Data Express 4.0

Data Masking Guide

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK
http://www.microfocus.com

2019-02-08

Contents

Data Masking Guide

Outlines the concepts and procedures used by the Data Masking module. Data Masking makes sensitive data anonymous in application test environments while maintaining the same characteristics as the original data.

The Data Masking module comprises a mainframe engine and a client configuration tool. The mainframe engine reads and catalogs the original information, and stores it in the Knowledge Base.

Data Masking enables seamless access to supported rDBMSes, by providing the additional connectivity, infrastructure, and interfaces for this environment.

Data Masking processing comprises the following elements:

- Setup processing that automatically identifies file layouts.
- The association of routines with data elements to mask data and create the modified data stores.

Who Should Read this Guide

This guide is for Micro Focus users who are interested in managing the privacy of data contained in databases of applications running on z/OS or distributed platforms that are either developed internally or purchased by external suppliers.

This guide explains the Data Masking configuration procedure, and describes the steps required to migrate the data stores in your application.

Before using this product, we recommend you analyze all the files to be processed and associate them with their copybooks using the Data Express Data module. We also recommend that you carefully read the *Process Guide for z/OS* and *Front End Guide*.

Data Masking Considerations

Bear in mind the following points about the Data Masking module for Data Express:

- This product is designed to read data from data stores containing the information to be analyzed, and to rewrite the content of these data stores to output data stores specified by the user.
- To use this product with z/OS, you must have the correct licence. See the *Installation Guide* for more details.
- You do not need to analyze the sources of the programs in the application in order to use Data Masking.
- Data Masking does not perform any operation capable of altering the analyzed data.

The tables defined during the installation phase that are updated by Data Masking are described in *Appendix A. Data Masking Tables*.

Getting Started

Provides an overview of the Data Masking module. You should read it before using the module.

General Characteristics of Use

Data Express for z/OS can be used in either Standard or Client/Server mode, whereas Data Express for Distributed Systems can only be used in Standard mode:

- Standard – Supports direct access to the data resident on RDBMS from your PC.
- Client/Server – Supports access to the data resident on RDBMS through the kbde-Server: a product based on a three-tier architecture where the second tier handles data transfer optimization. Client/ Server mode lets you access data over the Internet with the lowest amount of administration necessary and the best utilization of network capacity.

Note: Client/Server mode can use the same functions as Standard mode without you having to install additional software on your PC, such as the DB2 client or the Borland Database Engine (BDE).

For Data Express configuration information and instructions, see:

- *Appendix A. Standard Mode* in the *Front End Guide*.
- *Appendix B. Client/Server Mode* in the *Front End Guide* and in *Appendix C. Client/Server Mode* of this guide.

The following guidelines apply to windows and screen items within the Data Masking module:

- The main window: **Data Masking - [Active Files]** cannot be closed except when exiting the module.
- Multiple MDI windows may be opened simultaneously.
- Right-click an item or area to view a list of applicable commands.

Starting Data Masking

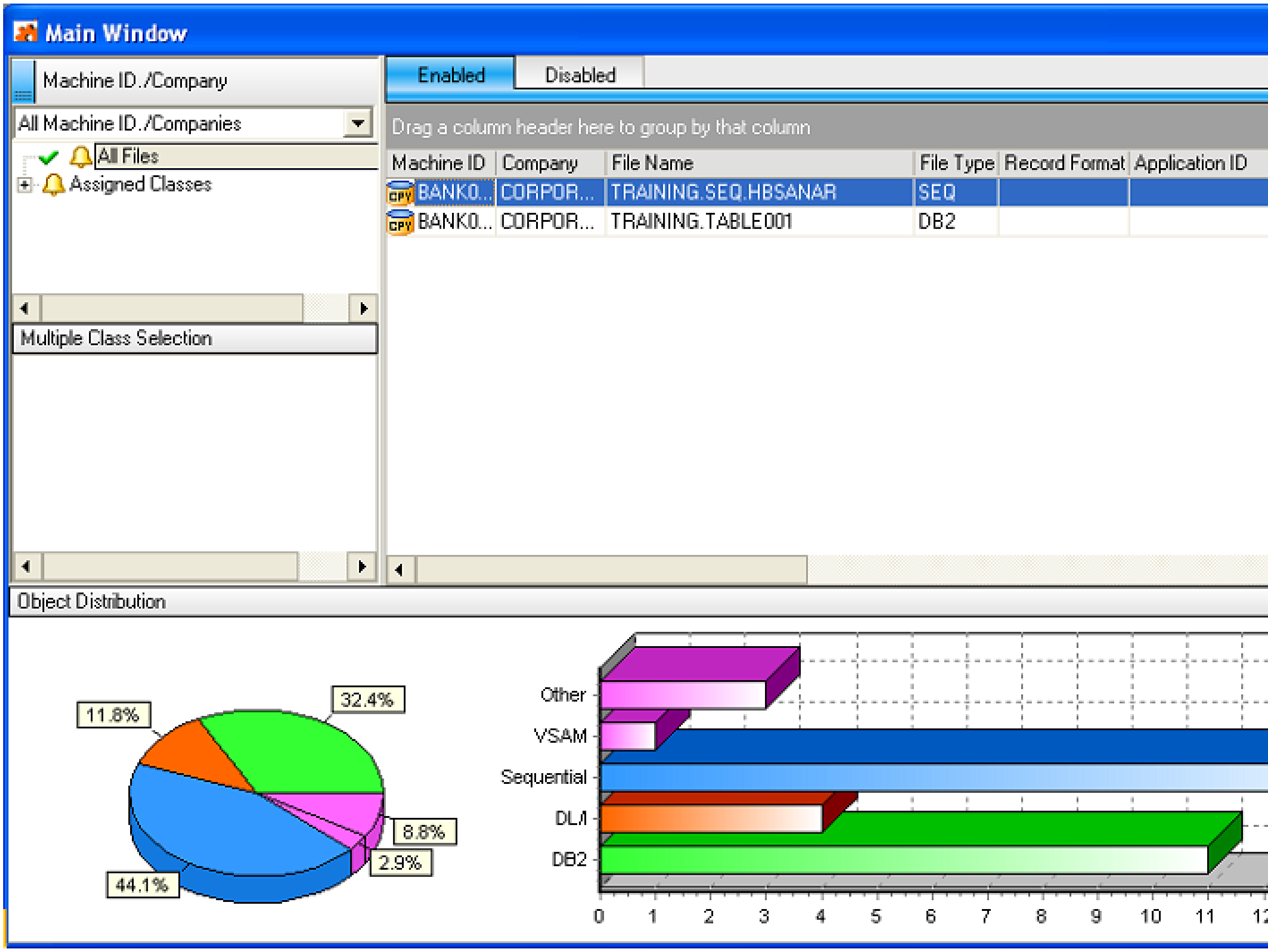
To start the Data Masking module:

1. Click **Start > All Programs > Micro Focus Data Express 4.0 > Data Masking**.

   Note: If you are using Data Express on Windows Vista and User Access Control is enabled, you must run Data Masking as Administrator.
2. Select the required database with which to launch the connection. By default, the highlighted item in the list of available databases is the last database you connected to using the Data Masking module.

   Note: To connect to the database for first time, you need your user ID and password.

Select the required schema. The **Main Window** appears as shown:



Machine ID	Company	File Name	File Type	Record Format	Application ID
BANKO...	CORPOR...	TRAINING.SEQ.HBSANAR	SEQ		
BANKO...	CORPOR...	TRAINING.TABLE001	DB2		

Figure 1: Data Masking - Main Window

Tip: Once you are connected to a database, you can open and close a database from the **Main Window** by clicking **File** > **Open Database** > **File** > **Close Database**.

Menus and Toolbars

This section describes the menus and common toolbars displayed in the **Main Window**.

File Menu

The **File** menu contains the following commands:

- **Open Database** – Opens a new database connection
- **Close Database** – Closes database connection
- **Exit** – Terminates the Data Masking module

Tools Menu

The **Tools** menu contains the following commands:

- **Enable All Data Stores** – Specifies to enable all disabled data stores for the currently selected machine ID and company combination that have data elements assigned to the data masking routine.

- **Options** – Opens the Option window for setting the help language, security options, and database user ID and password.
- **Connection Options** – Opens the Connection Option window for changing the type of connection.

View Menu

The **View** menu contains the following commands:

- **Hide ToolBar** – Displays or hides the tool bar
- **Hide Status Bar** – Displays or hides the status bar

Windows Menu

The set of open windows can be organized according to the standard Windows arrangements. The **Windows** menu contains the following commands:

- **Tile Horizontally** – Arranges all open windows from top to bottom.
- **Tile Vertically** – Arranges all open windows from left to right.
- **Cascade** – Arranges all open windows so that they will not overlap.
- **Minimize All** – Minimizes all open windows with the exception of the main window.
- **Arrange** – Arranges all minimized windows.

Help Menu

The Help menu contains the following commands:

- Help Topics, Index – Opens online help information for the product in the language selected in the Options window.
- Micro Focus SupportLine – Contains commands for connecting to Micro Focus Product Support for information or technical assistance.
- Micro Focus - Data Express Home Page – Connects to the Micro Focus Data Express web site.
- About – Displays production information for the Data Masking module.

Toolbar

The icons listed in default toolbars are described as follows:

- - Open database connection
- - Close database connection
- - Refresh information
- - Work with classes
- - Distributed Exporter

Context Menus

This section describes the commands included in context menus.

Column Sorting

A context menu to display a list of sort options can be displayed by right-clicking a column header.

Note:

All changes to the grid layout are saved in the Windows system registry. For example, groups, column width attributes, column order attributes, and column font and color information. However, filtering and sorting preferences are not saved.

The column sorting context menu contains the following commands:

- Sort Descending – Sorts the list contents in descending order. The same result can be obtained by simply clicking the column heading.
- Sort Ascending – Sorts the list contents in ascending order. The same result can be obtained by simply clicking the column heading.
- Group by This Column – Groups the list according to the contents of the selected column if the Group Panel has been enabled.
- Remove This Column – Deletes the selected column.
- Column Selector – Displays a Customize window that contains a list of columns that have been removed previously. A column can be restored by performing a drag-and-drop operation to the desired position in the list. A column can be removed from a list by performing a drag-and-drop operation from the list to the Customize window.
- Alignment – Aligns the content of the columns to the center, right, or left as specified by the user.
- Best Fit – Resizes the selected column to allow the content of each record to be displayed. The same result can be obtained by double-clicking the column's external margin.
- Pick Color – Allows the color of the selected column to be changed.
- Pick Font – Allows the font, font style, size, color, and script to be changed.
- Best Fit (All Columns) – Resizes all the columns from the list to allow the content of each record to be displayed.

  Note: Columns can also be sorted by clicking or performing a drag-and-drop operation on the column heading.
- Information displayed can be reduced by clicking a column heading, and then typing the first letter of the items you are interested in.

Grid Functions

A context menu to display a list of functions for a grid can be displayed by right-clicking any place on the grid.

Note: All changes to the grid layout are saved in the Windows registry. For example, groups , column width attributes, column order attributes , and column font and color information . However, filtering and sorting preferences are not saved.

The grid functions context menu contains the following commands:

- **Column Selector** - Displays a **Customize** window that contains a list of columns that have been removed previously. A column can be restored by performing a drag-and-drop operation to the desired position in the list. A column can be removed from a list by performing a drag-and-drop operation from the list to the **Customize** window.
- **Best Fit (All Columns)** - Resizes all the columns from the list to allow the content of each record to be displayed.
- **Export to File…** - Exports the information to a file using one of the following compatible formats:
  - comma separated value (`.csv`)
  - Hyper Text Markup Language (`.htm`)
  - text (`.txt`)
  - Excel (`.xls`)
  - Extensible Markup Language (`.xml`)
- **Show Summary Footer** - Displays the summary footer beneath the grid. By selecting a row and right-clicking the summary footer, a context menu displaying arithmetic operations you can perform on the selected row appears.

  Note: Arithmetic operations are only possible on numeric contents.

  Possible arithmetic operations include:

  - **Summary** - Displays the sum of all quantities in the column . T his option is active only for columns with integer or percentage values ; if the value of the cells in a specific column are represented as a percentage, their sum does not always equal 100% .
  - **Minimum** - Displays the minimum value of all values in the column. The option is active only for columns with numerical values.
  - **Maximum** - Displays the maximum value of all values in the column. The option is active only for columns with numerical values.
  - **Count** - Displays the number of elements in a column.
  - **Average** - Displays the arithmetic mean of all values in the column. The option is active only for columns with numerical values.
  - **None** - Specifies using no arithmetic operation on a column.
- **Show Group Panel** - Displays the column name by which data is grouped above the grid. By dragging the column headers into this window, a group hierarchy can be defined.

  Note: If data is being grouped by column, a footer panel is displayed beneath the grid for each group.
- **Filter Active** - If enabled, lets you choose data to be displayed on the grid; a downward arrow icon will appear in the column headers.

  Click an arrow to choose the data to be displayed on the grid. This function lets you generate a personal filter.

Note: The time taken to display the window increases with the number of records to be loaded.

