IBM

What happened to the '&' ?

A guide to data transformation map functions

Lisa Lubowiecki
IBM
Business Exchange Services
Data Transformation

The Next Generation
DI
WDI
2002 User Conference

Agenda

- Overview
  - The '&' Literal Keywords and what replaced them
  - Special Handling in DT maps
  - Arithmetic Operators in DT maps
  - Special Operator - Scaling in DT maps
  - Brand New Functions available in DT maps
  - Mapping Generic Service Elements in DT maps
- Objectives
  - To ease the transition of using DT maps for the mapper by providing a cross-reference guide relating the "old way" to the "new way"
- Considerations
  - It is highly recommended that attendees of this workshop have previously attended a demo of the new 4.1 Client which highlights DT maps.

Send and Receive Map Literal Keywords

- &ACFIELD
- &ASSERT*n*
- &DATE
- &DEFERRED
- &E
- &ERR
- &FORCE
- &FORMAT
- &IF
- &IFDATA
- &IFNODATA
- &LOOPBREAK
- &LOOPCHECK
- &LSAVE
- &LSET
- &LSID
- &SAMEAS
- &SAVE
- &SET
- &THANDLE
- &TIME
- &TPID
- &TPNICKN
- &USE
- &ZEROSIG

Literal Keywords

- &ACFIELD
  - There is no equivalent command/function for the DT maps in version 4.1. This functionality will likely be included with a future release supporting transaction store.
- &ASSERT*n*
  - The closest equivalent would be the use of the Else or ElseIf as part of the If condition block, since with IF statements action is taken if the expression is evaluated to true and with &ASSERT, action is taken if the expression is evaluated to false.
  - However, the If/Else/EndIf does not have a level associated with it and cannot be turned off at translation time.

For example, for a send or receive map:

```
&ASSERT1(UOM = 'CS') &ERR(2,5000,,'Invalid unit of measure')
```

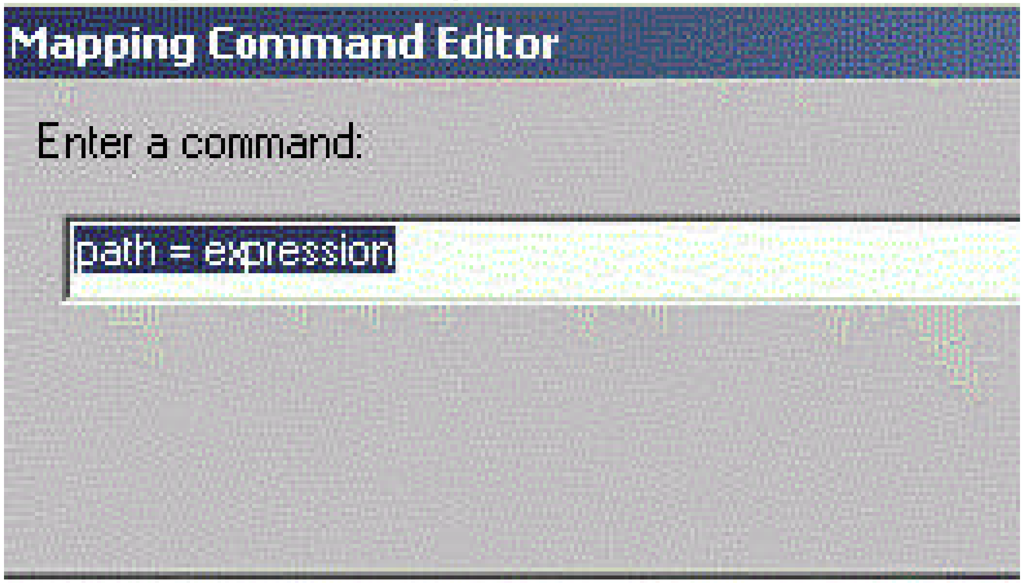
For a DT map:

```
If(UOM='CS')
        mapping command here
Else
        Error(2, 5000, 'Invalid unit of measure')
EndIf
```

Literal Keywords continued...

&DATE

- Use function Date(). For example:



Mapping Command Editor

Enter a command:

```
path = expression
```

```
\Table 1\20 M BIG\1 M 373\\ = Date()
```

&DEFERRED

- There is no deferred function to code for the DT maps. Instead, you can place the appropriate mapping in your map after the needed value is known, which can refer back to an element towards the top of your map.

Literal Keywords continued...

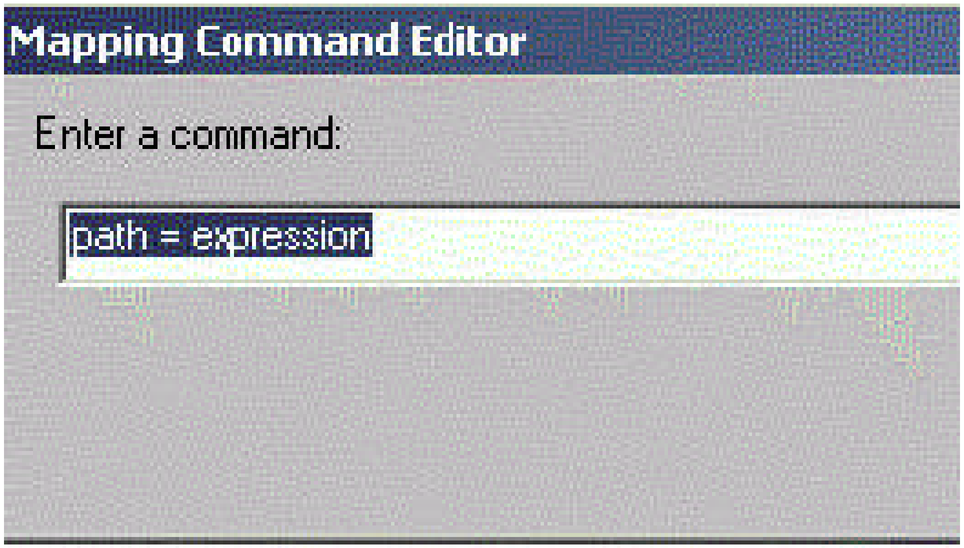
★ &E

- Insert gives you the choice of inserting **Before**, **After**, or **Within**

Insert >

Command >

Assignment...



