Database Management API

This Database Management API can be used by anyone that wants to develop an interface to add/edit/delete data in a WinDSX or WinDSX SQL system. It can be used to place the initial data into the system and it can be used as a real time interface for ongoing data manipulation.

The API is designed to insulate other systems from future changes in the DSX database structure by providing a consistent interface method regardless of changes made to the DSX software. As the WinDSX database schema changes your interface stays the same.

Building this interface requires teaching the other system to save the data into a text file formatted to the DSX Markup Language (DML) specification. In this document there will be clear examples of adding a person to the system, giving that person a card, changing and deleting cards, assigning access levels and more.

Below are the Four main points of Interest when developing this API. The rest of the document outlines the DSX Markup Language and it special commands accompanied with many examples covering different situations.

1. When trying to coordinate card holder data between 2 separate systems there must be a unique identifier for each person that both systems can agree on. The unique ID (UID) could be anything that is guaranteed to be unique, such as an employee ID number. This UID will be the key that both systems use to identify the card holder records. Further manipulation of the card holder information will require the use of this number. The UID is stored in a User Defined Field (UDF) in the DSX system. User Defined Fields are used in the DSX software to store attributes of a person.

2. The Card holder information including the "Identifying Number" is written in a text file according to the DSX Markup Language Specifications and placed into the WinDSX shared database folder. Multiple records could be included in a single DML text file. It is acceptable to create a file that adds 5 people, edits two cards, and deletes 3 people. The best method is to create the text file locally and then copy it into the shared directory. Each file can have an incrementing number in the file name so that many files can be placed in the "Shared Folder" at the same time. Once the file is placed in the shared folder Do Not Append data to the file! Place new or additional data into a new file. Once the data has been read from the file by the DSX system the file is deleted. For example file names such as the following could be used: ^Imp01.txt, ^Imp02.txt, ^Imp03.txt

3. The Comm Server PC typically processes the DML text files. The processing of the files on the comm server PC is intentionally slowed down so that it does not interfere with the comm servers other duties. For most systems the performance is acceptable. Systems that will be repeatedly sending a large number of records in a single file can specify a DSX Workstation on the system to process the DML text files at a much higher rate. Refer to the Daily Ops page at the end of this document.

4. There are specific DSX database fields used in the Interface. The names and numbers of these fields must be coordinated between the two systems. The Coordinated Fields section of this document deals with where and how to find this information.
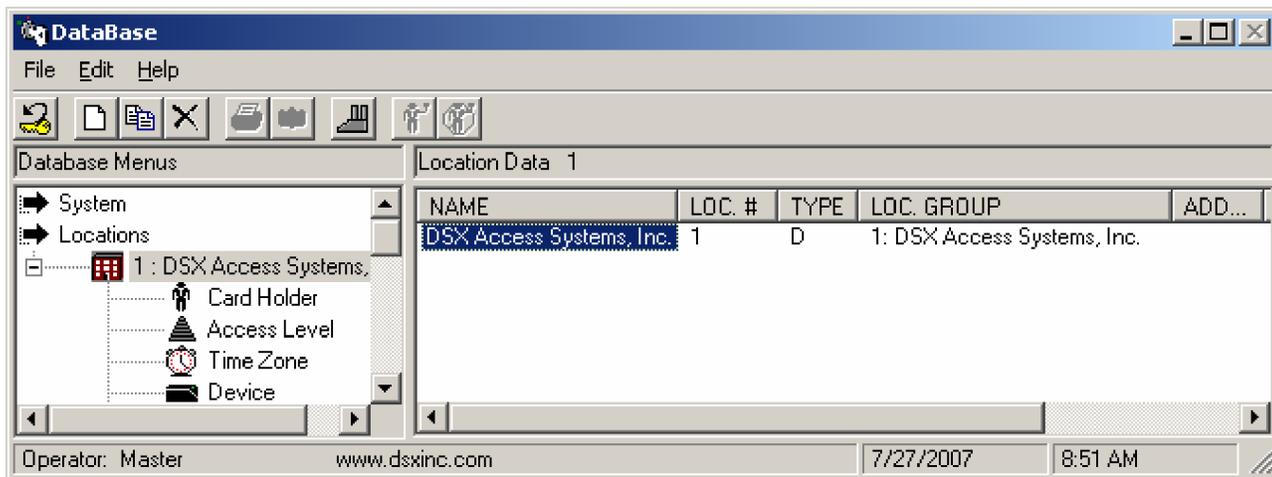
Coordinated Fields:

There are several fields that are referenced in the API. The data actually used in the interface would be coordinated with the DSX Administrator and easily found in the WinDSX database program.

L# - This Loc Group Number is found in the WinDSX Database at the beginning of the Loc.Group Field.
**Loc #** - Loc Number is also shown in the image below.

