

Target Server Support for Macros {ewc HLP25632,HLP256_TILE,water.bmp}

You can use ERwin macros in the templates you create for triggers, stored procedures, and pre-schema and post-schema generation scripts when these features are supported by the target DBMS.

All macros, except those specifically noted, can be used on all of the target servers supported by ERwin. See your server documentation for detailed information on its support for RI triggers, stored procedures, and/or scripts.

Using the Macro Reference {ewc HLP25632,HLP256_TILE,water.bmp}

The macro reference information is organized alphabetically by macro name. The information for each macro includes a description, macro syntax, the *scope* of the macro (i.e., the statements or code segments in which it can be used), the return value; and, typically, an example of the macro template code and corresponding expanded code. A description of each element follows:

- n **Macro Syntax.** Provides the syntax for each ERwin macro, including keywords, user-specified variables, and required punctuation.
- n **Scope.** For each macro, this element lists:
 - n The "%ForEach" commands in which a given macro can be inserted.
 - n Whether the macro can be used as the predicate in a conditional "%If" statement.
 - n The template types in which a given macro can be inserted. Entries include: *RI or Rel Override*, *Trigger Override*, and *Global*. The specific templates of each type are listed in the following table:

Scope	Templates	
RI or Rel Override	PARENT INSERT RESTRICT	CHILD INSERT RESTRICT
	PARENT INSERT CASCADE	CHILD INSERT CASCADE
	PARENT INSERT SET NULL	CHILD INSERT SET NULL
	PARENT INSERT SET DEFAULT	CHILD INSERT SET DEFAULT
	PARENT UPDATE RESTRICT	CHILD UPDATE RESTRICT
	PARENT UPDATE CASCADE	CHILD UPDATE CASCADE
	PARENT UPDATE SET NULL	CHILD UPDATE SET NULL
	PARENT UPDATE SET DEFAULT	CHILD UPDATE SET DEFAULT
	PARENT DELETE RESTRICT	CHILD DELETE RESTRICT
	PARENT DELETE CASCADE	CHILD DELETE CASCADE
	PARENT DELETE SET NULL	CHILD DELETE SET NULL
	PARENT DELETE SET DEFAULT	CHILD DELETE SET DEFAULT
	Stored procedures attached to tables (macros used within %ForEachFKAtt); pre- and post-scripts (macros used within %ForEachEntity).	
Trigger Override	TRIGGER UPDATE HEADER	TRIGGER INSERT FOOTER
	TRIGGER DELETE HEADER	CUSTOM TRIGGER HEADER
	TRIGGER INSERT HEADER	CUSTOM TRIGGER FOOTER
	TRIGGER UPDATE FOOTER	CUSTOM TRIGGER DEFAULT BODY
	TRIGGER DELETE FOOTER	
	Stored procedures attached to tables (macros used within %ForEachAtt); pre- and post-scripts (macros used within %ForEachEntity).	
Global	All triggers, stored procedures, and pre- and post-scripts.	

- n **Return Value.** Explains what SQL code the macro returns when it is generated.
- n **Example.** Includes sample template syntax and the corresponding expanded code. All expanded code examples are based on a subset of the tables in the ERwin MOVIES model (SQL Server target) shown in the ERwin Template Toolbox. An example is not included if the macro returns TRUE or FALSE.

Note: The sample model above shows the physical names for both tables and columns, indicated by the underscore character in the object names for example, MOVIE_COPY, movie_copy_number. Different ERwin macros are available to return either the ERwin logical name (which can contain hyphens or spaces) or database physical names (in which hyphens and spaces are replaced by underscores) depending on your requirements.

Reading ERwin Macro Syntax {ewc HLP25632,HLP256_TILE,water.bmp}

The ERwin macro syntax commands are composed of keywords, punctuation, and variables.

- n *Keywords* always start with a percent sign (%) and must be entered exactly as shown in the table (e.g. %AttFieldName).
- n Braces {} and parentheses () in the syntax must be included; commas are required if you include more than one variable in the command.
- n *Variables* are shown in angle brackets (e.g., <macro code1>). You replace variables with your own specific text, number, or macro code when you type the macro command in a trigger, stored procedure, or script.

Macro Variables {ewc HLP25632,HLP256_TILE,water.bmp}

The following list defines common variables used in the macro syntax:

- n <default value>, <value>, and <initial value> represent a user-specified text or numeric value, which appears exactly as written when the macro code is expanded.
- n < action> represents a valid SQL action (i.e., INSERT, UPDATE, or DELETE).
- n <macro code>, <macro code1>...<macro coden> represents any macro code sequence that is valid in the current scope. When multiple macro code sequences can be included in the same command, they are numbered from 1 to n.
- n <predicate> represents any valid Boolean expression that evaluates to TRUE or FALSE, including those supported directly by an ERwin macro (e.g., %AttIsFK) and those created using supported Boolean operators (e.g., %!= and other comparison operators, %And, %Or, and %Not).
- n <prefix>, <old prefix>, <new prefix> represents a text or numeric value (e.g., @ins) or expanded macro (e.g., %Parent) that is added to the beginning of a table or column name when the macro code is expanded. If both old and new prefix elements are specified, the list created by the macro has two sections: The first section declares <old prefix> <element>, the second section declares <new prefix> <element>.
- n <separator> represents one or more characters used to delimit a list created by the macro. Separators are set off by quotation marks (e.g., "OR ").
- n <table name>, <default name>, <domain name>, <validation> are replaced by the name of an ERwin table, default, domain, or validation rule or the corresponding macro (e.g., %TableName, %Child, %DefaultName, %DomainName, %ValidationName).
- n <variable> represents the name of a variable used within the trigger, stored procedure, or script.

%!= {ewc HLP25632,HLP256_TILE,water.bmp}

A Boolean predicate that returns "true" if the expansion of <macro code1> is not equal to the expansion of <macro code2>.

`%!=(<macro code1>, <macro code2>)`

Scope

Global, %If

Return Value

TRUE/FALSE

Example

Template Code

```
%ForEachAtt() {%If(%!=(%AttFK) {%AttName})}
```

Expanded Code

```
general-condition
```

%% {ewc HLP25632,HLP256_TILE,water.bmp}

Inserts a single percent (%) symbol in expanded trigger code.

`%%`

Scope

Global

Return Value

`%`

Example

Template Code

```
/* %%JoinFKPK
```

Expanded Code

```
/*%JoinFKPK
```

%* {ewc HLP25632,HLP256_TILE,water.bmp}

Multiplies the expansions of <macro code1> and <macro code2>.

`%*(<macro code1>,<macro code2>)`

Scope

Global

Return Value

Numeric

Example

Template Code

```
%*(5,9)
```

Expanded Code

```
45
```

```
%Decl(string1,aaaaa)
```

```
%Decl(string2,bbbbbbbbb)
```

```
%*(%Len(:string1),
```

```
%Len(:string2))
```

```
45
```

%+ {ewc HLP25632,HLP256_TILE,water.bmp}

Adds the expansions of <macro code1> and <macro code2>.

%+(<macro code1>,<macro code2>)

Scope

Global

Return Value

Numeric

Example

Template Code	Expanded Code
%+ (5, 9)	14
 %Decl (string1,aaaaa)	 14
%Decl (string2,bbbbbbbbbb)	
%+ (%Len(:string1),	
%Len(:string2))	

%- {ewc HLP25632,HLP256_TILE,water.bmp}

Subtracts the expansions of <macro code2> from <macro code1>.

%-(<macro code1>,<macro code2>)

Scope

Global

Return Value

Numeric

Example

Template Code	Expanded Code
%- (5, 9)	-4
 %Decl (string1,aaaaa)	 -4
%Decl (string2,bbbbbbbbbb)	
%- (%Len(:string1),	
%Len(:string2))	

%/ {ewc HLP25632,HLP256_TILE,water.bmp}

Divides the expansions of <macro code1> by <macro code2>.

%/(<macro code1>,<macro code2>)

Scope

Global

Return Value

Numeric

Example

Template Code	Expanded Code
%/ (5, 9)	.5556

```

%Decl(string1,aaaaa)                .5556
%Decl(string2,bbbbbbbbb)
%/(%Len(%:string1),
%Len(%:string2))

```

%: {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the value of <variable>.