Enter a command:

path = expression

- Assigns a path or variable equal to an expression

Literal Keywords continued...

- &ERR
  - Use the Error command as follows:

    Error(*level, code, text*)

    where:

    - level indicates the severity of the error (0, 1, or 2)
    - code is a unique error code assigned to the error (5000 - 5999)
    - text is the string value that will be included in the TR0026 error message
- &FORCE
  - Using a command to FORCE a value into a target element is not needed in DT maps since you can "insert" your assignment of the variable **before** or **after** an element instead on only within an element.
- &FORMAT
  - There is no equivalent command/function for the DT maps in 4.1. Support for this function will be introduced in a later release.

Literal Keywords continued...

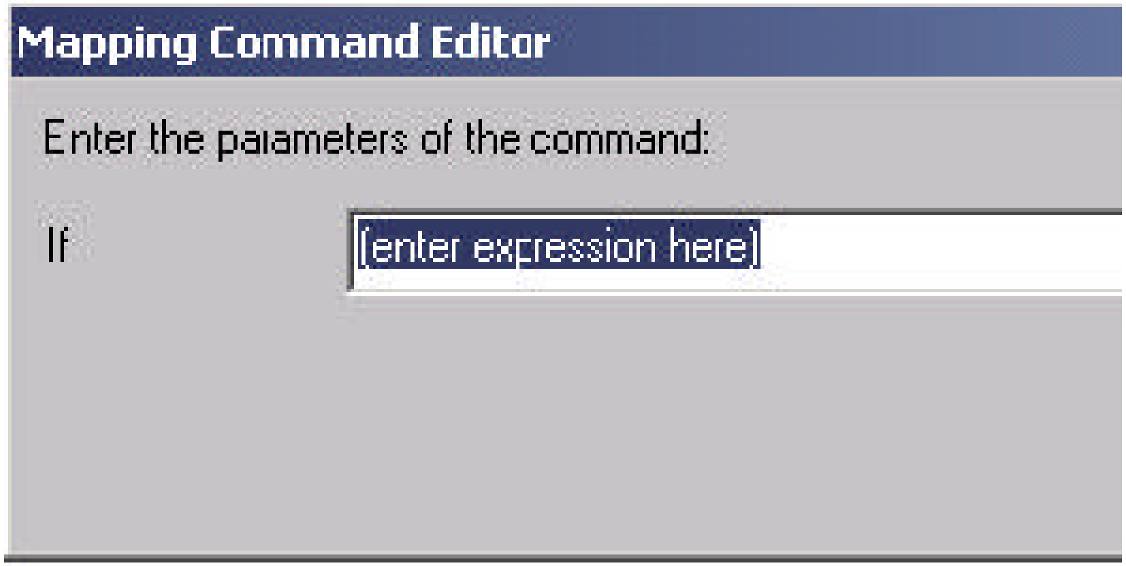
★ &IF

- DT maps have an If/EndIf block which can include optional ElseIf's and an optional Else.
- The following creates an If/EndIf block without an ElseIf or an Else

Insert >

Command >

If...



Mapping Command Editor

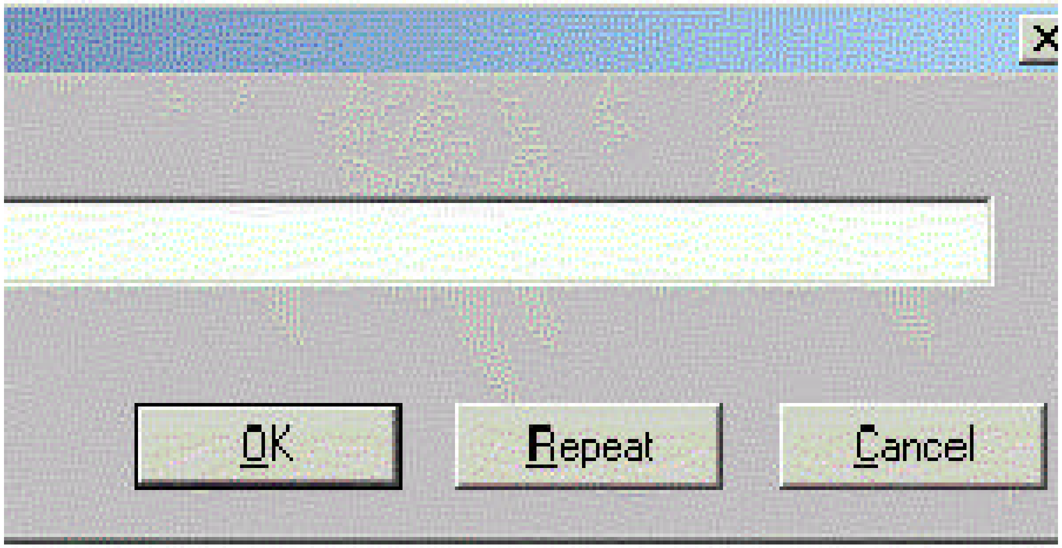
Enter the parameters of the command:

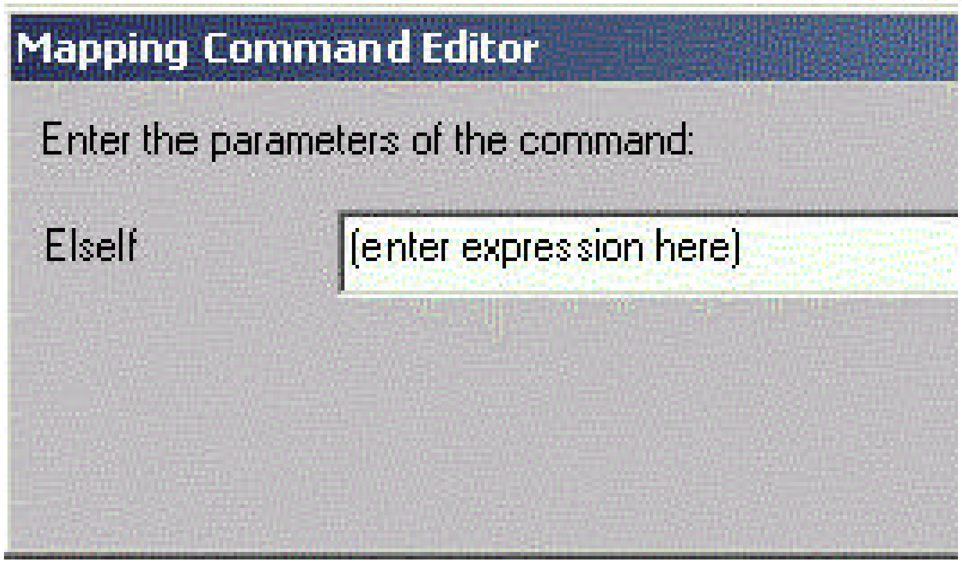
If [enter expression here]

- Click OK to create the If/EndIf block

Literal Keywords continued...

- &IF continued
  - To immediately create an ElseIf, click on the Repeat button instead of OK, and fill in the expression




Mapping Command Editor

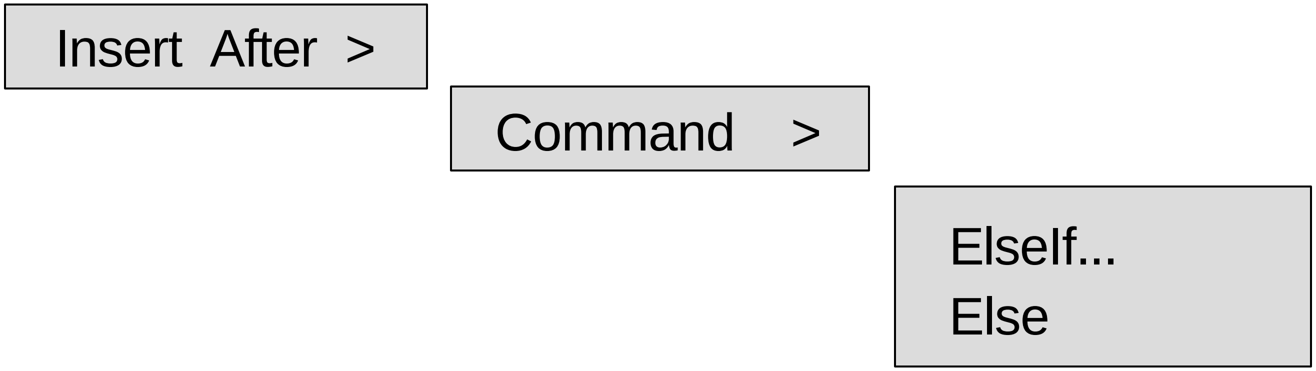
Enter the parameters of the command:

ElseIf (enter expression here)

Literal Keywords continued...

&IF continued...

- Once the If/EndIf block is created, you can do the following (and repeat if necessary) to add any desired ElseIf's or an Else
  - Click on the If statement to select it



Insert After >

Command >

ElseIf...

Else

  - If you select ElseIf, you can click the Repeat button to continue adding ElseIf's to this If/EndIf block
  - You can go back later and insert additional ElseIf's by clicking on the If or ElseIf above where you want to add it and select Insert After, Command, ElseIf
  - You can only add an Else after the last ElseIf or after the If if no ElseIf's exist
  - You can create a new If/EndIf block within an existing If, ElseIf, or Else

Literal Keywords continued...