Data Masking Main Window

Introduction

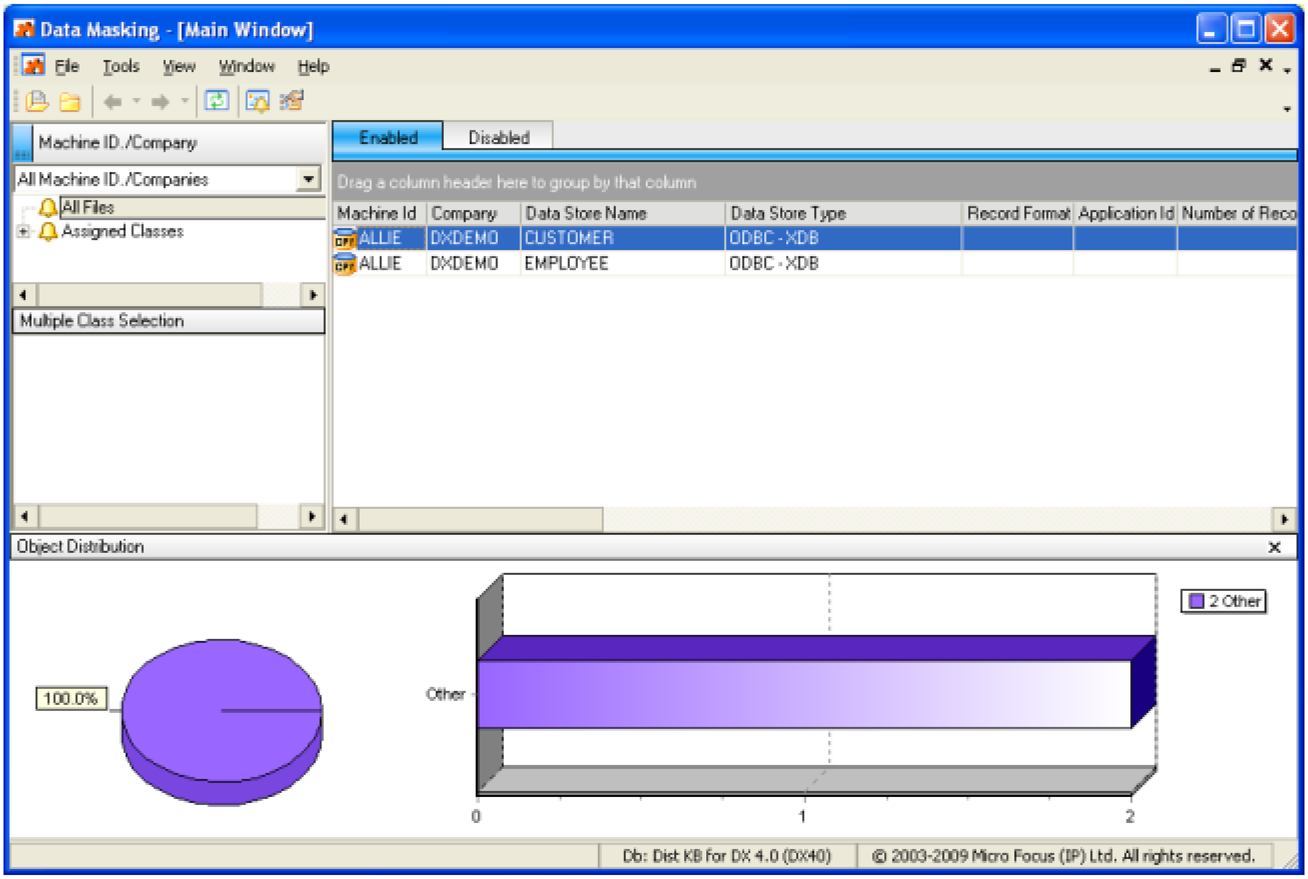
After starting Data Masking, the window: Main Window appears.

From this window, you can enable or disable data stores, and view a report of how the classification or the partitioning of your data stores is being performed for the specified Machine ID and Company names.

Window Components

This section describes each component of the Main Window area.

The Main Window comprises four main areas: **Machine ID./Company** grid, **All Machine ID./Companies**, **Multiple Class Selection**, and **Object Distribution** as shown here:



Machine Id	Company	Data Store Name	Data Store Type	Record Format	Application Id	Number of Reco
ALLIE	DXDEMO	CUSTOMER	ODBC - XDB			
ALLIE	DXDEMO	EMPLOYEE	ODBC - XDB			

Db: Dist KB for DX 4.0 (DX40)

© 2003-2009 Micro Focus (IP) Ltd. All rights reserved.

Machine ID./Company Grid

The area containing the grid and the tabs **Enabled** and **Disabled** is referred to as the **Machine ID./ Company** area or the **Grid** area. The grid lists data stores related to the selected class and to the selected machine ID and company. The names of the forms, their meaning and the included information are outlined below.

The **Enabled** tab shows a list of all active data stores (enabling a data stores means that it is visible in the export functionality in order to be prepared for masking), and all data stores, which have one or more specific classes related to their data elements.

Select and right-click a data store from the list to view a list of available commands as listed below:

- **Open Selected Data Store** – Displays a window with information about the data elements of the data store.
- **Data Store Properties** – Displays a window with all the information about the data store.
- **Enable** or **Disable** – Enables or disables the masking function.
- **Data Changer properties** – Displays a window with information about the Data Changer function for the selected data store.
- **Set Application ID** – Open a window to select the application identifier.

Note: The Data Changer (also referred to as the Data Masker) is the function within Data Masking that performs the masking.

- The **Work with Data Changer Files** function is available in Data Builder.
- For more information on working with the Data Changer function, see the *Front End Guide*.

All Machine ID./Companies

The **Machine ID./Companies** area contains drop-down lists from which you can choose the machine ID and the company ID for the specified assigned class. The hierarchical structure shows the grouping associated with each root-level node. Click **All Files** or select the assigned class you are interested in, and use the **Enabled** and **Disabled** tabs to view a list of active or inactive data stores respectively in the Grid area (which is part of the **Machine ID./Company** area).

Multiple Class Selection

The Multiple Class Selection area enables you to drag and drop two or more classes into it for class selection. Click the appropriate (x) icon to remove a class.

Tip: The same function can also be executed from a context menu, which is accessible by right-clicking this area.

Object Distribution

The **Object Distribution** area displays a pie chart and bar chart representation of the data store types for the classes selected in the **Machine ID./Companies** or **Multiple Class Section** area. The pie chart shows the percentage distribution of data store types, while the bar chart shows the numerical distribution of data store types. For both charts, both enabled and disabled data stores are included. Click the minimize button (_) to minimize the **Object Distribution** area.

Column Headings in the Grid Area

The following list describes the column headings in the **Grid** area for data stores that have been loaded in the Data Express Knowledge Base:

- **Machine ID** – Identifier of the machine.
- **Company** – Name of the company.
- **Data Store Name** – Name of the data store.
- **Data Store type** – Type of data store. For example, DB2, VSAM, DL/I, GDG, SEQ, ODBC, or ORA.
- **Record Format** – Name of the record format.
- **Application ID** – Identifier (if it exists) of the application area the data store belongs to.
- **Number of Records** – Number of records contained in the data store.
- **Record Length** – Length (in bytes) of the record.
- **Unload Input Data Store Type** – Type of the Unload Input data store.
- **Unload Input Data Store Name** – Name of the Unload Input data store.
- **Unload Input Data Store Version** – Version of the Unload Input data store.
- **Format Selector** – Exit routine for data store format recognition.
- **Select Program** – Selection program of the individual record.
- **Unload Output Data Store Type** – Type of the Unload Output data store.
- **Unload Output Data Store Name** – Name of the Unload Output data store.
- **Unload Output Data Store Version** – Version of the Unload Output data store.
- **Assigned Copy PDS** – Name of the PDS, which contains the copybook associated with the data store.
- **Assigned Copy Name** – Name of the copybook associated with the data store.
- **Database** – Name of the database.

Available Grid Operations

Right-click a data store name in the grid to view a list of available commands as described below:

- **Open the Selected Data Store** – Displays a window that lists all data elements as well as the classes assigned to them.
- **Data Store Properties** – Displays a window listing data store information.
- **Data Changer Properties** – Displays a window listing information for the associated Data Changer routine.

- **Disable or Enable** – Enables a data store (if it is actively disabled) or disables a data store (if it is actively enabled).
- **Set File Category** – Displays a simple form to specify data store category and a brief description of the category for the selected files in the Machine Id./Companies or Multiple Class Section area.
- **Set Application ID** – Displays a simple form to specify the application ID and a brief description of the ID.

Data Changer Data Store Properties

The Properties - File window is a tabbed view that displays all attributes for a selected data store.

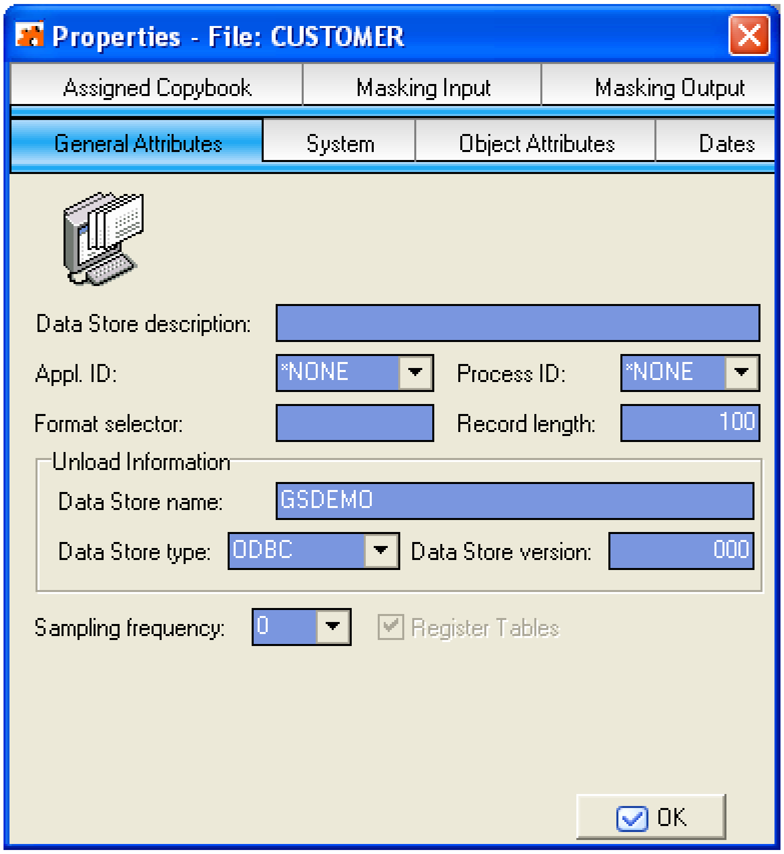
Introduction

To access the Properties - File window, right-click the file in the **Grid** area of the Data Masking main window, and then click **Data Store Properties**.

For more information about the context menu, see *Machine Id./Company Grid* in the *Data Masking Main Window* section.

Window Contents

This section describes the fields listed on each tab of the Properties – File window as shown below:



Data Store description:

Appl. ID: *NONE

Process ID: *NONE

Format selector:

Record length: 100

Unload Information

Data Store name: GSDEMO

Data Store type: ODBC

Data Store version: 000

Sampling frequency: 0

☑ Register Tables

Figure 2: Properties – File Window Listing File Attributes

Note: The **Masking Input** and **Masking Output** tabs display only when a Data Masking routine is associated with the data store. For more information about Changer Program properties, see *Window Contents* in the *Data Changer Properties* section.

General Attributes

The fields listed on the **General Attributes** tab are:

- **Data Store Description** – Description of the data store.
- **Appl. ID** – Identifier (if it exists) of the application area that the data store belongs to.
- **Process ID** – Process identifier.
- **Format selector** – Exit routine for file format recognition.
- **Record length** – Length (in bytes) of the record.
- **Data Store name** – Name of the data store.
- **Data Store type** – Type of data store. For example, DB2, VSAM, DL/I, GDG, SEQ, ODBC, or ORA.
- **Data Store version** – Version of the data store.
- **Sampling frequency** – Frequency (for example, daily or monthly) of sampling of the data elements for which the distribution of values is desired.
- **v** – Indicator of whether the data store contains demographic data.

System

The fields listed on the **System** tab are:

- **Machine ID** – Identifier of the machine
- **Company name** – Name of the company

Object Attributes

The fields listed on the **Object Attributes** tab are:

- **Data Store type** – Type of data store. For example, DB2, VSAM, DL/I, GDG, SEQ, ODBC, or ORA.
- **Data Store name** – Name of the data store.
- **Data Store description** – Description of the data store.
- **Application ID** – Identifier (if it exists) of the application area that the data store belongs to.
- **Record length** – Length (in bytes) of the record.
- **Record format** – Name of the record format.
- **Format selector** – Exit routine for file format recognition.
- **Number of records** – Number of records contained in the data store.
- **Analyzed records** – Number of analyzed records.
- **Database name** – Name of the database.
- **Tablespace name** – Name of the tablespace.
- **Object ID** – Object identifier.

Dates

The fields listed on the **Dates** tab are:

- **Date of Analysis** – Date of the analysis carried out on the data store.
- **Date of Last Calculation** – Date of the last recalculation carried out on the data store.
- **Date of Last Sample** – Date of the last calculation of the distribution of values of the data elements involved in the process.

Assigned Copybook

The fields on the **Assigned Copybook** tab are:

- **Copybook PDS** – Name of the area (level 01) within the copybook associated with the data store.
- **Copybook name** – Name of the copybook assigned to the data store.