:%:<variable>

Scope

Global

Return Value

Value of the variable <variable>.

Example

Template Code	Expanded Code
:%:@ins_customer_id	1213

%< {ewc HLP25632,HLP256_TILE,water.bmp}

A Boolean predicate that returns "true" if the expansion of <macro code1> is less than the expansion of <macro code2>.

%<(<macro code1>, <macro code2>)

Scope

Global, %If

Return Value

TRUE/FALSE

%<= {ewc HLP25632,HLP256_TILE,water.bmp}

A Boolean predicate that returns "true" if the expansion of <macro code1> is less than or equal to the expansion of <macro code2>.

%<= (<macro code1>, <macro code2>)

Scope

Global, %If

Return Value

TRUE/FALSE

%= {ewc HLP25632,HLP256_TILE,water.bmp}

Assigns the expansion of <macro code> to a declared <variable> (See [%Decl](#)). The result of the statement is variable assignment only, the result does not appear in the expanded code.

`%=(<variable>,<macro code>)`

Scope

Global

Return Value

None

Example

Template Code

`%=(@var1,%Tablename)`

Expanded Code

Internally, @var1 is set to the value of %Tablename

%== {ewc HLP25632,HLP256_TILE,water.bmp}

A Boolean predicate that returns "true" if the expansion of <macro code1> is equal to the expansion of <macro code2>.

`%==(<macro code1>, <macro code2>)`

Scope

Global, %If

Return Value

TRUE/FALSE

%> {ewc HLP25632,HLP256_TILE,water.bmp}

A Boolean predicate that returns "true" if the expansion of <macro code1> is greater than the expansion of <macro code2>

%>(<macro code1>, <macro code2>)

Scope

Global, %If

Return Value

TRUE/FALSE

%>= {ewc HLP25632,HLP256_TILE,water.bmp}

A Boolean predicate that returns "true" if the expansion of <macro code1> is greater than or equal to the expansion of <macro code2>

%>=(<macro code1>, <macro code2>)

Scope

Global, %If

Return Value

TRUE/FALSE

%Action {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the name of the action that caused the trigger to fire. Depending on the context of the command, %Action will return DELETE, INSERT, or UPDATE. For example, if the trigger was fired on a delete action, %Action returns DELETE.

`%Action`

Scope

Trigger Override

Return Value

DELETE, INSERT, or UPDATE

Example

Template Code	Expanded Code
<code>%Action</code>	<code>DELETE</code>

%Actions {ewc HLP25632,HLP256_TILE,water.bmp}

Returns a list of actions that were implemented either before or after the trigger fired. If the trigger is a multiple action trigger, the macro reports all actions.

`%Actions (<separator>)`

Scope

Trigger Override

Return Value

DELETE, INSERT, and/or UPDATE

Example

Template Code	Expanded Code
<code>%Actions(",")</code>	<code>DELETE,DELETE</code>

%And {ewc HLP25632,HLP256_TILE,water.bmp}

Lets you combine Boolean expressions to create more complex comparisons. %And performs a "logical and" on Boolean predicates defined by <macro code1> and <macro code2>. For example, the code %If {%And(%AttlsFK, %AttlsRolenamed)...} lets you execute additional steps if the given attribute is both a foreign key and has a rolename. Both elements must evaluate to "TRUE" for the %And condition to be satisfied.

`%And(<macro code1>, <macro code2>)`

Scope

Global, %If

Return Value

TRUE/FALSE

%AttDatatype {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the datatype of the current attribute. *Note:* Attributes can be assigned a user-defined datatype using ERwin domains. If an attribute has a user-defined datatype, the name of the datatype is provided by this macro.

`%AttDatatype`

Scope

`%ForEachAtt, %ForEachFKAtt`

Return Value

Attribute datatype

Example

Template Code

```
%ForEachAtt() {%AttName ->
%AttDatatype}
```

Expanded Code

```
master-number -> int
movie-copy-number -> int
general-condition -> varchar(10)
```

%AttDef {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the attribute definition

`%AttDef`

Scope

`%ForEachAtt, %ForEachFKAtt`

Return Value

Attribute definition.

Example

Template Code

```
/*If the current att. is
master_number, then */
%AttDef
```

Expanded Code

```
The unique id of a movie.
```

%AttDefault {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the name of the default attached to the attribute.

`%AttDefault`

Scope

`%ForEachAtt, %ForEachFKAtt`

Return Value

Name of the default attached to the attribute.

Example

Template Code

```
%AttDefault
```

Expanded Code

```
SPACES
```

%AttDomain {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the name of the domain attached to the attribute.

`%AttDomain`

Scope

`%ForEachAtt, %ForEachFKAtt`

Return Value

Name of the domain.

Example

Template Code	Expanded Code
<pre>/*If the current att. is cust_phone, then */ %AttDomain</pre>	<pre>PHONE NUMBER</pre>

%AttFieldName {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the column name associated with the current attribute.

`%AttFieldName`

Scope

`%ForEachAtt, %ForEachFKAtt`

Return Value

Column name.

Example

Template Code	Expanded Code
<pre>/*If the current att. is renting-customer, then */ %AttFieldName</pre>	<pre>renting_customer</pre>

%AttFieldWidth {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the length of the attribute datatype

`%AttFieldWidth`

Scope

`%ForEachAtt, %ForEachFKAtt`

Return Value

Numeric

Example

Template Code	Expanded Code
<pre>/* For a datatype of varchar(50) */ %AttFieldWidth</pre>	<pre>50</pre>

%AttId {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the attribute ID.

%AttId

Scope

%ForEachAtt, %ForEachFKAtt

Return Value

Integer

%AttIsFK {ewc HLP25632,HLP256_TILE,water.bmp}

A Boolean predicate that lets you determine whether or not the current attribute is a foreign key member.

`%AttIsFK`

Scope

`%ForEachAtt, %ForEachFKAtt, %If`

Return Value

TRUE/FALSE

Example

Template Code

```
%ForEachAtt() {  
    %If{  
        %AttIsFK}  
    %AttName}
```

Expanded Code

```
master-number movie-copy-number
```

%AttIsRolenamed {ewc HLP25632,HLP256_TILE,water.bmp}

A Boolean predicate that lets you determine whether or not the current attribute is rolenamed. See [%AttIsFK](#) for expansion information.

`%AttIsRolenamed`

Scope

`%ForEachAtt, %ForEachFKAtt, %If`

Return Value

TRUE/FALSE

%AttIsPK {ewc HLP25632,HLP256_TILE,water.bmp}

A Boolean predicate that lets you determine whether or not the current attribute is a primary key member. See [%AttIsFK](#) for expansion information.

%AttIsPK

Scope

%ForEachAtt, %ForEachFKAtt, %If

Return Value

TRUE/FALSE

%AttName {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the logical name of the current attribute.

`%AttName`

Scope

`%ForEachAtt, %ForEachFKAtt`

Return Value

Attribute name.

Example

Template Code	Expanded Code
<pre>/*If the current att. is master- number, then */ %AttName</pre>	<pre>master-number</pre>

%AttNullOption {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the null option of the current attribute.

`%AttNullOption`

Scope

`%ForEachAtt, %ForEachFKAtt`

Return Value

NULL/NOT NULL

%AttPhysDatatype {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the physical datatype of the current attribute, even for user-defined datatypes.

`%AttPhysDatatype`

Scope

`%ForEachAtt, %ForEachFKAtt`

Return Value

Datatype

Example

Template Code	Expanded Code
<pre>/*If the current att. is master- number, then */ %AttPhysDatatype</pre>	<pre>int</pre>

%AttProp {ewc HLP25632,HLP256_TILE,water.bmp}

Expands to the user-defined property (UDP) values of the attribute name specified.

`%AttProp()`

Scope

`%ForEachAtt, %ForEachFKAtt`

Return Value

Attribute UDP value

Example

Template Code	Expanded Code
<pre>%AttProp(Status)</pre>	<pre>Temporary attribute</pre>

%Atts {ewc HLP25632,HLP256_TILE,water.bmp}

Lists all the attributes of the trigger entity, performing the specified function on each member.

`%Atts(<separator>,<action>,<prefix>)`

Scope

Trigger Override

Return Value

List of actions and attributes.