U# and UdfNum - UDF Number is the number of the UDF field containing the Unique ID of the Card Holders. The fields are listed in numeric ascending order. The first UDF field shown in the list is #1 and the fifth one from the top is #5. The image below shows where to find User Define Fields in the WinDSX DataBase.
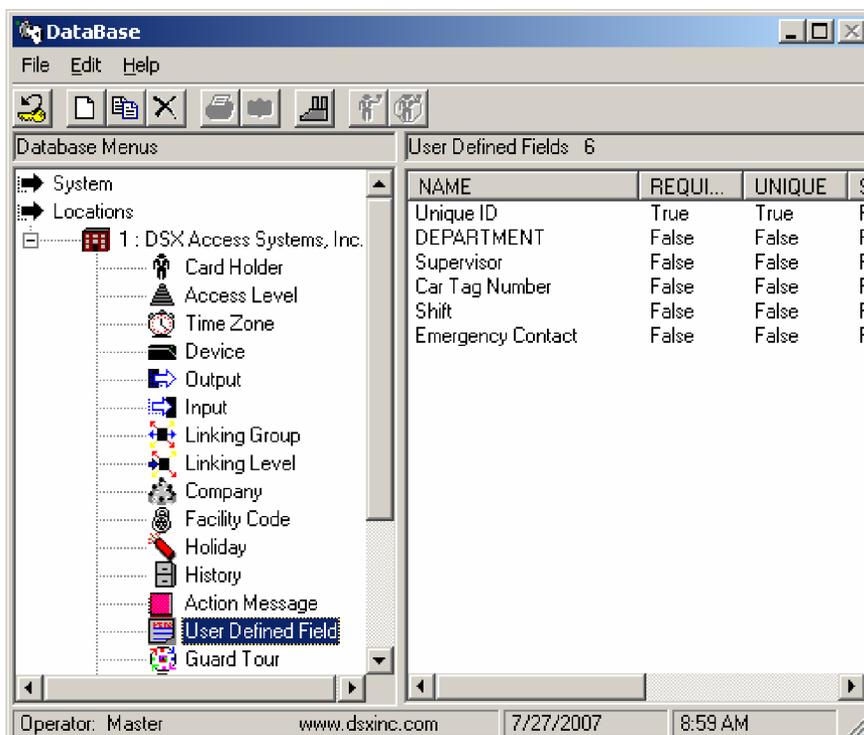
Company – Is the Company names as they are displayed in the WinDSX DataBase program. Companies are the Card Holder groups such as departments, vendors, contractors. Requires exact spelling.
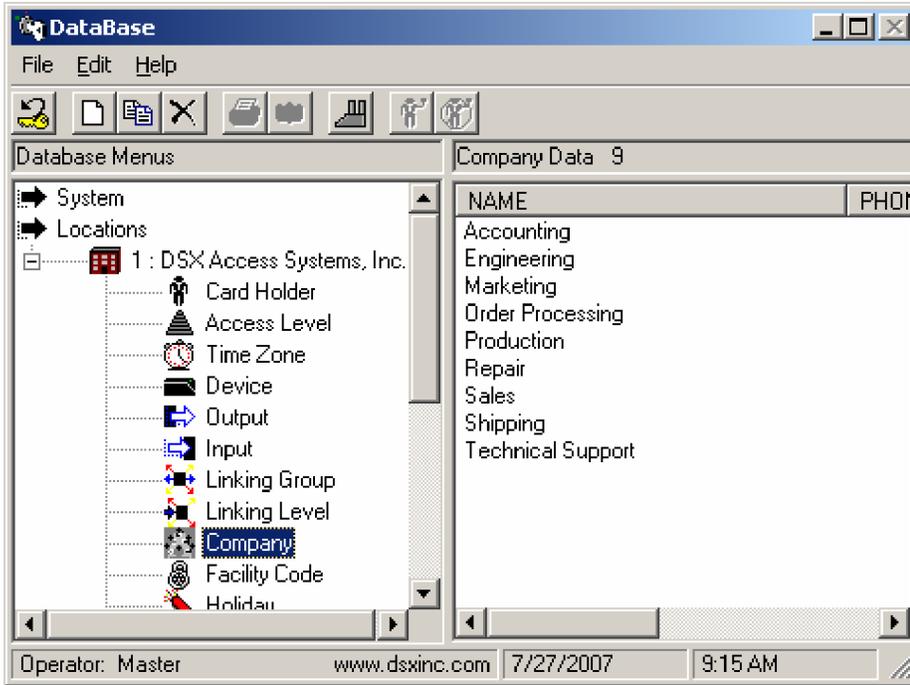