- **I/O Area Name** – Name of the area present on the copybook that describes the layout of the file.
- **I/O area progressive** – Progressive number of the area within the copybook associated with the file.
- **Information provider** – Supplier of the association information; PGM is information obtained from batch processing, USR is information supplied by the user.

Note: The **Assigned Copybook** tab is not visible for DB2 tables.

Data Changer Data Element Selection

The **Data Store Information** tab in the **Main Window** is similar to the **Work with Data Elements** area in Data Builder. Both areas can be used to associate modification routines to a data element in order to enable the routine associated with the data element, and to view the information about the data element. However, the **Data Store Information** tab filters information for the selected data store by the attributes **Machine ID**, **Company ID**, **Data Store Type**, and **Data Store Name**.

Introduction

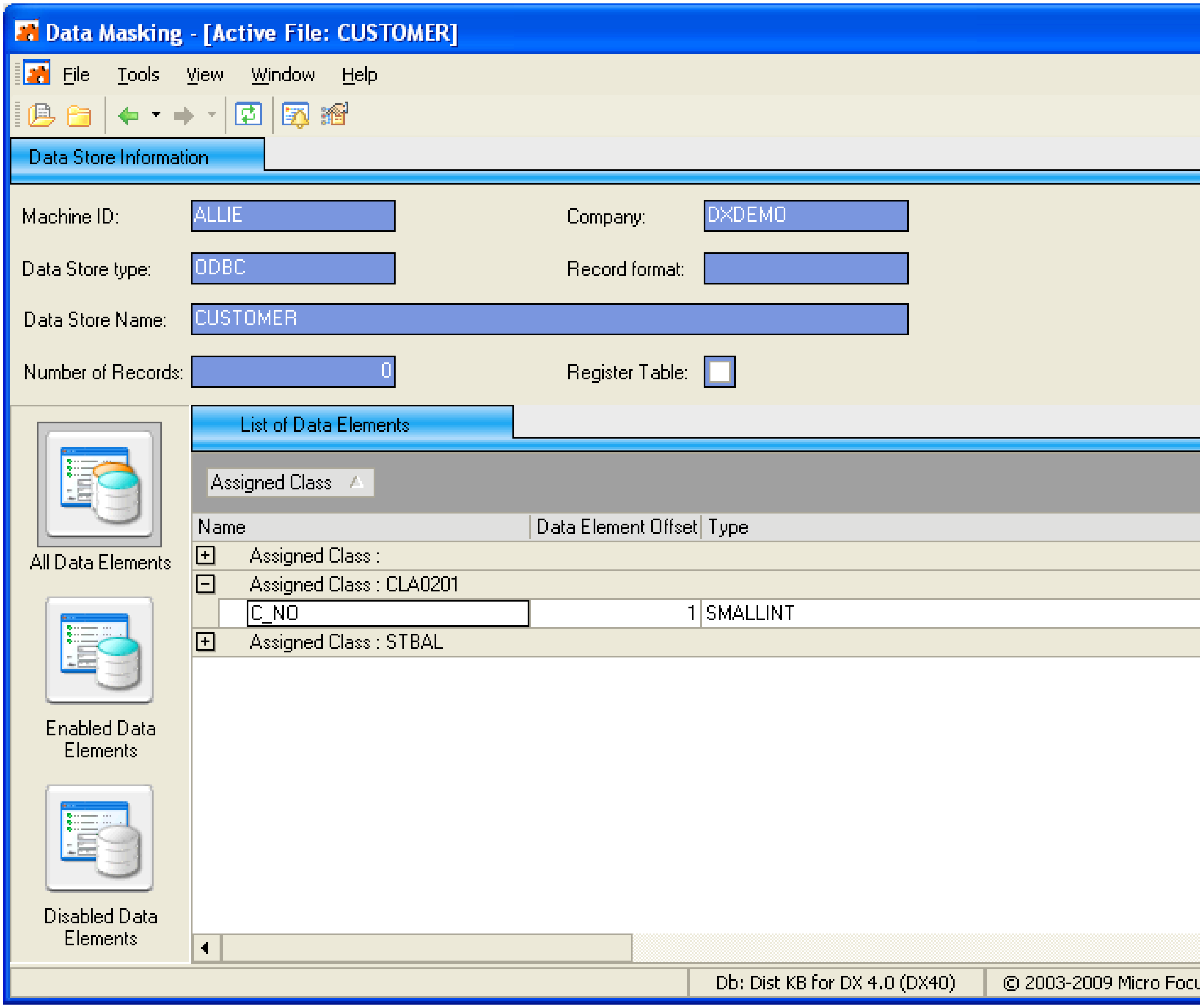
The Data Store Information tab can be used to associate modification routines to a data element in order to enable the routine associated with the data element, and to view the information about the data element. This tab supplies all available information about the selected data store and all the classes associated with it.

To access the **Data Store Information** tab, right-click the data store in the **Grid** area of the Data Masking main window, and click **Open Selected Data Store**.

For more information about the context menu, see *Machine Id./Company Grid* in the *Data Masking Main Window* section.

Window Contents

This section describes the fields and items listed in the **Data Store Information** area as shown below:



Machine ID:	ALLIE	Company:	DXDEMO
Data Store type:	ODBC	Record format:	
Data Store Name:	CUSTOMER		
Number of Records:	0	Register Table:	□

Name	Data Element Offset	Type
+ Assigned Class :		
− Assigned Class : CLA0201		
C_NO	1	SMALLINT
+ Assigned Class : STBAL		

Db: Dist KB for DX 4.0 (DX40) | © 2003-2009 Micro Focu

Figure 3: Data Masking – Data Store Information Tab

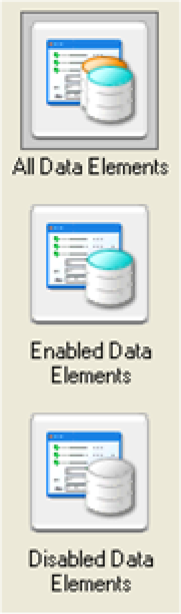
Data Store Information

The fields listed in the **Data Store Information** tab are:

- **Machine ID** – Identifier of the machine
- **Company** – Name of the company.
- **Data Store type** – Type of data store. For example, DB2, VSAM, DL/I, GDG, SEQ, ODBC, or ORA.
- **Record format** – Name of the record format.
- **Data Store Name** – Name of data store.
- **Number of Records** – Number of records contained in the data store.
- **Register Table** – Set/Reset table as Register.

Icon Bar

The buttons listed in the button bar are:



- **All Data Elements** – Shows all data elements with Data Changer enabled and disabled
- **Enabled Data Elements** – Shows the data elements with Data Changer enabled
- **Disabled Data Elements** – Shows the data elements with Data Changer disabled

List of Data Elements

The column names listed in the **List of Data Elements** tab are:

- **Name** – The name of the data element within the data store.
- **Data Element Offset** – The offset of the data element.
- **Type** – The data type of the data element. For example, Numeric or Alphanumeric.
- **Nature** – The nature of the data element. For example, Alphabetical.
- **Integer** – The number of integers within a numeric data element.
- **Length** – The length of the data element in bytes.
- **Decimal** – The number of digits included to the right of the decimal point.
- **Enabled** – An indicator of whether the data element modification routines have been enabled (Yes, No).
- **Changer Program** – The name of the modification routine associated with the data element.
- **Assigned By** – The changer program assignment (By Class, By Field).
- **Assigned Class** – Description of the associated class.

Note: A list of available values for the Type column is provided in the following table:

Symbol	Type	Database(s)
1	Bit	MVS file type, ODBC-enabled[*]
A	Alphanumeric	MVS file type
B	Binary	MVS file type
C	Combined	MVS file type
D	RowID	Oracle
E	Edited Field	MVS file type
F	Float	MVS file type, Oracle, ODBC-enabled*
G	Long	Oracle

[*] For the ODBC-enabled database, the contents of the Type column reflect the data name reported by the ODBC driver.

Symbol	Type	Database(s)
H	Char	Oracle, ODBC-enabled*
I	Integer	ODBC-enabled*
K	CLOB	Oracle
L	Date	MVS file type
M	Money	ODBC-enabled*
N	Numeric	MVS file type, ODBC-enabled*
O	BLOB	Oracle
P	Packed	MVS file type
R	Raw	Oracle
S	BigInt	ODBC-enabled*
T	Time	MVS file type
U	Number	Oracle
V	Varchar	Oracle, ODBC-enabled*
W	Varchar2	Oracle
X	Text	ODBC-enabled*
Z	Timestamp	MVS file type, Oracle, ODBC-enabled*
b	VarBinary	ODBC-enabled*
d	Decimal	ODBC-enabled*
e	XML	ODBC-enabled*
f	BFile	Oracle
g	Image	ODBC-enabled*
i	TinyInt	ODBC-enabled*
l	DateTime	ODBC-enabled*
m	SmallMoney	ODBC-enabled*
r	Real	ODBC-enabled*
s	SmallInt	ODBC-enabled*
t	SmallDateTime	ODBC-enabled*
u	UniqueIdentifier	ODBC-enabled*
w	Long Raw	Oracle
x	Text	ODBC-enabled*

*For the ODBC-enabled database, the contents of the Type column reflect the data name reported by the ODBC driver.

Data Changer Data Element Properties

This section describes the general and additional Data Changer properties for a selected data element.

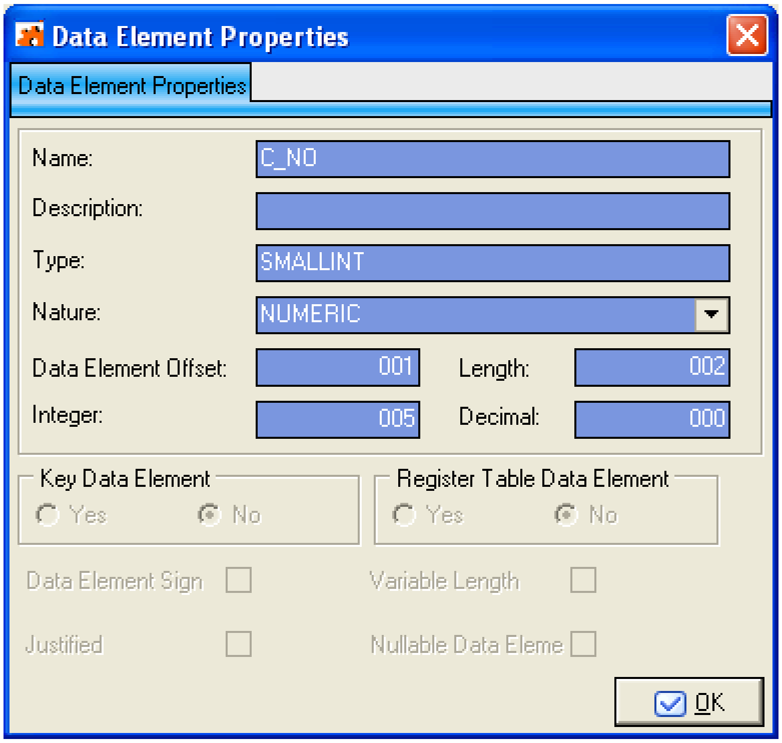
Introduction

Menu commands that enable you to view these properties can be viewed by right-clicking the appropriate class in the grid from the **Data Store Information** tab in the Main Window.

For more information about the Data Store Information tab in the Main Window, see the *Introduction* in the *Data Changer Data Element Selection* section.

General Data Element Properties

General data element properties are displayed in the Data Element Properties window. To display the Data Element Properties window, click **Data Element** properties from the context menu. An example of the Data Element Properties window is shown below:



Name: C_NO

Description:

Type: SMALLINT

Nature: NUMERIC

Data Element Offset: 001 Length: 002

Integer: 005 Decimal: 000

Key Data Element: ○ Yes ● No

Register Table Data Element: ○ Yes ● No

Data Element Sign □ Variable Length □

Justified □ Nullable Data Eleme □

Figure 4: Data Element Properties Window

The fields and items listed in the Data Element Properties window are:

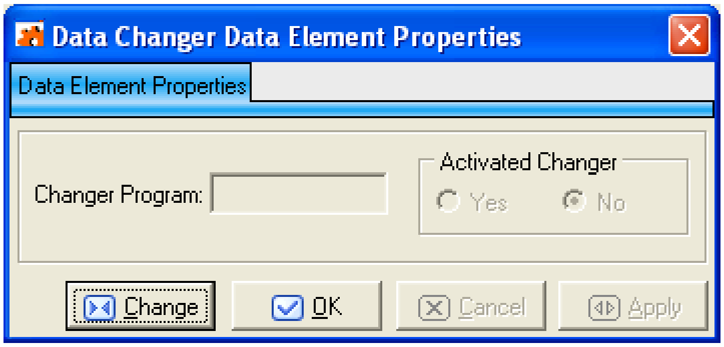
- **Name** – The name of the data element within the data store.
- **Description** – A description of the data store.
- **Type** – The type of field. For example, NUMERIC or ALPHANUMERIC.
- **Nature** – The nature of the data element. For example, ALPHABETICAL.
- **Displacement** – The offset of the data element offset.
- **Length** – The length of the data element in bytes.
- **Integer** – The number of integers within a numeric data element.
- **Decimal** – The number of digits included to the right of the decimal point.
- **Key Field** – An indicator of whether the data element is considered a key data element (Yes, No).
- **Register Table Field** – An indicator of whether the data element is unique in the data store (Yes, No).
- **Changer Program** – The name of the modification routine associated with the data element.

