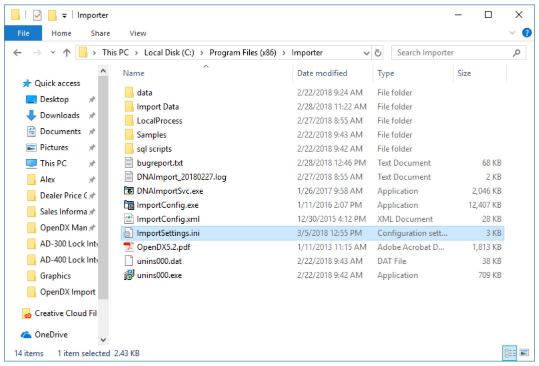
Watchdog Timer

The watchdog timer monitors package execution in OpenDX. If a package does not finish processing before the timer expires, the watchdog automatically assumes that a software failure caused the utility to freeze.

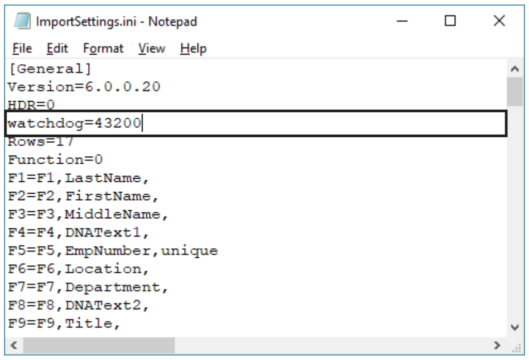
By default, this timer is set to six (6) hours. However, beginning with OpenDX version 5.0.0.12, the value can be configured by adding a line to the [General] section of the ImportSettings.ini file. The value must be entered in seconds.

1. In the [General] section, **add** the following line: watchdog=43200.

   This entry will force the watchdog timer to wait 12 hours before attempting to restart the import thread.



Name	Date modified	Type	Size
data	2/22/2018 9:24 AM	File folder	
Import Data	2/28/2018 11:22 AM	File folder	
LocalProcess	2/27/2018 8:55 AM	File folder	
Samples	2/22/2018 9:43 AM	File folder	
sql scripts	2/22/2018 9:42 AM	File folder	
bugreport.txt	2/28/2018 12:46 PM	Text Document	68 KB
DNAImport_20180227.log	2/27/2018 8:55 AM	Text Document	2 KB
DNAImportSvc.exe	1/26/2017 9:58 AM	Application	2,046 KB
ImportConfig.exe	1/11/2016 2:07 PM	Application	12,407 KB
ImportConfig.xml	12/30/2015 4:12 PM	XML Document	28 KB
ImportSettings.ini	3/5/2018 12:55 PM	Configuration sett...	3 KB
OpenDX5.2.pdf	1/11/2013 11:15 AM	Adobe Acrobat D...	1,813 KB
unins000.dat	2/22/2018 9:43 AM	DAT File	38 KB
unins000.exe	2/22/2018 9:42 AM	Application	709 KB



```
[General]
Version=6.0.0.20
HDR=0
watchdog=43200
Rows=17
Function=0
F1=F1,LastName,
F2=F2,FirstName,
F3=F3,MiddleName,
F4=F4,DNAText1,
F5=F5,EmpNumber,unique
F6=F6,Location,
F7=F7,Department,
F8=F8,DNAText2,
F9=F9,Title,
```

Log File Management

Application Log File

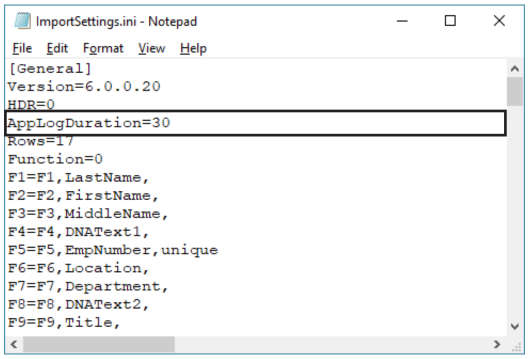
Each night at midnight (or during startup), the OpenDX service deletes the log file (.txt) that is 30 days old.