ACL – ACL refers to the Access Level(s) to be assigned to the Card of the Card Holder. The names of the Access Levels as they are shown in the program are used in the Interface. Requires exact Spelling. This applies to AddAcl / AddTempAcl / DelAcl / DelTempAcl
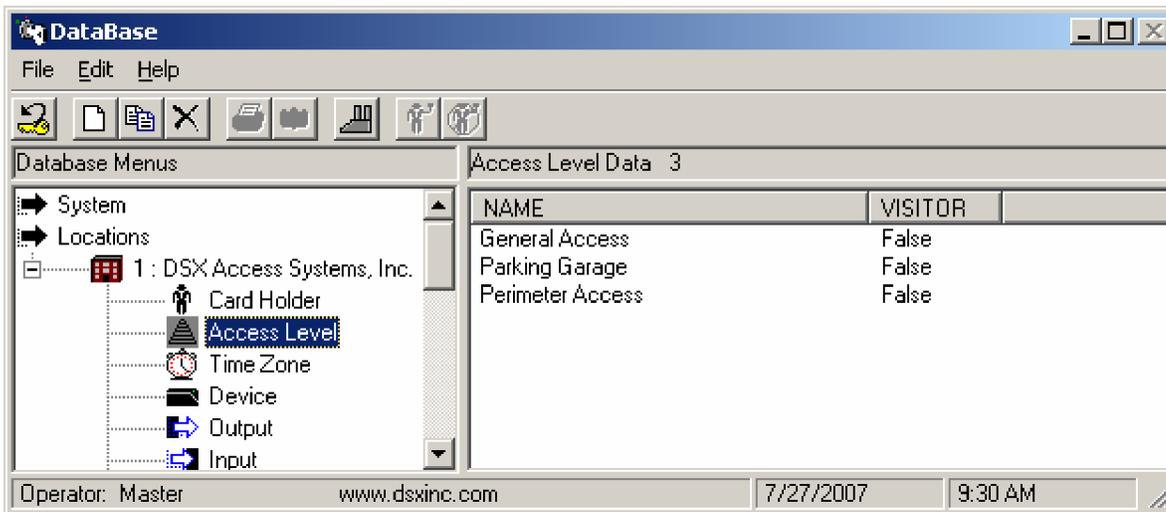
OLL – Refers to the Linking Level to be assigned the card. This is optional and usually for special applications such as handicap doors and Elevator Control. The Linking Level Number is the **Number** shown in the Linking Level List below next to the names.
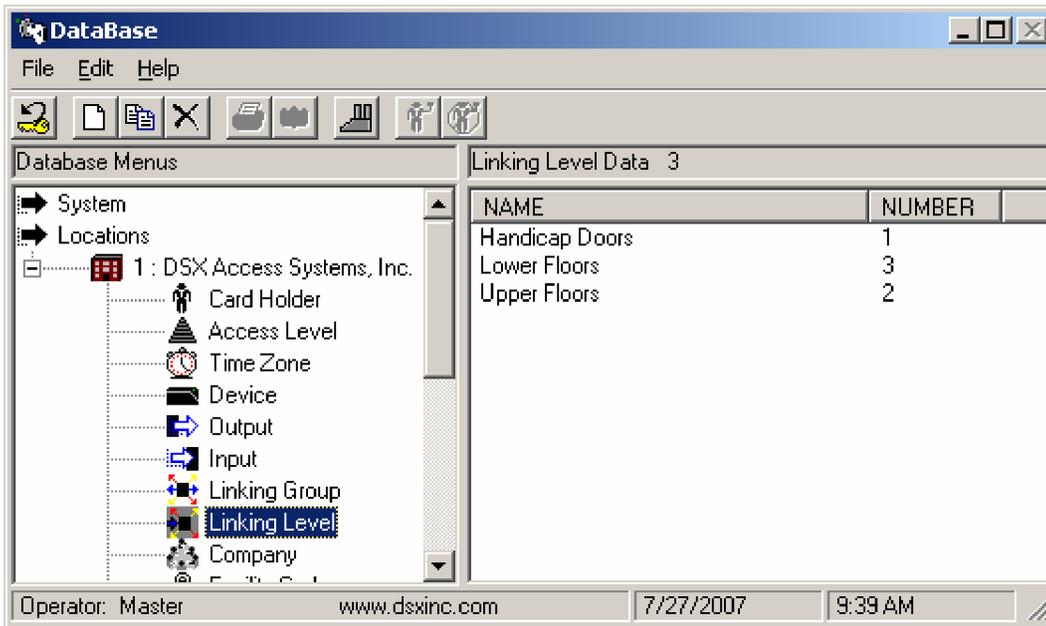
ImgType - The Badge Image Type is shown in the image below. Typically set to 1.
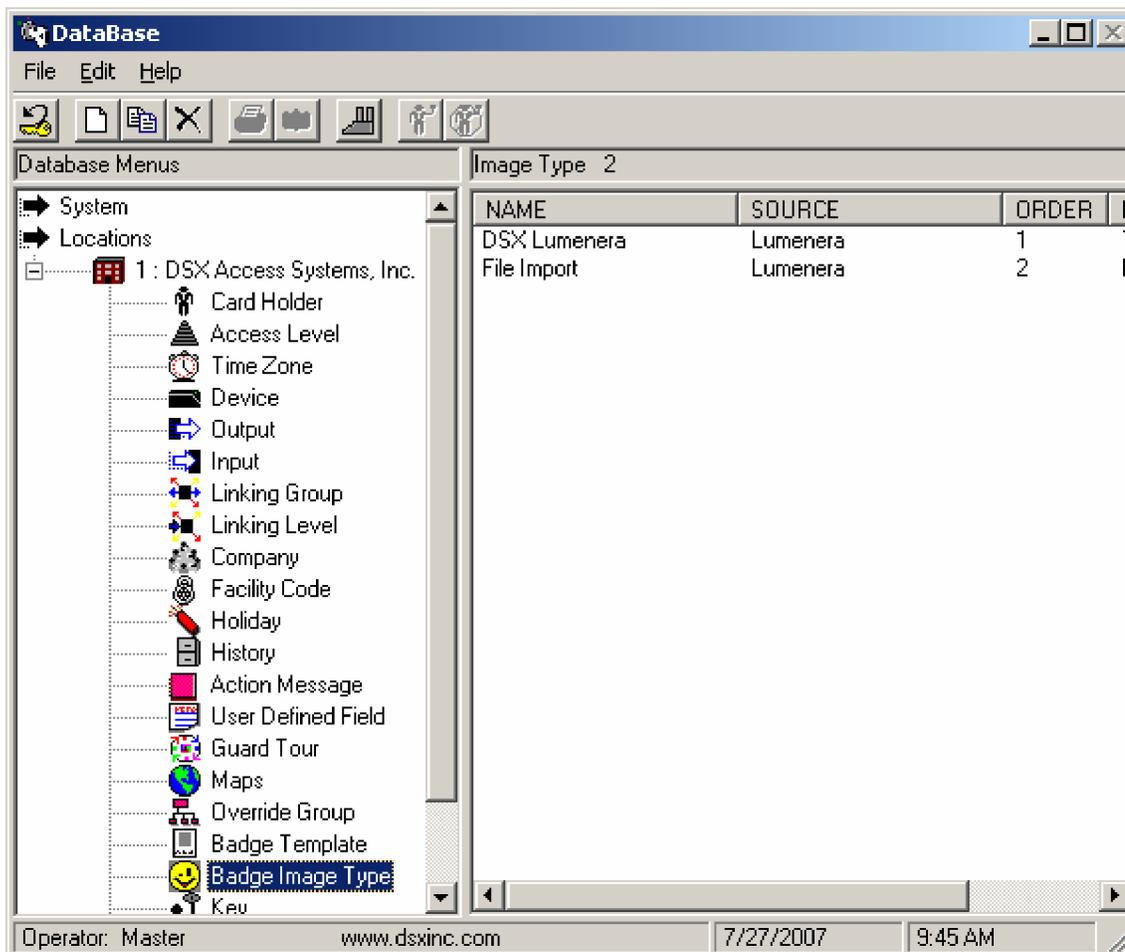
DML - DSX Markup Language