- &IFDATA and &IFNODATA
  - use the IsEmpty function

    IsEmpty(*value*)

    - The IsEmpty function will return a boolean **True** if the value is empty and a boolean **False** if value contains data.
    - Example:

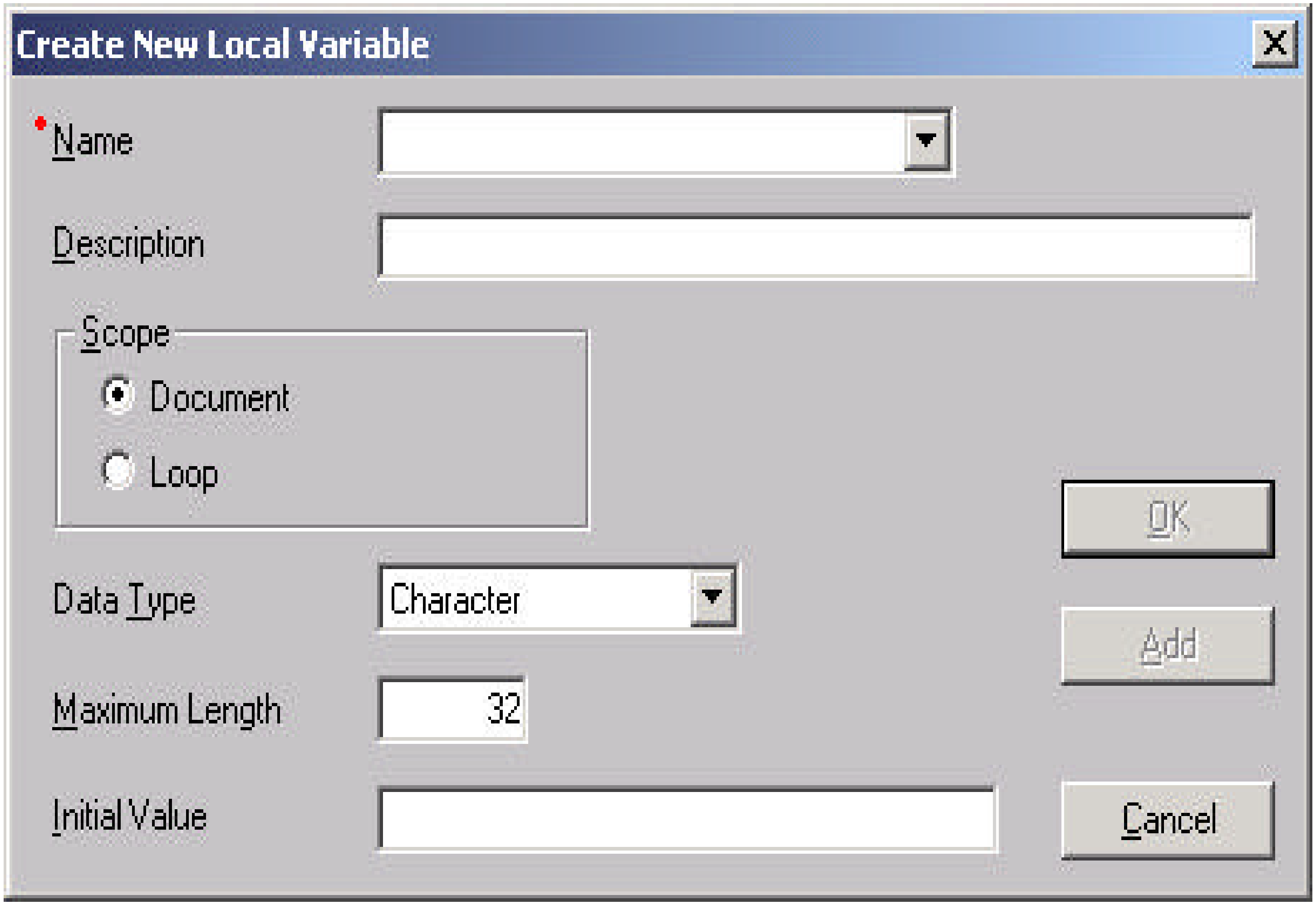
```
If(IsEmpty(variable1))
      mapping command
EndIf
```

- &IFNOVAR
  - There is no equivalent function for the DT maps, because it is not needed. All variables are set up in one of the variable sections in the bottom right-hand corner of the Mapping window.
- &LOOPBREAK and &LOOPCHECK
  - There is no equivalent command/function for the DT maps in version 4.1. This functionality will be introduced in a later release.

Literal Keywords continued...

&LSAVE, &SAVE, &LSET, and &SET

- This is accomplished in the DT maps by creating a variable and assigning it to a value or path



Create New Local Variable

Name

Description

Scope

● Document

○ Loop

Data Type: Character

Maximum Length: 32

Initial Value

Literal Keywords continued...

- &LSID
  - Returns the LS01 value on a receive map
  - This can be accomplished in a DT map where EDI is the source by:
    - Creating a new variable and mapping that variable to the LS01

      LoopInstance = \Table 2\10 M PO1 Loop\320 O LS\1 M 447\\

    - Use Occurrence function to obtain the current occurrence number
    - Use StrComp function to test the LS01 value in an If statement
- &SAMEAS
  - There is no equivalent command/function. It is not needed since copying and pasting functionality has been added.
    - To COPY a mapping: Shift + Click, keep both depressed, drag to desired location and let go of both.
    - To MOVE a mapping: Just click on the mapping and drag to desired location then let go of the mouse button.
- &THANDLE
  - There is no equivalent command/function for the DT maps in version 4.1. This functionality will likely be included with a future release supporting transaction store.

Literal Keywords continued...