1. **Open** the ImportSettings.ini file.

   Default location:

   - 32-bit — C:/Program Files/Importer
   - 64-bit — C:/Program Files (x86)/Importer
2. In the [General] section, **add** the following line: AppLogDuration=30.
3. If needed, **edit** the value to change the duration.

   A negative value will prevent the log files from being deleted.



```
[General]
Version=6.0.0.20
HDR=0
AppLogDuration=30
Rows=17
Function=0
F1=F1,LastName,
F2=F2,FirstName,
F3=F3,MiddleName,
F4=F4,DNAText1,
F5=F5,EmpNumber,unique
F6=F6,Location,
F7=F7,Department,
F8=F8,DNAText2,
F9=F9,Title,
```

Change Log File

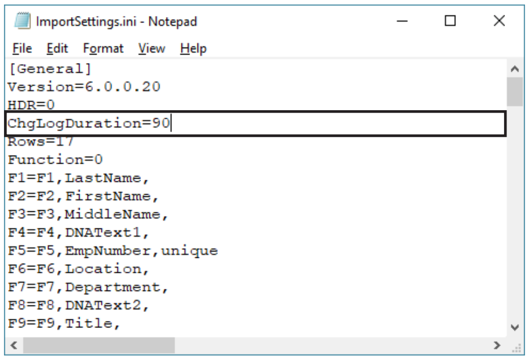
Each night at midnight (or during startup), the OpenDX service deletes the Change Log file (.csv) that is 90 days old.

1. **Open** the ImportSettings.ini file.

   Default location:

   - 32-bit — C:/Program Files/Importer
   - 64-bit — C:/Program Files (x86)/Importer
2. In the [General] section, **add** the following line: ChgLogDuration=90.
3. If needed, **edit** the value to change the duration.

   A negative value will prevent the log files from being deleted.



```
[General]
Version=6.0.0.20
HDR=0
ChgLogDuration=90
Rows=17
Function=0
F1=F1,LastName,
F2=F2,FirstName,
F3=F3,MiddleName,
F4=F4,DNAText1,
F5=F5,EmpNumber,unique
F6=F6,Location,
F7=F7,Department,
F8=F8,DNAText2,
F9=F9,Title,
```

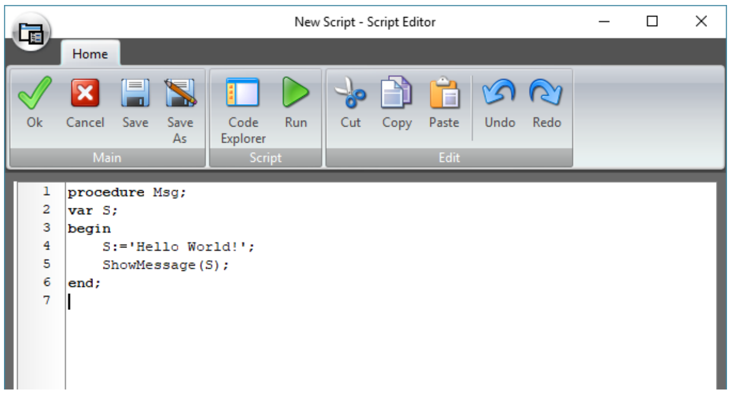
Script Editor

The Script Editor in OpenDX provides considerable flexibility for importing data. It supports both Pascal and Basic programming languages.

To open the Script Editor**:**

1. From the Main Menu, **click** the Script Editor button. Script Editor

   The New Script - Script Editor dialog appears.



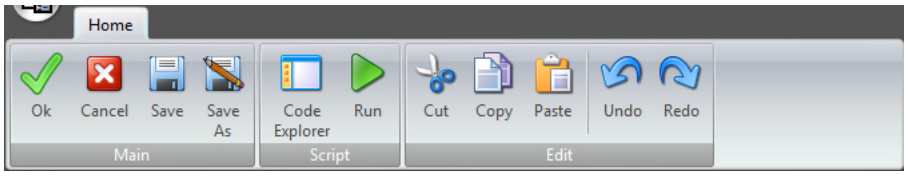
```
procedure Msg;
var S;
begin
    S:='Hello World!';
    ShowMessage(S);
end;
```

2. **Enter** the desired Pascal or Basic script and **click** OK or Save.