Example

Template Code

```
%Atts(" ,
      ",update,%Parent)
```

Expanded Code

```
update(MOVIE_COPY.master_number),
update(MOVIE_COPY.movie_copy_number)
```

%AttValidation {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the name of the validation attached to a given attribute.

`%AttValidation`

Scope

%ForEachAtt, %ForEachFKAtt

Return Value

Validation Name

Example

Template Code

```
/*If the current att. is
due_date, then */
%AttValidation
```

Expanded Code

```
DUE_DATE_RULE
```

%Cardinality {ewc HLP25632,HLP256_TILE,water.bmp}

Returns relationship cardinality:

- n *<nothing>* = 1:0,1, or more
- n P = 1:1 or more
- n Z = 1:0 or 1
- n *n* = 1:n, where *n* is an integer.

`%Cardinality`

Scope

RI or Rel Override

Return Value

<nothing>, P, Z, *<any integer>*

%Child {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the physical table name of the child entity in a relationship.

`%Child`

Scope

RI or Rel Override

Return Value

Table name.

Example

Template Code	Expanded Code
<code>/*If the current relationship is</code>	<code>MOVIE_RENTAL_RECOR</code>
<code><is in stock as>, then */</code>	
<code>%Child</code>	

%ChildAtts {ewc HLP25632,HLP256_TILE,water.bmp}

Lists all the attributes of the child entity in a relationship, and performs the specified action on each.

`%ChildAtts(<separator>,<action>,<prefix>)`

Scope

RI or Rel Override

Return Value

Attribute names and actions.

Example

Template Code	Expanded Code
<code>/*If the current relationship is</code>	<code>update(movie_copy_number) or</code>
<code><is in stock as>, then */</code>	<code>update(master_number) or</code>
<code>%ChildAtts(" or ",update)</code>	<code>...</code>

%ChildFK {ewc HLP25632,HLP256_TILE,water.bmp}

Lists the foreign keys of the child entity in a relationship and performs the specified action on each.

`%ChildFK(<separator>,<action> <prefix>)`

Scope

RI or Rel Override

Return Value

Attribute names and actions.

Example

Template Code	Expanded Code
<code>/*If the current relationship is</code>	<code>update(master_number) or</code>
<code><is in stock as>, then */</code>	<code>update(movie_copy_number)</code>
<code>%ChildFK(" or ",update)</code>	

%ChildFKDecl {ewc HLP25632,HLP256_TILE,water.bmp}

Lists the foreign keys of the child entity in the relationship followed by their datatypes.

```
%ChildFKDecl(<old prefix>, <new  
prefix>,<separator>>,<attribute/type separator>)
```

Scope

RI or Rel Override

Return Value

FK attribute names and datatypes

Example

Template Code

```
/*If the current relationship is  
<is in stock as>, then */  
%ChildFKDecl(,@ins_," , ")
```

Expanded Code

```
@ins_master_number int,  
@ins_movie_copy_number int
```

%ChildNK {ewc HLP25632,HLP256_TILE,water.bmp}

Lists non-key elements of the child entity and performs the specified action on each.

```
%ChildNK(<separator>,<action>,<prefix>)
```

Scope

RI or Rel Override

Return Value

Non-key attribute names and action

Example

Template Code

```
/*If the current relationship is  
<is in stock as>, then */  
%ChildNK(" or ",update,@)
```

Expanded Code

```
update(@general_condition)
```

%ChildNKDecl {ewc HLP25632,HLP256_TILE,water.bmp}

Lists the non-key elements of the child entity in the relationship followed by their datatypes.

```
%ChildNKDecl(<old prefix>, <new  
prefix>,<separator>>,<attribute/type separator>)
```

Scope

RI or Rel Override

Return Value

Non-key attribute names and datatypes

Example

Template Code

```
/*If the current relationship is  
<is in stock as>, then */  
%ChildNKDecl(,@ins_," , ")
```

Expanded Code

```
@ins_general_condition  
varchar(10)
```

%ChildParamDecl {ewc HLP25632,HLP256_TILE,water.bmp}

Lists all the attributes of the child entity in the relationship followed by their datatypes. This is useful when declaring parameters in stored procedures.

```
%ChildParamDecl(<old prefix>, <new prefix>,  
<separator>>,<attribute/type separator>)
```

Scope

RI or Rel Override

Return Value

Attribute names and datatypes

Example

Template Code

```
/*If the current relationship is  
<is in stock as>, then */  
%ChildParamDecl(@ins_," ")
```

Expanded Code

```
@ins_master_number int,  
@ins_movie_copy_number int,  
@ins_general_condition  
varchar(10),
```

%ChildPK {ewc HLP25632,HLP256_TILE,water.bmp}

Lists primary key elements of the child entity, and performs the specified action on each.

```
%ChildPK(<separator>,<action>, <prefix>)
```

Scope

RI or Rel Override

Return Value

Non-key attribute names and actions

Example

Template Code

```
/*If the current relationship is  
<is in stock as>, then */  
%ChildPK(" or ",update,@)
```

Expanded Code

```
update(@movie_copy_number) or  
update(@master_number)
```

%ChildPKDecl {ewc HLP25632,HLP256_TILE,water.bmp}

Lists the primary key elements of the child entity in the relationship followed by their datatypes.

```
%ChildPKDecl(<old prefix>, <new  
prefix>,<separator>>,<attribute/type separator>)
```

Scope

RI or Rel Override

Return Value

Attribute names and datatypes

Example

Template Code

```
/*If the current relationship is  
<is in stock as>, then */
```

Expanded Code

```
@ins_master_number int,  
@ins_movie_copy_number int,
```

```
%ChildPKDecl(,@ins_," ")
```

%ColProp {ewc HLP25632,HLP256_TILE,water.bmp}

Expands to the user-defined property (UDP) values of the column name specified.

```
%ColProp()
```

Scope

%ForEachAtt, %ForEachFKAtt

Return Value

Column UDP value

Example

Template Code

```
%ColProp(Status)
```

Expanded Code

```
Temporary
```


%Concat {ewc HLP25632,HLP256_TILE,water.bmp}

Concatenates <value1> and <value2> and returns the result.

%Concat (<value1>,<value2>)

Scope

Global

Return Value

String

%CurrentDatabase {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the name of the database in use when the script was generated, as entered in the <DB> Connection dialog.

`%CurrentDatabase`

Scope

Global

Return Value

String

Example

Template Code	Expanded Code
<code>%CurrentDatabase</code>	<code>testdb</code>

%CurrentFile {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the name of the ERwin file from which the script was generated.

`%CurrentFile`

Scope

Global

Return Value

String

Example

Template Code	Expanded Code
<code>%CurrentFile</code>	<code>C:\MYFILES\MOVIES.ER1</code>

%CurrentServer {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the name of the target server in use when the script was generated, as entered in the <DB> Connection dialog.

`%CurrentServer`

Scope

Global

Return Value

String

Example

Template Code	Expanded Code
<code>%CurrentServer</code>	<code>SQL6</code>

%CurrentUser {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the name of the user who generated the script, as entered in the <DB> Connection dialog.

`%CurrentUser`

Scope

Global

Return Value

String

Example

Template Code

```
%CurrentUser
```

Expanded Code

```
ssmith
```

%CustomTriggerDefaultBody {ewc HLP25632,HLP256_TILE,water.bmp}

Inserts the code from the trigger body template into the trigger in expanded form.

```
%CustomTriggerDefaultBody
```

Scope

Trigger Override

Return Value

All code contained in the CustomTriggerDefaultBody trigger template.

%CustomTriggerDefaultFooter {ewc HLP25632,HLP256_TILE,water.bmp}

Inserts the code from the trigger footer template into the trigger in expanded form.

`%CustomTriggerDefaultFooter`

Scope

Trigger Override

Return Value

All code contained in the CustomTriggerDefaultFooter trigger template.

%CustomTriggerDefaultHeader {ewc HLP25632,HLP256_TILE,water.bmp}

Inserts the code from the trigger header template into the trigger in expanded form.

%CustomTriggerDefaultHeader

Scope

Trigger Override

Return Value

All code contained in the CustomTriggerDefaultHeader trigger template.