- **Activated changer** – An indicator of whether or not the Changer Program has been activated.
- **Field Sign** – An indicator of whether the data element is considered signed or not signed.
- **Variable Length** – An indicator of whether the length of the data element is variable or not.
- **Justified** – An indicator of whether the data element is justified or not.
- **Nullable Field** – An indicator of whether the data element is NULL or not.

Additional Data Element Properties

Additional properties are displayed in the **Data Changer Data Element Properties** window.

To display the **Data Changer Data Element Properties** window, click **Data Changer Data Element Properties** from the context menu. An example of the **Data Changer Data Element Properties** window is shown below:



Changer Program:

Activated Changer: ○ Yes ● No

Figure 5: Data Changer Data Element Properties Window

Note: If data element properties have not been set previously, the data element properties can be edited; otherwise, the properties can only be viewed.

The fields and buttons listed in the **Data Changer Data Element Properties** window are:

- **Changer Program** - The program that will be used when data will be published.
- **Activated Changer** - Indicates whether the Data Changer is active or not (**Yes**, **No**).
- **Change** - Enables control and goes into edit mode.
- **OK** – Saves any changes that have not been applied and closes the window.
- **Cancel** – Cancels any changes applied to the selected class. Any actions confirmed with the **Apply** button cannot be restored with **Cancel** but must be entered and confirmed again.
- **Apply** - Confirms the changes.

Data Changer - Work with Classes

The **Work with Classes** area enables you to categorize data elements with homogeneous characteristics and to write generalized routines to check data.

Introduction

A class identifies a user-defined group of data elements with homogeneous characteristics. Typical classes of data can be quantities, prices, client code, totals, or currency flag. Each data element can be cataloged as a member of a class.

Classes are grouped into sets called Super Classes. The set of Super Classes is as follows:

- Numbers (N)
- Codes (C)
- Date (D)

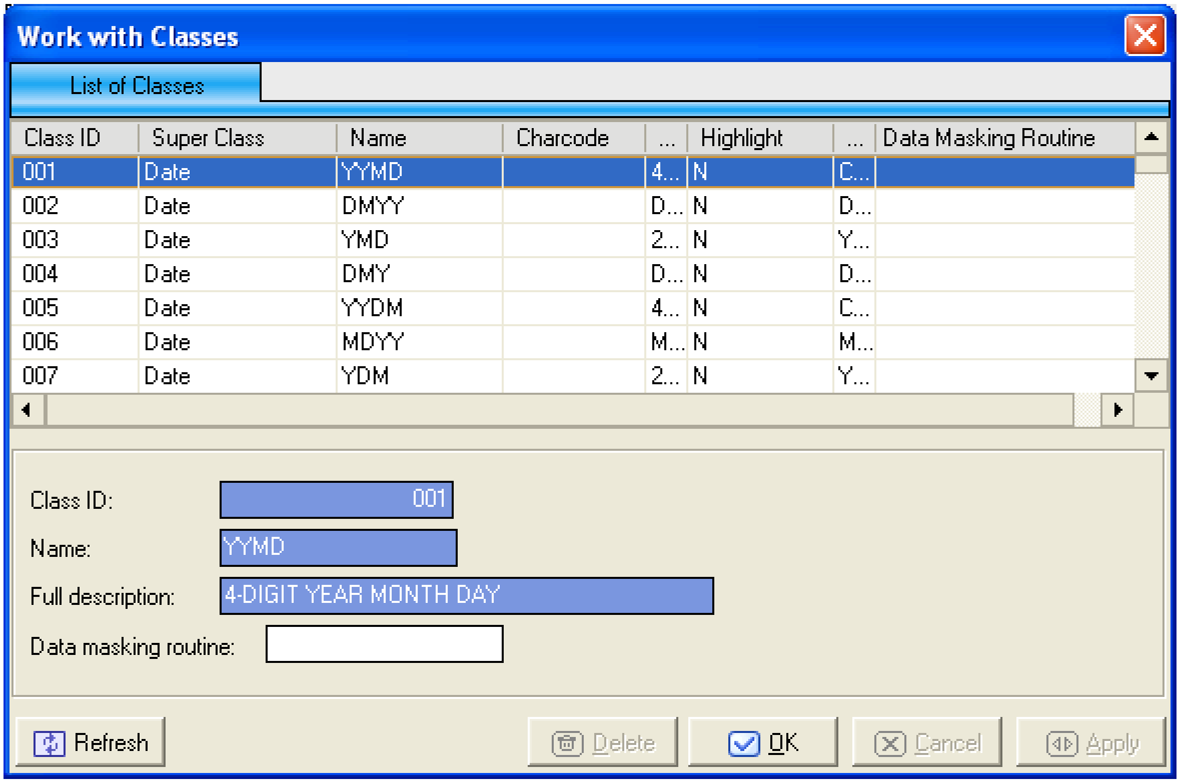
- Complemented Date (9)
- Descriptions (X)
- Privacy (P)
- Other (O)
- None

Note: The Super Class Other is used for new user classes that might not fall into any of the previous Super Classes. Conversely, the Super Class None is used for classes that do not belong to a SuperClass.

To access the **Work with Classes** area, click (Work with Classes) from the Data Masking main window.

Window Contents

This section describes the fields and items listed in the **Work with Classes** area as shown below:



Class ID	Super Class	Name	Charcode	...	Highlight	...	Data Masking Routine
001	Date	YYMD		4...	N	C...	
002	Date	DMYY		D...	N	D...	
003	Date	YMD		2...	N	Y...	
004	Date	DMY		D...	N	D...	
005	Date	YYDM		4...	N	C...	
006	Date	MDYY		M...	N	M...	
007	Date	YDM		2...	N	Y...	

Class ID: 001

Name: YYMD

Full description: 4-DIGIT YEAR MONTH DAY

Data masking routine:

Figure 6: Work with Classes Window

The columns and fields listed in the Work with Classes window are:

- **Class ID** – Identifier of the class.
- **Super Class** – SuperClass that has been associated with the class.
- **Name** – Name of the class.
- **Charcode** – Character identifier of the class.
- **Description** – Brief description of the class. The information in this column matches the information in the Full Description field.
- **Highlight** – Primary class flag.
- **Edit Mask** – Class editing mask
- **Full Description** – Brief description of the class. The information in this field matches the information in the Description column.
- **Data Masking Routine** – Program that will be used when the class is assigned as a changer for a selected data element.

Available Operations

The buttons at the bottom of the **Work with Classes** window are:

- **Refresh** – Refreshes the information listed in the grid.
- **Delete** – Deletes the Changer Program for the selected class.
- **OK** – Saves any changes that have not been applied and closes the window.
- **Cancel** – Cancels any changes applied to the selected class. Any actions confirmed with the Apply button cannot be restored with Cancel but must be entered and confirmed again.
- **Apply** – Confirms the changes made to the Changer Program of the selected class.

Data Changer Properties

The **Data Changer properties** window lets you view Data Changer information for an enabled data store, or lets you define information for a newly enabled data store that has no Data Changer information.

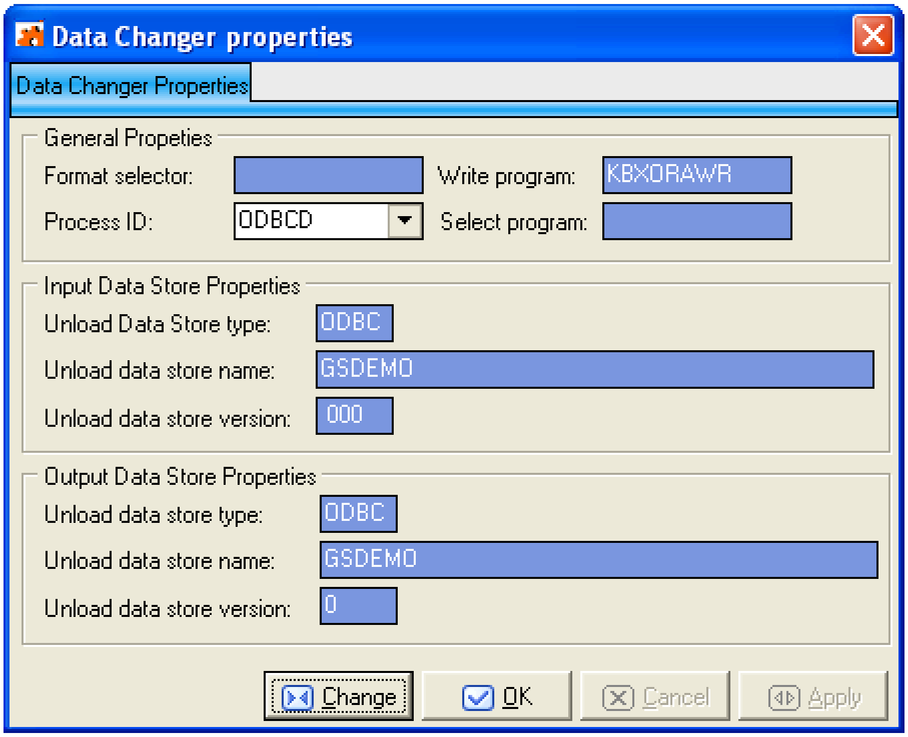
Introduction

To access the Data Changer properties window, right-click the data store in the **Grid** area of the Data Masking main window, and click **Data Changer properties**.

For more information about the context menu, see *Machine ID./Company Grid* in the *Data Masking Main Window* section.

Window Contents

This section describes the fields listed in the Data Changer properties window as shown below:



General Propeties

Format selector:

Write program: KBXORAWR

Process ID: ODBCD

Select program:

Input Data Store Properties

Unload Data Store type: ODBC

Unload data store name: GSDEMO

Unload data store version: 000

Output Data Store Properties

Unload data store type: ODBC

Unload data store name: GSDEMO

Unload data store version: 0

Figure 7: Data Changer properties Window

The fields listed in the Data Changer properties window are:

- **General Properties**
  - **Format selector** – Name of the exit routine for the format recognition.
  - **Write program** – Write program of the Output file.
  - **Process ID** – Identifier of the process.
  - **Select program** – Selection program of the individual record.
- **Input Data Store Properties**
  - **Unload data store type** – Type of the Unload Input data store.
  - **Unload data store name** – Name of the Unload Input data store. This is for the z/OS environment only. It can be different from the original name loaded into the Knowledge Base. In order to do this, the process ID associated with the elaboration should be created with characteristics for an Unload File Access (For example: GENUN).
  - **Unload data store version** – Version of the Unload Input data store.
- **Unload Data Store Properties**
  - **Unload data store type** – Type of the Unload Output data store.
  - **Unload data store name** – Name of the Unload Output data store.
  - **Unload data store version** – Version of the Unload Output data store.

Available operations

The buttons in the **Output Data Store Properties** section of the **Data Changer properties** window are:

- **Change** – Enables control and goes into edit mode.
- **OK** – Saves any changes that have not been applied and closes the window.
- **Cancel** – Cancels any changes. Any actions confirmed with the Apply button cannot be restored with Cancel but must be entered and confirmed again.
- **Apply** – Confirms the changes.

Set Application ID

The Set Application ID window is used to set an application ID to one or more selected data stores.

Introduction

To access the Set Application ID window, right-click the data store in the **Grid** area of the Data Masking main window, and click **Set Application ID**.

For more information about the context menu, see *Machine Id./Company Grid* in the *Data Masking Main Window* section.

Window Contents

This section describes the fields listed in the Set Application ID window as shown below:

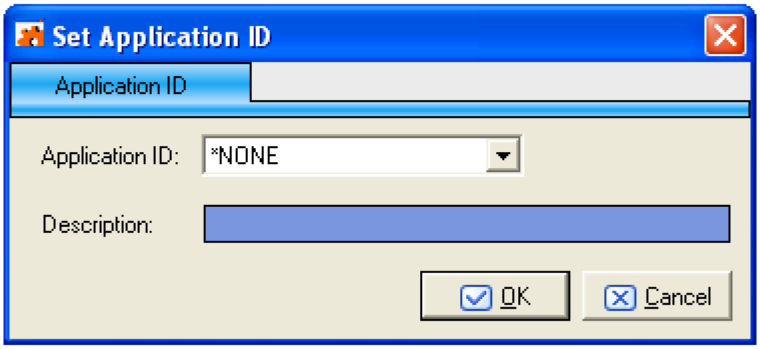


Figure 8: Set Application ID

The fields listed in the Set Application ID window are:

- **Application ID** – Available application identifier. If *NONE is selected, all selected data stores will have no associated application ID.
- **Description** – Brief description of the selected application ID.

Available Operations

The buttons at the bottom of the **Set Application ID** window are:

- **OK** – Sets the selected application to all selected data stores, and closes the window.
- **Cancel** – Cancels any changes. Any actions confirmed with the Apply button cannot be restored with Cancel but must be entered and confirmed again.

Define Unload Output Data Store

The **Define Unload Output Data Store Name** window is used to enable one or more selected data stores with a specified output data store name.