   See pages 4-5 through 4-14 for information on Pascal and Basic syntax.

Home Ribbon

The Home Ribbon in the Script Editor dialog contains the Main, Script, and Edit toolbars to help manage and edit a script.



Main Toolbar

- Ok - Applies the active Pascal or Basic script and closes the Script Editor without saving.
- Cancel - Closes the Script Editor without saving the changes.
- Save - Saves the script.
- Save As - Opens the Save As dialog to save the script in the desired file location.

Script Toolbar

- Code Explorer - Opens a navigation pane that displays script objects, i.e. procedures, methods, and properties.
- Run - Opens the built-in Script Debugger to test and debug workflow scripts. The interface allows the user to set breakpoints and/or step through the execution of script code line by line.

Contact Open Options Technical Support for information on using the Script Debugger.

Edit Toolbar

- Cut - Cuts the selected script from the Script Editor and copies it to the clipboard.
- Copy - Copies the selected script to the clipboard.
- Paste - Pastes the clipboard item(s) into the Script Editor.
- Undo - Reverses the user's last script action.
- Redo - Reverses the user's last Undo action.

NOTES:

Pascal Syntax

Pascal syntax currently supports:

- **begin .. end** constructors
- **procedure** and **function** declarations
- **if .. then .. else** constructors
- **for .. to .. do .. step** constructors
- **while .. do** constructors
- **repeat .. until** constructors
- **try .. except and try .. finally** blocks
- **case** statements
- **array** constructors (x:=[1, 2, 3]:)
- **^, *, /, and, +, -, or, <>, >=, <=, =, >, <, div, mod, xor, shl, shr** operators
- access to object properties and methods (**ObjectName.SubObject.Property**)

Script Structure

Two major blocks comprise the Pascal script structure: (a) procedure and function declarations, and (b) the main block. Both are optional; however, at least one should be present in the script. The main block is not required to be inside the Begin...End statement; it can be a single statement.

Script 1

```
procedure DoSomething;
begin
    CallSomething;
end;
begin
    CallSomethingElse;
end;
```

Script 2

```
begin
    CallSomething;
end;
```

Script 3

```
function MyFunction;
begin
    result:='Ok!';
end;
```

Script 4

```
CallSomethingElse;
```

Use a semicolon (;) to terminate Pascal script statements. Begin...End blocks can be used to group statements.

Identifiers

Identifiers, such as variable, function, and procedure names, must begin with a character (a..z or A..Z) or underscore (_), and can be followed by alphanumeric or underscore characters. The identifier cannot contain any other characters or spaces.

Valid Identifiers:

```
VarName
_Some
V1A2
___Some___
```

Invalid Identifiers:

```
2Var
My Name
Some-more
This,is,not,valid
```

Assign Statements

Assign statements, which assign a value or expression to a variable or object property, are built using ":=". See examples below.

```
MyVar:=2
Button.Caption:="This' + 'is ok.';
```

Character Strings

Strings (character sequences) are declared in Pascal by using a pair of single quotation marks. Double quotation marks are not used. To declare a character inside a string, use #nn. The '+" operator is not necessary to add the character(s) to the string. See examples below.

```
A:='This is a text';

Str:='Text' + 'concat';

B:='String with CR and LF char at the end' #13#10;

C:='String with '#33#34' characters in the midle';
```

Comments

To add user comments within the Pascal script, insert two forward slashes (//) before the comment line or wrap the comment in (* *) or { } blocks. Comments will have no effect on the script. See examples below.

```
// This is a comment before ShowMessage
 ShowMessage('Ok');

(* This is another comment *)
 ShowMessage('Ok');

{ And this is a comment
 with two lines }
 ShowMessage('Ok');
```

Indexes

Strings, arrays, and array properties can be indexed using square brackets ([ ]). For example, if Str denotes a string variable, the expression Str[3] returns the third character in the string denoted by Str, while Str[I + 1] returns the character immediately after the one indexed by I. See examples below.

```
MyChar:=MyStr[2];

MyStr[1]:='A';

MyArray[1,2]:=1530;

Lines.Strings[2]:='Some text';
```

Variables

Pascal script does not require the user to declare variable types; instead, the variable is declared using only the var directive and its user-defined name (var Name).