%DatatypeName {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the name of the datatype.

```
%DatatypeName()
```

Scope

Global

Return Value

Datatype

Example

Template Code	Expanded Code
<pre>%ForEachAtt(%Tablename," ") {</pre>	<pre>varchar</pre>
<pre>%DatatypeName(%AttPhysDatatype) }</pre>	

%DatatypeScale {ewc HLP25632,HLP256_TILE,water.bmp}

For decimal datatypes, %DatatypeScale returns the number of decimal places.

```
%DatatypeScale()
```

Scope

Global

Return Value

Integer

Example

Template Code	Expanded Code
<pre>/* For example, in</pre>	<pre>2</pre>
<pre>decimal(10,2) */</pre>	
<pre>%DatatypeScale()</pre>	

%DatatypeWidth {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the field width of the datatype.

```
%DatatypeWidth()
```

Scope

Global

Return Value

Integer

Example

Template Code	Expanded Code
<pre>/* For example, in</pre>	<pre>10</pre>
<pre>decimal(10,2) */</pre>	
<pre>%DatatypeWidth()</pre>	

%Datetime {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the current date and time.

`%Datetime`

Scope

Global

Return Value

String

Example

Template Code

`%Datetime`

Expanded Code

Wed Feb 17 12:56:22 1999

%DBMS {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the database name.

`%DBMS`

Scope

Global

Return Value

String

Example

Template Code

`%DBMS`

Expanded Code

ORACLE

%DBMSDelim {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the statement delimiter of the DBMS.

`%DBMSDelim`

Scope

Global

Return Value

, or ; or other delimiter

%DBMSTriggerDelim {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the trigger delimiter strings of the DBMS.

%DBMSTriggerDelim

Scope

Global

Return Value

, go; or other server specific delimiter

%Decl {ewc HLP25632,HLP256_TILE,water.bmp}

Declares a <variable>and, optionally, initializes it with <initial value>. For example, if you include the %Decl(var1,0) statement, on expansion, var1 is declared and initialized with the value 0. No expansion code is generated.

```
%Decl(<variable>,<initial value>)
```

Scope

Global

Return Value

None

Example

Template Code

```
%Decl(var1,0)
```

Expanded Code

Internally, var1 is set to the value of 0.

%DefaultName {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the default's name.

```
%DefaultName
```

Scope

%ForEachDefault

Return Value

Name of the default.

Example

Template Code

```
%ForEachDefault(", "){  
%DefaultName}
```

Expanded Code

```
SPACES, PERSON-NAME, ADDRESS
```

%DefaultValue {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the default's value.

```
%DefaultValue(<default name>) or %DefaultValue
```

Scope

Global with an argument (<default name>) or %ForEachDefault without an argument

Return Value

Value of the default.

Example

Template Code

```
%ForEachDefault(", "){  
%DefaultValue}
```

Expanded Code

```
NEW RELEASE, COMEDY, DRAMA,  
CHILDREN, HORROR
```

%DiagramProp {ewc HLP25632,HLP256_TILE,water.bmp}

Expands to the user-defined property (UDP) values of the diagram name specified.

`%DiagramProp()`

Scope

Global

Return Value

Diagram UDP value

Example

Template Code	Expanded Code
<code>%DiagramProp(Department)</code>	Accounting

%DomainDatatype {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the physical datatype of the domain.

`%DomainDatatype(<domain name>)` or `%DomainDatatype`

Scope

Global with an argument (<domain name>) or `%ForEachDomain`

Return Value

Datatype

Example

Template Code

```
%ForEachDomain(", "){  
  %DomainDatatype}
```

Expanded Code

```
varchar(10), int, char
```

%DomainDef {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the name of the default attached to the domain.

`%DomainDef(<domain name>)` or `%DomainDef`

Scope

Global with an argument (<domain name>) or `%ForEachDomain`

Return Value

Domain's default value.

Example

Template Code

```
%ForEachDomain(", "){  
  %DomainDef}
```

Expanded Code

```
NEW RELEASE
```

%DomainDefault {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the name of the default value attached to an ERwin domain.

`%DomainDefault`

Scope

`%ForEachDomain`

Return Value

Default domain name

Example

Template Code

```
%ForEachDomain() {  
  %DomainDefault  
}
```

Expanded Code

```
Address
```

%DomainLogProp {ewc HLP25632,HLP256_TILE,water.bmp}

Expands to the user-defined property (UDP) values of the logical domain specified.

`%DomainLogProp`

Scope

`%ForEachDomain`, `%ForEachAtt`, `%ForEachFKAtt`

Return Value

Logical Domain UDP value

Example

Template Code	Expanded Code
<code>%DomainLogProp(description)</code>	<code>Standard street address</code>

%DomainName {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the name of the domain.

`%DomainName`

Scope

`%ForEachDomain`

Return Value

Domain name.

Example

Template Code	Expanded Code
<code>%ForEachDomain{ %DomainName}</code>	<code>PERSON NAME ADDRESS</code>

%DomainNullOrOption {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the null option of the domain.

`%DomainNullOrOption(<domain name>)` or `%DomainNullOrOption`

Scope

Global with an argument (`<domain name>`) or `%ForEachDomain`

Return Value

NULL, NOT NULL, IDENTITY, WITH NULL, or NOT NULL WITH DEFAULT

%DomainPhysProp {ewc HLP25632,HLP256_TILE,water.bmp}

Expands to the user-defined property (UDP) values of the physical domain specified.

`%DomainPhysProp()`

Scope

`%ForEachDomain, %ForEachAtt, %ForEachFKAtt`

Return Value

Physical Domain UDP value

Example

Template Code	Expanded Code
<code>%DomainPhysProp(description)</code>	<code>text column</code>

%DomainValidation {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the name of a validation attached to the domain.

`%DomainValidation(<domain name>)` or `%DomainValidation`

Scope

Global with an argument (<domain name>) or %ForEachDomain

Return Value

Domain name.

Example

Template Code	Expanded Code
<code>%ForEachDomain{ %DomainValidation}</code>	<code>IS VALID CUSTOMER, DUE DATE</code>

%EntityId {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the ID of the entity or table.

`%EntityId(<entity or tablename>)`

Scope

RI or Rel OverrideTrigger Override

Return Value

Integer

Example

Template Code	Expanded Code
<code>%EntityId(%Parent)</code>	<code>1234</code>

%EntityName {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the name of the entity or table.

`%EntityName(<entity or tablename>)`

Scope

RI or Rel OverrideTrigger Override

Return Value

String

Example

Template Code	Expanded Code
<code>%EntityName(%Parent)</code>	<code>MOVIE-COPY</code>

%EntityProp {ewc HLP25632,HLP256_TILE,water.bmp}

Expands to the user-defined property (UDP) values of the entity name specified.

`%EntityProp()`

Scope

%ForEachParentRel, %ForEachChildRel, RI or REL override, Trigger Override

Return Value

Entity UDP value

Example

Template Code

`%EntityProp(DeveloperName)`

Expanded Code

John Smith

%File {ewc HLP25632,HLP256_TILE,water.bmp}

Expands the macro code within the braces and writes the result to the given file. This macro provides a workaround to the Windows 3.x 64K string size limit.

```
%File(<filename>) {<macro code>}
```

Scope

Global

Return Value

String

Example

Template Code	Expanded Code
<pre>%File(C:\erw\myfile.tst) {%ForEachAtt(Parent, ", ") {%AttFieldName} ...}</pre>	<pre><creates a file and includes the specified information></pre>

%Fire {ewc HLP25632,HLP256_TILE,water.bmp}

Specifies when the trigger is fired. Available on INFORMIX, ORACLE 7/8.x, and Rdb only.

```
%Fire
```

Scope

Trigger Override

Return Value

BEFORE or AFTER

Example

Template Code	Expanded Code
<pre>%Fire</pre>	<pre>BEFORE</pre>

%ForEachAtt {ewc HLP25632,HLP256_TILE,water.bmp}

Expands macro code for each of the attributes in the given table.

```
%ForEachAtt(<table name>,<separator>) { <macro code> }
```

Scope

Trigger Override

Return Value

Result of <macro code> for each attribute in a given table.