Formatting/Syntax:

- All commands are single character and case insensitive
- Commands must appear as the 1st non-space character in the line
- Any line whose first non-space character is not a recognized command is ignored
- Any command requiring additional fields must have <u>one</u> space between each field.
- All Data Fields must start with a single ^ and end with three ^^^

Commands:

I   ID Command

Contains the data used to identify a particular person. The ID command marks the beginning of a new set of commands. All table commands following an ID command will be associated with the person specified by the ID command. There are **3** fields required to identify the person. **L#** defines the location group number. **U#** defines the number of the UDF field that will store the unique identifier and the UID String which represents the data expected to be in the UDF field. These three items allow the program to establish which person the following table commands will be related to.

Example: I L1 U2 ^1ABC^^^

L1 = LocGrp 1, U2 = UDF 2, 1ABC = unique identifying number

T   Table Command

Identifies the table that the following field commands will be applied to. Valid tables are Names, UDF, Images, and Cards.

Example: T Names

The following field commands will be assigned to the Names table

F   Field Command

Identifies a fieldname and data to be applied to that field. Must be a valid field name of the current table.

Example: F LName ^Smith^^^

Assign the LName field the value of Smith

W   Write Command

Used at the end of table field definitions to commit data to the database.

DML - DSX Markup Language

D   Delete Command

Used at the end of table field definitions to delete from the database. Using the Delete command on a card number removes the card record from the system.

P   Print Command

Used at the end of table field definitions to print this person with the badge template that is assigned to the company they are associated with.

U   Update Command

Use the Update command to edit an existing record. The difference between the [**U**]pdate and the [**W**]rite command is that the Write command can be used to either add a new record or update an existing record. The Update command can only Update an existing record it will not add a new record.

For example: You send a command to edit the PIN field of a code that does not exist. If you use the Write command it will create a new card with the code you specified and set the PIN to the value you specified. If you use the Update command there will be no change to the database.

DML - DSX Markup Language

Field Specifications:

Names:

Field Name	Type	Length	Description
FName	String	30	First Name (Optional)
LName	String	30	Last Name (Required)
Company	String	30	Company Name (Required)
Visitor	Boolean		1 or 0 = true or false (Optional default = False)
Trace	Boolean		1 or 0 = true or false (Optional default = False)
Notes	String	200	Notes to be stored with Name record (Optional)

UDF:

Field Name	Type	Length	Description
UdfNum	Integer		The UDF number that this data is assigned to
UdfText	String	50	The string to be assigned to this UdfNum

Images: (images are optional)

Field Name	Type	Length	Description
ImgType	Integer		The number of the badge image type this file is assigned
FileName	String	50	Path / name of file to import (must be proper format)

Cards: (In general Cards are optional, but if you define a card there are some fields that are required)

Field Name	Type	Length	Description
Code	Double	11	Card number (Must be first field defined)(Required)
PIN	Long	7	PIN number (Optional Default = 0)
StartDate	Date		Activation mm/dd/yyyy HH:MM (Optional default = Today) 24:hr
StopDate	Date		De-activation mm/dd/yyyy HH:MM (Optional default 12-31-9999) 24:hr
CardNum	String	15	Imprinted card number (Optional)
NumUses	Integer	4	# of card uses allowed. 1-9999 /9999=unlimited (Optional default=9999)
GTour	Boolean		Guard Tour card. 1 or 0 = true or false (Optional default = false)
APB	Boolean		Override AntiPassBack 1 or 0 = true or false (Optional default = false)
AddAcl	String	30	exact spelling of the access level being added (Optional)
AddTempAcl	String	30	exact spelling of the temporary access level added (Optional)
AclStartDate	Date		Temporary ACL supersedes access level (Optional) mm/dd/yyyy
AclStopDate	Date		Temporary ACL relinquish control back to access level (Optional)
Loc	Integer		Location number that OLL will be set for. (Optional)
OLL	Integer		Output Linking Level number. 0 = none (Optional)
Notes	String	200	Notes to be stored with card data (Optional)

DML - DSX Markup Language

Field Definitions:

I L5 U1 ^123456789^^^	(Associate records with Name in LocGrp 5 with 123456789 in UDF 1)
T Names	(Following field records are for the Names table)
F FName ^Jennifer^^^	(FName field of the Names table is set to Jennifer)
F LName ^Rose^^^	(LName field of the Names table is set to Rose)
F Company ^DSX Access^^^	(Company is set to "DSX Access" company) Company name not number.
F Visitor ^0^^^	(Visitor is set to False) False =0 True=1
F Trace ^1^^^	(Trace is set to True) False =0 True=1
F Notes ^These are notes^^^	(String to be stored in the notes field of the name record)
W	(Write Names data)
T UDF	(Following field records are for the UDF table)
F UdfNum ^1^^^	(UdfNum is set to 1)
F UdfText ^123456789^^^	(Text for UdfNum 1 is "123456789") This is adding the unique identifier into the system
W	(Write UDF 1)
T UDF	(Following field records are for the UDF table)
F UdfNum ^2^^^	(UdfNum is set to 2)
F UdfText ^More Text^^^	(Text for UdfNum 2 is "Some More Text")
D	(Delete UDF 2)
T Images	(Following field records are for the Images table)
F ImgType ^1^^^	(Image Type is set to 1)
F FileName ^c:\MyPic.jpg^^^	(Import file at "c:\MyPic.jpg" [System will change the file name])
W	(Write Image data)
T Cards	(Following field records are for the Cards table)
F Code ^6347^^^	(Card code is set to 6347)
F PIN ^1234^^^	(PIN is set to 1234)
F StartDate ^1/1/2008 09:00^^^	(Start Date is MM/DD/YYYY and Start Time is 24 hour format)
F StopDate ^12/31/9999 10:00^^^	(Stop Date is MM/DD/YYYY and Stop Time is 24 hour format)
F CardNum ^6347^^^	(Imprinted Card Number is set same as code)(Can be same or different as Code – Optional)
F NumUses ^9999^^^	(NumUses is set to 9999 / 9999 - does not decrement)
F GTour ^0^^^	(Is this card to be used with Guard Tour 0=False /1=True)
F APB ^0^^^	(Is this card to override AntiPassback 0=False /1=True)
F AddAcl ^Front Door ACL^^^	(exact spelling of the access level being added)
F AddTempAcl ^Back Door ACL^^^	(exact spelling of the temporary access level being added)
F AclStartDate ^05/01/2008^^^	(Temporary Access Level 14 is set to commence and supersede access level 12 on May 1st 2008)
F AclStopDate ^05/05/2008^^^	(Temporary Access Level 14 is set to relinquish control to access level 12 on May 5th 2008)
F Loc ^5^^^	(Location 5 is specified for the following Output Linking Level)
F OLL ^3^^^	(Output Linking Level is set to 3 for location 5. Other Location numbers and output linking level numbers can be inserted here to assign different linking levels to this card for other locations in the loc group)
F Notes ^These are notes^^^	(String to be stored in the notes field of the card)
W	(Write Card data) (required)
P	(Print this name record with the badge template assigned to the company they are in) Optional