- &TIME
  - Use the Time function in an assignment. For example:

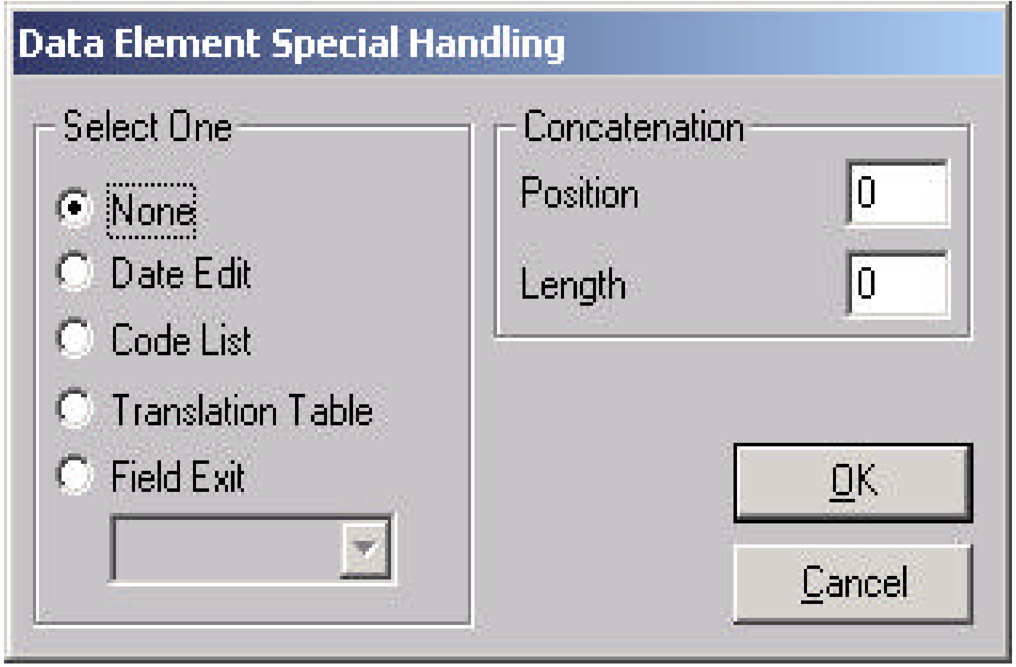
    ```
    \Table 1\140 O DTM\3 C 337\\ = Time()
    ```

- &TPID and &TPNICKN
  - There is no equivalent command/function for the DT maps in version 4.1. This functionality will be introduced in a later release.
- &USE
  - There is no equivalent command/function for the DT maps since you can insert the variable assignments anywhere.
- &ZEROSIG
  - There is no equivalent command/function for the DT maps in version 4.1. This functionality will likely be introduced in a later release.

Special Handling in Send and Receive Maps

- Date Edit
- Code List
- Translation Table
- Field Exit

- Concatenation (send only)



Data Element Special Handling

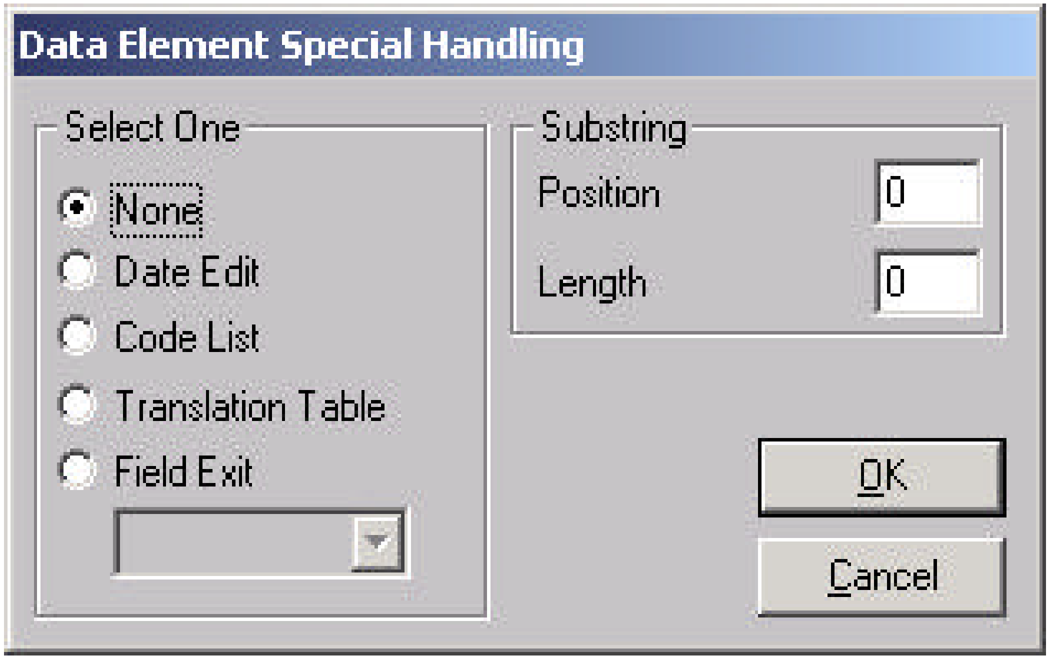
Select One

● None
○ Date Edit
○ Code List
○ Translation Table
○ Field Exit

Concatenation

Position: 0

Length: 0



Data Element Special Handling

Select One

● None
○ Date Edit
○ Code List
○ Translation Table
○ Field Exit

Substring

Position: 0

Length: 0

- Substring (receive only)

- **Note: Special Handling has not changed for Send and Receive maps in Version 4.1**

Special Handling in 4.1 for DT maps

Date Edit

- Purpose: Convert dates from the format listed in the drop-down list to the format set in the standards (yymmdd) for send maps and reverse the process for receive map.
- The any-to-any flexibility in a data transformation map does not insure that the target or source will be a standard. For DT maps, you must enter the following command:

DateCnv(*sourceDate, frommask, tomask*)

Note: The above function replaces

&E(*variable* FD *mask* TD *mask*)

Field Exit

- There is no corresponding function for this in the DT maps for version 4.1. This functionality will be introduced in a later release.