Example

Template Code	Expanded Code
<pre>%ForEachAtt(%Parent, ", ") {%AttFieldName}</pre>	<pre>master_number, movie_copy_number, general_condition</pre>

%ForEachChildRel {ewc HLP25632,HLP256_TILE,water.bmp}

Expands the <relationship code> for every relationship in which the trigger entity is a child.

```
%ForEachChildRel(<separator>) {<relationship code>}
```

Scope

Trigger Override

Return Value

Result of <macro code> for each relationship.

Example

Template Code

```
%ForEachChildRel("go") {  
    insert %Parent values  
    (%ParentAtts(", ", @ins_) )
```

Expanded Code

```
insert MOVIE_COPY  
values(@ins_master_number,  
@ins_movie_copy_number,  
@ins_general_condition)  
go
```

%ForEachDefault {ewc HLP25632,HLP256_TILE,water.bmp}

Expands the macro code for each default.

```
%ForEachDefault(<separator>) {<macro code>}
```

Scope

Global

Return Value

Result of <macro code> for each default.

Example

Template Code

```
%ForEachDefault(  
) {%DefaultName }
```

Expanded Code

```
FILL_WITH_SPACES  
SET_TO_NULL  
SET_TO_ZERO
```

%ForEachDomain {ewc HLP25632,HLP256_TILE,water.bmp}

Expands the macro code for all domains that are implemented as user-defined datatypes.

```
%ForEachDomain(<separator>) {<macro code>}
```

Scope

Global

Return Value

Result of <macro code>

Example

Template Code

```
%ForEachDomain(",")  
{%DomainName }
```

Expanded Code

```
ADDRESS,  
PERSON-NAME
```

%ForEachEntity {ewc HLP25632,HLP256_TILE,water.bmp}

Expands the macro code for all entities in the current subject area.

`%ForEachEntity(<separator>) {<macro code>}`

Scope

Trigger Override

Return Value

Result of <macro code> for each entity.

Example

Template Code	Expanded Code
<code>%ForEachEntity() {%TableName}</code>	<code>MOVIE_COPY</code>
	<code>MOVIE_RENTAL_RECOR</code>

%ForEachFKAtt {ewc HLP25632,HLP256_TILE,water.bmp}

Expands macro code for each of the foreign key attributes migrated through the current relationship.

`%ForEachFKAtt(<separator>) {<macro code>}`

Scope

RI or Rel Override

Return Value

Result of <macro code> for each FK attribute.

Example

Template Code	Expanded Code
<code>%ForEachFKAtt(", ")</code>	<code>master_number, movie_copy_number</code>
<code>{%AttFieldName}</code>	

%ForEachIndex {ewc HLP25632,HLP256_TILE,water.bmp}

Expands the macro code for all indexes in the current subject area. <table> defaults to the table in the current scope; can be used to name another table (%Parent, MOVIE_COPY).<type>filters type of index (AK, IE, IF, PK, AK1, IE2, etc.). Defaults to all.

`%ForEachIndex([<table>],[<type>],[<name>],[<separator>]) {<macro code>}`

Scope

Trigger Override

Return Value

Result of <macro code> for each index.

Example

Template Code	Expanded Code
<code>%ForEachIndex { %IndexName</code>	<code>XPKMOVIE_COPY PK</code>
<code>%IndexType}</code>	

%ForEachIndexMem {ewc HLP25632,HLP256_TILE,water.bmp}

Expands the macro code for all members of the index in the current subject area. <sequence> can be used to enter the assigned number of a specific index member. Defaults to all members.

`%ForEachIndexMem([<sequence>],[<separator>]) {<macro code>}`

Scope

Trigger Override

Return Value

Result of <macro code> for each index member.

Example

Template Code	Expanded Code
<code>%ForEachIndex</code>	<code>int</code>
<code>{ %ForEachIndexMem()</code>	<code>int</code>
<code>{ %AttDatatype }}</code>	<code>varchar(10)</code>

%ForEachKey {ewc HLP25632,HLP256_TILE,water.bmp}

Expands the macro code for all alternate keys and inversion entries in the current subject area.

`%ForEachKey([<table>],[<type>],[<separator>]) {<macro code>}`

Scope

Trigger Override

Return Value

Result of <macro code> for each key.

Example

Template Code	Expanded Code
<code>%ForEachKey(%Parent,,,</code>	<code>AK1,</code>
<code>) { %KeyName}</code>	<code>IE1</code>

%ForEachKeyMem {ewc HLP25632,HLP256_TILE,water.bmp}

Expands the macro code for all members of the key.

`%ForEachKeyMem([<sequence>],[<separator>]) {<macro code>}`

Scope

Trigger Override

Return Value

Result of <macro code> for each key member.

Example

Template Code	Expanded Code
<code>%ForEachKey { %ForEachKeyMem()</code>	<code>renting-customer</code>
<code>{ %AttName} }</code>	

%ForEachParentRel {ewc HLP25632,HLP256_TILE,water.bmp}

Expands the <relationship code> for every relationship in which the trigger entity is a parent.

`%ForEachParentRel(<separator>) { <relationship code> }`

Scope

Trigger Override

Return Value

Result of <macro code> for each relationship.

Example

Template Code

```
%ForEachParentRel() {insert
%Parent values(%ParentAtts(",
",,@ins_)
}
```

Expanded Code

```
insert MOVIE_COPY
values(@ins_master_number,
@ins_movie_copy_number,
@ins_general_condition)
```

%ForEachValidValue {ewc HLP25632,HLP256_TILE,water.bmp}

Expands the macro code for all valid values within the validation rule.

```
%ForEachValidValue(<separator>) {<macro code>}
```

Scope

Trigger Override, %ForEachValidation

Return Value

Result of <macro code> for each valid value.

Example

Template Code

```
%ForEachValidValue(",") {
%ValidValue =>%ValidValueDef }
```

Expanded Code

```
N =>New (recent release),
S =>Standard release,
O =>Old movie,
C =>Classic
```

%ForEachValidation {ewc HLP25632,HLP256_TILE,water.bmp}

Expands the macro code for all validation rules.

```
%ForEachValidation(<separator>) {<macro code>}
```

Scope

Trigger Override

Return Value

Result of <macro code> for each validation rule.

Example

Template Code

```
%ForEachValidation()
{ %ValidationName }
```

Expanded Code

```
Due Date Validation
```

%If %Else {ewc HLP25632,HLP256_TILE,water.bmp}

Conditionally expands either the %If<macro code1> if the predicate evaluates to "true" or the %Else <macro code2> if the predicate evaluates to "false". The %Else clause is optional.

```
%If (<predicate>) {<macro code1>} %Else {<macro code2>}
```

Scope

Global

Return Value

Result of <macro code1> each time <predicate> evaluates to TRUE.
Result of <macro code2> each time <predicate> evaluates to FALSE.

Example

Template Code	Expanded Code
<code>%ForEachAtt() { %If(%AttIsPK)</code>	<code>master-number -> PK</code>
<code>{ %AttName -> PK}}</code>	<code>movie-copy-number -> PK</code>

%Include {ewc HLP25632,HLP256_TILE,water.bmp}

Allows you to include trigger macro code across multiple files or trigger templates.

`%Include("<path name>")`

`%Include(<trigger template name>)`

Scope

Global

Return Value

String

Example

Template Code	Expanded Code
<code>%Include("<c:\er\trg.tdl")</code>	<code><All code included in the trg.tdl file></code>

%IndexName {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the index name.

`%IndexName`

Scope

Trigger Override, %ForEachIndex, %ForEachIndexMem

Return Value

String

Example

Template Code	Expanded Code
<code>%IndexName</code>	<code>XPK_MOVIE_COPY</code>

%IndexProp {ewc HLP25632,HLP256_TILE,water.bmp}

Expands to the user-defined property (UDP) values of the index name specified.

`%IndexProp()`

Scope

Trigger Override, %ForEachKey, %ForEachKeyMem

Return Value

Index UDP values

Example

Template Code

`%IndexProp(CustID)`

Expanded Code

based on customer ID number

%IndexType {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the index type.

`%IndexType`

Scope

Trigger Override, %ForEachIndex, %ForEachIndexMem

Return Value

FK, PK, AK, IE1, etc.

Example

Template Code	Expanded Code
<code>%IndexType</code>	<code>PK</code>

%JoinFKPK {ewc HLP25632,HLP256_TILE,water.bmp}

Lets you join the foreign key of a child entity to the primary key of the parent entity in a relationship.