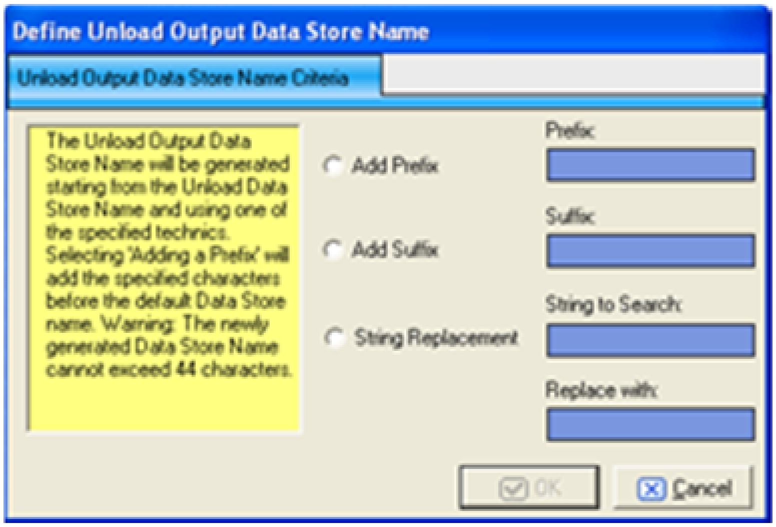
Introduction

To access the Define Unload Output Data Store window:

1. From the Data Masking main window, click the **Disabled** tab.
2. Select one or more data stores you want to enable.
3. Right-click to display the context menu, and then click **Enable**.

Window Contents

This section describes the fields listed in the Define Unload Output Data Store Name window as shown below:


Define Unload Output Data Store Name
Unload Output Data Store Name Criteria

The Unload Output Data Store Name will be generated starting from the Unload Data Store Name and using one of the specified technics. Selecting 'Adding a Prefix' will add the specified characters before the default Data Store name. Warning: The newly generated Data Store Name cannot exceed 44 characters.

Figure 9: Define Unload Output Data Store Name Window

The fields listed in the Define Unload Output Data Store Name window are:

- **Add Prefix** –Activates the Prefix box so that a prefix string can be entered, and then added to the beginning of every data store name that does not have associated Data Changer information.
- **Add Suffix** – Activates the Suffix box so that a suffix string can be entered, and then added to the end of every data store name that does not have associated Data Changer information.
- **String Replacement** – Activates the String to Search and Replace with boxes so that all data store names to do not have associated Data Changer information can be changed to the string listed in the Replace with box.

Available Operations

The buttons at the bottom of the **Define Unload Output Data Store Name** window are:

- **OK** – Enables the selected data stores using the selected criteria, and closes the window.
- **Cancel** – Cancels any changes. Any actions confirmed with the Apply button cannot be restored with Cancel but must be entered and confirmed again.

Options

Data Masking lets you customize the product in order to meet the requirements of different users.

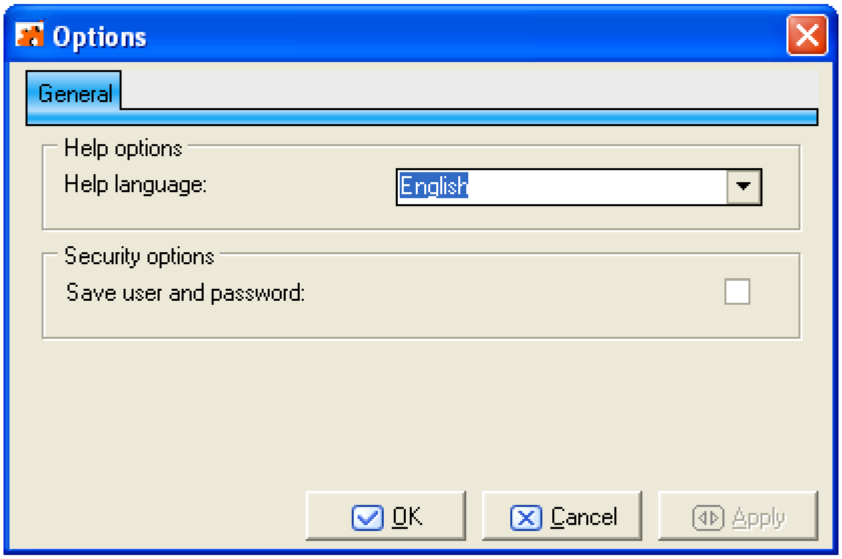
Introduction

This section describes the general and mandatory options in the Data Masking module.

General Options

General options can be edited in the Options window.

To display the Options window, click **Tools > Options**. The Options window is shown below:



Options

General

Help options

Help language: English

Security options

Save user and password: □

Figure 10: Options Window

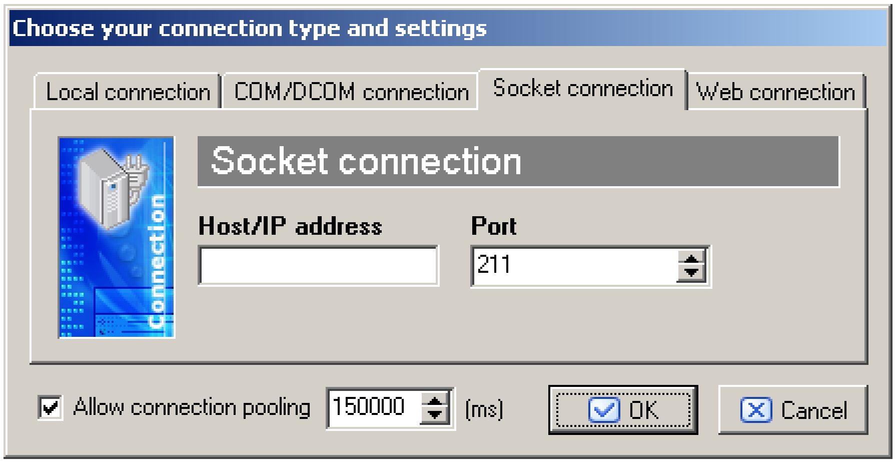
The fields listed in the Output window are:

- **Help language** – Language used for online help. The Help language list contains the languages available on your PC.
- **Save user and password** – Indicator of whether or not you want to save your user ID and password information for the current database connection.

Connection Options

Connection options to the kbde-Server can be edited in the **Choose your connection type and settings** window. Before opening the project with the Data Masking module, you must define the connection information.

To display the **Choose your connection type and settings** window, click **Tools > Connection Options**. An example of the **Choose your connection type and settings** window is shown below:



Choose your connection type and settings

Local connection | COM/DCOM connection | Socket connection | Web connection

Socket connection

Host/IP address:

Port: 211

☑ Allow connection pooling 150000 (ms)

Figure 11: Choose your connection type and settings Window

To define connection settings:

1. Click the tab that corresponds to your connection type, and complete the fields:
   - **Local connection** – Client connection to database located on the local machine.

Note: No additional settings are required for local connections.

- **COM/DCOM connection** – Client connection to the Data Express Application Server as specified on the client machine (`MIDAS.DLL` must be present on the client machine.BDE 5.x and DB2 Client 5.2 or above must be present on the Data Express Application Server machine).
    - **HostName** – The IP address of the NT server.

    Note: If the NT server and local computer are the same, you can leave this field blank. If the NT server and local computer are not the same, they must belong to the same local network.
- **Socket connection** – Client connection using TCP/IP protocol to the Borland Socket Server.
    - **Host/IP address** – IP address of the Borland socket server.
    - **Port** – Port number assigned to the Borland socket server.

    Note: For this connection type, the client and server machines do not need to be part of the same local network.
- **Web connection** – Client connection using HTTP protocol to the ISAPI-compatible web server.
    - **URL** – URL for the .dll file installed on the server machine.

    Note: For this connection type, the client and server machines do not need to be part of the same local network.

2. If desired, check **Allow connection pooling** to enable the specified number of MS as the minimum time interval for the database call to maintain an active connection when the Data Masking module is idle.

Work with Data Changer Jobs

The Work with Jobs area contains information about the function related to the execution of the Data Store Data Changer job (which is a batch job that executes the masking routines), and is part of the Data Builder module. These jobs are submitted using the **Work with Jobs** area of Data Builder.

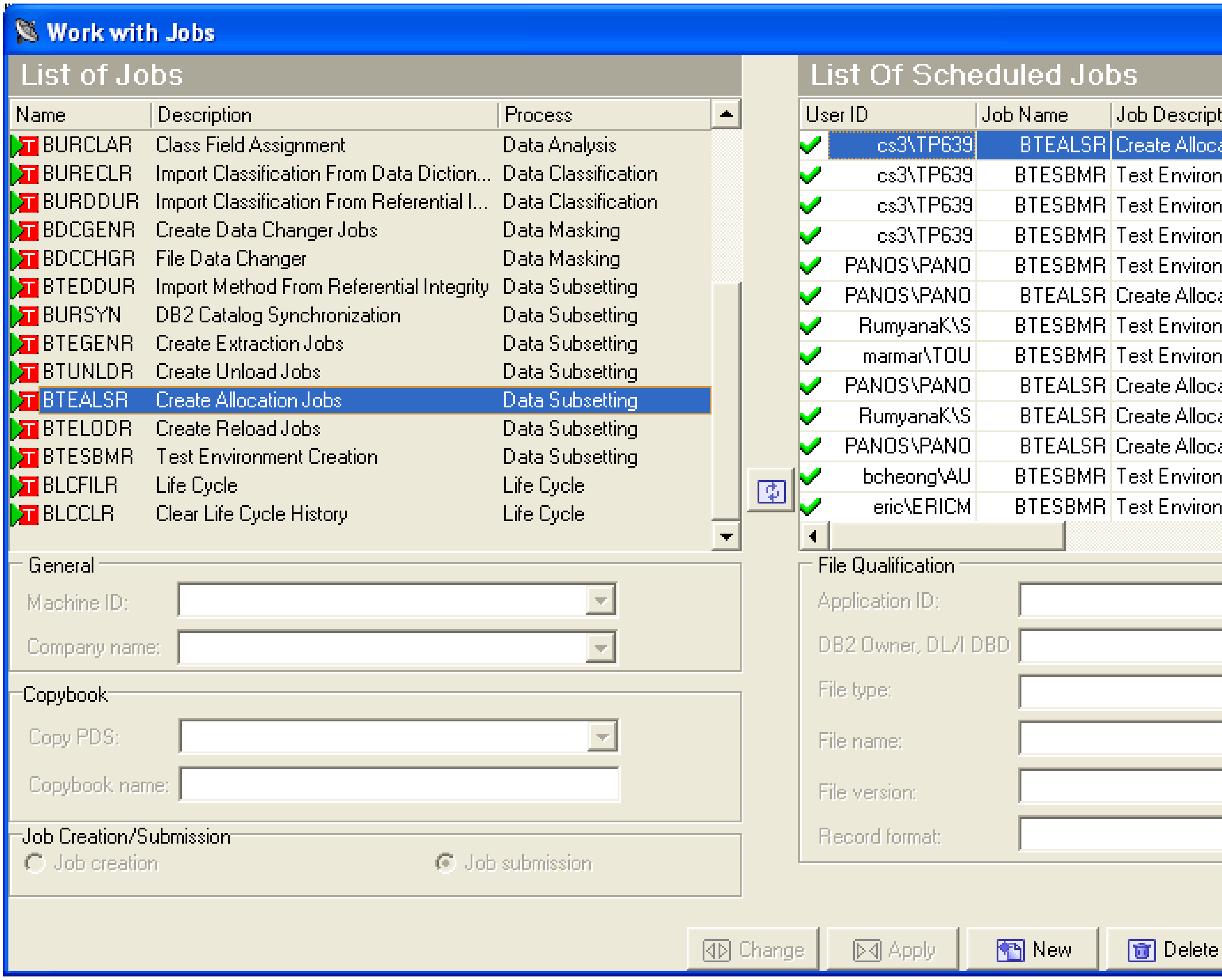
For more information about the **Work with Jobs** area, see *Work with Jobs* in the Front End Guide.

Introduction

To access the Work with Jobs area:

1. Start Data Builder.
2. Click ▶ (Work with Jobs) in the toolbar of the Data Builder main window.

   The **Work with Jobs** area is displayed as shown below:



Name	Description	Process
BURCLAR	Class Field Assignment	Data Analysis
BURECLR	Import Classification From Data Diction...	Data Classification
BURDDUR	Import Classification From Referential I...	Data Classification
BDCGENR	Create Data Changer Jobs	Data Masking
BDCCHGR	File Data Changer	Data Masking
BTEDDUR	Import Method From Referential Integrity	Data Subsetting
BURSYN	DB2 Catalog Synchronization	Data Subsetting
BTEGENR	Create Extraction Jobs	Data Subsetting
BTUNLDR	Create Unload Jobs	Data Subsetting
BTEALSR	Create Allocation Jobs	Data Subsetting
BTELODR	Create Reload Jobs	Data Subsetting
BTESBMR	Test Environment Creation	Data Subsetting
BLCFILR	Life Cycle	Life Cycle
BLCCLR	Clear Life Cycle History	Life Cycle

	User ID	Job Name	Job Descript
✓	cs3\TP639	BTEALSR	Create Alloca
✓	cs3\TP639	BTESBMR	Test Environ
✓	cs3\TP639	BTESBMR	Test Environ
✓	cs3\TP639	BTESBMR	Test Environ
✓	PANOS\PANO	BTESBMR	Test Environ
✓	PANOS\PANO	BTEALSR	Create Alloca
✓	RumyanaK\S	BTESBMR	Test Environ
✓	marmar\TOU	BTESBMR	Test Environ
✓	PANOS\PANO	BTEALSR	Create Alloca
✓	RumyanaK\S	BTEALSR	Create Alloca
✓	PANOS\PANO	BTEALSR	Create Alloca
✓	bcheong\AU	BTESBMR	Test Environ
✓	eric\ERICM	BTESBMR	Test Environ

Figure 12: Work with Jobs Area

Data Store Data Changer Job Creation

The **Data Store Data Changer** job executes a process to create a JCL that will execute data changing associated with a business protection (masking) procedure. The migration function modifies and transfers the content of an input sequential file into an output sequential file.