Special Handling in 4.1 for DT maps continued...

★ Code List

- Purpose: Verify values that appear in the field against a pre-defined code list, whether it be one supplied by the standard or created by the mapper.
- Use the Validate function.

Validate("*CodeList*",*value [,error]*)

where:

- CodeList is the name of the Code List table
- value is the string to look up in the table. It can be a variable name or path.
- error (value **true** or **false**) indicates whether a warning message should be returned in the case of "not found". **True** will return the warning and is the default.

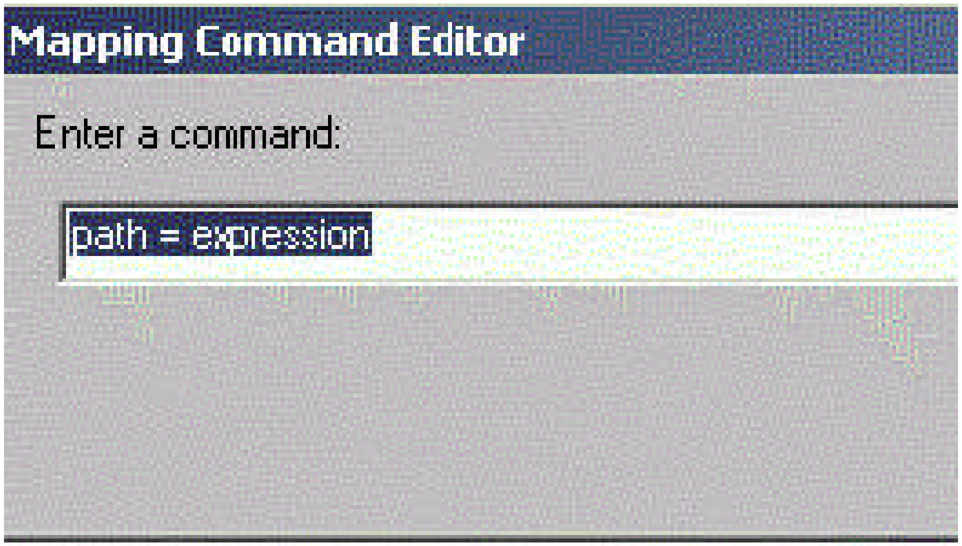
- Since the Validate function returns a boolean **true** or **false**, you will need to use this in conjunction with an If statement. For example:

```
If(Validate("Contractnum",\Table 1\20 M BEG\6 O 367\\))
     mapping command here
Else
     Error(1,5001,"Invalid value specified")
Endif
```

Special Handling in 4.1 for DT maps continued...

★ Translation Table

- Purpose: Translate the value that appears in the field or data element into another value.
- Use the Translate function in conjunction with an Assignment command.



Mapping Command Editor

Enter a command:

```
path = expression
```

- Drag the target field or element to the Mapping Command Editor window and replace "path".
- Replace "expression" with the Translate function:

  Translate("*Tablename*","*Tabledirection*",*value*)

Special Handling in 4.1 for DT maps continued...

Translation Table continued...

Translate("*Tablename","Tabledirection",value*)

where:

- Tabledirection is either **SOURCE** or **TARGET** (or abbreviated to **S** or **T**)
  - For **SOURCE**, DI looks in the *Local Value* column
  - For **TARGET**, DI looks in the *Standards or Trading Partner Value* column
- value is either a variable name or a source path

Translate("*Tablename","Tabledirection",value [,error [, default]]*

where:

- error is either **True** or **False** indicating whether a "not found" condition will result in a warning message. Default is **True**.
- default is a string that provides a default value to return in the event of a "not found" condition.

Replaces:

&E(*localvalue* TS '*Tablename*') or

&E(*stdvalue* TL '*Tablename*')

Special Handling in 4.1 for DT maps continued...

★ Concatenation

- Purpose in send map: Select two or more application fields in a business document you are sending and combine parts of each field into a single data element by specifying the starting position and number of characters to move to the new element. For Send maps only.
- Purpose in DT map: To concatenate two complete character strings together.
- Use the Concat function as follows:

Concat(*string1, string2*)

Special Handling in 4.1 for DT maps continued...

- **Concatenation continued....**
  - Or, use the **Overlay** function to overlay a portion of a string with data from another string.

    Overlay(*string1, string2, position [, length]*)

    where:

    - string1 is the string that will be overlaid with string2
    - string2 is the string that will overlay the data in string1
    - position indicates the position where the overlay will begin in string1
    - length indicates the number of characters to be overlaid in string1. If this optional parameter is omitted or less than 0, all of the characters in string2 will be used to overlay the characters in string1 starting at the specified position.

Special Handling in 4.1 for DT maps continued...

Substring

- Purpose: allows you to weed out unnecessary data received from a trading partner and use only the part you want.
- Use the SubString function as follows:

Substring(*string, position [,length]*)

where:

- string is the input string
- position indicates the starting position of the substring within the *string*
- length indicates the number of characters to be copied into the substring. If omitted, all of the characters will be copied starting at the specified position.

Arithmetic Operators in 4.1 for DT maps

- **CHAR** and **NUMBER** work the same way in mapping commands for data transformation maps

Instead of:

&SET *variable2* &E(CHAR(*variable1*))

it would look like this:

variable2 = CHAR(*variable1*)

Arithmetic Operators in 4.1 for DT maps cont...