Special Fields and Operational Examples:

ReplaceCode   Replace Code is used to replace the code number of an existing card. This would be used when a card is lost and must be replaced with a new one. All of the settings of the original code are transferred to the new code. ReplaceCode can be used without specifying the existing card number just the new card number. It will find the first active card that is assigned to the person specified in the I command line and will replace the code field of that card with a new card number. If there are multiple cards assigned to the person there is no way to determine which card will be replaced. If there are no cards assigned, the code specified by ReplaceCode will be added to the card holder as a new card. Any field could be modified after the ReplaceCode Code command.

```
I L5 U1 ^123456789^^^        'define the person
T Cards                      'define the table to be edited
F Code ^5457^^^              'define the code to be replaced
F ReplaceCode ^33445^^^      'define the new code
W                            'write the data
```

Copy2Code   This field will be used to copy an existing cards data to a new card. This would be used to assign an additional card to a person and give it the same characteristics as their existing card such as start and stop dates, access levels, and linking levels. If the existing code is not specified the system will try to copy an active card first, then an inactive. If there is no existing code the new number specified by Copy2Code will be added to the card holder and give the defaults. Below the Copy2Code line any other code information can be defined so that it is implemented instead of the copied data.

```
I L5 U1 ^123456789^^^        'define the person
T Cards                      'define the table to be edited
F Code ^5457^^^              'define the code to be copied
F Copy2Code ^33445^^^        'define the new code
W                            'write the data
```

AddAcl / AddTempAcl   These fields will be used to add an access level and/or temp access level to an existing card without deleting the access levels already assigned to the card. Multiple Access Levels can be assigned at the same time by repeating the AddAcl or AddTempAcl with the name of each new level desired.

```
I L5 U1 ^123456789^^^        'define the person
T Cards                      'define the table to be edited
F Code ^5457^^^              'define the code to be edited
F AddAcl ^Front Door ACL^^^  'exact spelling of the access level being added
F AddAcl ^Back Door ACL^^^   'exact spelling of the access level being added
F AddTempAcl ^Back Door ACL^^^   'exact spelling of the temporary access level being added
W                            'write the data
```

DML - DSX Markup Language

Special Fields and Operational Examples: continued…

DelAcl DelTempAcl   These fields will be used to delete an access level and/or temp access level from an existing card. This is used to remove a certain access level from the card without removing them all.

```
I L5 U1 ^123456789^^^          'define the person
T Cards                        'define the table to be edited
F Code ^5457^^^                'define the code to be edited
F DelAcl ^Front Door ACL^^^        'exact spelling of the access level being deleted
F DelAcl ^Back Door ACL^^^         'exact spelling of the access level being deleted
F DelTempAcl ^Back Door ACL^^^     'exact spelling of the temporary access level to delete
W                              'write the data
```

ClearAcl ClearTempAcl   These fields will be used to remove all of the access levels and/or temp access levels from an existing card without knowing what they are. This command can be immediately followed by the AddAcl / AddTempAcl commands.

```
I L5 U1 ^123456789^^^          'define the person
T Cards                        'define the table to be edited
F Code ^5457^^^                'define the code to be edited
F ClearAcl ^^^^                'removes all access levels without knowing what they are – requires the four ^^^^
F ClearTempAcl ^^^^            'removes all temporary levels without knowing what they are – requires the four ^^^^
F AddAcl ^Front Door ACL^^^ 'exact spelling of the access level being added
W                              'write the data
```

CopyPerm2Temp   This command will be used to add the access levels in the permanent list to the temporary list. Resulting in a list that contains everything from the permanent list AND everything that previously existed in the temporary list.

```
I L5 U1 ^123456789^^^          'define the person
T Cards                        'define the table to be edited
F Code ^5457^^^                'define the code to be edited
F CopyPerm2Temp ^^^^           'copies permanent access levels and adds them to the temporary access level list
W                              'write the data
```

DC   This field will be used to remove all of the cards assigned to this person. You do not have to know what the card or code numbers are in order to delete them with this field.

```
I L5 U1 ^123456789^^^          'define the person
T Names                        'establish the name object
DC                             'delete all cards/codes assigned to this person.
W                              'write the data
```

Application Examples:

The following examples typically show the optional fields, these fields do not have to be included in the text file if you are not using them. Any optional field that is not specified will assume its default value. The examples are designed to show some of the common applications, they do not represent every possible combination.