`%JoinFKPK([<child table>, <parent table>,<comparison op>,<separator>)`

Scope

RI or Rel Override, WHERE clause search condition

Return Value

Code supporting comparison of FK and PK values in a relationship.

Example

Template Code	Expanded Code
<code>%JoinFKPK(%Child, deleted," <> "," or")</code>	<code>MOVIE_RENTAL_RECOR.master_number <> deleted.master_number or MOVIE_RENTAL_RECOR.movie_copy_num ber <> deleted.movie_copy_number</code>

%JoinPKPK {ewc HLP25632,HLP256_TILE,water.bmp}

Lets you join the primary key of two correlations, or a base table and a correlation.

`%JoinPKPK(<table>, <correlation>, <comparison op>,<separator>)`

Scope

RI or Rel OverrideTrigger Override, WHERE clause search condition

Return Value

Code supporting comparison of PK values of two tables.

Example

Template Code	Expanded Code
<code>%JoinPKPK(%Child,ins," <> "," or")</code>	<code>MOVIE_RENTAL_RECOR.renting_custo mer <> ins.renting_customer or MOVIE_RENTAL_RECOR.master_number <> ins.master_number or ...</code>

%KeyGroupProp {ewc HLP25632,HLP256_TILE,water.bmp}

Expands to the user-defined property (UDP) values of the keygroup name specified.

`%KeyGroupProp()`

Scope

Trigger Override, %ForEachKey, %ForEachKeyMem

Return Value

Key group UDP value

Example

Template Code

`%KeyGroupProp(CustID)`

Expanded Code

Keygroup based on CustomerID
number

%KeyName {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the key name.

`%KeyName`

Scope

Trigger Override, %ForEachKey, %ForEachKeyMem

Return Value

String

Example

Template Code

`%KeyName`

Expanded Code

Alternate Key 1

%KeyType {ewc HLP25632,HLP256_TILE,water.bmp}

Expands to the key's IDEF1X type.

`%KeyType`

Scope

%ForEachKey, %ForEachIndex

Return Value

String

Example

Template Code

`%KeyType`

Expanded Code

PK1

%Len {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the string length of <macro code>.

`%Len(<macro code>)`

Scope

Global

Return Value

Integer

Example

Template Code	Expanded Code
<code>string1='aaaaa'</code>	5
<code>%Len(%.string1)</code>	

%Lookup {ewc HLP25632,HLP256_TILE,water.bmp}

Reads a two column lookup table from the text file <filename>. The first column contains a pattern string, and the second column <expr> contains a string to substitute for the pattern. The file columns must be delimited by a comma (,) and the second column must end with a carriage return. The <expr> string can also be a UDP macro that calls a valid lookup string.

`%Lookup(<filename>,<expr>)`

Scope

Global

Return Value

String

Example

Template Code	Expanded Code
<code>%Lookup(physname.txt,CUST)</code>	CUSTOMER
<code>%Lookup(physname.txt,SubjectAreaProp(Dept)</code> (where the UDP macro SubjectAreaProp(Dept) has a value of "CUST")	CUSTOMER
<code>%Lookup(Physname.txt,%Entityname)</code>	CUSTOMER, EMPLOYEE, MOVIES, etc.

A section of the physname.txt file might look like this:

```
CUST,CUSTOMER
EMPL,EMPLOYEE
MOVIE,MOVIES
.
.
.
```

%Lower {ewc HLP25632,HLP256_TILE,water.bmp}

Converts the expansion of <macro code> to lowercase.

`%Lower (<macro code>)`

Scope

Global

Return Value

Lowercase string

Example

Template Code	Expanded Code
<code>%Lower (%Parent)</code>	<code>movie</code>

%Max {ewc HLP25632,HLP256_TILE,water.bmp}

Compares <value1> and <value2> and returns the maximum.

`%Max (<value1>, <value2>)`

Scope

Global

Return Value

Larger value in comparison

Example

Template Code	Expanded Code
<code>var1=8</code>	<code>8</code>
<code>var2=5</code>	
<code>%Max (var1, var2)</code>	

%Min {ewc HLP25632,HLP256_TILE,water.bmp}

Compares <value1> and <value2> and returns the minimum.

`%Min (<value1>, <value2>)`

Scope

Global

Return Value

Smaller value in comparison

Example

Code	Expansion
<code>var1=8</code>	<code>5</code>
<code>var2=5</code>	
<code>%Min (var1, var2)</code>	

%NK {ewc HLP25632,HLP256_TILE,water.bmp}

Lists all the non-key attributes of the trigger entity and performs the specified action on each.

`%NK(<separator>,<action>,<prefix>)`

Scope

Trigger Override

Return Value

Attribute and action

Example

Template Code

```
%NK(" or ",update,%Parent)
```

Expanded Code

```
update(MOVIE_COPY.general_conditio  
n) or...
```

%NKDecl {ewc HLP25632,HLP256_TILE,water.bmp}

Lists the non-key attributes of the trigger entity followed by their datatypes.

`%NKDecl(<old prefix>,<new prefix>,<separator>,<attribute/type separator>)`

Scope

Trigger Override

Return Value

Attribute names and datatypes

Example

Template Code

```
%NKDecl(@ins_," ", " ")
```

Expanded Code

```
@ins_general_condition  
varchar(10)
```

%Not {ewc HLP25632,HLP256_TILE,water.bmp}

Lets you combine Boolean expressions to create more complex comparisons. %Not performs a "logical not" on the Boolean predicate defined by <macro code>.

`%Not(<macro code>)`

Scope

Global, %If

Return Value

Result of macro code.

Example

Template Code

```
%ForEachAtt()  
{  
    %If(%Not(%==(AttDatatype,  
int)))  
{AttName}}
```

Expanded Code

```
general-condition
```

%NotNullFK {ewc HLP25632,HLP256_TILE,water.bmp}

Compares the foreign key of a child entity <child table> in a relationship to <null expression>. This macro expands if and only if the relationship is non-identifying , nulls allowed.

`%NotNullFK(<child table>, <not null expression>,<prefix>,<separator>)`

Scope

RI or Rel Override, WHERE clause search condition

Return Value

Null expression for each FK allowing null values.

Example

Template Code

```
where %NotNullFK("is null",," or  
")
```

Expanded Code

```
where master_number is null  
or movie_copy_number is null
```

%Or {ewc HLP25632,HLP256_TILE,water.bmp}

Lets you combine Boolean expressions to create more complex comparisons. %Or performs a "logical or" on Boolean predicates defined by <macro code1> and <macro code2>. For example, %If {%Or(%AttIsFK,%AttIsRolenamed)...} lets you execute additional steps if the given attribute is either a foreign key or is rolenamed. The macro is executed if any element evaluates to "TRUE".

`%Or(<macro code1>,<macro code2>)`

Scope

Global, %If

Return Value

TRUE/FALSE

Example

Template Code

```
%If (%Or(%AttIsFK,  
%AttIsRolenamed) {  
%AttName}
```

Expanded Code

```
master_copy_number
```

%ParamDecl {ewc HLP25632,HLP256_TILE,water.bmp}

Lists all attributes of the trigger entity followed by their datatypes.

`%ParamDecl(<old prefix>,<new prefix>,<separator>,<attribute/type separator>)`

Scope

Trigger Override

Return Value

Attributes and datatypes

Example

Template Code

```
%ParamDecl(  
@ins_,"", "  
")
```

Expanded Code

```
@ins_master_number int,  
@ins_movie_copy_number int,  
@ins_general_condition  
varchar(10)
```

%ParamPass {ewc HLP25632,HLP256_TILE,water.bmp}

Assigns values to procedure parameters specified by the <old prefix> and/or the <new prefix>, for all attributes of the trigger entity. Available on Ingres only.

```
%ParamPass(<old prefix>,<new prefix>,<param/value separator>,<param separator>)
```

Scope

Trigger Override

Return Value

Parameter list for Ingres

Example

Template Code

```
%ParamPass(new," = ")
```

Expanded Code

```
master_number =  
new.master_number
```

%Parent {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the physical table name of the parent entity in a relationship.

```
%Parent
```

Scope

RI or Rel Override

Return Value

String

Example

Template Code

```
%Parent
```

Expanded Code

```
MOVIE_COPY
```

%ParentAtt {ewc HLP25632,HLP256_TILE,water.bmp}

Expands any attribute macro (e.g., %AttFieldName, %AttDatatype) on the parent primary key attribute from which the current attribute is migrated.

```
%ParentAtt(<attribute macro>)
```

Scope

%ForEachAtt, %ForEachFKAtt

Return Value

Result of <attribute macro>

Example

Template Code

```
%ForEachAtt() {%ParentAtt  
(%AttFieldName) }
```

Expanded Code

```
master_number
```

%ParentAtts {ewc HLP25632,HLP256_TILE,water.bmp}

Lists all the attributes of the parent entity in a relationship and performs the specified action on each.