If the OptionExplicit script property is set to False, the user does not need to declare variables; they are declared implicitly. To have greater control over the script, set the OptionExplicit property to True. This setting will raise a compile error if the variable is used (but not declared) in the script.

See below for examples.

Script 1

```
procedure Msg;
var S;
begin
    S:='Hello World!';
    ShowMessage(S);
end;
```

Script 2

```
var A;
begin
    A:=0;
    A:=A+1;
end;
```

Script 3

```
var S;
S:='Hello World!';
ShowMessage(S);
```

If the OptionExplicit script property is set to False, the example scripts above do not require var declarations.

Arrays

Pascal scripts support array constructors and variant arrays. An array is a data structure that can store a fixed-size sequential collection of elements of the same type. Use square brackets ([]) to construct an array, and nest array constructors to create multi-index arrays.

If the variable is a variant array, the script automatically supports indexing in that variable. A variable is a variant array if it meets one of three criteria:

- The variable was assigned using an array constructor
- The variable directly references a Delphi variable that is a variant array
- The variable was created using VarArrayCreate

Arrays in Pascal script contain a 0-based index. See examples below.

```
NewArray:=[2,4,6,8];

Num:=NewArray[1]; // Num receives "4"

MultiArray:=[['green','red','blue'],['apple', 'orange', 'lemon']];

Str:=MultiArray[0,2]; // Str receives 'blue'

MultiArray[1,1]:='new orange';
```

If Statements

Pascal script contains two types of "If" Statements:

- if...then
- if...then...else

When the "if" expression is true, the script statement (or block) is executed. When the "if" expression is false, the statement (or block) after the "else" expression is executed; however, if the script does not contain an "else" expression, nothing will be executed. See examples below.

```
if J <> 0 then Result:=I/J;

if J = 0 then Exit else Result:=I/J;

if J <> 0 then
  begin
    Result:=I/J;
    Count:=Count + 1;
  end;
else Done:=True;
```

While Statements

In Pascal script, a "While" Statement is used to repeat a statement or block while the control condition (expression) is evaluated as true. The control condition is evaluated before the statement. Hence, if the control condition is false during its first iteration, the statement sequence is never executed. The while statement executes its constituent statement (or block) repeatedly, testing the expression before each iteration. As long as the expression returns true, the statement will continue to execute. See examples below.

```
while Data[I] <> X do I:=I + 1;

while I > 0 do
  begin
    if Odd(I) then Z:=Z * X;
    I:=I div 2;
    X:=Sqr(X);
  end;

while not Eof(InputFile) do
  begin
    Readln(InputFile, Line);
    Process(Line);
  end;
```

Repeat Statements

The syntax for a “Repeat” Statement is repeat statement1;...statementX; until expression (where expression returns a Boolean value). The repeat statement executes its sequence of constituent statements continually, testing the expression after each iteration. When the expression returns true, the repeat statement terminates. Because the expression is not evaluated until after the first iteration, the sequence always executes at least once. See examples below.

```
repeat
  K:=I mod J;
  I:=J;
  J:=K;
until J=0;

repeat
  Write('Enter a value (0..9):');
  Readln(I);
until (I>=0) and (I<=9);
```

For Statements

Pascal script supports “For” Statements with the following syntax: for counter := initialValue to finalValue do statement. The for statement sets the counter to the initialValue, repeatedly executes the statement (or block), and increments the value of the counter until it reaches the finalValue. See examples below.

```
for c:=1 to 10 do a:=a+c;

for i:=a to b do
  begin
    j:=I^2;
    sum:=sum+j;
  end;
```