To create a **Data Store Data Changer** job:

1. From the **Work with Jobs** window, press the **New** button. The **Work with Jobs** window lets you insert parameters for the new job.
2. In the **List of Jobs** section, select **Data Store Data Changer**.
3. In the **General section**, select the appropriate **Machine ID** and **Company name**.
4. In the **Data Store Qualification** section, specify values for the following fields:

   - **Application ID** – The application ID, if applicable.
   - **DB2 Owner, DL/I DBD** – The owner of DB2 or DL/I DBD data store where data will be masked.
   - **Data Store type** – Type of data store. For example, DB2, VSAM, DL/I, GDG, SEQ, ODBC, or ORA.
   - **Data store name** – The name of the data store where data will be masked.
   - **Data store version** – The version number if the data store is GDG.

- **Record format** – Name of the record format.

5. Press **Apply** to submit the job.

Create Data Changer Job Creation

The **Create Data Changer** job executes a process that creates a JCL (the JCL is the base language for batch scripts in MVS, OS/390, or z/OS environment) that will execute the data changer process.

To create a Create Data Changer job:

1. From the **List of Jobs** section in the **Work with Jobs** window, select **Create Data Changer Jobs**.
2. In the **General** section, select the appropriate **Machine ID** and **Company name**.
3. In the **Data Store Qualification** section, specify values for the following fields:
   - **Application ID** – Application ID, if applicable.
   - **DB2 Owner, DL/I DBD** – Owner of DB2 or DL/I DBD data store where data will be masked.
   - **Data store type** – Type of data store (`ADA`, `VSAM`, `DB2`, `DL/I`, `GDG`, `SEQ`, or `ORA`).
   - **Data store name** – Name of the data store where data will be masked.
   - **Data store version** – Version number if the data store is GDG.
   - **Record format** – Name of the record format.
4. Press the **Apply** button. The **Secondary Options** window appears:

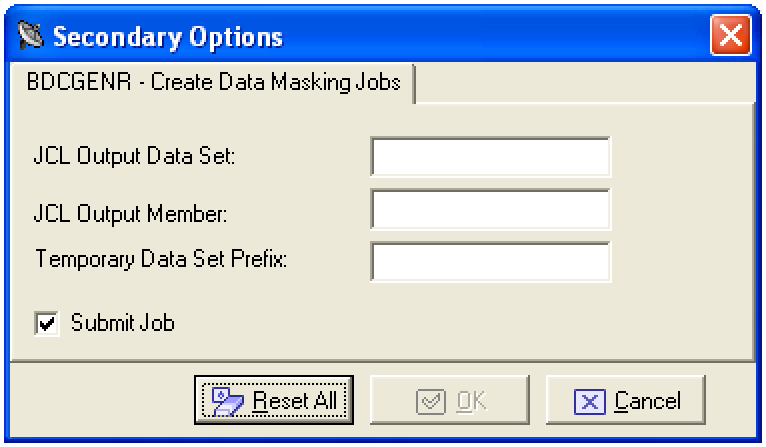


Figure 13: BDCGENR – Create Data Masking Jobs

5. In the **Secondary Options** window, specify values for the following fields:
   - **JCL Output Data Set** – Name of the data set containing the JCL. This field is mandatory.
   - **JCL Output Member** – Name of the member containing the JCL. This field is mandatory.
   - **Temporary Data Set Prefix** – Prefix of the required temporary data sets. This field is mandatory. The prefix must have a minimum of two qualifiers.
   - **Submit Job** – Indicator of whether or not the created JCL will be submitted.

Available Operations

The buttons in the **List of Scheduled Jobs** section of the **Work with Jobs** window and the **Secondary Options** window are:

- **Change** – Lets you modify File Qualification parameters for the selected job if the job is scheduled but has not yet been submitted (Flag Status is 0). Changes should be saved using the Apply button.
- **Apply** – Saves the information for the new job, but keeps the window open.
- **New** – Initializes fields for insertion when scheduling a new job.

- **Delete** – Cancels the execution of the selected scheduled job and removes the job from the List of Scheduled Jobs list. If the job is currently being executed (Flag Status is 1), you will be prompted to confirm the deletion.
- **Cancel** – Cancels any changes. Any confirmations made with the Apply button cannot be undone with the Cancel button but must be instead entered and confirmed again.
- **OK** – Applies all changes and closes the window.
- **Reset All** – Clears the values of all fields.

Distributed Exporter

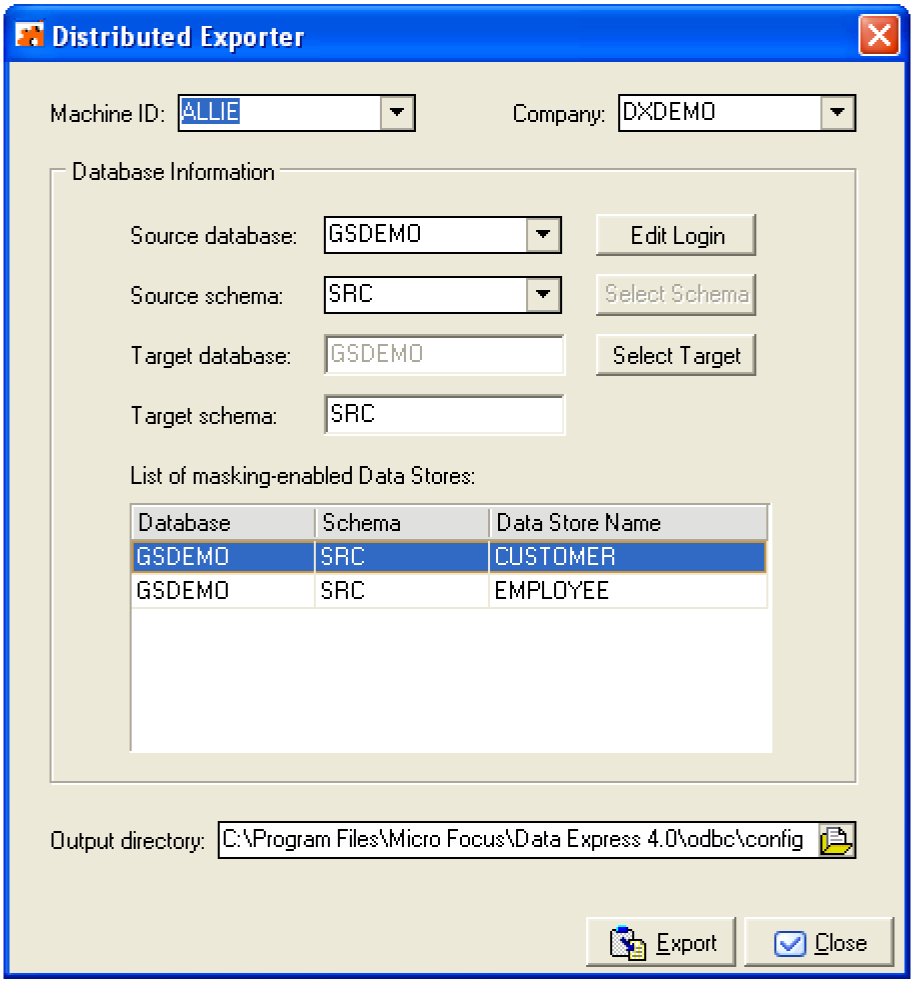
The Distributed Exporter is a utility that creates and provides necessary information to the Extension Technology so that actual masking of your distributed data store can occur. This utility generates the following files that are used for masking after you have successfully exported from the Data Masking module:

- `method.rc` – Coded content of source and target databases information.
- `elab.txt` – List of data stores and the enabled columns within those data stores that need to be masked.
- `CREATETABLE.sql` – list of the tables that need to be masked; a generic CREATE TABLE statement is provided for each table.
- `CREATEINDEX.sql` – list of all the indexes for the tables that need to be masked; a generic CREATE INDEX statement is provided for each index.
- `ALTERTABLE_RI.sql` – list of all the primary and foreign keys for the tables that need to be masked; a generic ALTER TABLE statement is provided for appropriate primary and foreign keys.
- There are two areas of interest when it comes to using the Distributed Exporter with distributed data stores.
- If you use ODBC-enabled data stores, you can mask across all your data stores with one invocation of the ODBC Extension. This feature is not available with the Oracle Extension.
- You also have the ability to mask within one distributed data store, as long as a distinct target schema name is provided.

To access the Distributed Exporter, click from the main window of the Data Masking module.

Window Contents

This section describes the fields listed in the Distributed Exporter window as shown below:



Machine ID: ALLIE

Company: DXDEMO

Database Information

Source database: GSDEMO

Source schema: SRC

Target database: GSDEMO

Target schema: SRC

List of masking-enabled Data Stores:

Database	Schema	Data Store Name
GSDEMO	SRC	CUSTOMER
GSDEMO	SRC	EMPLOYEE

Output directory: C:\Program Files\Micro Focus\Data Express 4.0\odbc\config

Figure 14: Distributed Exporter

To export from the Data Masking module, you need to provide the following input:

- your workspace (machine ID, company)
- your source data store credentials (source database, source schema, source user ID/password)
  - For a data store that you will access using the ODBC Extension, the source database is an ODBC DSN configured to access that data store.
  - For a data store that you will access using the Oracle Extension, the source database is a BDE alias configured to access that data store. For more information, see *Appendix Oracle Alias Creation* in your *Getting Started with Distributed Data Stores* guide.
- your target data store credentials (target database, target scheme, target user ID/password)
  - For a data store that you will access using the ODBC Extension, the target database is an ODBC DSN configured to access that data store.
  - For a data store that you will access using the Oracle Extension, the target database is an Oracle TNS service name configured to access that data store. For more information, see *Appendix Oracle Alias Creation* in your *Getting Started with Distributed Data Stores* guide.
- directory where you want to place output files for Extension Technology (Output directory)

Note: If the source database and the target database are the same, the schemas cannot be the same.

In addition, if you are using ODBC-enabled data stores, you can provide multiple sets of source and target data store credentials.

Note: If you use the Distributed Exporter utility on one machine, and do your masking on another, make sure that the target ODBC DSN matches the ODBC DSN used for the actual masking. Similarly, if you use the Oracle Extension, make sure your Oracle TNS service names match.

Available Operations

The buttons at the bottom of the **Distributed Exporter** window are:

- **Export** – Exports all data store information to the specified output directory.
- **Close** – Closes the window.

Exit Routines

Within Data Masking, a series of exit routines are available to manage file migration. The available programs are prototypes that can be customized for the management of your database.

Note: A user-defined exit routine definition must comply with the linkage parameters associated with the routine type.

Tip: If you modify an exit routine, we recommend that you make copies of the programs and edit the copies rather than the original versions.

There are four main types of exit routines:

- Writing
- Discard
- Field modification
- Logging

Regarding each type of routine, a short outline of the predefined routines, their purposes and the ways to create new routines will be provided.

Writing Exit Routines

A writing exit routine is a program that writes the migrated modified content of a sequential input file to a sequential output file.

The current migration procedure is a program written in C that performs the following tasks:

- Opens the output file.
- Closes the output file.
- Given the modified record layout, writes to the output file.

Writing Routine Parameters

The linkage of the exit routine identified in the product PDS COPY by the kDCWRTW source includes the following fields.

```
01  kDCWRT-REC.
     05  kDCWRT-FLOPER              PIC  X(001).
     05  kDCWRT-LREC                PIC S9(009) COMP.
     05  kDCWRT-DSNAME.
           10  kDCWRT-DSNAME-52        PIC  X(056) VALUE SPACE.
           10                          PIC  X(001) VALUE LOW-VALUE.
     05  kDCWRT-RETCODE             PIC  X(002).
01  kDCWRT-RECORD                   PIC  X(16000).
```

These fields are described in the following table:

Field name	Input/Output	Description
kDCWRT-FLOPER	Input	Flag describing the action to be performed. Possible values: O - indicates that the call has been performed in the File opening mode C - indicates that the call has been performed in the File closing mode W - indicates that the call has been performed in the File writing mode
kDWRT-LREC	Input	Length of the record layout that will be written.
kDCWRT-DSNAME	Input	Name of the output sequential file.
kDCWRT-RETCODE	Output	Return code. If the return code is left BLANK, it means that the routine has been run successfully.
kDCWRT-RECORD	Output	File new record.

Discard Exit Routines

A discard exit routine is a program that skips writing records with certain characteristics during the migration phase.

Discard Routine Parameters

The linkage of the exit routine identified in the product PDS COPY by the UDCUSSLW source includes the following fields:

```
01  UDCUSSL-RECORD                   PIC X(32760).
01  UDCUSSL-RETCODE                  PIC X(02).
```

These fields are described in the following table:

Field name	Input/Output	Description
UDCUSSL-RETCODE	Output	Return code. If the return code is left blank, it means that the routine has been performed successfully.
UDCUSSL-RECORD	Input	File record.