`%ParentAtts(<separator>, <action>,<prefix>)`

Scope

RI or Rel Override

Return Value

Attributes and actions

Example

Template Code

```
%ParentAtts(" or ",update,
%Parent)
```

Expanded Code

```
update(MOVIE.master_number) or
update(MOVIE.customer_number)...
```

%ParentNK {ewc HLP25632,HLP256_TILE,water.bmp}

Lists the non-key attributes of the parent entity in a relationship and performs the specified action on each.

`%ParentNK(<separator>, <action>,<prefix>)`

Scope

RI or Rel Override

Return Value

Non-key attributes and actions

Example

Template Code

```
%ParentNK(" or ",update,%Parent)
```

Expanded Code

```
update(MOVIE.movie_name) or
update(MOVIE.movie_rating) or
update(MOVIE.movie_year)
```

%ParentNKDecl {ewc HLP25632,HLP256_TILE,water.bmp}

Lists the non-key attributes of the parent entity in the relationship followed by their datatypes.

`%ParentNKDecl(<old prefix>, <new prefix>,<separator>,<attribute/type separator>)`

Scope

RI or Rel Override

Return Value

FK attribute names and datatypes

Example

Template Code

```
/*If the current relationship is
<is in stock as>, then */
%ParentNKDecl(,@ins_," , ")
```

Expanded Code

```
@ins_movie_name varchar(50),
@ins_movie_rating varchar(5),
@ins_movie_rental_rate money
```

%ParentParamDecl {ewc HLP25632,HLP256_TILE,water.bmp}

Lists the attributes of the parent entity in the relationship followed by their datatypes.

`%ParentParamDecl(<old prefix>, <new prefix>,<separator>,<attribute/type separator>)`

<separator>,<attribute/type separator>)

Scope

RI or Rel Override

Return Value

Attribute names and datatypes

Example

Template Code

```
/*If the current relationship is  
<is in stock as>, then */  
%ParentParamDecl(  
@ins_,"", "")
```

Expanded Code

```
@ins_movie_name varchar(50),  
@ins_movie_rating varchar(5),  
@ins_movie_rental_rate money
```

%ParentPK {ewc HLP25632,HLP256_TILE,water.bmp}

Lists the primary key elements of the parent entity in a relationship, and performs the specified action on each.

%ParentPK(<separator>,<action>)

Scope

RI or Rel Override

Return Value

Primary key attributes and actions

Example

Template Code

```
%ParentPK(,update,%Parent)
```

Expanded Code

```
update(MOVIE_COPY.master_number)  
update(MOVIE_COPY.movie_copy_number)
```

%ParentPKDecl {ewc HLP25632,HLP256_TILE,water.bmp}

Lists the primary key elements of the parent entity in the relationship followed by their datatypes.

%ParentPKDecl(<old prefix>, <new
prefix>,<separator>,<attribute/type separator>)

Scope

RI or Rel Override

Return Value

Primary key attribute names and datatypes

Example

Template Code

```
/*If the current relationship is  
<is in stock as>, then */  
%ParentPKDecl(,@ins_)
```

Expanded Code

```
@ins_master_number varchar(10)  
@ins_movie_copy_number  
varchar(10)  
@ins_customer_number varchar(10)
```

%PhysRelName {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the physical relationship name.

`%PhysRelName`

Scope

RI or Rel Override

Return Value

String

Example

Code	Expansion
<code>%PhysRelName</code>	<code>is_rented_under</code>

%PK {ewc HLP25632,HLP256_TILE,water.bmp}

Lists the primary key elements of the trigger entity and performs the specified action on each.

`%PK (<separator>,<action>,<prefix>)`

Scope

Trigger Override

Return Value

Primary key attributes and actions

Example

Template Code	Expanded Code
<code>%PK (,update,%Parent)</code>	<code>update (MOVIE_COPY.master_number)</code> <code>update (MOVIE_COPY.movie_copy_number)</code>

%PKDecl {ewc HLP25632,HLP256_TILE,water.bmp}

Lists the primary key elements of the trigger entity followed by their datatypes.

`%PKDecl (<old prefix>,<new prefix>,<separator>>,<attribute/type separator>)`

Scope

Trigger Override

Return Value

Primary key attribute names and datatypes

Example

Template Code	Expanded Code
<code>%PKDecl (,@ins_)</code>	<code>@ins_master_number varchar(10)</code> <code>@ins_movie_copy_number</code> <code>varchar(10)</code>

%RefClause {ewc HLP25632,HLP256_TILE,water.bmp}

Generates the references clause from the OLD and NEW information added in the Entity Trigger Editor. Available on INFORMIX, Ingres, ORACLE 7/8.x, and Rdb only.

`%RefClause`

Scope

Trigger Override

Return Value

Reference clause

Example

Template Code

```
%RefClause
```

Expanded Code

```
REFERENCES NEW as inserted OLD  
as deleted
```

%RelId {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the relationship ID.

```
%RelId
```

Scope

RI or Rel Override

Return Value

Integer

%RelIsNotNull {ewc HLP25632,HLP256_TILE,water.bmp}

Examines the relationship null expression and returns "true" if nulls are not allowed and "false" if nulls are allowed.

%RelIsNotNull

Scope

RI or Rel Override, %If

Return Value

TRUE/FALSE

%RelLogProp {ewc HLP25632,HLP256_TILE,water.bmp}

Expands to the user-defined property (UDP) values of the logical relationship specified.

`%RelLogProp()`

Scope

%ForEachParentRel, %ForEachChildRel, RI or Rel Override

Return Value

Logical relationship UDP value

Example

Template Code

`%RelLogProp(Designer)`

Expanded Code

John Smith

%RelPhysProp {ewc HLP25632,HLP256_TILE,water.bmp}

Expands to the user-defined property (UDP) values of the physical relationship specified.

`%RelPhysProp()`

Scope

%ForEachParentRel, %ForEachChildRel. RI or Rel Override

Return Value

Physical relationship UDP value

Example

Template Code	Expanded Code
<code>%RelPhysProp(DeveloperName)</code>	<code>Frank Jones</code>

%RelRI {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the given referential integrity option for the specified <action> (insert, delete, update) and <RI type> (parent or child).

`%RelRI(<action>,<RI type>)`

Scope

RI or Rel Override

Return Value

CASCADE, RESTRICT, SET NULL, SET DEFAULT, or NONE

Example

Template Code	Expanded Code
<code>%RelRI(Update, Child)</code>	<code>CASCADE</code>

%RelTemplate {ewc HLP25632,HLP256_TILE,water.bmp}

Expands the template code attached to the current relationship. If no code is attached, the corresponding diagram-wide referential integrity template is expanded.

`%RelTemplate`

Scope

RI or Rel Override

Return Value

Expands the code fragment for the relationship corresponding to trigger type.

%RelType {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the relationship type. Relationship type valid values and their definitions include: RT_ID (identifying), RT_NI (non-identifying), RT_SC (subtype), or RT_MM (many-to-many).

`%RelType`

Scope

RI or Rel Override

Return Value

RT_ID, RT_NI, RT_SC, or RT_MM

Example

Template Code	Expanded Code
<code>%RelType</code>	<code>RT_ID</code>

%Scope {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the specification of how the trigger is to be executed. Available on ORACLE7 only.

`%Scope`

Scope

Trigger Override

Return Value

FOR EACH ROW or FOR EACH TABLE

Example

Template Code	Expanded Code
<code>%Scope</code>	<code>FOR EACH ROW</code>

%SetFK {ewc HLP25632,HLP256_TILE,water.bmp}

Lists the foreign key elements of a child entity in a relationship and sets each member to the specified value.