Case Statements

Pascal script supports “Case” Statements with the following syntax:

```
case selectorExpression of
  caseexpr1: statement1;
  ...
  caseexprn: statementn;
else
  else statement;
end;
```

If selectorExpression matches one of the casexprn results, the respective statement (or block) will be executed. Otherwise, the else statement (optional) will be executed. Unlike Delphi, the case statement in Pascal script is not limited to ordinal values. Any type of expression can be used in both the selectorExpression and case expression. See example below.

```
case uppercase(Fruit) of
    'lime': ShowMessage('green');
    'orange': ShowMessage('red');
    caseexprn: statementn;
else
    ShowMessage('black');
end;
```

Function and Procedure Declarations

Function and Procedure declarations are similar to Object Pascal (OP) in Delphi; however, variable types are not specified. Like OP, to return function values, use implicitly declared variables or reference parameters. Include the restriction in the reference parameter. See examples below.

```
procedure HelloWorld;
  begin
      ShowMessage('Hello world!');
  end;

procedure UpcaseMessage(Msg);
  begin
      ShowMessage(Uppercase(Msg));
  end;

function TodayAsString;
  begin
      result:=DateToStr(Date);
  end;

function Max(A,B);
  begin
      if A>B then result:=A
      else result:=B;
  end;
```

NOTES:

Basic Syntax

Basic syntax currently supports:

- **sub .. end** and **function .. end** declarations
- **byref** and **dim** directives
- **if .. then .. else .. end** constructor
- **for .. to .. step .. next** constructor
- **do .. while .. loop** and **do .. loop .. while** constructors
- **do .. until .. loop** and **do .. loop .. until** constructors
- **^, *, /, and, +, -, or, <>, >=, <=, =, >, <, div, mod, xor, shl, shr** operators
- **try .. except** and **try .. finally** blocks
- **select case .. end select** constructor
- **array** constructors (x:=[1, 2, 3];)
- **exit** statement
- access to object properties and methods (**ObjectName.SubObject.Property**)

Script Structure

Two major blocks comprise the Basic script structure: (a) sub and function declarations, and (b) the main block. Both are optional; however, at least one should be present in the script. See examples below.

```
     Script 1              Script 2              Script 3

SUB DoSomething       CallSomethingElse     FUNCTION MyFunction
  CallSomething                               MyFunction = “OK!”
END SUB                                     END FUNCTION
```

Insert a colon (:) to separate statements in a single line.

Identifiers

Identifiers, such as variable, function, and procedure names, must begin with a character (a..z or A..Z) or underscore (_), and can be followed by alphanumeric or underscore characters. The identifier cannot contain any other characters or spaces.

Valid Identifiers:

```
VarName
_Some
V1A2
___Some___
```

Invalid Identifiers:

```
2Var
My Name
Some-more
This,is,not,valid
```

Indexes

Strings, arrays, and array properties can be indexed using square brackets ([ ]). For example, if Str denotes a string variable, the expression Str[3] returns the third character in the string denoted by Str, while Str[I + 1] returns the character immediately after the one indexed by I. See examples below.

```
MyChar = MyStr[2]

MyStr[1] = “A”

MyArray[1,2] = 1530

Lines.Strings[2]= “Some text”
```

Arrays

Basic scripts support array constructors and variant arrays. An array is a data structure that can store a fixed-size sequential collection of elements of the same type. Use square brackets ([]) to construct an array, and nest array constructors to create multi-index arrays.

If the variable is a variant array, the script automatically supports indexing in that variable. A variable is a variant array if it meets one of three criteria:

- The variable was assigned using an array constructor
- The variable directly references a Delphi variable that is a variant array
- The variable was created using VarArrayCreate

Arrays in Basic script contain a 0-based index. See examples below.

```
NewArray = [2,4,6,8]

Num = NewArray[1] // Num receives "4"

MultiArray = [["green","red","blue"],["apple", "orange", "lemon"]]

Str = MultiArray[0,2] // Str receives 'blue'

MultiArray[1,1] = "new orange"
```

If Statements

Basic script contains two types of "If" Statements:

- IF...THEN...END
- IF...THEN...ELSE..END IF

When the "IF" expression is true, the script statement is executed. When the "IF" expression is false, the statement after the "else" expression is executed. See examples below.

```
IF J <> 0 THEN Result = I/J END IF

IF J = 0 THEN Exit ELSE Result = I/J END IF

IF J <> 0 THEN
   Result = I/J
   Count = Count + 1
ELSE Done = True
END IF
```

While Statements

In Basic script, a "While" Statement is used to repeat statements while the control condition (expression) is evaluated as true. The control condition is evaluated before the statement. Hence, if the control condition is false during its first iteration, the statement sequence is never executed. The WHILE statement executes its constituent statement repeatedly, testing the expression before each iteration. As long as the expression returns true, the statement will continue to execute. See examples below.

```
WHILE (Data[I] <> X) I = I+1 END WHILE

WHILE (I > 0)
  IF Odd(I) THEN Z = Z*X END IF
  X = Sqr(X)
END WHILE

WHILE (not Eof(InputFile))
  Readln(InputFile, Line)
  Process(Line)
END WHILE
```

Loop Statements

Basic scripts support "Loop" Statements with the following syntax:

- DO WHILE expr statements LOOP
- DO UNTIL expr statements LOOP
- DO statements LOOP WHILE expr
- DO statements LOOP UNTIL expr

Statements will be executed while expr is true or until expr is true. If expr is before statements, the control condition will be tested before each iteration. Otherwise, the control condition will be tested after each iteration. See examples below.

```
DO
  K = I mod J
  I = J
  J = K
LOOP UNTIL J = 0

DO UNTIL I >= 0
  Write("Enter a value (0..9): ")
  Readln(I)
LOOP

DO
  K = I mod J
  I = J
  J = K
LOOP WHILE J <> 0

DO WHILE I < 0
  Write("Enter a value (0..9): ")
  Readln(I)
LOOP
```

For Statements

Basic script supports "For" Statements with the following syntax: FOR counter = initialValue TO finalValue STEP stepValue statements NEXT. The FOR statement sets the counter to initialValue, repeatedly executes the statement until NEXT, and increments the value of the counter by stepValue until the counter reaches finalValue. See examples below.

```
FOR c = 1 TO 10 STEP 2
  a = a + c
NEXT

FOR I = a TO b
  j = i ^ 2
  sum = sum + j
NEXT
```

The STEP expression is optional; if omitted, stepValue defaults to 1.

Select Case Statements

Basic script supports "Select Case" Statements with the following syntax:

```
SELECT CASE selectorExpression
   CASE caseexpr1
     statement1
   CASE caseexprn
     statementn
CASE ELSE
   elsestatement
END SELECT
```

If selectorExpression matches the result of one of the caseexpr expressions, the respective statements are executed. Otherwise, the CASE ELSE statement will be executed. The CASE ELSE statement is optional. See example below.

```
SELECT CASE uppercase(Fruit)
  CASE "lime"
    ShowMessage("green")
  CASE "orange"
    ShowMessage("orange")
  CASE "apple"
    ShowMessage("red")
CASE ELSE
  ShowMessage("black")
END SELECT
```

Function and Sub Declarations

Function and Sub declarations are similar to Basic. In functions to return function values, use an implicitly declared variable that has the same name as the function. Use a BYREF directive to include reference parameters. See examples below.

```
SUB HelloWorld
  ShowMessage("Hello world!")
END SUB

SUB UpcaseMessage(Msg)
  ShowMessage(Uppercase(Msg))
END SUB

FUNCTION TodayAsString
  TodayAsString = DateToStr(Date)
END FUNCTION

FUNCTION Max(A,B)
  IF A>B THEN
    MAX = A
  ELSE
    MAX = B
  END IF
END FUNCTION

SUB SWAPVALUES(BYREF A,B)
  DIM TEMP
  TEMP = A
  A = B
  B = TEMP
END SUB
```

This Page Intentionally Left Blank

This Page Intentionally Left Blank

OPEN OPTIONS®
ACCESS TECHNOLOGY