Add 2 People

Example: This example shows how to Add two people to the system with no card. The first line starts with the "I" command and specifies the Location Group, the number of the UDF field that stores the unique ID and this persons unique identifying number. If the system finds this Identifying number in UDF#1 this is an Edit, but if it does not find this number this becomes an Add.

The first and last name is defined followed by the Company. So that we can refer back to this person later we must also add their unique identifying number into the specified User Defined Field in this case UDF # 1.

```
I L5 U1 ^123456789^^^
T Names
F FName ^Jennifer^^^
F LName ^Rose^^^
F Company ^DSX Access Systems, Inc.^^^
F Visitor ^0^^^
F Trace ^1^^^
W
T UDF
F UdfNum ^1^^^
F UdfText ^123456789^^^
W
I L5 U1 ^ABC456789^^^
T Names
F FName ^John^^^
F LName ^Smith^^^
F Company ^Accounting^^^
W
T UDF
F UdfNum ^1^^^
F UdfText ^ ABC456789^^^
W
```

Add a person and a User Defined Field
Example: This example shows how to Add a Single person to the system with no card, the unique identifying number and an optional but in this case additional User Defined Field.

```
I L5 U1 ^123456789^^^
T Names
F FName ^Jennifer^^^
F LName ^Rose^^^
F Company ^DSX Access Systems, Inc.^^^
F Visitor ^0^^^
F Trace ^1^^^
F Notes ^These are notes^^^
W
T UDF
F UdfNum ^1^^^
F UdfText ^123456789^^^
W
T UDF
F UdfNum ^2^^^
F UdfText ^optional text^^^
W
```

Add a Person, a User Defined Field, and an Image
Example: This example shows how to Add a single Person to the system with no card, the unique identifying number and an optional User Defined Field and a Picture. The Badge Image Type is coordinated with the WinDSX administrator and the Picture file name must include a valid path.

```
I L5 U1 ^123456789^^^
T Names
F FName ^Jennifer^^^
F LName ^Rose^^^
F Company ^DSX Access Systems, Inc.^^^
F Visitor ^0^^^
F Trace ^1^^^
F Notes ^These are notes^^^
W
T UDF
F UdfNum ^1^^^
F UdfText ^123456789^^^
W
T UDF
F UdfNum ^2^^^
F UdfText ^optional text^^^
W
T Images
F ImgType ^1^^^
F FileName ^c:\MyPic.jpg^^^
W
```

Add a Person, a User Defined Field, an Image, a Card, and print the badge
Example: This example shows how to Add a single Person to the system, the unique identifying number, an optional User Defined Field, a Picture, a Card, and to print the persons Badge.

```
I L5 U1 ^123456789^^^
T Names
F FName ^Jennifer^^^
F LName ^Rose^^^
F Company ^DSX Access Systems, Inc.^^^
F Visitor ^0^^^
F Trace ^1^^^
F Notes ^These are notes^^^
W
T UDF
F UdfNum ^1^^^
F UdfText ^123456789^^^
W
T UDF
F UdfNum ^2^^^
F UdfText ^optional text^^^
W
T Images
F ImgType ^1^^^
F FileName ^c:\MyPic.jpg^^^
W
T Cards
F Code ^6347^^^
F PIN ^123456^^^
F StartDate ^1/1/2008 09:00^^^
F StopDate ^12/31/9999 10:00^^^
F CardNum ^6347^^^
F NumUses ^9999^^^
F GTour ^0^^^
F APB ^0^^^
F AddAcl ^Front Door ACL^^^
F AddAcl ^Back Door ACL^^^
F AddTempAcl ^Back Door ACL^^^
F AclStartDate ^05/01/2008^^^
F AclStopDate ^05/05/2008^^^
F Loc ^5^^^
F OLL  ^3^^^
F Notes ^These are notes^^^
W
P
```

Add a Person and a Card Without optional Fields

Example: This very typical example shows how to Add a single person to the system, the Unique Identifying Number, and a card with a 2 Access Levels.

I L5 U1 ^123456789^^^
T Names
F FName ^Jennifer^^^
F LName ^Rose^^^
F Company ^DSX Access Systems, Inc.^^^
W
T UDF
F UdfNum ^1^^^
F UdfText ^123456789^^^
W
T Cards
F Code ^6347^^^
F AddAcl ^Front Door ACL^^^
F AddAcl ^Back Door ACL^^^
W

Add an Image to an Existing Person

Example: This example shows how to Add an Image to an existing person. The Badge Image Type is coordinated with the WinDSX administrator and the Image file name must include a valid path.

I L5 U1 ^123456789^^^
T Images
F ImgType ^1^^^
F FileName ^c:\MyPic.jpg^^^
W

Add a Person and a Card with multiple Access Levels
Example: This typical example shows how to Add a single Person to the system, the Unique Identifying Number, and a card with multiple Access Levels.

```
I L5 U1 ^123456789^^^
T Names
F FName ^Jennifer^^^
F LName ^Rose^^^
F Company ^DSX Access Systems, Inc.^^^
F Visitor ^0^^^
F Trace ^1^^^
F Notes ^These are notes^^^
W
T UDF
F UdfNum ^1^^^
F UdfText ^123456789^^^
W
T Cards
F Code ^6347^^^
F PIN ^123456^^^
F StartDate ^1/1/2008 09:00^^^
F StopDate ^12/31/9999 10:00^^^
F CardNum ^6347^^^
F NumUses ^9999^^^
F GTour ^0^^^
F APB ^0^^^
F AddAcl ^Front Entrances^^^
F AddAcl ^Employee Doors^^^
F AddAcl ^Parking Garage^^^
F AddAcl ^Cafeteria Doors^^^
F Notes ^These are notes^^^
W
```