`%SetFK(<child table>,<value>)`

Scope

RI or Rel Override

Return Value

Foreign key attribute name and value

Example

Template Code	Expanded Code
<code>%SetFK(%Child,NULL)</code>	<code>MOVIE_RENTAL_RECOR.master_number = NULL</code> <code>MOVIE_RENTAL_RECOR.movie_copy_number =</code> <code>NULL</code> <code>MOVIE_RENTAL_RECOR.customer_number = NULL</code> <code>MOVIE_RENTAL_RECOR.rental_record_id = NULL</code>

%SetPK {ewc HLP25632,HLP256_TILE,water.bmp}

Lists the primary key elements of the specified table and sets each member to the specified value.

```
%SetPK(<table>,<value>)
```

Scope

RI or Rel OverrideTrigger Override

Return Value

Primary key attribute name and value

Example

Template Code

```
%SetPK(%Parent,0)
```

Expanded Code

```
MOVIE_COPY.master_number = 0,  
MOVIE_COPY.movie_copy_number = 0
```

%SubjectAreaProp {ewc HLP25632,HLP256_TILE,water.bmp}

Expands to the user-defined property (UDP) values of the subject area specified.

```
%SubjectAreaProp()
```

Scope

%ForEachParentRel, %ForEachChildRe, RI or Rel Override

Return Value

Subject area UDP value.

Example

Template Code

```
%SubjectAreaProp(DeveloperName)
```

Expanded Code

```
John Smith
```

%Substitute {ewc HLP25632,HLP256_TILE,water.bmp}

Substitutes string <pattern> in string <value> with string <substitute>.

```
%Substitute(<value>,<pattern>,<substitute>)
```

Scope

Global

Return Value

String

Example

Template Code

```
/*If the name of the child table is movie_copy_number*/  
%Substitute(%Child,  
movie,ins_movie)
```

Expanded Code

```
ins_movie_copy_number
```

%Substr {ewc HLP25632,HLP256_TILE,water.bmp}

Generates a substring of the expansion of the given <macro code>. <length> is not required.

`%Substr(<macro code>,<initial pos>,<length>)`

Scope

Global

Return Value

String

Example

Template Code	Expanded Code
<code>%Substr(macro,1,3)</code>	<code>mac</code>

%Switch {ewc HLP25632,HLP256_TILE,water.bmp}

Evaluates the argument against the specified choices. ERwin expands the macro code corresponding to the matching choice. If no match is found, ERwin expands the default macro code.

`%Switch(<argument>) {%Choose(<choice 1>) {<macro code 1>}
 %Choose(<choice 2>) {<macro code 2>} <etc...> %Default
{<default macro code>}}`

Scope

Global

Return Value

Macro code value

Example

Template Code	Expanded Code
<code>/* If %TableName is MOVIE_COPY */ %Switch(%TableName) { %Choose(MOVIE_COPY) { %FK} %Choose(MOVIE_RENTAL_RECOR) { %PK} %Default{%PK}}</code>	<code>master_number</code>

%TableName {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the physical table name of the trigger entity.

`%TableName`

Scope

Trigger Override

Return Value

Name of table

Example

Template Code

`%Tablename`

Expanded Code

`MOVIE_COPY`

%TemplateName {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the name of the trigger, stored procedure, or script template.

`%TemplateName`

Scope

Trigger Override

Return Value

Template name

Example

Template Code

`%TemplateName`

Expanded Code

`CHILD_DELETE_CASCADE`

%TriggerName {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the physical trigger name.

`%TriggerName`

Scope

Trigger Override

Return Value

Trigger name

Example

Template Code

`%TriggerName`

Expanded Code

`MOVIE_COPY_CHILD_DELETE_CASCADE`

%TableProp {ewc HLP25632,HLP256_TILE,water.bmp}

Expands to the user-defined property (UDP) values of the table name specified.

`%TableProp()`

Scope

Trigger Override

Return Value

Table UDP value

Example

Template Code

`%TableProp(Name)`

Expanded Code

`Customer`

%TriggerRelRI {ewc HLP25632,HLP256_TILE,water.bmp}

Boolean predicate that is "true" if the given trigger and relationship are of the given <action> (Update/Delete/Insert), <RI type> (Child/Parent), and <integrity > (Cascade/Restrict/Set Null/Set Default).

`%TriggerRelRI(<action>, <RI type>, <integrity>)`

Scope

RI or Rel Override, %If

Return Value

TRUE/FALSE

Example

Template Code

```
%ForEachChildRel()  
{ %If(%TriggerRelRI  
(Update, Child, Cascade))  
{%VerbPhrase Child Update  
Cascade}}
```

Expanded Code

```
is rented under Child Update  
Cascade
```

%UpdateChildFK {ewc HLP25632,HLP256_TILE,water.bmp}

Lists the foreign key elements of the child entity in a relationship and performs the update action on each member. Available on ORACLE7, SQL Server, and SYBASE only.

`%UpdateChildFK()`

Scope

RI or Rel Override

Return Value

Foreign key value and action

Example

Template Code

```
%UpdateChildFK()
```

Expanded Code

```
update(master_number) or  
update(movie_copy_number)  
...
```

%UpdateParentPK {ewc HLP25632,HLP256_TILE,water.bmp}

Lists the primary key elements of the parent entity in a relationship and performs the update action on each member. Available on ORACLE7, SQL Server, and SYBASE only.

`%UpdateParentPK()`

Scope

RI or Rel Override

Return Value

Primary key value and action

Example

Template Code

```
%UpdateParentPK()
```

Expanded Code

```
update (master_number)
```

%UpdatePK {ewc HLP25632,HLP256_TILE,water.bmp}

Lists the primary key elements of the trigger entity and performs the update action on each member. Available on ORACLE7, SQL Server, and SYBASE only.

```
%UpdatePK()
```

Scope

Trigger Override

Return Value

Primary key value and action

Example

Template Code

```
%UpdatePK()
```

Expanded Code

```
update (master_number)
update (movie_copy_number)
```

%Upper {ewc HLP25632,HLP256_TILE,water.bmp}

Converts the expansion of <macro code> to uppercase.

```
%Upper(<macro code>)
```

Scope

Global

Return Value

String

Example

Template Code

```
%Upper(%EntityName)
```

Expanded Code

```
MOVIE COPY
```

%ValidationHasValidValues {ewc HLP25632,HLP256_TILE,water.bmp}

A Boolean predicate that lets you determine whether or not the current validation has valid values.

```
%ValidationHasValidValues(<validation name>)
```

Scope

Global with an argument (<validation name>) or %ForEachValidation, %If

Return Value

TRUE/FALSE

%ValidationName {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the name of the validation.

`%ValidationName`

Scope

`%ForEachValidation`

Return Value

Validation Name

Example

Template Code

`%ValidationName`

Expanded Code

`Due Date Validation`

%ValidationRule {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the server-side validation rule.

`%ValidationRule(<validation name>)` or `%ValidationRule`

Scope

Global with an argument (<validation name>) or `%ForEachValidation`

Return Value

Validation rule string

Example

Template Code

`%ValidationRule
(Valid CustNumber)`

Expanded Code

`@col BETWEEN 1000 AND 9999`

%ValidValue {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the data value of a valid value.

`%ValidValue`

Scope

`%ForEachValidValue`

Return Value

String

Example

Template Code

`%ValidValue`

Expanded Code

`CHILDREN`

%ValidValueDef {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the definition of a valid value.

`%ValidValueDef`

Scope

%ForEachValidValue

Return Value

String

Example

Template Code

```
/* Where valid values are:
COMEDY, HORROR, DRAMA, NEW
RELEASE and CHILDREN, one
definition could read: */
%ValidValueDef
```

Expanded Code

```
A movie that is of interest to
children.
```

%VerbPhrase {ewc HLP25632,HLP256_TILE,water.bmp}

Returns the verb phrase of a relationship.

%VerbPhrase

Scope

RI or Rel Override

Return Value

Relationship verb phrase

Example

Template Code

%VerbPhrase

Expanded Code

```
is rented under
```