★ RD (Round)

- For Send and Receive maps:

  value RD 2

  will round *value* to 2 decimal places or
  will return the first 2 characters from *value* if nonnumeric

- For DT maps:

  Round(*value, decimals*)

  where:

  - value is the value that will be rounded
  - decimals is the number of decimals to round to
  - Note: For nonnumeric values, use the Left function to return the left-most characters

Or Use:

NumFormat(*value,decimals*,"R")

Arithmetic Operators in 4.1 for DT maps cont...

★ TU (Truncate)

- For Send and Receive maps:

  value TU 3

  will truncate *value* to 3 decimal places if numeric or
  will result in the last 3 characters of *value* if nonnumeric

- For DT maps:

  Truncate(*value, decimals*)

  where:

  - value is the value that will be truncated
  - decimals is the number of decimals to truncate to
  - Note: For nonnumeric values, use the Right function to return the right-most characters

  Or Use:

  NumFormat(*value,decimals*,"T")

Special Operators - Scale

SC

- Real Numbers:
  - In send and receive maps: &E(*value* SC *max.decimals*)
    - Truncates to a specified number of decimal places
    - Does not truncate from the left side of the decimal point
  - For DT maps: Use the **Truncate** function to accomplish the same thing

    Truncate(*value, decimals*)
- Strings:
  - In send and receive maps: &E(*string* SC *pos.length*)
    - Works like Substringing returning a *length* characters of the string starting in the given position
  - For DT maps: Use the **Substring** function

    Substring(*string, position [,length]*)

SR

- In send and receive maps: &E(*value* SR *max.decimals*)
  - Works the same was as SC except SR rounds instead of truncates to the specified number of decimals
- For DT maps: Use the **Round** function

  Round(*value, decimals*)

Brand New Functions Available for DT maps

Created

- Checks the target to see if a particular path (or mapping) has data and returns a boolean **True** if it does, or a boolean **False** if it doesn't.

```
Created(targetPath)
```

- Use with an If to insure mandatory elements are populated. For example:

```
If(!Created(\Table 1\20 M BIG\2 M 76\\))
    mapping command
EndIf
```

Brand New Functions Available for DT maps

Find

- Can be used to determine if a string is contained within another string

  Find(*toSearch, searchVal, startPos*)

  where:
  - toSearch is the string to be searched
  - searchVal is the string that is to be searched for within toSearch
  - startPos is the starting position within toSearch where the search will be started

- If the *searchVal* is found in *toSearch*, it returns the starting position where *searchVal* occurs
- Returns a 0 if
  - the requested string is not contained in the searched string
  - the string to be searched is empty
  - the length of the string to be searched is less than the starting position specified

Brand New Functions Available for DT maps

Found

- Checks the source to see if a particular path (or mapping) has data and returns a boolean **True** if it does, or a boolean **False** if it doesn't.

  Found(*sourcePath*)

- Similar to Created function but searches the source instead of target.
- For example:

```
If(!Found(\Table 1\20 M BIG\2 M 76\\))
    mapping command
EndIf
```

Brand New Functions Available for DT maps

★ HexEncode

- Translates binary data into a character string

HexEncode(*binaryval*)

★ HexDecode

- Translates a character string into binary data

HexDecode(*charval*)

★ Length

- Returns the length of a string

Length(*value*)

Brand New Functions Available for DT maps

- **Left**
  - Returns the left-most characters from a string.

    Left(*stringIn, length*)

    where:

    – stringIn is the input string

    – length indicates the number of characters to be copied.

- **Right**
  - Returns the right-most characters from a string.

    Right(*stringIn, length*)

    where:

    – stringIn is the input string

    – length indicates the number of characters to be copied.

Brand New Functions Available for DT maps

★ TrimLeft

- Removes the left-most characters from a string.

  TrimLeft(*stringIn [, trimList]*)

  where:

  - stringIn is the input string to be trimmed
  - trimList includes all the characters that are to be removed. The function ends when the first character not contained in trimList is encountered. If trimList is not included, this function just removes leading whitespace.

★ TrimRight

- Removes the right-most characters from a string.

  TrimRight(*stringIn [, trimList]*)

  where:

  - stringIn is the input string to be trimmed
  - trimList includes all the characters that are to be removed. The function ends when the first character not contained in trimList is encountered. If trimList is not included, this function just removes trailing whitespace.

Brand New Functions Available for DT maps

- **Lower**
  - Converts a string to lowercase

    Lower(*value*)

- **Upper**
  - Converts a string to uppercase

    Upper (*value*)

Brand New Functions Available for DT maps

★ Occurrence

- Returns the current occurrence number of the current repeating compound element or simple element OR the current occurrence number of a specific repeating compound element or simple element.

Occurrence() or Occur() *current
Occurrence(*sourcePath*) or Occur(*sourcePath*) *specific

For example:

```
        If(Occur(\Table 2\10 M PO1 Loop\10 M PO1\1 O 350\\)=2)
            mapping command
        EndIf
or
        If(Occurrence() != 1)
            mapping command
        EndIf
```

Brand New Functions Available for DT maps

NumFormat

- Formats a real number or integer as a character string using a specified number of decimal positions.
- Unused digits are truncated or rounded.

NumFormat(*value, decimals [,flag]*)

where:

- value is the number to be formatted into the character string
- decimals is the number of decimal places to be included in the character string
- flag is either "ROUND" or "TRUNCATE" (can be shortened to "R" or "T"). If not supplied, the default is "ROUND".