Add Multiple Access Levels to an existing Card
Example: This shows how to add multiple Access Levels to a card already in the database. This adds these four levels to this card.

```
I L5 U1 ^123456789^^^
T Cards
F Code ^6347^^^
F AddAcl ^Front Entrances^^^
F AddAcl ^Employee Doors^^^
F AddAcl ^Parking Garage^^^
F AddAcl ^Cafeteria Doors^^^
W
```

Add a Person and a Card with a Permanent and Temporary Access Level
Example: This typical example shows how to Add a single Person to the system, the Unique Identifying Number, and a card with both a Permanent and Temporary Access Level. Unless the Permanent Level is also added in as a Temporary Level the card will be operating solely by the Temporary Level between the AclStart and AclStop dates.

```
I L5 U1 ^123456789^^^
T Names
F FName ^Jennifer^^^
F LName ^Rose^^^
F Company ^DSX Access Systems, Inc.^^^
F Visitor ^0^^^
F Trace ^1^^^
F Notes ^These are notes^^^
W
T UDF
F UdfNum ^1^^^
F UdfText ^123456789^^^
W
T Cards
F Code ^6347^^^
F PIN ^123456^^^
F StartDate ^1/1/2008 09:00^^^
F StopDate ^12/31/9999 10:00^^^
F CardNum ^6347^^^
F NumUses ^9999^^^
F GTour ^0^^^
F APB ^0^^^
F AddAcl ^Front Entrances^^^
F AddTempAcl ^Back Door ACL^^^
F AclStartDate ^05/01/2008^^^
F AclStopDate ^05/05/2008^^^
F Notes ^These are notes^^^
W
```

Add Temporary Access Levels to an existing Card
Example: This shows how to add Temporary Access Levels to a card already in the database. The following Temporary Levels are to be used ***instead of*** the Permanent ones between the AclStart and AclStop dates.

```
I L5 U1 ^123456789^^^
T Cards
F Code ^6347^^^
F AddTempAcl ^Back Door ACL^^^
F AddTempAcl ^Garage ACL^^^
F AclStartDate ^05/01/2008^^^
F AclStopDate ^05/05/2008^^^
W
```

Add Temporary Access Levels to an existing Card
Example: This shows how to add Temporary Access Levels to a card already in the database. The following Temporary Levels are to be used ***in addition to*** the Permanent ones between the AclStart and AclStop dates.

```
I L5 U1 ^123456789^^^
T Cards
F Code ^6347^^^
F CopyPerm2Temp ^^^^
F AddTempAcl ^Back Door ACL^^^
F AddTempAcl ^Garage ACL^^^
F AclStartDate ^05/01/2008^^^
F AclStopDate ^05/05/2008^^^
W
```

Replace a Card
Example: This shows how to easily replace a lost or stolen card. The Code specified is the old card number and the ReplaceCode is the new card number.

```
I L5 U1 ^123456789^^^
T Cards
F Code ^5457^^^
F ReplaceCode ^33445^^^
W
```

Delete a Person
Example: This example shows how to Delete the Person and all associated data. Deleting a Person from the Names Table causes a cascading delete of all associated UDFs, Cards, and Images.

```
I L5 U1 ^123456789^^^
T Names
F FName ^Jennifer^^^
F LName ^Rose^^^
D
```

Delete a Specific Card from a Person
Example: This example shows how to remove a specific card assigned to a person leaving the person and all other data such as UDFs and Images.

```
I L5 U1 ^123456789^^^
T Cards
F Code ^6347^^^
D
```

Delete All Cards from a Person
Example: This example shows how to remove all cards assigned to a person without knowing what the card numbers are while leaving the person and all other data such as UDFs and Images.

```
I L5 U1 ^123456789^^^
T Names
DC
W
```

Delete a User Defined Field from a Person
Example: This example shows how to remove a specific UDF from a person.

```
I L5 U1 ^123456789^^^
T UDF
F UdfNum ^2^^^
F UdfText ^optional text^^^
D
```

Delete a Specific Access Level from a Card
Example: This example shows how to remove a specific Access Level or Temporary Access Level from a card.
This is used to remove a certain access level from the card without removing them all. It is necessary to use the exact spelling of the access level.

```
I L5 U1 ^123456789^^^
T Cards
F Code ^5457^^^
F DelAcl ^Front Door ACL^^^
F DelTempAcl ^Back Door ACL^^^
W
```


Delete All Access Levels from a Person and add new ones
Example: This example shows how to remove all Access Levels or Temporary Access Level from a person.
These fields will be used to remove all of the access levels and/or temp access levels from an existing card without knowing what they are. It can be immediately followed by the AddAcl / AddTempAcl commands to assign new access levels.

```
I L5 U1 ^123456789^^^
T Cards
F Code ^5457^^^
F ClearAcl ^^^^
F ClearTempAcl ^^^^
F AddAcl ^Front Door ACL^^^
F AddAcl ^Back Door ACL^^^
W
```


Assign Linking Levels in Multiple Locations to a card
Example: This typical example shows how to Add a Single person to the system, the Unique Identifying Number, a card with a single Access Level and multiple Linking Levels.

```
I L5 U1 ^123456789^^^
T Names
F FName ^Jennifer^^^
F LName ^Rose^^^
F Company ^DSX Access Systems, Inc.^^^
F Visitor ^0^^^
F Trace ^1^^^
F Notes ^These are notes^^^
W
T UDF
F UdfNum ^1^^^
F UdfText ^123456789^^^
W
T Cards
F Code ^6347^^^
F PIN ^123456^^^
F StartDate ^1/1/2008 09:00^^^
F StopDate ^12/31/9999 10:00^^^
F CardNum ^6347^^^
F NumUses ^9999^^^
F GTour ^0^^^
F APB ^0^^^
F AddAcl ^Front Door ACL^^^
F Loc ^5^^^
F OLL ^3^^^
F Loc ^6^^^
F OLL ^2^^^
F Notes ^These are notes^^^
W
```