UDCUSSL Discard Exit Routine Example

In the following example of the UDCUSSL discard exit routine, a condition is tested. If this condition is not complied with, the return code is set to “ER”, thus disabling the writing of the record on the output file.

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID.   UDCUSSL.
      ******************************************************************
      *
      *      Source name: UDCUSSL
      *      Function   : DATA CHANGER - USER EXIT FOR RECORD SELECTION
      *
      ******************************************************************
       ENVIRONMENT DIVISION.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
       DATA DIVISION.
       FILE SECTION.
      ****************************************************************
       WORkING-STORAGE SECTION.
```

```
****************************************************************
*
 01  WS-COMODI.
     05  FILLER                 PIC  X.
*
 01  WS-AREA-RECORD.
     05  FILLER                 PIC  X(30).
     05  WS-FIELD-SELECT        PIC  X(10).
     05  FILLER                 PIC  X(82).
****************************************************************
 LINkAGE SECTION.
****************************************************************
 COPY UDCUSSLW.
****************************************************************
 PROCEDURE DIVISION  USING  UDCUSSL-RECORD UDCUSSL-RETCODE.
****************************************************************
*
 START-PGM.
*
     MOVE SPACE                     TO UDCUSSL-RETCODE.
*
     MOVE UDCUSSL-RECORD(1:122)     TO WS-AREA-RECORD.
*
     IF   WS-FIELD-SELECT        NOT = "CBL       "
          MOVE "ER"                 TO UDCUSSL-RETCODE
     END-IF.
*
 END-PGM.
*
     GOBACK.
*
```

Modification Exit Routines

A modification exit routine includes a series of programs aimed at handling the information contained in the individual fields.

Predefined Modification Exit Routines List

Note: Unless otherwise stated, all routines are available for use with all data store types.

A list of predefined modification exit routines are described in the following table:

Routine Name	Description
BPMAADG	Sample routine for masking company name or surname according to data element content.
BPMNIC1	Sample routine for changing dates.
BPMNICC	Generic alphanumeric code (can be used for telephone numbers). This routine changes numbers only, leaving all other characters untouched.
UDCADDC	Address masking routine.
UDCBLKC	Routine initializing the field with which it is associated. You can use it when you do not want some information to be displayed in the migrated data store.
UDCBPMK	Sample for masking complex codes.
UDCCDFC	Italian taxpayer code number masking routine.

Routine Name	Description
UDCCFPI	Italian VAT number or taxpayer's code number masking routine. The routine associated with the mixed field containing both the VAT number and the taxpayer code number recognizes the content of the field and applies either one or the other routine.
UDCCHLC	Chilean tax code masking routine.
UDCCIFC	CIF (Spanish tax codes) masking routine.
UDCCOMC	Company name masking routine.
UDCCDCC	Credit card number masking routine.
UDCCDF1	Sample routine for Italian tax code that considers date change.
UDCCFP1	Sample routine for Italian tax code and VAT number that considers date change.
UDCEMAC	Email address masking routine.
UDCIBNC	International bank account number (IBAN).
UDCLOB1	Template routine LOB/BLOB/CLOB columns containing positional data.
UDCNACC	Advanced name masking routine that preserves gender, and guarantees more distinct masked values than the standard name masking routine.
UDCNAMC	People's names masking routines to be used to mask personal detail files.
UDCNIFC	NIF (Spanish tax codes) masking routine.
UDCNUMC	Account number masking routine.
UDCPIVC	Italian VAT number masking routine.
UDCRANAG	Sample for forcing a fixed value in output.
UDCSNC	Surname and name masking routine.
UDCSNC2	Surname and name masking routine specific to the Italian language.
UDCSNR2C	Surname and name or company name (in the same field) masking routine specific to the Italian language.
UDCSUCC	Advanced surname masking routine that guarantees more distinct masked values than the standard surname masking routine.
UDCSURC	Surname masking routine.
UDCTELC	Telephone number masking routine.
UDCUCST	String masking routine. It inverts the characters and then transcodes them according to a transcoding table.
UDCUSSN	Masking routine for USA Social Security Number.
UDCXML0	Template routine for XML or LOB/BLOB/CLOB columns not containing data to be masked.
UDCXML1	Template routine for XML or LOB/BLOB/CLOB columns containing XML data.

Modification Exit Routines Parameters

The linkage of the exit-routine identified in the product PDS COPY by the UDCUSFLW source includes the following fields:

```
01  UR-ACTION                             PIC  X(04).
01  UR-FILE-INFO.
    03 UR-FILE-NAME                     PIC  X(128).
    03 UR-FILE-TYPE                       PIC  X(04).
01  UR-RECORD                             PIC  X(32760).
01  UR-FIELD.
    03 UR-INP-VAL                            PIC  X(256).
    03 UR-POS                                     PIC S9(05) COMP.
    03 UR-LEN                                     PIC S9(05) COMP.
    03 UR-TYPE                                   PIC  X(01).
    03 UR-INT                                      PIC S9(03) COMP.
    03 UR-DEC                                     PIC S9(03) COMP.
    03 UR-SIGN                                   PIC  X(01).
    03 UR-NULL                                  PIC  X(01).
    03 UR-VARCHAR                         PIC  X(01).
    03 UR-INULL-VAL                       PIC  X(01).
       88 UR-VAL-NULL     VALUE  X"6F".
    03 UR-OTHER-FIELDS-NBR       PIC S9(04) COMP.
01  UR-OTHER-FIELDS.
    03 OCCURS 50.
       05 UR-P-FILE-NAME                PIC  X(128).
       05 UR-P-FILE-TYPE                  PIC  X(04).
       05 UR-P-INP-VAL                      PIC  X(256).
       05 UR-P-POS                               PIC S9(05) COMP.
       05 UR-P-LEN                               PIC S9(05) COMP.
       05 UR-P-TYPE                             PIC  X(01).
       05 UR-P-INT                                PIC S9(03) COMP.
       05 UR-P-DEC                               PIC S9(03) COMP.
       05 UR-P-SIGN                              PIC  X(01).
       05 UR-P-NULL                             PIC  X(01).
       05 UR-P-VARCHAR                    PIC  X(01).
       05 UR-P-INULL-VAL                  PIC  X(01).
          88 UR-P-VAL-NULL    VALUE  X"6F".
01 UR-OUTPUT.
    03 UR-OUT-VAL                             PIC  X(256).
    03 UR-OUT-FLDTYPE                    PIC  X(01).
    03 UR-OUT-FLDLEN                      PIC S9(03) COMP.
    03 UR-OUT-FLDINT                       PIC S9(03) COMP.
    03 UR-OUT-FLDDEC                      PIC S9(03) COMP.
    03 UR-OUT-FLDSIGN                     PIC  X(01).
    03 UR-RETCODE                            PIC  X(02).
    03 UR-MSG-CODE                         PIC  X(07).
    03 UR-MSG-PARM                         PIC  X(128).
    03 UR-OUT-TYPE                           PIC  X(01).
    03 UR-DESCR                                  PIC  X(80).
```

These fields are described in the following table:

Field name	Input/Output	Description
`UR-ACTION`	Input	Flag describing the action to be performed. Possible values: `INIT` - (not used with this product but included for future use) indicates that the call processes a record read from an original file. LAST - indicates that the call has been made at the end of the record processing operation. It is

Field name	Input/Output	Description
		only available to routines customized by the user. AUTH - (not used with this product but included for future use) indicates that the call returns a string containing the author of the exit routine in UR-DESCR VERS - (not used with this product but included for future use) indicates that the call returns a string containing the version of the exit routine in UR-DESCR INFO - (not used with this product but included for future use) indicates that the call returns a string containing the title of the exit routine in UR-DESCR DESC - (not used with this product but included for future use) indicates that the call returns, in UR-DESCR, a row in a table containing the description of the exit routine INPT - (not used with this product but included for future use) indicates that the call returns, in UR-DESCR, a row in a table containing: string 1 (to indicate that the user exit has only one input parameter), the type and the length of the input field OUTP - (not used with this product but included for future use) indicates that the call returns, in UR-DESCR, a row in a table containing: the type and the length of the output field
UR-FILE-NAME	Input	Not used
UR-FILE-TYPE	Input	Not used
UR-RECORD	Input	File Record
UR-INP-VAL	Input	Value contained in the field to be migrated.
UR-POS	Input	Position of the field to be migrated inside the record.
UR-LEN	Input	Length of the field to be migrated.
UR-TYPE	Input	Type of the field to be migrated.
UR-INT	Input	Number of integers (for numeric fields only).
UR-DEC	Input	Number of decimals (for numeric fields only).
UR-SIGN	Input	Sign of the field to be migrated.
UR-NULL	Input	Content of any null byte (MVS only).
UR-VARCHAR	Input	Length of the VARCHAR field (MVS only).
UR-INULL-VAL	Input	Byte value possibly containing the null value (MVS only).
UR-OTHER-FIELDS-NBR	Input	Not used

Field name	Input/Output	Description
`UR-OTHER-FIELDS`	Input	Source encoding type in its UR-P-FILE-NAME field
`UR-OUT-VAL`	Output	Value contained in the migrated field
`UR-OUT-FLDTYPE`	Output	Not used
`UR-OUT-FLDLEN`	Output	Length of the masked value.
`UR-OUT-FLDINT`	Output	Not used
`UR-OUT-FLDDEC`	Output	Not used
`UR-OUT-FLDSIGN`	Output	Not used
`UR-RETCODE`	Output	Return code (Space = OK)
`UR-MSG-CODE`	Output	Not used
`UR-MSG-PARM`	Output	Not used
`UR-OUT-TYPE`	Output	Flag describing the location of the processed value. Possible flags: `R` - The change worked directly on `UR-RECORD F` - The output value is `in UR-OUT-VAL`
`UR-DESCR`	Output	(Not used now, but included for future use.) It contains additional information about the exit routine.

Logging Exit Routines

A logging exit routine includes two programs aimed at managing the errors found during the masking process.

Predefined Masking and Logging Exit Routines List

Note: Unless otherwise stated, all routines are available for use with all data store types.

A list of predefined logging exit routines is described in the following table:

Routine Name	Description
UDCMASPG	Generic routine for masking IBAN or Credit Card (MVS only).
UDCLOGPG	Manage logging messages.

Masking Routine UDCMASPG - Prototype

The Data Express routine UDCMASPG can now be used to change an existing customer’s masking routines. This routine contains new rules to follow in case of an error. Please see the following steps:

- Move “ER” to linkage field UR-RETCODE.
- Assign different values to field UR-MSG-CODE depending on the values used to populate the table HSLOGTAB.
- Move the routine's name to field UR-MSG-PARM.

All of the information will be used in order to access HSLOGTAB and if there is a logging routine to be called, it will be called.

Masking Routine UDCLOGPG – Prototype

The Data Express routine UDCLOGPG contains an example of managing logging errors. The routine produces an output file managing the masking routine, name that finished in error, the type of error, the related descriptions, and the data store name.

The following example shows a row produced by the routine:

```
UDCMASPG FUNCTION ERROR THE STRING DOES NOT CONTAIN A VALID VALUE THE INPUT
VALUE:FR121234512345123456789 0112 TABLE:LOGGING
```

Note: In order to correctly manage the prototype routine provided (UDCLOGPG), it is advisable to add into JCLs BDCCHGR and BTESBMR the previous JCL card in order to work correctly with the Logging file produced by that routine, according to the way you choose to develop your Logging routine.

The JCL's card showed below is only an example to follow in case you decide to write an output file to log the masking's errors as is performed into UDCLOGOPG:

```
//LOGSRC DD DISP=(NEW,CATLG,CATLG),
//          DSN=LOGGING OTUPUT FILE NAME,
//          DCB=(RECFM=FB,LRECL=327),
//          SPACE=(CYL,(10,10))
```

Logging Table HSLOGTAB

This table contains the information to call the logging routine depending on the Msgcode used in the masking routine.

For more information about this table, see the *Table and column description* in the *Data Model Guide*.

In order to populate this table, please follow the example below:

```
INSERT INTO tableowner.HSLOGTAB (MSGCODE, FLAGENVI, ROUTNAME, VALRCODE)
VALUES ('LCI0001', 'Y', 'UDCLOGPG', '00');
```

The population of the table is very important since it allows associating one different logging routine for each MSGCODE or associating the same logging routine for all MSGCODE inserted into the table.

These differences provide choices between creating and managing of one routine and creating and managing several routines

Appendix A: COBOL Masking Routines Technical Description

This appendix provides a technical description of how masking routines work; particular detail is given to the meaning of the fields within the Data Masking interface.

A.1. Masking Routines Interface

All masking routines have the same interface. This interface is described, in the product, with a COBOL copybook named `UTEUSFLW`.

A.2. Interface Source

A.2.1. Data Express for z/OS

```

      01 UR-ACTION                    PIC  X(32760).
      01 UR-FILE-INFO.
         03 UR-FILE-NAME             PIC  X(44).
         03 UR-FILE-TYPE             PIC  X(04).

      01 UR-RECORD                   PIC  X(32760).

      01 UR-FIELD.
          03 UR-INP-VAL               PIC  X(256).
          03 UR-POS                   PIC S9(05) COMP.
          03 UR-LEN                   PIC S9(05) COMP.
          03 UR-TYPE                  PIC  X(01).
          03 UR-INT                   PIC S9(03) COMP.
          03 UR-DEC                   PIC S9(03) COMP.
          03 UR-SIGN                  PIC  X(01).
          03 UR-NULL                  PIC  X(01).
          03 UR-VARCHAR               PIC  X(01).
          03 UR-INULL-VAL             PIC  X(01).
             88 UR-VAL-NULL           VALUE  X"6F".
          03 UR-OTHER-FIELDS-NBR      PIC S9(04) COMP.