Brand New Functions Available for DT maps

StrComp and StrCompI

- Compares two character strings
- The StrCompI function compares the strings with regards to case

StrComp(*string1, string2*)
StrCompI(*string1, string2*)

- Returns:
  - -1 if *string1 < string2*
  - 0 if *string1 = string2*
  - 1 if *string1 > string2*

```
If(StrComp(\Table 1\40 M N1 Loop\40 M N1\1 M 98\\,'BY')=0)
        mapping command
EndIf
```

Brand New Functions Available for DT maps

StrCompN and StrCompNI

- Compares the first *length* characters of two character strings
- The StrCompNI function compares the first *length* characters of the strings with regards to case

StrCompN(*string1, string2, length*)
StrCompNI(*string1, string2, length*)

Where:

- length is the number of characters to be compared

- Returns:
  - -1 if the first *length* characters of *string1* < the first *length* characters of *string2*
  - 0 if the first *length* characters of the two strings are equal
  - 1 if the first *length* characters of *string1* > the first *length* characters of *string2*

Generic Service Element Field Cross-Reference

Interchange

Receive map (retrieve only)	Data Transformation Maps (to retrieve)	Data Transformation Maps (to set)	Description
&I	GetProperty("Ichg")	SetProperty("Ichg")	The entire envelope interchange segment
&ICN	GetProperty("IchgCtlNum")	SetProperty("IchgCtlNum")	Interchange Control Number
&IIS	GetProperty("IchgSndrid")	SetProperty("IchgSndrid")	Interchange Sender ID
&IIR	GetProperty("IchgRcvrid")	SetProperty("IchgRcvrid")	Interchange Receiver ID
&IDT	GetProperty("IchgDate")	SetProperty("IchgDate")	Interchange Date
&ITM	GetProperty("IchgTime")	SetProperty("IchgTime")	Interchange Time
&IPW	GetProperty("IchgPswd")	SetProperty("IchgPswd")	Interchange Password
not available	GetProperty("IchgUsgInd")	SetProperty("IchgUsgInd")	Interchange Usage Indicator
&IAP	GetProperty("IchgAppRef")	SetProperty("IchgAppRef")	Interchange Application Reference
&IVR	GetProperty("IchgVerRel")	SetProperty("IchgVerRel")	Interchange Version/Release
&IGT	GetProperty("IchgGrpCnt")	SetProperty("IchgGrpCnt")	Interchange Group Count
&ICT	GetProperty("IchgCtlTotal")	SetProperty("IchgCtlTotal")	Interchange Control Total
&ITT	GetProperty("IchgTrxCnt")	SetProperty("IchgTrxCnt")	Interchange Transaction Count
&ISA*nn*	GetProperty("ISA*nn*")	SetProperty("ISA*nn*")	A specific ISA element *nn*
&IEA*nn*	GetProperty("IEA*nn*")	SetProperty("IEA*nn*")	A specific IEA element *nn*

Generic Service Element Field Cross-Reference

Group

Receive map (retrieve only)	Data Transformation Maps (to retrieve)	Data Transformation Maps (to set)	Description
&G	GetProperty("Grp")	SetProperty("Grp")	Entire envelope group segment
&GCN	GetProperty("GrpCtlNum")	SetProperty("GrpCtlNum")	Group Control Number
&GFG	GetProperty("GrpFuncGrpId")	SetProperty("GrpFuncGrpId")	Functional Group ID
&GAS	GetProperty("GrpAppSndrId")	SetProperty("GrpAppSndrId")	Group Application Sender ID
&GAR	GetProperty("GrpAppRcvrId")	SetProperty("GrpAppRcvrId")	Group Application Receiver ID
&GDT	GetProperty("GrpDate")	SetProperty("GrpDate")	Group Date
&GTM	GetProperty("GrpTime")	SetProperty("GrpTime")	Group Time
&GPW	GetProperty("GrpPswd")	SetProperty("GrpPswd")	Group password
&GVR	GetProperty("GrpVer")	SetProperty("GrpVer")	Group Version
&GLV	GetProperty("GrpRel")	SetProperty("GrpRel")	Group Release
&GTT	GetProperty("GrpTrxCnt")	SetProperty("GrpTrxCnt")	Number of transactions in group
&GS*nn*	GetProperty("GS*nn*")	SetProperty("GS*nn*")	A specific GS element *nn*
&GE*nn*	GetProperty("GE*nn*")	SetProperty("GE*nn*")	A specific GE element *nn*

Generic Service Element Field Cross-Reference

Transaction

Receive map (retrieve only)	**Data Transformation Maps (to retrieve)**	**Data Transformation Maps (to set)**	**Description**
&T	GetProperty("Trx")	SetProperty("Trx")	Entire envelope transaction header segment
&TCN	GetProperty("TrxCtlNum")	SetProperty("TrxCtlNum")	Transaction Control Number
&TTC	GetProperty("TrxCode")	SetProperty("TrxCode")	Transaction Code
&TVR	GetProperty("TrxVer")	SetProperty("TrxVer")	Transaction Version
&TLV	GetProperty("TrxRel")	SetProperty("TrxRel")	Transaction Release
&TTS	GetProperty("TrxSegCnt")	SetProperty("TrxSegCnt")	Number of segments in the transaction
&ST*nn*	GetProperty("ST*nn*")	SetProperty("ST*nn*")	A specific ST element *nn*
&SE*nn*	GetProperty("SE*nn*")	SetProperty("SE*nn*")	A specific SE element *nn*

Questions?