Add Multiple People and Cards without optional fields
Example: This very typical example shows how to Add multiple people to the system, the Unique Identifying Number, and a card with multiple Access Levels. Note that the optional fields are omitted. For instance the StartDate and StopDate fields are omitted for the card definition. This will cause the system to assign the default values of "Today" to the StartDate and 12/31/9999 (infinite) to the StopDate. The results of this example would be 3 people added to the database each with a single card assigned to them.

```
I L5 U1 ^123456789^^^
T Names
F FName ^Jennifer^^^
F LName ^Rose^^^
F Company ^DSX Access Systems, Inc.^^^
W
T UDF
F UdfNum ^1^^^
F UdfText ^123456789^^^
W
T Cards
F Code ^6347^^^
AddAcl ^Front Door ACL^^^
AddAcl ^Back Door ACL^^^
W
I L5 U1 ^ABC456789^^^
T Names
F FName ^John^^^
F LName ^Smith^^^
F Company ^Accounting^^^
W
T UDF
F UdfNum ^1^^^
F UdfText ^ABC456789^^^
W
T Cards
F Code ^12347^^^
AddAcl ^Parking ACL^^^
AddAcl ^Back Door ACL^^^
W
I L5 U1 ^XYZ456789^^^
T Names
F FName ^Mark^^^
F LName ^Johnson^^^
F Company ^Security^^^
W
T UDF
F UdfNum ^1^^^
F UdfText ^XYZ456789^^^
W
T Cards
F Code ^44563^^^
AddAcl ^Security ACL^^^
AddAcl ^Parking ACL^^^
W
```

Import Holidays and Companies

Overview:

The WinDSX API accommodates managing Holidays and Companies through the same method as Card Holders.

Below is a description of the file that other programs can create to Add Holidays and Companies into the WinDSX DataBase.

1. The DB.exe program on the Comm Server or optionally on the DailyOps PC watches for a ^imp##.txt file in the shared database directory every 15 seconds. If the file exists the data is imported and the file deleted.

2. Command Rules
❖ All Commands are single character.
❖ Command Characters are case insensitive.
❖ Commands must appear as the 1st non space character in the line.
❖ Any line whose first non-space character is not a recognized command is ignored.
❖ Any command requiring additional fields must have one space between each field of the command.
❖ The text file is created and placed in the shared WinDSX database folder.

File Name and Structure

File name = ^imp01.txt
File Structure for Holidays:

T Hol (following data is for the Holiday table)
F Loc ^1^^^ (specifies location number)
F Date ^12/25/2007^^^ (specify date in US format)
F Type ^1^^^ (specify the type as it relates to time zones)
F Name ^New Years^^^ (the name of the holiday)
F Reoccurring ^1^^^ (reoccurs each year) 0 = Off and 1 = On
W (write the data)
(Location and Date are order specific and must be first in the file. It is recommended that all fields are in this order and all fields are complete.)

File name = ^imp01.txt
File Structure for Companies:

T Company (specifies the Company Table)
F LocGrp ^1^^^ (specifies the Location Group)
F Name ^XYZ Corp^^^ (Name of the Company)
F Phone ^1^^^ (phone number)
F Fax ^0^^^
F Badge ^DSX^^^ (name of the associated badge template)
F Contact ^ME^^^ (contact person for company)
F Suite ^201^^^   (suite #)
F NoUseDays ^12^^^^ (use it or lose it days of inactivity)
F Notes ^These are notes^^^
W   (write the data)

WinDSX DailyOps

DailyOps occurs once a day at midnight traditionally from the Communications Server PC. When Daily Ops runs each day several automated processes are initiated. Cards that have pending Start or Stop Dates and Times, and Cards with Temporary Access Levels are evaluated and modified. Cards that are assigned to a Company with a "Use it or Lose it" parameter are evaluated and de-activated if appropriate. Backups of DataBase and/or History are created if the system is configured for automated backups. The DataBase Management Interface is another function of Daily Ops that is performed by the Comm Server PC with considerable delays built in to prevent the interface from occupying all of the Comm Server's processor time. All of these processes are performed at the Communications Server.

With Version 3.6.4 / 4.6.4 and higher the Daily Ops routine can be assigned to any Workstation instead of the Comm Server which is the default. This can be beneficial, for example, in situations where the automated backups need to be run on a PC other than the Comm Server. Another example would be moving the DataBase Management Interface to an administrative PC that could be dedicated to the process. This is ideal for sites that want to send large amounts of data through the API and want it to be processed quickly. This PC must be left on at all times just like the Comm Server PC as it will now perform all of the automated duties.

To do this only requires a text file named DailyOps.txt that contains the Workstation number of the PC required to run Daily Ops. This will cause that Workstation to check for Imp01.txt files constantly and will remove the delays that are normally built into the routines used by the Comm Server. The PC that will be performing this task should not be used for anything else. The DataBase Management Interface will completely consume the CPU but will drastically reduce the amount of time it takes to move large amounts of data. The PC that will run Daily Ops needs to be left on at all times so that the Interface will run as well as automated backups and card evaluations.

Data Logging

Recording of the data being imported can be important and because of that there are records of the files being processed. In Location/History there is an Event recorded each time a file is processed. The event is "ASCII File Imported". This lets a user know when there was a file processed.

For a complete report of Who and What was imported use the "DataBase History" report located under System/Reports. This report shows the time and date along with the card holder name, udf information, and card information imported for each card holder and that they were added to the system using the import feature instead of the database program.

This type of logging must be enabled at the Comm Server. Navigate in the DataBase program to System/Setup/System Parameters/Yes-No Options and select "Record DataBase Changes".