      01 UR-OTHER-FIELDS.
          03 OCCURS 50.
             05 UR-P-FILE-NAME        PIC  X(44).
             05 UR-P-FILE-TYPE        PIC  X(04).
             05 UR-P-INP-VAL          PIC  X(256).
             05 UR-P-POS              PIC S9(05) COMP.
             05 UR-P-LEN              PIC S9(05) COMP.
             05 UR-P-TYPE             PIC  X(01).
             05 UR-P-INT              PIC S9(03) COMP.
             05 UR-P-DEC              PIC S9(03) COMP.
             05 UR-P-SIGN             PIC  X(01).
             05 UR-P-NULL             PIC  X(01).
             05 UR-P-VARCHAR          PIC  X(01).
             05 UR-P-INULL-VAL        PIC  X(01).
                88 UR-P-VAL-NULL      VALUE  X"6F".

      01 UR-OUTPUT.
          03 UR-OUT-VAL               PIC  X(256).
          03 UR-OUT-FLDTYPE           PIC  X(01).
          03 UR-OUT-FLDLEN            PIC S9(03) COMP.
          03 UR-OUT-FLDINT            PIC S9(03) COMP.
          03 UR-OUT-FLDDEC            PIC S9(03) COMP.
          03 UR-OUT-FLDSIGN           PIC  X(01).
     *
          03 UR-RETCODE               PIC  X(02).
     *
          03 UR-MSG-CODE              PIC  X(07).
          03 UR-MSG-PARM              PIC  X(128).
          03 UR-OUT-TYPE              PIC  X(01).
     *
          03 UR-DESCR                 PIC  X(80).
```

A.2.2. Data Express for Distributed Systems

```
01  UR-ACTION                   PIC  X(04).

       01  UR-FILE-INFO.
           03 UR-FILE-NAME             PIC  X(44).
           03 UR-FILE-TYPE             PIC  X(04).

       01  UR-RECORD                   PIC  X(32760).

       01  UR-FIELD.
           03 UR-INP-VAL               PIC  X(256).
           03 UR-POS                   PIC  S9(9) COMP-5.
           03 UR-LEN                   PIC  S9(9) COMP-5.
           03 UR-TYPE                  PIC  X(01).
           03 UR-INT                   PIC  S9(9) COMP-5.
           03 UR-DEC                   PIC  S9(9) COMP-5.
           03 UR-SIGN                  PIC  X(01).
           03 UR-NULL                  PIC  X(01).
           03 UR-VARCHAR               PIC  X(01).
           03 UR-INULL-VAL             PIC  X(01).
              88 UR-VAL-NULL           VALUE  X"6F".
           03 UR-OTHER-FIELDS-NBR      PIC  S9(4) COMP-5.

       01  UR-OTHER-FIELDS.
           03 OCCURS 50.
              05 UR-P-FILE-NAME        PIC  X(44).
              05 UR-P-FILE-TYPE        PIC  X(04).
              05 UR-P-INP-VAL          PIC  X(256).
              05 UR-P-POS              PIC  S9(9) COMP-5.
              05 UR-P-LEN              PIC  S9(9) COMP-5.
              05 UR-P-TYPE             PIC  X(01).
              05 UR-P-INT              PIC  S9(9) COMP-5.
              05 UR-P-DEC              PIC  S9(9) COMP-5.
              05 UR-P-SIGN             PIC  X(01).
              05 UR-P-NULL             PIC  X(01).
              05 UR-P-VARCHAR          PIC  X(01).
              05 UR-P-INULL-VAL        PIC  X(01).
                 88 UR-P-VAL-NULL      VALUE  X"6F".

       01 UR-OUTPUT.
           03 UR-OUT-VAL               PIC  X(256).
           03 UR-OUT-FLDTYPE           PIC  X(01).
           03 UR-OUT-FLDLEN            PIC  S9(9) COMP-5.
           03 UR-OUT-FLDINT            PIC  S9(9) COMP-5.
           03 UR-OUT-FLDDEC            PIC  S9(9) COMP-5.
           03 UR-OUT-FLDSIGN           PIC  X(01).
      *
           03 UR-RETCODE               PIC  X(02).
      *
           03 UR-MSG-CODE              PIC  X(08).
           03 UR-MSG-PARM              PIC  X(128).
           03 UR-OUT-TYPE              PIC  X(01).
      *
           03 UR-DESCR                 PIC  X(80).
```

A.3. Interface Description

Field	Type	Meaning
`UR-ACTION`	Input	Type of operation (see *Values of UR-ACTION* table).
`UR-FILE-NAME`	Unused	For future use.
`UR-RECORD`	Input-Output	The first 32,760 bytes (left-aligned and padded with spaces) of the read record, in the same format as a sequential unload.
`UR-INP-VAL`	Input	The field to be masked.
`UR-POS`	Input	The position of the field to be masked, in the file layout.
`UR-LEN`	Input	The length of the field to be masked.

Field	Type	Meaning
UR-TYPE	Input	The type of the field to be masked. Options are: A - Alphanumeric P - Packed N - Numeric zoned B - Binary
UR-INT	Input	The integer number of the field to be masked.
UR-DEC	Input	The decimal number of the field to be masked.
UR-SIGN	Input	The sign of the field to be masked. Options are: Y - Signed N - Unsigned space - Non-numeric field
UR-NULL	Input	The null-capability of the field to be masked. Options are: Y - Null capable space - Not null capable
UR-VARCHAR	Input	The variable length property of the field to be masked. Options are: Y - Variable length space - Not variable length
UR-INULL-VAL	Input	If a column accepts NULL values and it is actually NULL, then the UR-INULL-VAL field is set to "o" (6f), if it is different than NULL, the UR-INULL-VAL is blank " ". The "?" sign means that the column is NULLABLE and the value is NULL. If a column accepts NULL values, then after it is set in the UR-RECORD, we add a byte with the "?".
UR-OTHER-FIELDS-NBR	Unused	For future use.
UR-OTHER-FIELDS	Unused	Source encoding type in its UR-P-FILE-NAME field
UR-OUT-VAL	Output	The value of the masked field, if UR-OUT-TYPE is equal to F .
UR-OUT-FLDTYPE	Unused	For future use.
UR-OUT-FLDLEN	Output	Output length of the masked value. Data Express for z/OS initializes this field value to the actual length of UR-INP-VAL, but you can explicitly set the value for UR-OUT-FLDLEN.
UR-FLDINT	Unused	For future use.
UR-FLDDEC	Unused	For future use.
UR-FLDSIGN	Unused	For future use.
UR-RETCODE	Output	Routine return code. Options are: space character(s) -masking routine works as expected. 01 or 02 - a problem was encountered and the JCL return code is set to 8. If the UR-RETCODE value is 01 or 02 and UR-OUT-TYPE is equal to F, the field is not masked.

Field	Type	Meaning
`UR-MSG-CODE`	Unused	For future use.
`UR-MSG-PARM`	Unused	For future use.
`UR-OUT-TYPE`	Output	Flag describing the location of the processed value. Possible flags: `F` - The output value is in `UR-OUT-VAL` `R` - The change worked directly on `UR-RECORD`
`UR-DESCR`	Unused	For future use.

A.4. UR-ACTION Values

Value(s)	Meaning
`INIT`	The routine is called with this value at the beginning of each file elaboration. The code executed in this case can be an initialization code. The predefined routines set the return code to a space character.
`EXEC`	The routine is called with this value for each read record. This is the masking routine.
`LAST`	The routine is called with this value at the end of each file elaboration. The code executed in this case can be a final elaboration code. The predefined routines do not use this value because no final elaboration is needed.
`AUTH` , `VERS` , `INFO` , `DESC` , `INPT` , `OUTPT`	The routine is never called for these values of the parameter. The code included in the predefined routines is written for documentation purposes. We suggest you write similar code for customized routines.

Appendix B: C Masking Routines for ODBC - Technical Description

This appendix provides a technical description of how C masking routines for ODBC work; particular detail is given to the meaning of the fields within the Data Masking interface.

For ODBC extension, Data Express provides two types of masking routines: standard masking and extended masking. Standard masking routines have limited capabilities as opposed to extended masking routines, meaning that they mask only one value, are not based on dependencies, and use only the source value and its length. Also, when two or more source values are related, the masking of one could depend on the other.

Note: This limitation does not apply to the Oracle extension because its masking routines exist as part of a working chain rather than within a DLL. Hence, the routine itself has information about all columns.

The extended masking routines, provided in Data Express starting with Update 17, have the same interface as the COBOL masking routine.

Interface Details

The interface for the extended C masking routines allows you to decide exactly what kind of masking to perform, based on additional parameters including:

UR-RECORD String-based variable that contains all record values.

Source value parameters A structure, containing all metadata of the source value like type, length, int, dec, position (the starting position in the UR-RECORD), including the value itself.

Output values structure A structure that stores the output value and the masking type, which can be either by a single masked value or extracted from the UR-RECORD.

For complete information, see *Masking Routines* in *Getting Started with Distributed Data Stores*

Examples

The following examples show the complexity of extended masking as opposed to standard masking:

Standard Masking

```
char* UDCNAMC(char *szValue, unsigned int *nLength)
```

Extended Masking

```
char * UDCNAMC(char *action, char *fileinfo, char *record,
   struct fldstruct *chgparams_fstruct, struct fldodata
*chgparams_odata,
   struct fldoutput *chgparams_out)
```

action `INIT`, `EXEC`, or `LAST`. For more information, see *Modification Exit Routines Parameters* in your *Data Masking Guide*.

fileinfo Used only in z/OS.

record UR-RECORD

fldstruct The structure containing the input source information.

```
struct fldstruct
    {
        unsigned char f_inpval[UR_INP_VAL_LEN];
        int f_posn;
        int f_len;
        unsigned char f_fldtype;
        short f_inttype;
        short f_dectype;
        unsigned char f_sign;
        unsigned char f_null;
        unsigned char f_varchar;
        unsigned char f_inull;
        short f_otherfld;
    } ;
```

f_inpval	The source value.
f_posn	Its position in the UR-RECORD.
f_len	The column length.
f_fldtype	The column type.
f_inttype	`int` type.
f_dectype	`dec` type.
f_sign	Whether value is signed, or unsigned.
f_null	Whether column is nullable, or not.
f_varchar	Whether varchar is signed, or unsigned.
f_inull	Whether the value is NULL, or not.

f_otherfld Used only in z/OS.

fldodata Unused structure in the distributed engine.

```
struct fldodata
    {
        unsigned char oth_filename[UR_FILE_NAME_LEN];
        unsigned char oth_filetype[4];
        unsigned char oth_inpval[UR_INP_VAL_LEN];
        short oth_posn; /*short*/
        short oth_len; /*short*/
        unsigned char oth_fldtype;
        short oth_inttype; /*short*/
        short oth_dectype; /*short*/
        unsigned char oth_sign;
        unsigned char oth_null;
        unsigned char oth_varchar;
        unsigned char oth_inull;
        short oth_otherfld;
    } ;
```

fldoutput Output structure that is being returned to the DE engine.

```
struct fldoutput
    {
        char out_outval[UR_INP_VAL_LEN];
        char out_fldtype;
        short out_fldlen; /*short*/
        short out_inttype; /*short*/
        short out_dectype; /*short*/
        char out_fldsign;
        char out_rtncode[2];
        char out_msgcode[7]; /*7*/
        char out_msgparm[UR_OUT_MSGPARM_LEN];
        char out_type;
        char out_descr[80];
    };
```

out_outval The masked value.

out_fldtype The output column type.

out_fldlen The output column length.

out_inttype The output `int` type.

out_dectype The output `dec` type.

out_fldsign The output value sign.

out_rtncode The return code of the masking routine. If the masking is successful, the return code is " " (space).

out_msgcode Used only in z/OS

out_msgparm Used only in z/OS

out_type Identifies the output type. One of:

R	The out_outval value is ignored and what is in the UR-RECORD is inserted in the target environment.

F	The masked value used is the one stored in out_outval.

out_desrc Used only in z/OS

```
*
#define UR_INP_VAL_LEN    256
#define UR_RECORD_LEN    32760
#define UR_FILE_NAME_LEN    128
#define UR_OUT_MSGPARM_LEN 128
```

Appendix C: C Masking Routines for OCI - Technical Description

This appendix provides a technical description of how C masking routines for OCI work.

Oracle OCI extension of Data Express works as a chain of processes that passes information about each targeted column, such as value, type, length, precision, scale and so on. In contrast to ODBC masking routines, C masking routines for OCI are built as separate processes, and are not included in a DLL. Thus, each masking process (masking routine) becomes a part of the main Data Express processing chain, and can use all the information it passes.

Functional Analysis C masking routines provide additional functionality over ODBC masking routines in that each routine can:

- Access and use all of the data related to a column
- Modify data values on the fly
- Pass all changed data back to Data Express for insertion into the target

Technical Analysis The Data Express OCI extension uses predefined data structures that hold column information. It also uses functions that read and write the metadata passed to and from each part of the processing chain. We also provide these functions in a library file so that you can use them in your own written routines just as Data Express uses them.